

IC Unidad 1: Problemas arquitecturas vectoriales

Ejercicio 1

Entrelazado inferior con 8 módulos de memoria → 2^3 bloques → 3 bits para direccionar bloques: **m=3**.

Resto de bits para direccionar posiciones de memoria dentro de los módulos: **n=resto de bits**

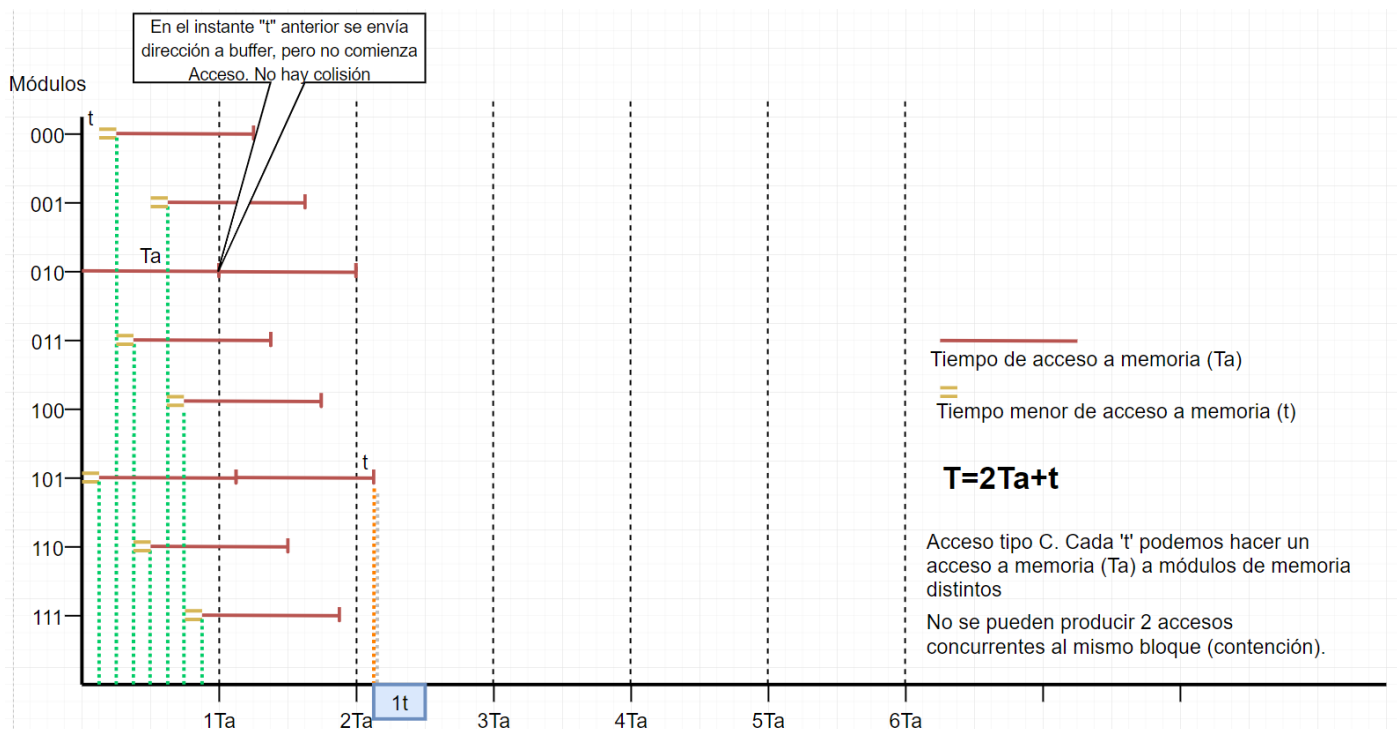
Dirección inicio: 01C2h → 0000 0001 1100 0**010**

Almacenamiento del vector V en memoria a partir de la dirección de inicio con stride 3 (elementos del vector almacenados cada 3 posiciones, 2 huecos entre cada componente):

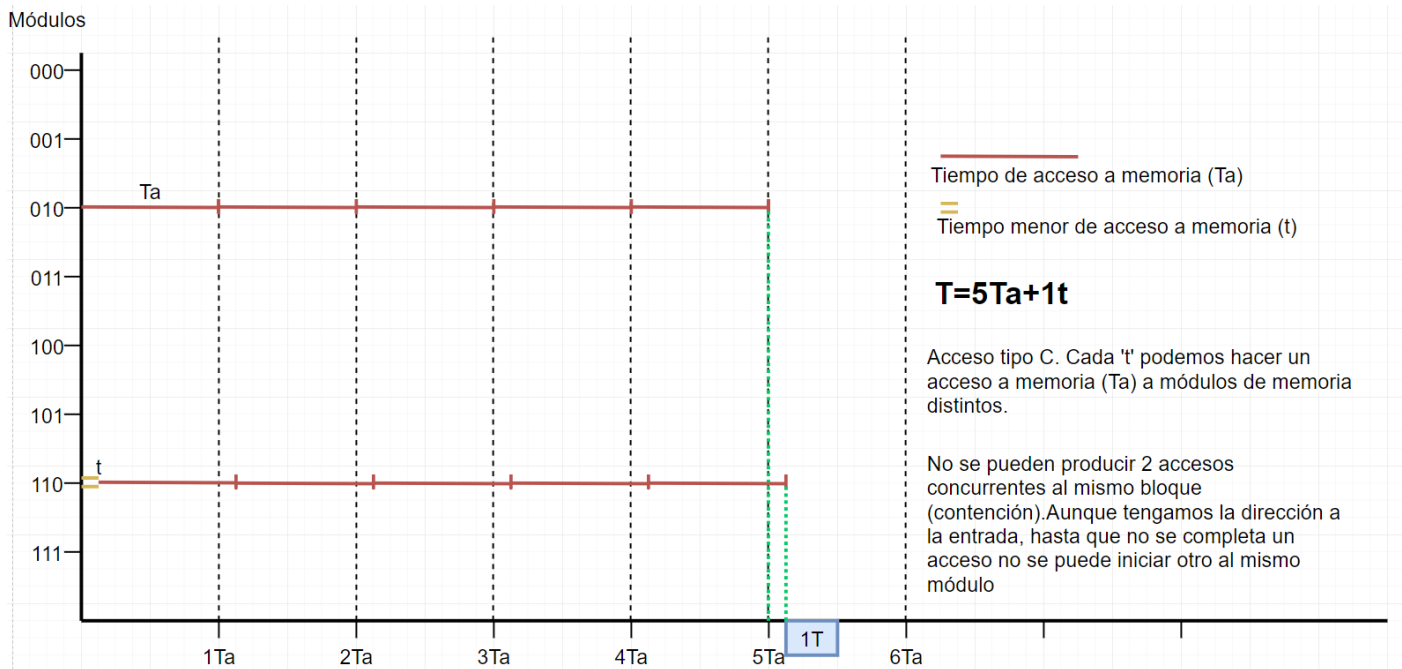
DIRECCIONES	MÓDULOS MEMORIA							
	M0 000	M1 001	M2 010	M3 011	M4 100	M5 101	M6 110	M7 111
0000 0001 1100 0			V0			V1		
0000 0001 1100 1	V2			V3			V4	
0000 0001 1101 0		V5			V6			V7
0000 0001 1101 1			V8			V9		

- Apartado a) Acceso simultáneo: $T = 4T_a + 2t$

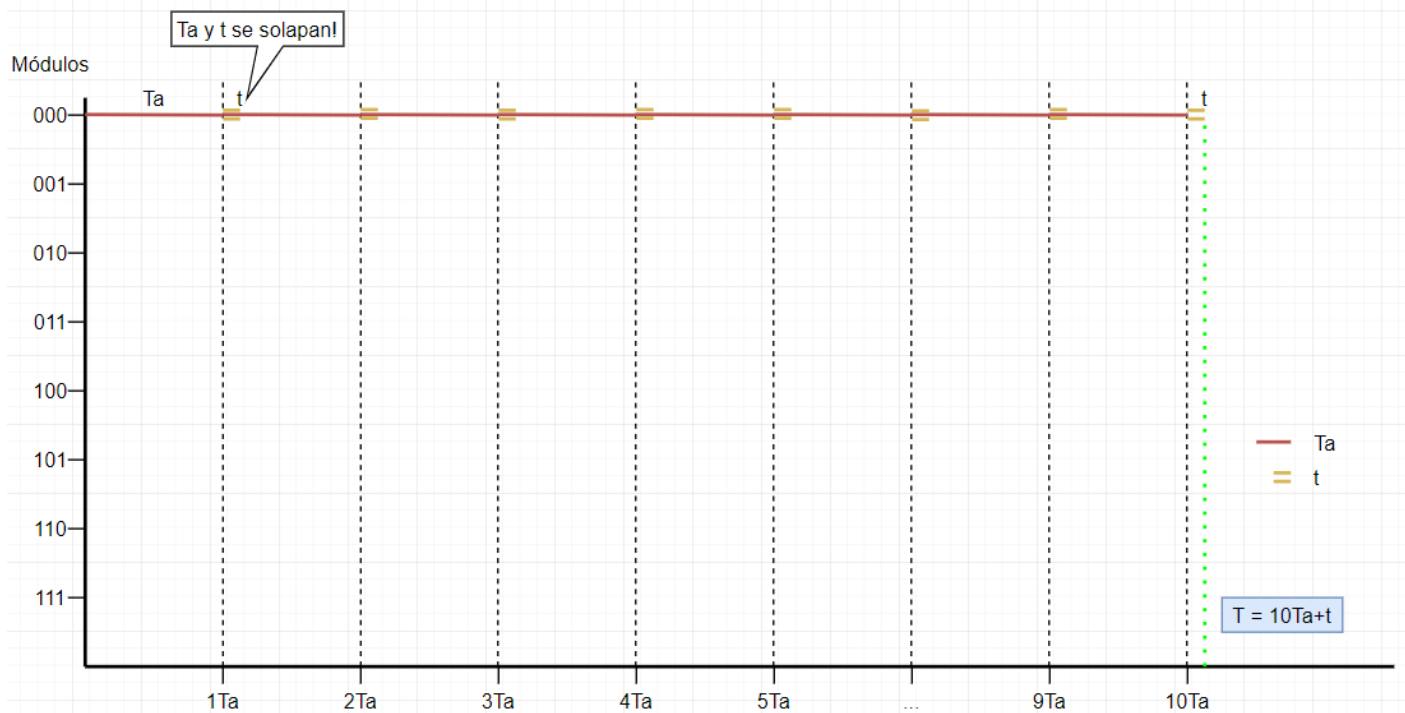
- Apartado b) Acceso concurrente:



- Apartado b), cómo sería con stride 4 (3 posiciones de memoria entre elementos del vector)



- Apartado c)



Entrelazado superior

Posiciones consecutivas de memoria estarán ubicadas dentro del mismo módulo. Si la dirección de comienzo de almacenamiento del vector es: 01C2h = 0000 0001 1100 0010
Los 3 bits de mayor peso marcarán el módulo de memoria dónde está almacenada la 1a posición del vector. En este caso 000, por tanto el módulo 0, en la dirección 0 0001 1100 0010. A partir de esa, siguiendo el stride marcado por el enunciado, estarán almacenadas las posiciones restantes del vector.

Acceso tipo S

Recuperamos posiciones de memoria de todos los bloques a la vez en cada acceso (T_a), pero de cada bloque solo se puede recuperar una palabra por cada acceso. Tras cada acceso tardamos un tiempo menor (t) en poner el dato recuperado en el bus de datos y completar la recuperación.

Ejercicio 2

Latencias:

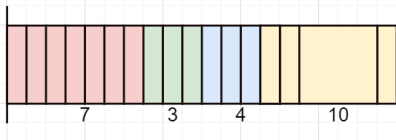
- Cargas: 7 ciclos
- Sumas: 3 ciclos
- Desplazamientos: 4 ciclos

Código ejemplo:

- $V0 = \text{load (mem)}$
- $V2 = V0 + V1$
- $V3 = V2 \ll A1$

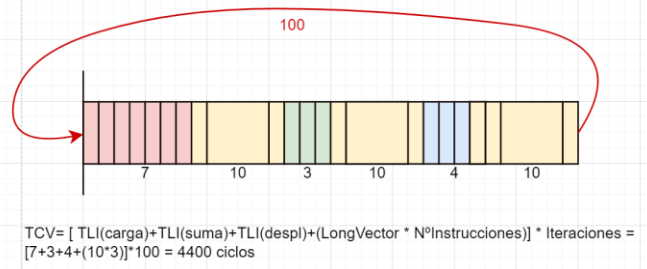
a)

$TCV = TLI(\text{carga}) + TLI(\text{suma}) + TLI(\text{despl}) + \text{LongVector} = 7 + 3 + 4 + 10 = 24$ ciclos



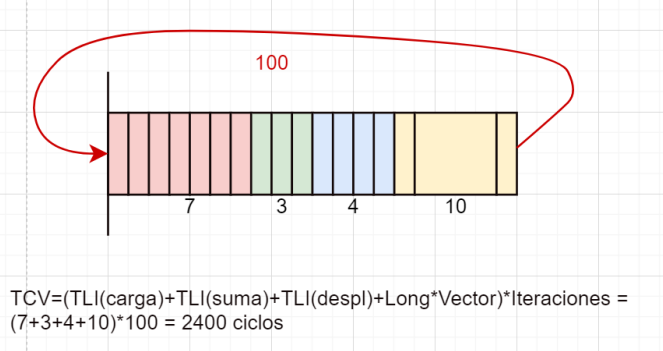
b) Nos piden la ganancia por lo que habrá que estimar el número de ciclos empleado en el caso sin encadenamiento frente al caso en el que sí disponemos de un cauce vectorial con encadenamiento. Se indica que el código anterior están dentro de un bucle que se repetirá en 100 ocasiones por lo que a partir de los TCV iniciales (como el calculado en 2.a) obtendremos los de este caso.

TCV para el cauce sin encadenamiento iterando 100 veces:



$TCV = [TLI(\text{carga}) + TLI(\text{suma}) + TLI(\text{despl}) + (\text{LongVector} * N^{\circ} \text{Instrucciones})] * \text{Iteraciones} = [7 + 3 + 4 + (10 * 3)] * 100 = 4400$ ciclos

TCV para el cauce con encadenamiento iterando 100 veces:



$TCV = (TLI(\text{carga}) + TLI(\text{suma}) + TLI(\text{despl}) + \text{Long} * \text{Vector}) * \text{Iteraciones} = (7 + 3 + 4 + 10) * 100 = 2400$ ciclos

Ganancia:

$$G = \frac{T_0}{T_{enc}} = \frac{4400}{2400} = 1.83$$