

## Acceso a memoria.

### Ejercicio 1.

Se requiere sumar 2 vectores de 10 elementos, ordenados de forma consecutiva desde la dirección: 01C2 HEX.

$$A[10] + B[10].$$

Cuántos accesos se debe hacer? Cuándo duraría?

- Acceso de tipo S.
- 8 Bloques.
- Entrelazado INF = [POS][MOD]
- Datos empiezan en: 01C2 HEX.

#### Pasos a seguir:

1) Pasamos de HEX BIN la dirección de inicio.

0 = 0000

1 = 0001

C = 1100

2 = 0010

Dirección de inicio: 0000 0001 1100 0010

2) Cuántos Bloques hay? 8. Cuántos bits necesitamos para direccionar 8 Modulos?  
 $\log_2(8) = 3$ .

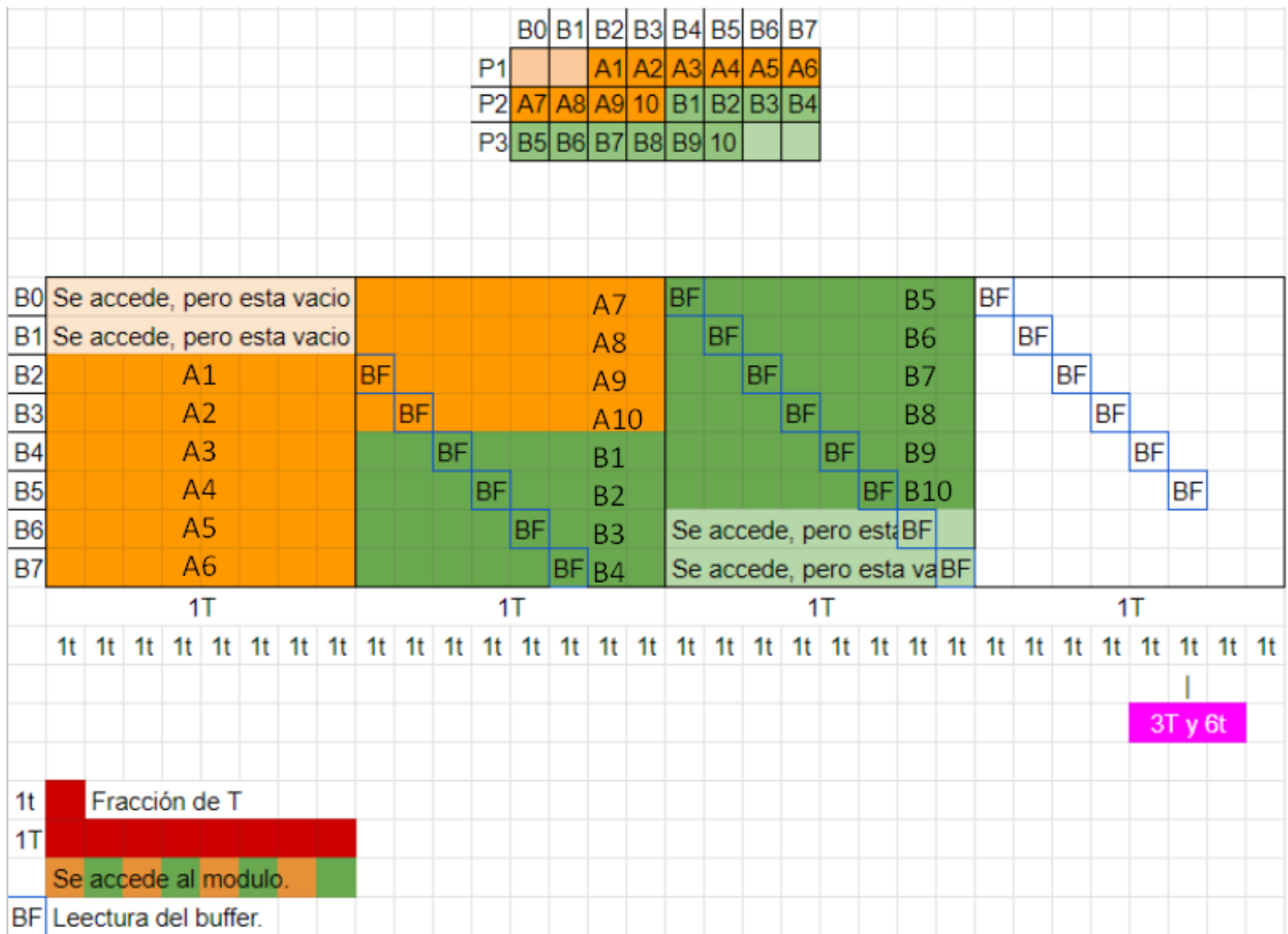
El entrelazado al ser inferior, significa que escogeremos los ultimos 3 bits para direccionar ( 010 ), y el resto ( 00000001110000 ) para especificar una posicion dentro de cada Modulo.

3) Modulo 010 = N° 2 (de 0 a 2 ).

Entonces, nuestros bloques tendrian este aspecto:

	B0	B1	B2	B3	B4	B5	B6	B7
P1			A1	A2	A3	A4	A5	A6
P2	A7	A8	A9	10	B1	B2	B3	B4
P3	B5	B6	B7	B8	B9	10		

- 4) Accedemos de manera Simultanea como nos dice el ejercicio, accedemos a todos los bloques a la vez, iterando posicion por posicion. Hacemos un esquema.



- 5) *Cuántos accesos se debe hacer? Cuándo duraría?*

Accesos: Se ha accedido 3 veces a memoria.

Duración: 3T y 6t.



## Ejercicio 1 - Parte 3.

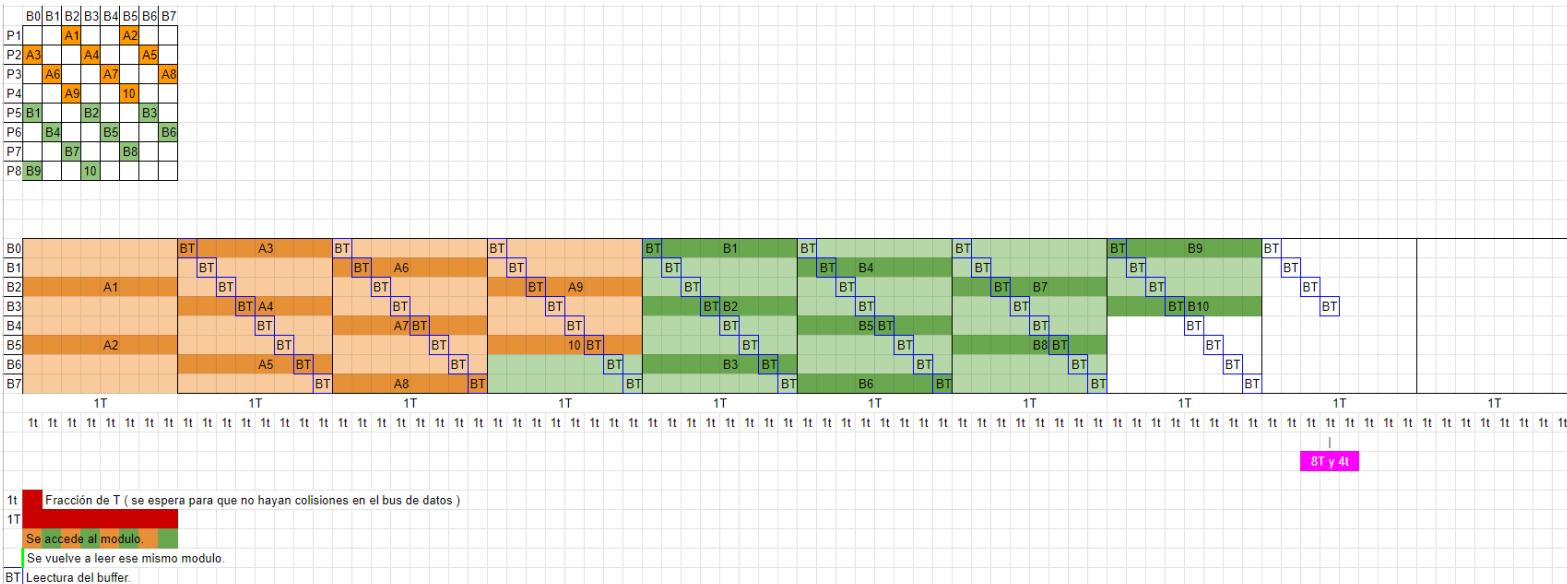
Si los elementos de los vectores se almacenan con  $\text{stride} = 3$ , cual seria el tiempo con Tipo C?

2 STRIDES es la separación DE 1 POS, entre 1 posicion y otra concurrentes de datos, en un bloque.

Stride = 3.

	B0	B1	B2	B3	B4	B5	B6	B7
P1			A1			A2		
P2	A3			A4			A5	
P3		A6			A7			A8
P4			A9			10		
P5	B1			B2			B3	
P6		B4			B5			B6
P7			B7			B8		
P8	B9			10				

4) Accedemos de manera Simultanea como nos dice el ejercicio, accedemos a todos los bloques a la vez, iterando posicion por posicion. Hacemos un esquema.



Accesos: Se ha accedido 8 veces a memoria.

Duración: 8T y 4t.

**El valor de tiempo final no es correcto. debería ser  $8T_a + 2t$ . Si te fijas, en el último acceso a memoria sólo quedan por recuperar 2 posiciones del vector B, que recuperamos de bloques distintos (B0 y B3). Por lo tanto, al terminar el acceso a memoria a dichos bloques, sólo tendremos que leer 2 buffers de salida, por lo que sólo necesitaremos 2t, un t por cada buffer de salida de bloque al que hemos accedido.**

## Ejercicio 2.

Se dispone de una arquitectura vectorial en la que se utiliza memoria entrelazada con orden inferior y a la que se accede de forma simultánea.

Se quiere evaluar el tiempo de acceso en operaciones de tipo Vector Escalar como:

MULTSV V2,F0,V1 que realiza  $V_j = f \times V_i$ .

La memoria se estructura en 8 bloques y el vector operando V1, de 10 elementos, se encuentra almacenado con un Stride = 3 a partir de la dirección 01C2 HEX.

*¿Cuántos accesos a memoria serán necesarios para recuperar el contenido del vector V1?*

- Acceso de tipo S.
- 8 Bloques.
- Entrelazado INF.
- Datos empiezan en 01C2 HEX.

**Pasamos de HEX BIN la dirección de inicio.**

0 = 0000

1 = 0001

C = 1100

2 = 0010

Dirección de inicio: 0000 0001 1100 0010

**Cuántos Bloques hay? 8. Cuántos bits necesitamos para direccionar 8 Modulos?**

$\log_2(8) = 3$ .

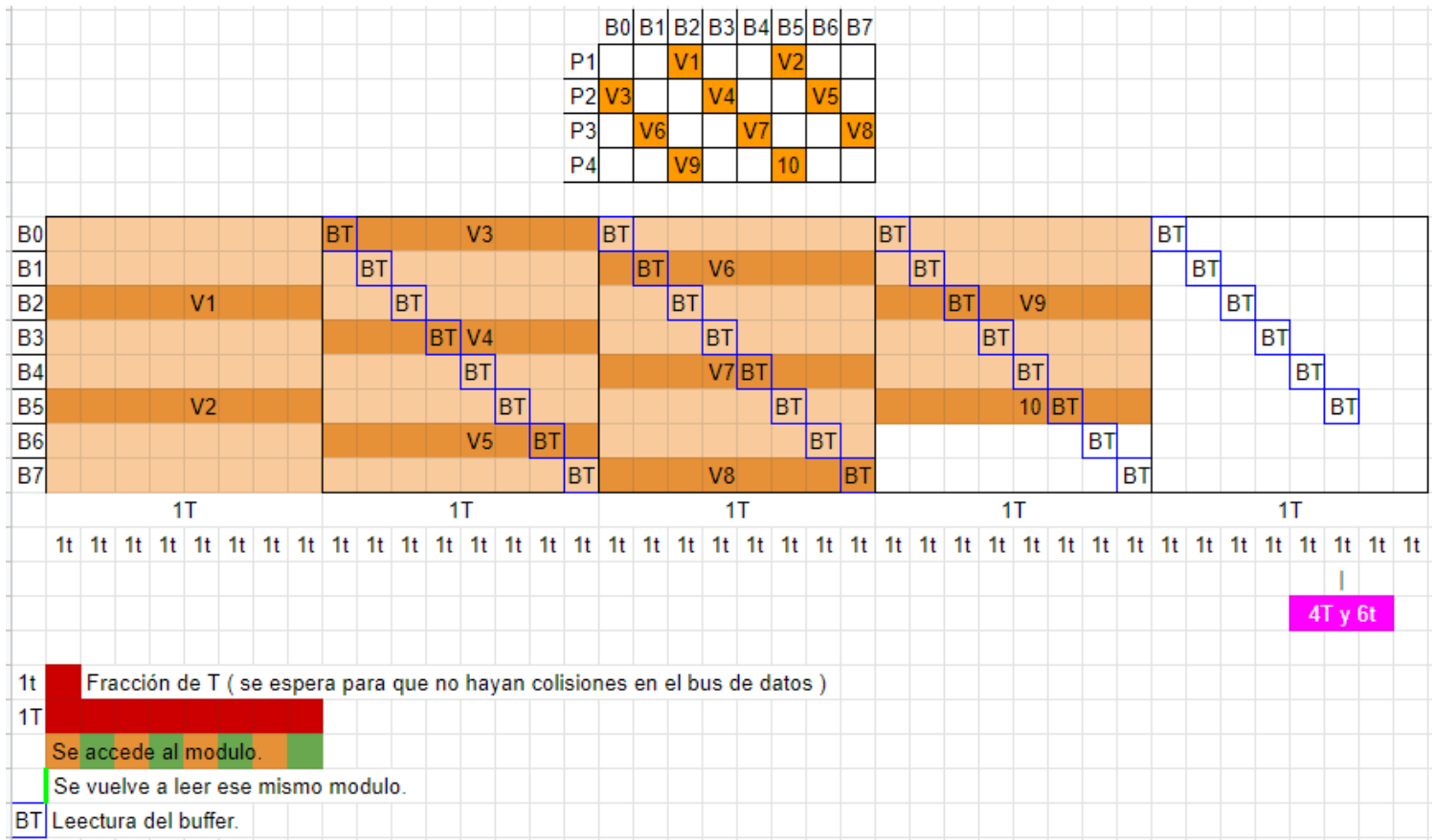
**El entrelazado al ser inferior, significa que escogeremos los ultimos 3 bits para direccionar ( 010 ), y el resto ( 00000001110000 ) para especificar una posicion dentro de cada Modulo.**

Modulo 010 =  $N^{\circ} 2$  (de 0 a 2 ).

**Entonces, nuestros bloques tendrian este aspecto:**

	B0	B1	B2	B3	B4	B5	B6	B7
P1			V1			V2		
P2	V3			V4			V5	
P3		V6			V7			V8
P4			V9			10		

Accedemos de manera Simultanea como nos dice el ejercicio, accedemos a todos los bloques a la vez, iterando posicion por posicion. Hacemos un esquema.



¿Cuántos accesos a memoria serán necesarios para recuperar el contenido del vector V1?

Accesos: Se ha accedido 4 veces a memoria.

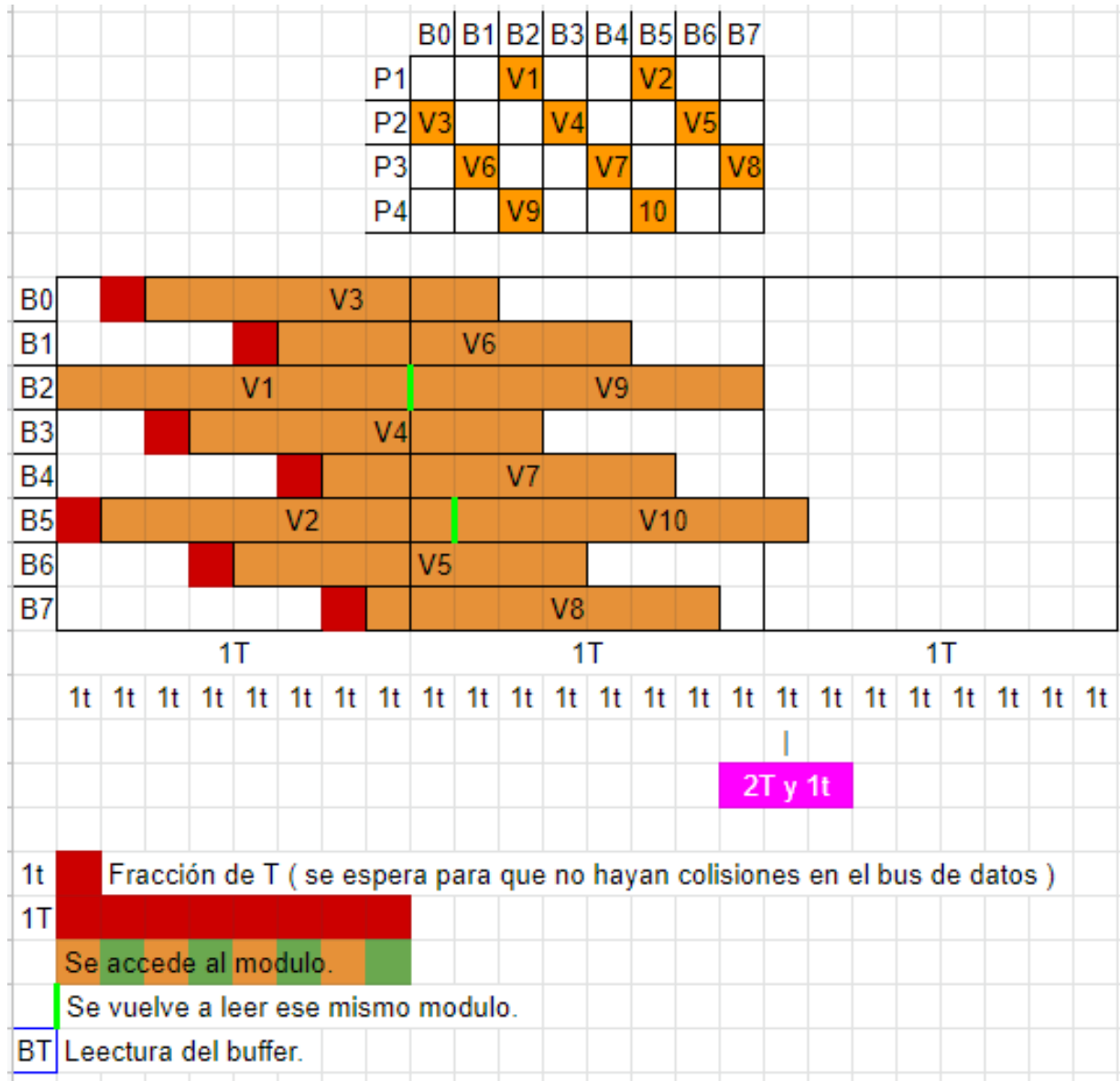
Duración: 4T y 6t.

**Ocurre lo mismo que en el anterior, el tiempo final es  $T = 4T_a + 2t$**

## Ejercicio 2 - Parte 2.

*¿Y si el acceso fuera de tipo concurrente?*

Accedemos de forma Concurrente. Accedemos a cada modulo de forma concurrente y consecuente ( uno detras de otro ). Iterando de Modulo a Modulo. Hacemos un esquema.



Accesos: Se ha accedido 2 veces a memoria.

Duración: 2T y 1t.

## Ejercicio 2 - Parte 3.

*Razonar las implicaciones del cambio de tipo de entrelazado a orden superior.*

Si el entrelazado fuera de orden superior, tanto en tipo C como S, en vez de leer una posición en concreto de todos los bloques a la vez, podríamos leer un bloque entero, ya que los datos estarían ordenados de forma consecutiva.

No es correcto. Está bien que hayas razonado como se distribuirían en ese caso las posiciones contiguas de memoria. Sin embargo, lo que dices no se cumple. En el acceso C nunca leeremos una misma posición de todos los bloques a la vez, porque siempre habrá un intervalo "t" entre 2 accesos seguidos. Además, en ningún caso podremos leer un bloque entero a la vez, sea con acceso S o C, por que recuerda que eso supondrían colisiones ya que no podemos realizar 2 accesos al mismo bloque al tiempo. Hay que esperar que acabe uno para que pueda producirse. Para verlo te sugiero que repitas los apartados 1 y 2 pero con la memoria dividida con entrelazado superior.

Por otra parte , en cuanto al formato, en la tabala donde se muestra como están almacenados los vectores en memoria, es útil incluir la parte de la dirección (bits) para direccionar los bloques y las posiciones de memoria dentro de cada bloque.

## Ejercicio 2

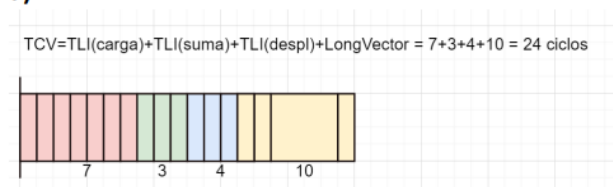
Latencias:

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

Código ejemplo:

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

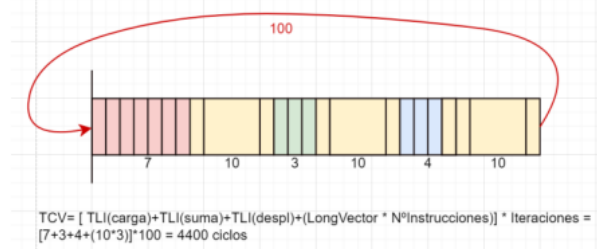
a)



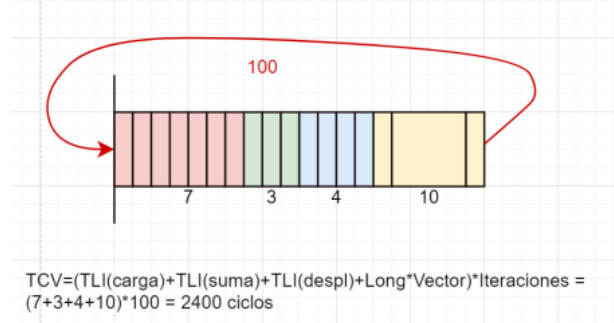


**b)** Nos piden la ganancia por lo que habrá que estimar el número de ciclos empleado en el caso sin encadenamiento frente al caso en el que sí disponemos de un cauce vectorial con encadenamiento. Se indica que el código anterior están dentro de un bucle que se repetirá en 100 ocasiones por lo que a 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 para el cauce con encadenamiento iterando 100 veces:



Ganancia:

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