

Ampliación de Arquitecturas de Computadoras (Cuestiones y Solución)

Las soluciones aparecen en cursiva y en la mayoría de las ocasiones se dan indicaciones de como llegar al resultado, se entiende que el alumno debe explicar cada cosa; aquí sólo se da un resumen como referencia..

Examen de febrero (13/02/04) **Teoría y problemas**

Durante el examen no se permiten apuntes ni nada encima de la mesa. El tiempo para la realización de este ejercicio es de **2 horas**.

1. (3 puntos) Supongamos una máquina vectorial con registros vectoriales de 64 elementos, una unidad funcional de cada tipo (ADDV, SUBV, MULTV y DIVV) y un cauce de lectura (LV) y otro de escritura (SV) con la memoria. Cada registro vectorial tiene dos cauces de lectura y uno de escritura. Los tiempos de arranque son 12 para LV y SV; 6 para ADDV y SUBV; 7 para MULTV; y 16 para DIVV. El tiempo de bucle es de 15 ciclos y la frecuencia de reloj 200 MHz. Dado el siguiente código vectorial (parte interna de un bucle):

12	[LV	V2 , R1]	12
12	[LV	V1 , R2]	12+6+6=24
6	[ADDV	V3 , V2 , V1]	
6	[SUBV	V4 , V3 , V2]	
12	[MULTV	V5 , V4 , V2]	12
	[SV	R1 , V3]	
12	[SV	R2 , V4]	12

Calcula R_{728} en dos casos: uno suponiendo que la máquina no soporta encadenamiento y otro suponiendo que sí que lo soporta. Debe incluirse el código con la representación gráfica de la separación en convoyes y los tiempos de arranque.

Sin encadenamiento: El tiempo de arranque es $12+12+6+6+12+12=60$ y hay 6 convoyes.

$T_{728}=[728/64](60+15)+6*728=12*75+4368=5268$ ciclos.*

*$R_{728}=728*3*200/5268=82.9$ MFlops.*

Con encadenamiento: Tiempo de arranque es $12+24+12+12=60$ y hay 4 convoyes.

$T_{728}=[728/64](60+15)*4*728=900+3640=4540$ ciclos*

*$R_{728}=728*3*200/4540=114.59$ MFlops.*

2. (1,8 puntos) Supongamos dos máquinas A y B. La máquina A es un procesador vectorial cuyos registros vectoriales tienen 64 elementos, la máquina B es un procesador matricial con 32 elementos de proceso. Las unidades funcionales del vectorial están completamente segmentadas mientras que las del matricial están sin segmentar. En todos los casos el tiempo de arranque de las unidades funcionales es 12. Si las dos las venden al mismo precio ¿con cual te quedarías? Debes exponer **todas** las razones que justifican tu decisión y calcular cuanto tardaría cada máquina en ejecutar el siguiente código (los vectores son de 64 elementos):

ADDV V1 , V2 , V3

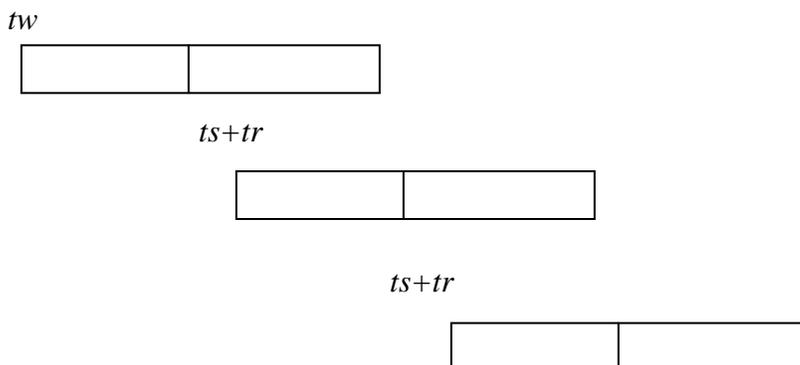
Desde el punto de vista de la programación no es muy diferente un vectorial de un matricial pues ambos tienen juegos de instrucciones similares. La ventaja del matricial es que los cálculos sobre un vector se realizan en paralelo sobre cada elemento, mientras que en el vectorial se hace uno tras otro, de forma segmentada para ganar velocidad, pero uno tras otro en cualquier caso. El matricial también tiene como ventaja el poder interconectar los diferentes elementos de proceso según varios patrones, por lo que se pueden implementar operaciones que el vectorial no admite. Eso sí, un procesador matricial, al tener todas sus unidades replicadas suele tener un mayor coste. Pero todo esto es en general.

En este caso particular el matricial tiene 32 EP, pero todos trabajan en paralelo. Una instrucción en el vectorial tarda el tiempo de arranque y luego un ciclo por elemento, o sea, 12 ciclos más 64 (o estrictamente hablando 63) es decir 72 ciclos. En cambio, el matricial calcula 32 elementos en paralelo, así que para calcular 64 lo tiene que hacer en dos vueltas, y como las unidades no están segmentadas tardaría 24 ciclos (12+12); si estuvieran segmentadas aun tardaría menos.

En cuanto al código que se presenta, como no hay dependencias de datos ni estructurales, las tres instrucciones se ejecutarían en paralelo, el vectorial lo haría entonces en unos 72 ciclos y el matricial en unos 24 (sin contar en ambos casos la demora en la búsqueda de la instrucción, etc.)

Por lo tanto, a pesar de estar comparando un matricial “recortado” con un vectorial al completo, sigue ganando el matricial. La cuestión es que seguramente no nos lo venderían al mismo precio (no hay más que contar las unidades funcionales de uno y otro).

3. (2 puntos) Se propone un nuevo mecanismo de control de flujo segmentado para una red de un multicomputador. Este mecanismo funciona de la siguiente manera: los encaminadores tienen memorias internas donde se guardan los flits, pero sólo cabe medio paquete por canal. Además, un encaminador no manda flits al nodo siguiente si no tiene al menos la mitad del paquete en memoria, o dicho de otra manera: sólo se envían flits al nodo siguiente cuando el buffer interno del canal correspondiente está lleno. De alguna manera es como la conmutación de paquetes pero en este caso se conmutan medios paquetes, de manera que un paquete está en dos nodos a un tiempo como mucho. Dibujar el diagrama de tiempos de un paquete que atraviesa tres enlaces (4 nodos) que siga este mecanismo de conmutación. Explicar los retrasos que sufre el paquete y dar una expresión para la latencia total de un paquete que recorra D enlaces.



Sin contar con los retrasos del nodo origen, lo que tarda la mitad del paquete en llegar a su destino será $D(tr+(ts+tw)[(L+W)/(2W)])$ y lo que tarda en llegar el resto de la mitad del paquete será $\max\{ts,tw\}[(L+W)/(2W)]$ donde L es la longitud de la parte de datos, W la de a cabecera y por tanto $[(L+W)/W]$ es el número de flits del paquete. La latencia total sería entonces:

$$D(tr+(ts+tw)[(L+W)/(2W)]) + \max\{ts,tw\}[(L+W)/(2W)]$$

Aun es más, podemos generalizar lo anterior y suponer que el paquete no sale hasta que se tengan F flits en el buffer. En ese caso se tiene que el tiempo es

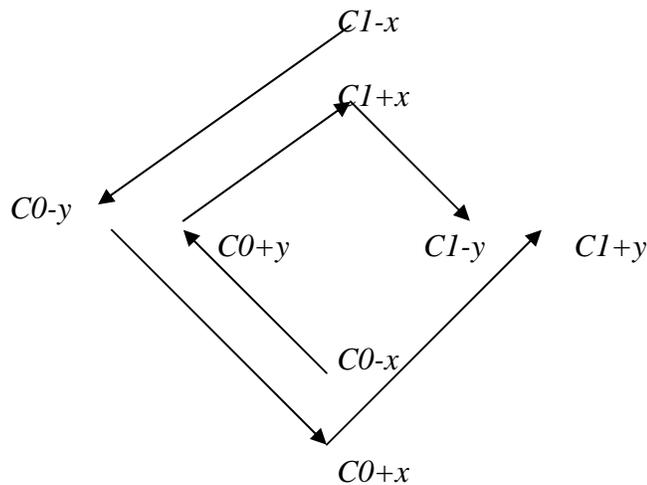
$$D(tr+(ts+tw)[F]) + \max\{ts,tw\}[(L+W)/W - F]$$

Donde es fácil comprobar que si esperamos tantos flits como tenga el paquete ($F=(L+W)/W$) tenemos la latencia de la conmutación de paquetes, y si sólo tenemos que esperar un flit ($F=1$) entonces tenemos la latencia de la conmutación de lombriz, tal como debe ser.

Si $F=(L+W)/(2W)$ entonces tenemos la mitad de flits del paquete, es decir, el caso del problema que nos piden.

Se podría incluir el retraso que se produce en el primer encaminador, pero no es necesario refinar tanto.

4. (1.5 puntos) Dado el algoritmo parcialmente adaptativo de encaminamiento de Primero Oeste para mallas, dibuja el grafo de dependencia de canales suponiendo una red malla 2x2 bidireccional. ¿Presenta ciclos este grafo? ¿Que conclusión puedes sacar?



No tiene ciclos. La conclusión que podemos sacar es que es bastante probable que no tenga bloqueos mortales, pero no lo podemos asegurar puesto que el teorema visto en clase sólo sirve para algoritmos deterministas. No obstante, aplicando el teorema general sí que se demuestra que esta libre de bloqueos puesto que el grafo extendido no tiene ciclos, aunque este teorema no lo hemos visto en clase.

5. (1,7 puntos) Dibuja la arquitectura de una máquina de control de flujo y explica el funcionamiento de la misma y de los diferentes componentes que la forman. Dibuja el grafo de flujo para realizar la operación $R=(A+B)*C$.

La arquitectura y funcionamiento se ha explicado en clase y había que exponerlo aquí tal cual.

*Un grafo de flujo para la operación $R=(A+B)*C$ podría ser el siguiente:*

