

Problema

Dispone de un procesador vectorial VECTOR-A con las siguientes características:

- ⤴ Tbase = 10 ciclos.
- ⤴ Tbucle = 15 ciclos.
- ⤴ Dos unidades de carga-almacenamiento vectorial con 12 ciclos de tiempo de arranque,
- ⤴ Una unidad de suma vectorial en coma flotante con tiempo de arranque de 6 ciclos
- ⤴ Unidad de multiplicación vectorial en coma flotante con tiempo de arranque de 7 ciclos.

y de un segundo procesador vectorial, denominado VECTOR-B, en el que, con la salvedad del número de unidades de carga/almacenamiento, se dobla el número de unidades aritméticas y, como consecuencia del incremento de la complejidad en las operaciones de asignación a las unidades, los gastos. Se tiene así:

- ⤴ Tbase = 20 ciclos.
- ⤴ Tbucle = 30 ciclos.
- ⤴ Dos unidades de carga-almacenamiento vectorial con 24 ciclos de tiempo de arranque,
- ⤴ Dos unidades de suma vectorial en coma flotante con tiempo de arranque de 12 ciclos
- ⤴ Dos unidades de multiplicación vectorial en coma flotante con tiempo de arranque de 14 ciclos.

Con el fin de realizar una comparativa, en ambos procesadores se van a ejecutar las dos siguientes secuencias de código:

```
// Secuencia de código 1
for (i=0; i<10000; i++) {
    A[i+1] := x*A[i] + y*A[i];
}

// Secuencia de código 2
for (i=0; i<100; i++) {
    A[i+1] := x*A[i];
}
```

Se pide

- a) Escriba el código vectorial que corresponde a las dos secuencias de código.
- b) Calcule el número de ciclos consumidos por la secuencia 1 en el procesador VECTOR-A.
- c) Calcule el número de ciclos consumidos por la secuencia 1 en el procesador VECTOR-B.
- d) Calcule el número de ciclos consumidos por la secuencia 2 en el procesador VECTOR-A.
- e) Calcule el número de ciclos consumidos por la secuencia 2 en el procesador VECTOR-B
- f) Compare y explique, razonadamente, las diferencias obtenidas al realizar ejecución de la secuencia 1 y la 2 en ambos procesadores.

Solución

a) El código correspondiente a la secuencia de código 1 es

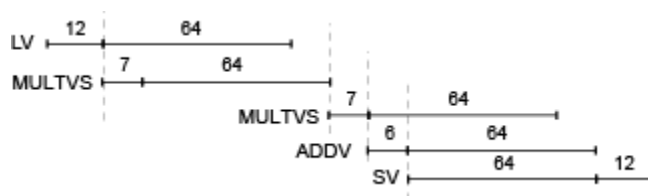
```
LV      V1, A(i)
MULTVS  V2, Rx, V1
MULTVS  V3, Ry, V1
ADDV    V4, V2, V3
SV      V4, A(i+1)
```

y el correspondiente a la secuencia de código 2 es:

```
LV      V1, A(i)
MULTVS  V2, x, V1
SV      V2, A(i+1)
```

donde $A(i)$ y $A(i+1)$ representan las posiciones de memoria en las que se ubican esos elementos y R_x y R_y corresponden a los registros en los que se almacenan los valores de x e y .

b) Tal y como se puede apreciar en el siguiente esquema, el total de ciclos consumidos es 172 ciclos para un vector de 64 elementos considerando que el encadenamiento de resultados entre unidades es posible.



El *Telemento* es de 2 ciclos dado que se tienen dos convoyes. El *Tarranque* total se obtiene de sumar los tiempos de arranque visibles de las unidades funcionales. Si se analiza el esquema previo se tiene

$$\text{Tarranque} = \text{Tarranque LV} + 2 \cdot \text{Tarranque MULTVS} + \text{Tarranque ADDV} + \text{Tarranque SV}$$

$$\text{Tarranque} = (12 + 2 \cdot 7 + 6 + 12) \text{ ciclos} = 44 \text{ ciclos}$$

Con estos valores y dado que el tiempo base es 10, la expresión del tiempo total de ejecución del bucle queda

$$T_{10000} = 10 + \left\lceil \frac{10000}{64} \right\rceil \cdot (15 + 44) + 2 \cdot 10000 = 29273 \text{ ciclos}$$

c) Tal y como se puede apreciar en el siguiente esquema, el total de ciclos consumidos es 138 ciclos para un vector de 64 elementos considerando que el encadenamiento de resultados entre unidades es posible.



El *Telemento* es de 1 ciclo dado que se tiene un único convoy. El *Tarranque* total se obtiene de sumar los tiempos de arranque visibles de las unidades funcionales. Si se analiza el esquema previo se tiene

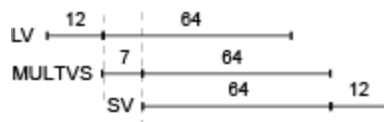
$$\text{Tarranque} = \text{Tarranque LV} + \text{Tarranque MULTVS} + \text{Tarranque ADDV} + \text{Tarranque SV}$$

$$\text{Tarranque} = (24 + 14 + 12 + 24) \text{ ciclos} = 74 \text{ ciclos}$$

Con estos valores y dado que el tiempo base es 20, la expresión del tiempo total de ejecución del bucle interior queda

$$T_{10000} = 20 + \left\lceil \frac{10000}{64} \right\rceil \cdot (30 + 74) + 1 \cdot 10000 = 26348 \text{ ciclos}$$

d) Tal y como se puede apreciar en el siguiente esquema, el total de ciclos consumidos es 95 ciclos para un vector de 64 elementos considerando que el encadenamiento de resultados entre unidades es posible.



El *Telemento* es de 1 ciclo dado que se tiene un único convoy. El *Tarranque* total se obtiene de sumar los tiempos de arranque visibles de las unidades funcionales. Si se analiza el esquema previo se tiene

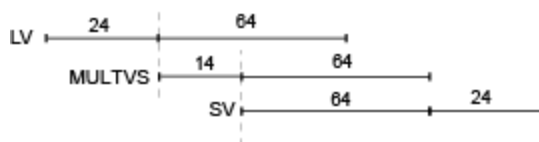
$$Tarranque = Tarranque \text{ LV} + Tarranque \text{ MULTVS} + Tarranque \text{ SV}$$

$$Tarranque = (12 + 7 + 12) \text{ ciclos} = 31 \text{ ciclos}$$

Con estos valores y dado que el tiempo base es 10, la expresión del tiempo total de ejecución del bucle interior queda

$$T_{100} = 10 + \left\lceil \frac{100}{64} \right\rceil \cdot (15 + 31) + 1 \cdot 100 = 202 \text{ ciclos}$$

e) Tal y como se puede apreciar en el siguiente esquema, el total de ciclos consumidos es 126 ciclos para un vector de 64 elementos considerando que el encadenamiento de resultados entre unidades es posible.



El *Telemento* es de 1 ciclo dado que se tiene un único convoy. El *Tarranque* total se obtiene de sumar los tiempos de arranque visibles de las unidades funcionales. Si se analiza el esquema previo se tiene

$$Tarranque = Tarranque \text{ LV} + Tarranque \text{ MULTVS} + Tarranque \text{ SV}$$

$$Tarranque = (24 + 14 + 24) \text{ ciclos} = 62 \text{ ciclos}$$

Con estos valores y dado que el tiempo base es 20, la expresión del tiempo total de ejecución del bucle interior queda

$$T_{100} = 20 + \left\lceil \frac{100}{64} \right\rceil \cdot (30 + 62) + 1 \cdot 100 = 304 \text{ ciclos}$$

f) Se puede apreciar que para la secuencia 1 el VECTOR-A es más rápido que el VECTOR-B mientras para la secuencia 2 sucede lo contrario, el VECTOR-A es el más lento. Es decir, que las diferencias entre rendimiento máximo y rendimiento sostenido pueden ser grandes dependiendo del problema a procesar.

A mayor tamaño del vector de datos a procesar, los tiempos por gastos de sobrecarga pesan menos en el tiempo total de procesamiento ya que quedan difuminados por el tiempo consumido en el procesamiento de las operaciones vectoriales, teniendo más impacto el incremento del número de unidades funcionales.