

Laboratorio de Arquitectura e Ingeniería de Computadores PRÁCTICA III

COMPUTADORES VECTORIALES (DLXV)

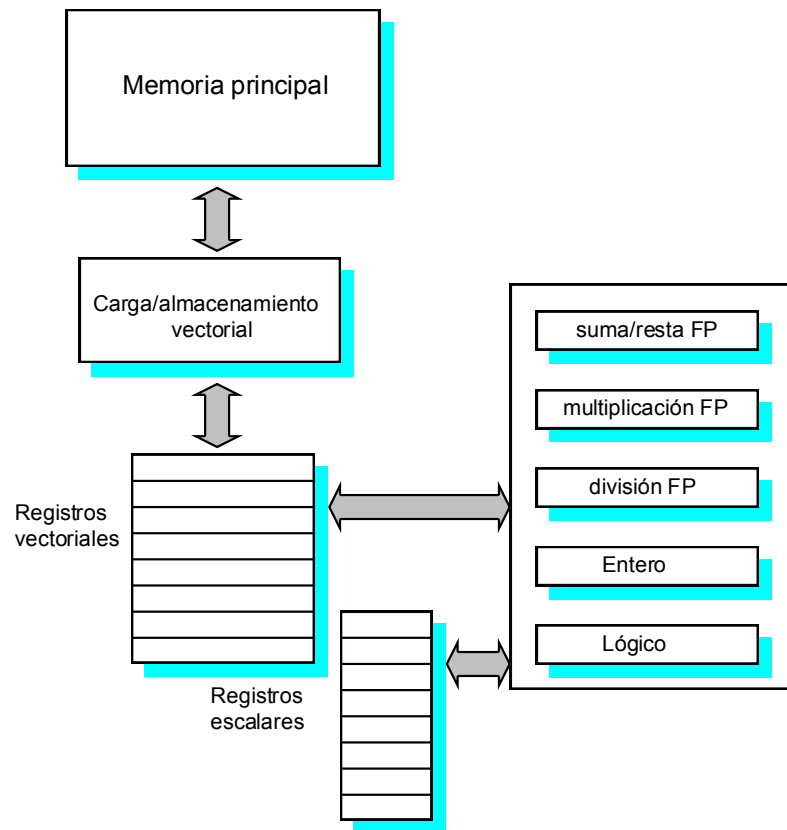
OBJETIVO

Conocer la programación de computadores vectoriales y sus mejoras de rendimiento frente a computadores escalares. Para el desarrollo de la práctica usaremos el simulador DLXV que simula el comportamiento de un computador vectorial. Realizaremos la programación de la operación DAXPY ($Y = a \cdot X + Y$ en doble precisión) tanto para programación escalar como vectorial y compararemos resultados. También se estudiará el manejo de los módulos de memoria y se realizará un programa de multiplicación de matrices con instrucciones vectoriales.

INTRODUCCIÓN TEÓRICA.

Arquitectura del computador DLXV

En la figura se muestra el diagrama de bloques del computador DLXV. Es un computador con las características escalares del DLX, ya estudiado, al que se han añadido capacidades vectoriales: registros vectoriales, unidades funcionales vectoriales y una unidad de carga/almacenamiento de vectores.



Las características de este computador son las siguientes:

- 8 registros vectoriales de 64 dobles palabras cada uno (V0 a V7).
- 5 unidades funcionales vectoriales.
- Unidad de carga/almacenamiento vectorial completamente segmentada para que puedan ser transferidas las palabras entre los registros vectoriales y la memoria con un ancho de banda de una palabra por ciclo de reloj, después de una latencia inicial..
- Conjunto de registros escalares formado por los 32 registros enteros y los 32 registros punto flotante del computador DLX.

Instrucciones vectoriales

En la siguiente tabla se describen las instrucciones vectoriales del computador

Instrucción	Operandos	Función
ADDV	V1,V2,V3	Suma los elementos de V2 y V3 y pone el resultado en V1.
ADDSV	V1,F0,V2	Suma F0 a cada elemento de V2 y pone el resultado en V1.
SUBV	V1,V2,V3	Resta los elementos de V3 a V2 y pone el resultado en V1.
SUBVS	V1,V2,F0	Resta F0 a los elementos de V2 y pone el resultado en V1.
SUBSV	V1,F0,V2	Resta a F0 los elementos de V2 y pone el resultado en V1.
MULTV	V1,V2,V3	Multiplica los elementos de V2 y V3; resultado en V1.
MULTSV	V1,F0,V2	Multiplica los elementos de V2 por F0; resultado en V1.
DIVV	V1,V2,V3	Divide los elementos de V2 por los de V3, resultados a V1.
DIVVS	V1,V2,F0	Divide los elementos de V2 por F0, resultados a V1.
DIVSV	V1,F0,V2	Divide F0 por los elementos de V2, resultados a V1.
LV	V1,A(R1)	Carga el registro vectorial V1 desde la posición de memoria indicada por A+R1.
SV	A(R1),V1	Almacena el registro vectorial V1 en memoria a partir de la posición indicada por A+R1.
LVWS	V1,R1,R2	Carga V1 desde la dirección en R1 con separación en R2.
SVWS	R1,R2,V1	Almacena V1 desde la dirección en R1 con separación en R2.
LVI	V1,R1,V2	Carga V1 con un vector cuyos elementos están en la dirección $R1+V2(i)$.
SVI	R1,V2,V1	Almacena V1 en las posiciones indicadas por la dirección $R1+V2(i)$.
CVI	V1,R1	Crea un vector de índices, esto es, almacena en V1 los valores $0, R1, 2 \cdot R1, 3 \cdot R1, \dots, 63 \cdot R1$.
S_V S_SV	V1,V2 F0,V1	Compara (EQ, NE, GT, LT, GE, LE) los operandos; si la condición se cumple, pone a 1 el bit correspondiente en el registro de mascara vectorial VM y, si no se cumple, lo pone a 0.
POP	R1,VM	Cuenta el número de 1s en el registro VM y almacena el resultado en R1.
CVM		Pone todo el registro de mascara vectorial a 1.
MOVI2S	VLR,R1	Transfiere el contenido de R1 al registro de longitud vectorial.
MOVS2I	R1,VLR	Transfiere el contenido del registro de longitud vectorial a R1.
MOVF2S	VM,F0	Transfiere el contenido de F0 al registro de máscara vectorial.
MOVS2F	F0,VM	Transfiere el contenido del registro de máscara vectorial a F0.

Tiempos en un computador vectorial

En un computador vectorial podemos distinguir dos tiempos en la ejecución de instrucciones en las unidades funcionales:

Tiempo de arranque: El que transcurre desde el comienzo hasta obtener el resultado de la primera componente; depende de la latencia de la unidad funcional.

Velocidad de iniciación: El que transcurre entre la obtención de cada una del resto de componentes después del tiempo de arranque. En computadores totalmente segmentados, como es el caso del DLXV, su valor es 1 ciclo.

Los tiempos de arranque de las unidades funcionales del DLXV se muestran en la siguiente tabla. Si dos operaciones consecutivas son dependientes se debe añadir al tiempo de arranque 4 ciclos.

Operación	Arranque
Suma vectorial	6
Multiplicación vectorial	7
División vectorial	20
Carga vectorial	12

NOTA: Los valores mostrados corresponden a los que el simulador DLXV asigna por defecto. Se pueden alterar a voluntad mediante la opción "Configuración".

Las cargas vectoriales se realizan desde la memoria dividida en bancos, en los que se realiza una carga simultánea de posiciones de memoria consecutivas, de forma que se pueda suministrar una componente del vector por cada ciclo después del tiempo de acceso a memoria (12 ciclos). Para que esto sea posible el número de bancos debe ser superior al tiempo de acceso. En el caso de que las componentes del vector no estén en posiciones consecutivas de memoria se pueden dar colisiones cuando a dos componentes del vector les corresponde el mismo banco de memoria produciéndose retardos por este motivo. La solución a este problema es aumentar lo suficiente los bancos de memoria o disponer su número de forma que se minimice el número de colisiones.

DESARROLLO DE LA PRÁCTICA

- 1.- Realizar un programa que calcule la operación $Y = a \cdot X + Y$ donde X e Y son vectores de 100 elementos y a un escalar, primero con instrucciones escalares y después con instrucciones vectoriales. Comparar los resultados obtenidos. Hacer un estudio de las estadísticas obtenidas en el programa vectorial. (Todos los operandos deben ser de doble precisión).
- 2.- Realizar la parte vectorial del apartado 1 cambiando el número de bancos de memoria y, con vectores cuyas componentes no estén almacenadas en memoria de forma consecutiva, analizar resultados. Realizar las pruebas y analizar los resultados para las siguientes situaciones:

Bancos de memoria	Distancia en memoria entre componentes
64 bancos	Posiciones consecutivas
	.space 8
	.space 24
32 bancos	Posiciones consecutivas
	.space 8
	.space 24
16 bancos	Posiciones consecutivas
	.space 8
	.space 24

- 3.- Realizar un programa que multiplique dos matrices de 10 x 10 usando instrucciones vectoriales.

Sugerencia: Obsérvese que se verifica la siguiente igualdad:

$$(c_{i1}, \dots, c_{in}) = \sum_{j=1, n} a_{ij} \cdot (b_{j1}, \dots, b_{jn})$$

donde A , B son las matrices a multiplicar, C la matriz producto y n el orden de las matrices. Por lo tanto, se trata de reiterar una operación del tipo DAXPY como la comentada en el principio de la práctica.