Arquitecturas Avanzadas (plan 1993) y Ampliación de Arquitecturas de Computadores (plan 2000)

Solución al examen de febrero de 2002

Fernando Pardo, 13 de febrero de 2002

Teoría

Las preguntas 3 y 4 han sido explicadas en clase y están en los apuntes. Las respuestas que aquí se dan no tienen por qué recoger la totalidad de lo que se pregunta, sólo se han incluido aquellas partes que requerían pensar y no se encuentran de forma explícita en la teoría.

Pregunta 1

Se tienen 5 bits para seleccionar el módulo de memoria, los dos altos no nos importan puesto que no producen ningún tipo de entrelazado (que es de lo que se trata). Por lo tanto, son los tres bits de orden bajo los que nos dan el número de módulos entrelazados que hay, por lo tanto tenemos 8 módulos entrelazados. Los accesos a la memoria se pueden ver gráficamente a continuación:

Mód. 0	Mód. 1	Mód. 2	Mód. 3	Mód. 4	Mód. 5	Mód. 6	Mód. 7
1							
						2	
				3			
		4					
5!!!							

El quinto acceso lo vuelve a hacer al mismo módulo donde empezó, por lo tanto, y suponiendo que la latencia de cada módulo es mayor de 6 ciclos, se pueden leer 4 elementos del vector de forma entrelazada sin que se produzca una detención en la memoria.

Esto mismo se podía haber razonado de otra manera: El mínimo común múltiplo del número de bancos y del paso entre elementos nos dice cuantas casillas de memoria se recorren sin conflictos al no repetir módulo. En este caso, el m.c.m. de 14 y 8 es 56, si ahora dividimos por 14 (que es el paso entre elementos) obtenemos el número de elementos (accesos) del vector que se pueden realizar sin conflicto; efectivamente: 56/14=4.

Pregunta 2

Realmente esta pregunta está contestada en teoría, si acaso comentar que es necesaria una red de tipo hipercubo, ya que en una red de este tipo existen dos nodos por dimensión de manera que en cada paso del algoritmo se consume una de las dimensiones (log n) pasando el resultado almacenado en un nodo al otro nodo de su misma dimensión. En otro tipo de redes esto no sería posible y el algoritmo no se podría implementar al faltar conexiones.

Problema

Primera parte

No hay encadenamiento ni demasiados recursos (ver enunciado). La división en convoys y los tiempos de arranque correspondientes son lo que siguen:

12	LV	V1,R1
12	LV	V2,R2
12	ADDV LV	V2,V2,V1 V3,R3
12	SV SUBV MULTV	R4,V2 V2,V2,V3 V3,V3,V1
6	SUBV	V2,V3,V1
12	MULTV SV	V1,V2,V3 R1,V2
12	SV MULTV	R2,V1 V3,V3,V4
12	SV	R3,V3

Por lo tanto hay 8 convoys y 6 instrucciones en coma flotante. El tiempo de arranque será:

Tarranque=12+12+12+12+6+12+12+12=90

Tn=|n/32|(15+90)+8n

Tn(n>>)=11.28n+105

A partir de aquí es sencillo calcular el rendimiento para vectores con un número grande de elementos:

R∞=6*100/11.28=53.1856 MFLOPS

Para calcular Nv necesitamos saber el tiempo que necesita un escalar para realizar las operaciones anteriores sobre un elemento, pero esto es un dato que nos dan, por lo tanto, el tiempo invertido por un escalar en procesar un vector de longitud n es 14n. Igualando lo que tarda el escalar por elemento y lo que tarda el vectorial se puede calcular Nv (suponemos que Nv<32):

14Nv=105+8Nv y despejando:

Nv=105/(14-8)=17.5

Así que a partir de 18 elementos el procesador vectorial es más rápido que el escalar.

Segunda parte

Ahora ya se permiten más cosas, pero hay restricciones, como el número de cauces de escritura a los registros, que son las que principalmente nos marcan el número de convoys:

12+6=18	LV	V1,R1	12
	LV	V2,R2	12
	ADDV	V2,V2,V1	6
12+7+6+12=37	LV	V3,R3	12
	SV	R4,V2	12
	SUBV	V2,V2,V3	6
	MULTV	V3,V3,V1	7
	SUBV	V2,V3,V1	6
	MULTV	V1,V2,V3	7
	SV	R1,V2	12
7+12=19	SV	R2,V1	12
	MULTV	V3,V3,V4	7
	SV	R3,V3	12

El número de convoy es de 3 y el Tarranque=18+37+19=74, con lo que el tiempo de ejecución por elemento para n grande queda:

Tn(n>>)=5.78125n+89

Y el Rendimiento cuando n tiende a infinito será:

R∞=6*100/5.78125=103.783 MFLOPS

Para calcular el $N_{1/2}$ suponemos en primer lugar que $N_{1/2}$<=32, por lo que el tiempo por elemento queda como Tn(n<=32)=3n+89. El rendimiento para $N_{1/2}$ es la mitad de R∞, por lo tanto:

$R(N_{1/2})=6*100*N_{1/2} /(3*N_{1/2} + 89)=103.783/2$

Despejando tenemos:

$N_{1/2}$ = 89*51.89/(600-3*51.89)=10.3936

Que podemos redondear a 10 que es el entero más cercano

Para calcular R_{77} tenemos que sustituir n por 77 en la fórmula de Rn y Tn. Por lo tanto tendremos:

R_{77} = (6*100*77)/(3*89+3*77)=92.77 MFlops

(No vale utilizar la fórmula de Tn cuando n es grande porque n en este caso no es grande).