

Arquitectura de Computadores. Examen extraordinario 1/09/2015. Teoría.

1. Sabiendo que el conjunto de programas patrón SPEC_FP2006 está formado por 17 programas

a) (1 punto) Explica cómo se calcula el valor de SPEC_FP2006 para un determinado computador C.

b) (1 punto) Supongamos que el valor de SPEC_FP2006 para el computador C es 25. Si el período de reloj de C se reduce al 80% de su valor inicial ¿se puede afirmar que el nuevo valor de SPEC_FP2006 para C será 31,25? Justifica la respuesta.

a)

$$SPEC_FP2006(C) = (r_1 \times r_2 \times r_3 \times \dots \times r_{16} \times r_{17})^{1/17}$$

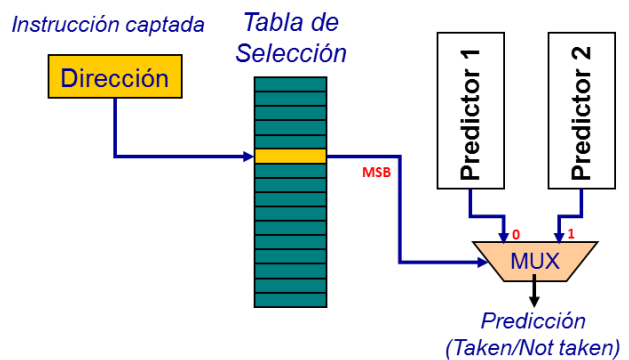
Siendo: $r_i = (T \text{ ejec patrón } i \text{ en máquina de referencia}) / (T \text{ ejec patrón } i \text{ en } C)$

b) Llamemos C' al computador mejorado)

$$r'_i = (T \text{ ejec patrón } i \text{ en máquina de referencia}) / (T \text{ ejec patrón } i \text{ en } C') = (T \text{ ejec patrón } i \text{ en máquina de referencia}) / (0,8 \times T \text{ ejec patrón } i \text{ en } C) = (1/0,8) \times r_i$$

$$SPEC_FP2006(C') = [(1/0,8)^{17} \times r_1 \times r_2 \times r_3 \times \dots \times r_{16} \times r_{17}]^{1/17} = (1/0,8) SPEC_FP2006(C) = 25/0,8 = 31,25. \text{ Luego la afirmación es cierta.}$$

2. (1 punto) La figura representa el esquema general de un predictor híbrido. Explica cómo se suele implementar la tabla de selección. Explica cómo se actualiza la tabla de selección en todos los casos posibles: si los dos predictores fallan, si los dos aciertan, si acierta el Predictor 1 y falla el 2, y si falla el Predictor 1 y acierta el 2.



(Directamente de las transparencias de teoría)

3. (1 punto) En un procesador con Planificación Dinámica mediante el algoritmo de Tomasulo sin especulación, explica brevemente cuál es la función de los Store Buffers, y qué información almacena cada una de sus entradas.

Función: Alojamiento temporal de instrucciones de STORE que han completado la fase ISSUE, pero todavía no han realizado la escritura en memoria.

Información almacenada:

- Vi: Valor que debe ser almacenado en memoria (si está calculado).
- Dir: Dirección de almacenamiento
- Qi: Nº de la ER que está produciendo el resultado. Si Qi=0 => valor calculado
- Busy: 1 bit que indica si la entrada del SB está libre u ocupada.

4. (1 punto) Explica brevemente en qué consiste el multithreading de grano grueso y cuáles son sus principales ventajas y desventajas. Cita algún ejemplo de un procesador que lo implemente.

-Conmuta entre threads solo en caso de largos stalls, como fallos de cache L2

Ventajas:

- No necesita conmutación entre thread muy rápida.
- No retarda cada thread, la conmutación solo se produce cuando un thread no puede avanzar.

Desventajas: no elimina pérdidas por stalls cortos. La conmutación es costosa en ciclos.

Ejemplo: Itanium o SPARC64 VI

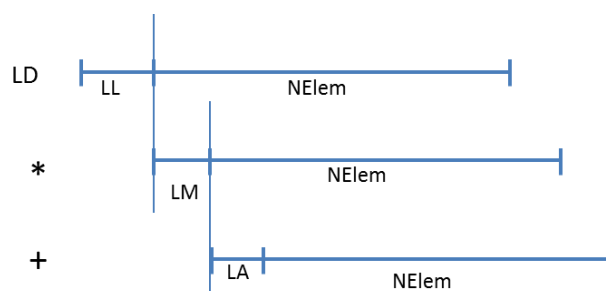
5. (1 punto). En un computador vectorial, explica brevemente en qué consiste el encadenamiento de operaciones vectoriales. Para una secuencia de operaciones vectoriales de la forma

LV	V1, R1
MULVV.D	V3,V1,V2
ADDVV.D	V5,V3,V4

dibuja el diagrama de tiempo que muestra su ejecución en forma encadenada y explica el significado de cada uno de los tramos o intervalos que aparecen el mismo.

Es un mecanismo para acelerar la ejecución de instrucciones vectoriales con dependencias LDE, que consiste en conectar directamente la salida de la UF que está produciendo un vector resultado a la entrada de la UF donde se ejecuta la operación que necesita dicho resultado. (Tb se debe admitir la explicación mediante un gráfico que muestre la conexión entre las UFs, como está en las transp)

Diagrama



donde: LL=Latencia de la unidad de Load; LM=Latencia (nº etapas) del multiplicador; LA=latencia (nº etapas) del sumador; Nelem=nº de elementos que se procesan (valor del VLR)

6. (1 punto) Describe brevemente las principales similitudes y diferencias entre un computador vectorial y una GPU tipo NVIDIA.

Similitudes

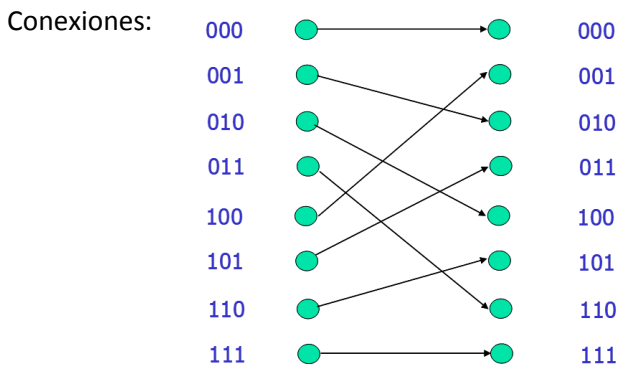
- Funciona bien en problemas con paralelismo de datos
- Transferencias con memoria tipo dispersar/reunir (scatter/gather)
- Registros de máscara
- Existencia de grandes ficheros de registros

Diferencias

- No hay un procesador escalar
- Utilización de multithreading para ocultar la latencia de memoria
- Existencia de gran cantidad de UFs: Contrasta con la reducida cantidad de UFs muy segmentadas, que es típica de los procesadores vectoriales

7. (1 punto) Define la permutación de baraje perfecto (perfect shuffle) con 2^n entradas y salidas. Dibuja las conexiones para una red de baraje perfecto (en base 2) con 8 entradas y salidas.

Def: $\sigma(x_{n-1} x_{n-2} \dots x_1 x_0) = (x_{n-2} \dots x_1 x_0 x_{n-1})$ (se asume que la base k es 2)



8. En una red tipo malla, explica brevemente en qué consiste el encaminamiento de orden dimensional (dimension-order routing). Qué tipo de problema se pretende evitar usando este encaminamiento.

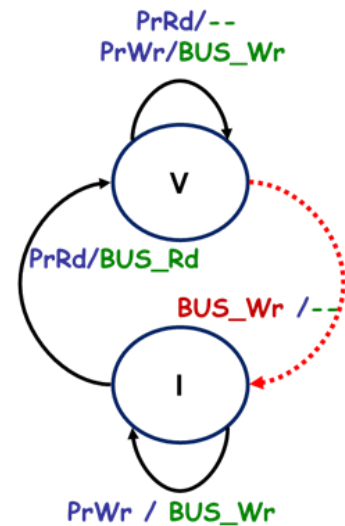
- En la propagación del paquete las dimensiones de la red (X, Y, Z, ...) se siguen en un orden estricto.
- Los enlaces en una cierta dimensión no pueden ser utilizados por un paquete hasta que no ha recorrido todos los enlaces necesarios (para alcanzar el destino) en todas las dimensiones precedentes.

Este método se utiliza para evitar situaciones de interbloqueo.

9. (1 punto) 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):

¿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?



- 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 (acuerdo de escritura) se genera una transacción Bus_Wr para actualizar la Mp.