

Arquitectura de Computadores. Examen final 4/02/2014. Teoría.

Puntuación.- Cada apartado vale un punto.

1. Para ejecutar el programa P un procesador invierte un promedio de 2 ciclos de reloj por instrucción. El 10% de las instrucciones son de punto flotante.

a) Si los cálculos enteros consumen el 60% del tiempo de ejecución, calcula los ciclos de reloj en promedio consumidos por cada instrucción de PF.

b) En el mismo supuesto del apartado anterior, supongamos que mejoramos el hardware de PF de tal manera que las instrucciones de PF reducen su tiempo de ejecución a una cuarta parte ¿Cuál es el Speedup?

$$a) T = N \text{ CPI } t = 2 N t$$

$$T_{PF} = 0,4 T; \text{ Sustituyendo: } T_{PF} = 0,4 (2 N t) = 0,8 N t$$

$$\text{Pero también: } T_{PF} = N_{PF} \text{ CPI}_{PF} t = 0,1 N \text{ CPI}_{PF} t; \text{ Igualando:}$$

$$0,8 N t = 0,1 N \text{ CPI}_{PF} t ; \text{ De donde: } \text{CPI}_{PF} = 8$$

$$b) F = 0,4; x = 4$$

$$S = 1 / [(1-F) + (F/x)] = 1 / [0,6 + (0,4 / 4)] = 1 / 0,7 = 1,43$$

2. En una arquitectura con planificación dinámica de instrucciones que utiliza el algoritmo de Tomasulo sin especulación, explica razonadamente las operaciones que se realizan al ejecutar la fase WRITE de una instrucción de la forma: ADDD F4, F2, F0

Esperar hasta que la ejecución esté completa en ER(x) Y el CDB esté disponible. Entonces:

$$\text{Si } (F4.Q_i = x) \Rightarrow (F4.Q_i = 0) \text{ Y } (F4.V_i = \text{RESUL})$$

$$\forall z (\text{Si } ER(z).Q_j = x) \Rightarrow (ER(z).Q_j = 0) \text{ Y } (ER(z).V_j = \text{RESUL})$$

$$\forall z (\text{Si } ER(z).Q_k = x) \Rightarrow (ER(z).Q_k = 0) \text{ Y } (ER(z).V_k = \text{RESUL})$$

$$\forall z (\text{Si } SB(z).Q_i = x) \Rightarrow (SB(z).Q_i = 0) \text{ Y } (SB(z).V_i = \text{RESUL})$$

$$ER(x).Busy = \text{No}$$

3. Supongamos la arquitectura básica del DLX, segmentado en 5 etapas, que tiene una Branch Target Address Cache (BTAC) en la etapa IF, con 16K entradas y emplazamiento directo. La anchura del PC es 32 bits. Los saltos se resuelven en la etapa DE.

a) Cuando se realiza la etapa IF de una instrucción que está almacenada en la dirección X de memoria, describe detalladamente cómo se accede a la BTAC, y cómo se determina si la instrucción se encuentra o no registrada en la BTAC como una instrucción de salto.

b) Explica razonadamente a qué distancia tienen que estar almacenadas en memoria dos instrucciones de salto para que compitan por la misma entrada de la BTAC. Se asume que cada instrucción ocupa una palabra de memoria y que la memoria se direcciona por palabras.

a) Dado que la BTAC tiene 16K entradas ($=2^{14}$) con emplazamiento directo, tomamos los 14 bits menos significativos de la dirección de la instrucción (que está almacenada en el PC) y los usamos para seleccionar una entrada de la BTAC. En esta entrada habrá un campo de TAG formado por $32-14=18$ bits. Comparamos esos 18 bits con los 18 bits más significativos del PC. Si hay coincidencia, entonces la instrucción ha sido previamente registrada en la BTAC como de salto, y el campo DIR nos dirá la dirección de destino del salto (en caso de que se tome)

b) Dado que la entrada de la BTAC a la que se accede viene determinada por los 14 bits menos significativos del PC, dos instrucciones de salto almacenadas respectivamente en las direcciones X e Y, competirán por la misma entrada si tienen idénticos los 14 bits menos significativos. Y esto ocurre si la distancia $|X-Y|$ es un múltiplo entero de 2^{14} .

4. Describe brevemente en qué consiste y cuáles son las principales características del multithreading de grano fino. Explica sus principales ventajas y desventajas. Cita un ejemplo de procesador que lo implemente.

Tema 3. Tranp 62: Conmuta entre threads en cada instrucción, entrelazando la ejecución de los diferentes thread.

Generalmente en modo "round-robin", los threads bloqueados se saltan

La CPU debe ser capaz de conmutar de thread cada ciclo.

Ventaja; puede ocultar stalls de alta y baja latencia, cuando un thread está bloqueado los otros usan los recursos.

Desventaja; retarda la ejecución de cada thread individual, ya que un thread sin stall es retrasado por reparto de recursos (ciclos) entre threads

Ejemplo Niagara y Niagara 2 (SUN)

5. En arquitectura vectorial;

a) explica para qué usamos la siguiente expresión: $T_n = \left\lceil \frac{n}{MVL} \right\rceil \times (T_{loop} + T_{start}) + n \times T_{chime}$

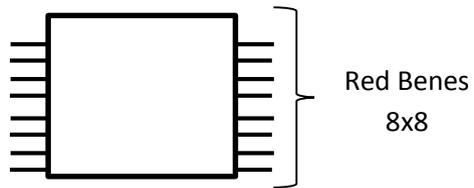
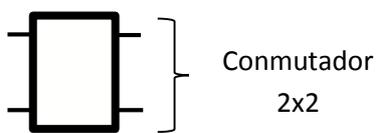
y define el significado de cada una de las variables que aparecen en ella.

b) Idem para la expresión: $mcm(\text{espaciamento}, n^{\circ} \text{ bancos}) / \text{espaciamento} \geq T \text{ acceso a banco}$

a)

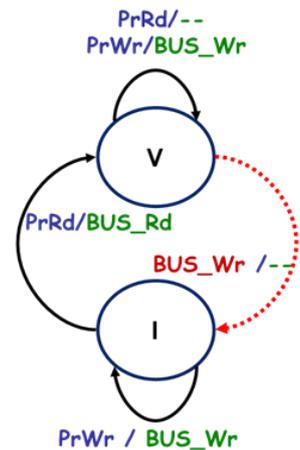
b)

6. Suponiendo que dispones de los tipos de componentes dibujados a continuación, construye con ellos una red Benes 16x16. Nota: Debes usar 2 redes Benes 8x8 y los conmutadores 2x2 que estimes necesarios.



7. La figura de la derecha muestra el diagrama de estados del protocolo “snoopy” de dos estados. A la vista del diagrama responde razonadamente a las siguientes preguntas (no se obtendrá puntuación por respuestas no razonadas):

- Cuando un procesador tiene un fallo de escritura ¿se lleva el bloque correspondiente a su cache? ¿Por qué?
- ¿Qué política de coherencia de cache se usa: invalidación o actualización en escritura? ¿Por qué?
- Para la actualización de la memoria principal ¿se usa escritura directa o pos-escritura? ¿Por qué?
- En un fallo de lectura ¿quién proporciona la copia válida del bloque referenciado? ¿Por qué?



- No, porque el diagrama muestra que cuando un bloque está en estado I y se hace una escritura (fallo de escritura) la escritura provoca un Bus_Wr que actualiza la Mp y el bloque sigue en estado I. Luego no se copia a la cache.
- Invalidación, porque cuando un controlador de cache tiene un bloque en estado V y detecta un Bus_Wr a ese bloque, lo pone en estado I, tal como muestra el diagrama
- Se usa escritura directa, ya que cuando un bloque está en la cache de un procesador en estado V, y se hace una escritura (acierto de escritura) se genera una transacción Bus_Wr para actualizar la Mp.
- Como la Mp está siempre actualizada, lo proporciona la Mp como respuesta al correspondiente Bus_Rd.