

¿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

¿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 GTB) 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

1- En cuanto a los riesgos por dependencia de datos en las arquitecturas superescalares...

Tiene una:

Los riesgos WAW y WAR son riesgos por dependencia de datos que se pueden solucionar por adelantamiento

Los riesgos RAR son riesgos por dependencia de datos que se pueden solucionar utilizando renombrado

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 no se pueden solucionar utilizando renombrado

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

Tiene una:

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.

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.

Si la predicción es explícita, no se utilizan bits de predicción

3- El mejor tipo de benchmark para evaluar un sistema es:

Tiene una:

Depende de lo que se desee evaluar.

Los benchmarks sintéticos porque son independientes de las aplicaciones.

Los kernels porque permiten evaluar aspectos concretos.

Las aplicaciones reales porque evalúan la realidad.

4- ¿Qué diferencias existen entre las estaciones de reserva y la ventana de instrucciones?

Tiene una:

Las instrucciones se cargan en la ventana de instrucciones una vez decodificadas 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 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

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

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

Tiene una:

Para nada. En superescalares no se usa ese algoritmo.

Para evitar los efectos de los **riesgos WAR** y **WAW** dependencias

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

6- El buffer de reorden en una arquitectura superescalar permite:

Tiene una:

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.

Completar (sacar del cauce) las instrucciones en un orden diferente al orden de programa.

La ejecución de las instrucciones fuera de orden.

El buffer de reorden no se utiliza en las arquitecturas superescalares.

7- 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?

Tiene una:

7+3+4+5

7+3+4+**10**+3*10

7+3+4+6+10

7+3+4+10

8- Una ventana de instrucciones alineada es:

Tiene una:

Una ventana de instrucciones que carga instrucciones conforme se va vaciando (sin esperar a vaciarse del todo)

Una ventana de instrucciones que tiene una linea de separación entre los códigos de instrucción

Una ventana de instrucciones que carga instrucciones cuando está completamente vacía

Una ventana de instrucciones extraterrestre

9- ¿Cuál de las siguientes afirmaciones es correcta para la gestión de los riesgos de control?

Tiene una:

El bloqueo del procesamiento del salto es la estrategia más común en los procesadores superescalares

El procesamiento especulativo de los saltos es la estrategia más común en los procesadores superescalares

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

10- Una arquitectura vectorial es:

Tiene una:

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

Una arquitectura donde el cálculo de los componentes del vector se realiza de forma independiente obteniendo buenos rendimientos.

Una arquitectura orientada al procesamiento de vectores (suma de vectores, productos escalares, etc.).

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

Tiene una:

Almacenar la predicción en un buffer implícito

No se almacena la predicción sino la dirección del salto en el buffer correspondiente y en función de este almacenamiento se decide la predicción.

Escribir en una hoja de papel la predicción del salto

Se almacena la predicción en una tabla

12- ¿Cuál de las siguientes afirmaciones es correcta?

Tiene una:

Todas las afirmaciones son correctas

La tabla de historia de los saltos con bits desacoplados aumenta el HW necesario para gestionar los saltos.

Los campos de la BTB se actualizan después de ejecutar el salto.

La tabla de historia de los saltos con bits desacoplados permite predecir instrucciones que no estén en la BTAC

13- ¿Cuál de las siguientes afirmaciones es correcta?

Tiene una:

Las instrucciones se introducen en el buffer de reorden en orden de programa estricto

El buffer de renombrado puede estar mezclado con los registros de la arquitectura

Las instrucciones se retiran del buffer de reorden si han finalizado y todas las que le preceden también.

En el buffer de renombrado se utiliza un puntero de cabecera que apunta a la siguiente posición libre del buffer.

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

Seleccione una:

Permite varias escrituras pendientes a un mismo registro.

No tiene ninguna ventaja

Tiene un elemento que indica si el valor es válido o no

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

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

Seleccione una:

Prediccion implicita y explicita

La prediccion implicita no se puede utilizar ya que es necesario tener informacion sobre el estado anterior del salto y en un salto incondicional esto no es posible

La prediccion explicita no se puede utilizar ya que es necesario tener informacion sobre el estado anterior del salto y en un salto incondicional esto no es posible

La prediccion no tiene sentido cuando el salto es incondicional

16- El acceso a memoria concurrente o tipo C es:

Seleccione una:

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 simultanea

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

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

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(...)

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

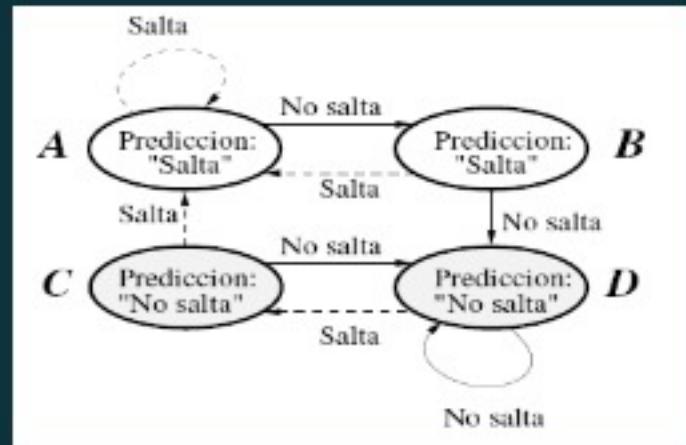
Puntúa como
0,50

¶ 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

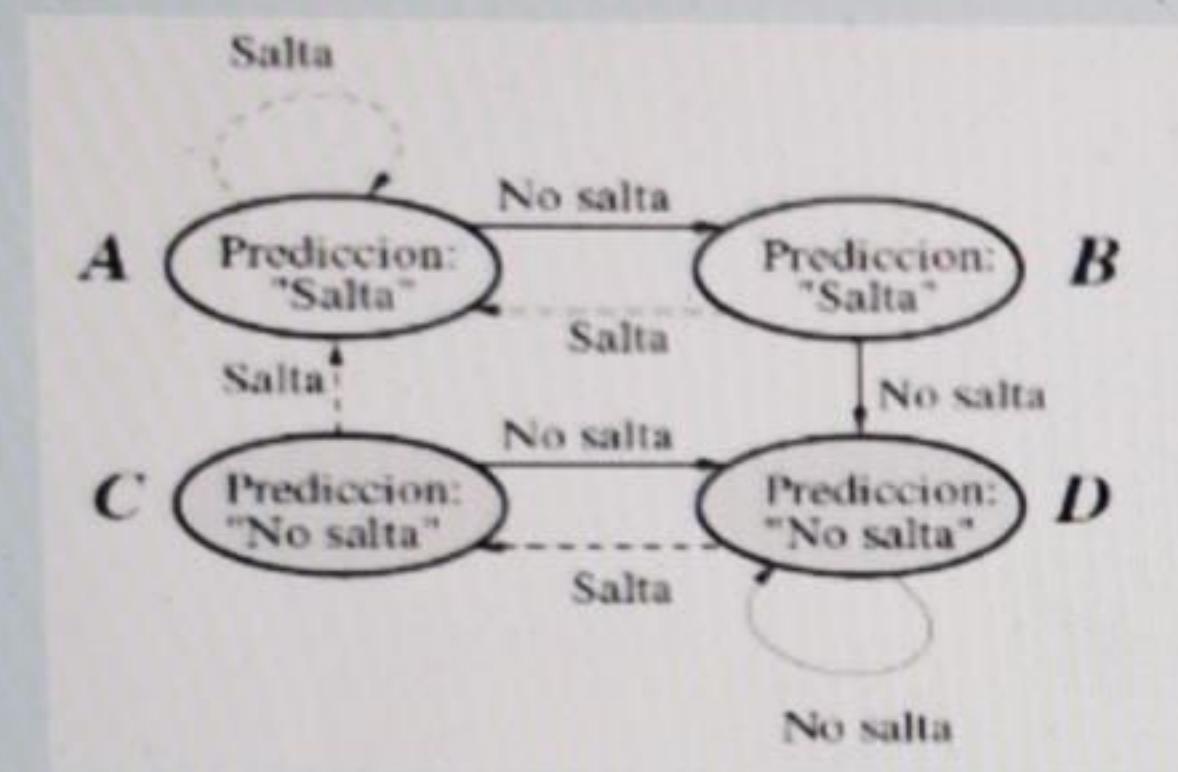
La ejecución de las instrucciones en una máquina superescalar con buffer de reorden o buffer de renombrado....

- 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:

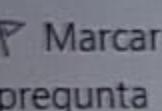
¿Cuántos **commutadores** hay en la etapa 1 de la implementación más económica de esta topología?

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

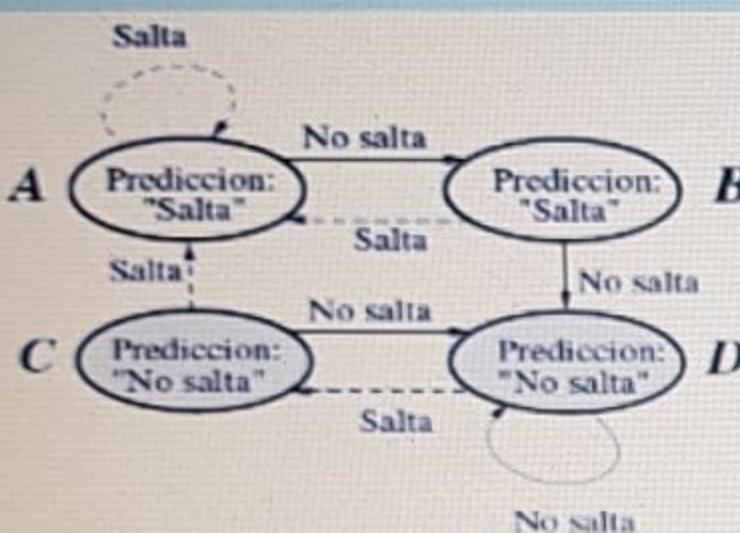
Pregunta 3

Respuesta
guardada

Puntúa como
1.50



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 serán los bits de predicción de la primera fila de la BTB una vez realizada la primera iteración?

88

¿Cuáles serán 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

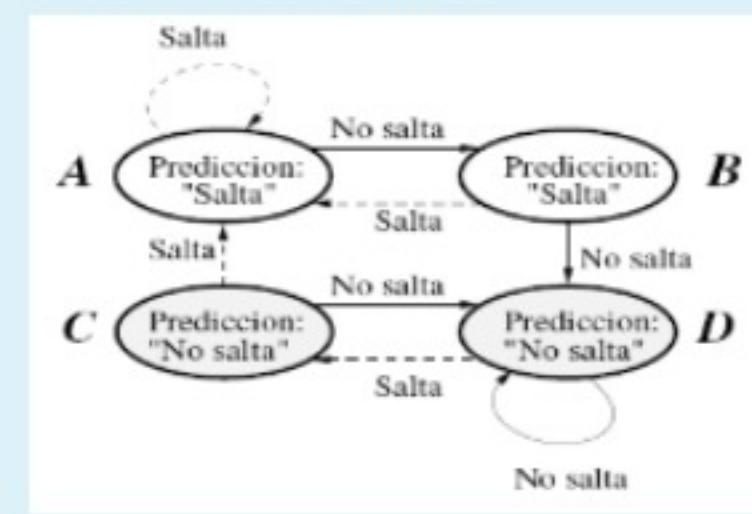
En una máquina superescalar con buffer de reorden o buffer de renombrado....

- 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)**?

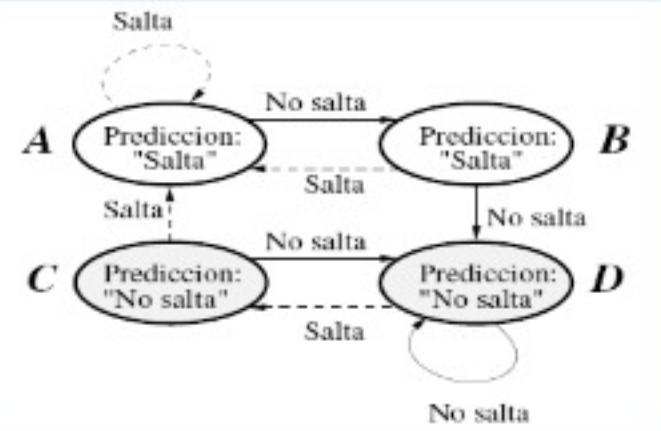
¿Cuántos ciclos tarda el código?

Pregunta 2Sin 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.

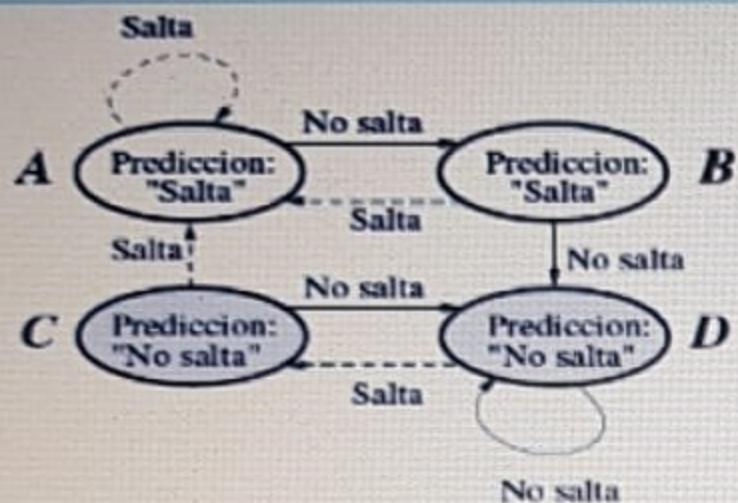
Respuesta:

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:



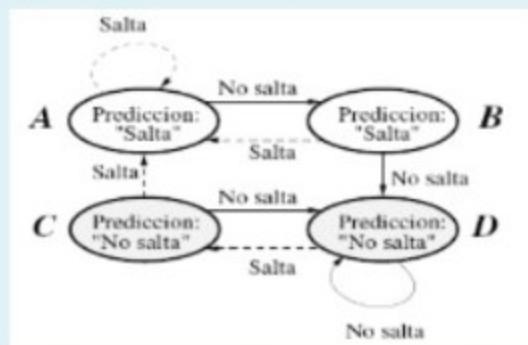
Pregunta 3

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.

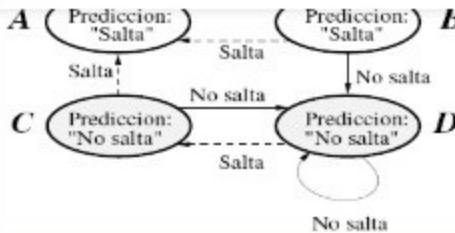
Recuerdo la notación (la misma que la que hemos visto en clase):

- **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

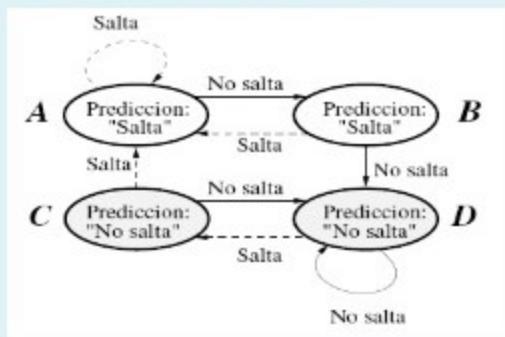
Sin responder

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:

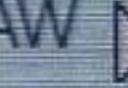
- b) ¿Qué fracción del problema total debe ejecutar cualquiera de los núcleos más rápidos?

[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

Markar
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`?

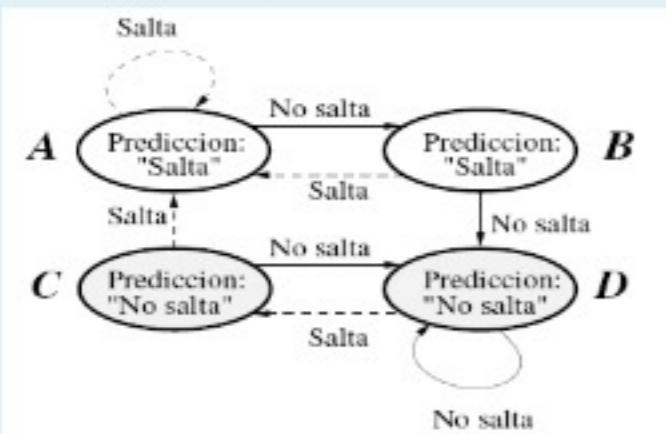
9

¿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**.

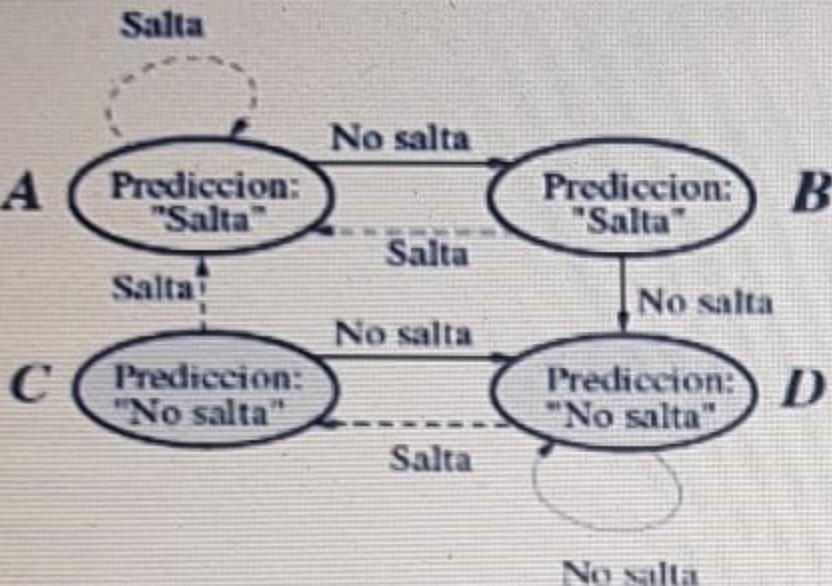
Respuesta:

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_i:Write(N|M,M)** significa: el procesador *i* escribe el valor N en la posición de memoria M (dada en hexadecimal)
- **P_i: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:

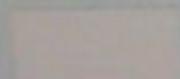
```
addi r3,r1,#800  
addi r1,r1,#8  
addi r2,r2,#8  
ld f0,coef  
ld f2,-8(r1)  
ld f4,0(r1)  
muld f8,f2,f0  
addd f4,f8,f4  
sd 0(r2),f4  
addi r1,r1,#8  
addi r2,r2,#8  
slt r4,r1,r3
```

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).

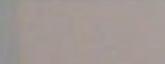
¿En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción `addi r2,r2,#8`?



¿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.

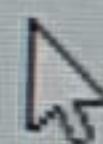
[ENGLISH]

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?**

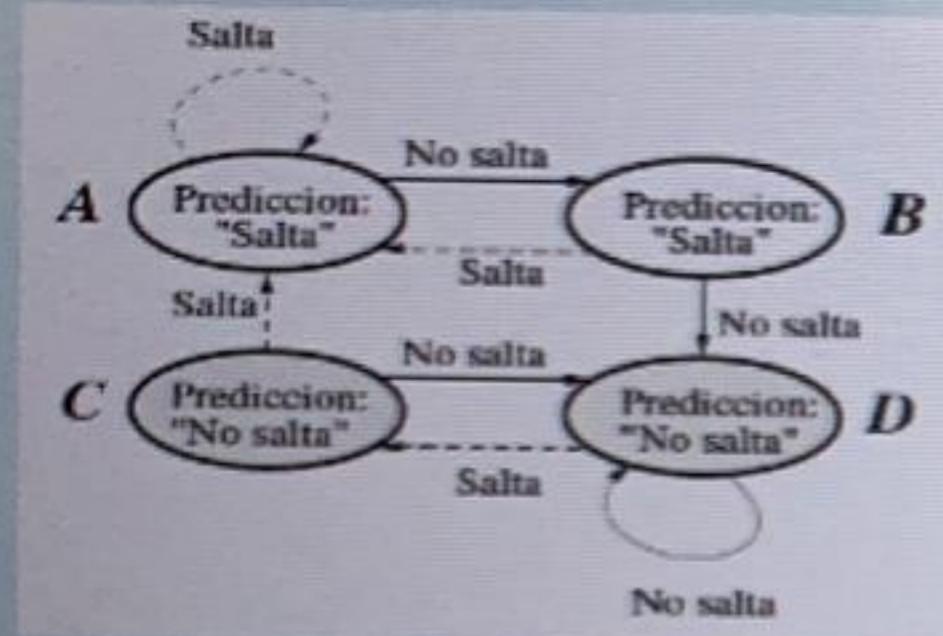
Sea un computador superescalar capaz de decodificar 2 inst/c, emitir 2 inst/c, escribir 2 inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión puede ser desordenada y contiene una unidad de adelantamiento. Teniendo el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

:En qué ciclo finaliza la etapa de ejecución (EX) de la primera instrucción UI: 2 - 3 #22

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									
Línea	dir	dato	Estado	dir	dato	Estado	dir	dato	Estado	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	6	S	2	4	I	0	0	S
	3	3		1	-6		3	7		1	-6	

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	BTB	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?