

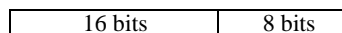


## Arquitectura de Computadores I

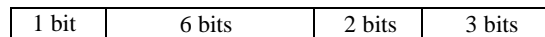
### Sistema de memoria 3 (Solución): Segmentación + Bancos

En un sistema de memoria, en el que el direccionamiento se realiza a nivel de byte, las direcciones de acceso a memoria tienen la siguiente estructura:

- Memoria virtual segmentada:



- Memoria principal:



**Contesta a las siguientes preguntas:**

- Esquema de traducción de direcciones. ¿Cuántos segmentos puede tener un programa? ¿Cuál es el tamaño máximo de segmento? ¿Cuántas entradas tendrá la tabla de segmentos? ¿Cuál es el tamaño de cada entrada? En el caso de la existencia de un TLB, ¿de qué tamaño es cada una de sus entradas?
- Esquema de la memoria principal. ¿Cuántas palabras tiene? ¿Cuál es el tamaño en bytes de una palabra? ¿Cómo está organizada la memoria? En caso de estar formada por varios módulos, ¿cuál es el tamaño de un módulo?
- En un instante determinado, el procesador lanza la siguiente secuencia de referencias lógicas: 148, 412, 1496 y 428. El sistema dispone de un TLB para la traducción de direcciones, inicialmente vacío, con un tiempo de acceso de 30 ciclos en caso de fallo y 1 ciclo en caso de acierto. ¿Cuánto tiempo es necesario para la traducción de esa secuencia de referencias? Teniendo en cuenta el contenido de la tabla de segmentos indicada a continuación, ¿cuánto tiempo es necesario para acceder a memoria principal para dicha secuencia de referencias? El tiempo de acceso a memoria principal es de 10 ciclos (1 ciclo desde el buffer de entrelazado).

Tabla de segmentos: segmento	10	5	1	2
@base	3000	600	2700	100

- ¿Cuál será el tiempo de traducción para todas las referencias de este programa si está formado por 4 segmentos (instrucciones, vector A, vector B y vector C)?

```

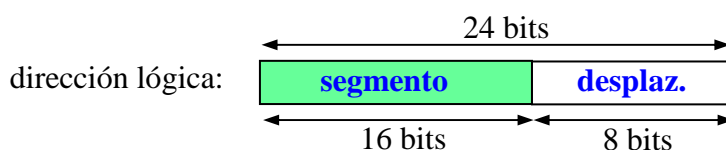
movi r1,#16
movi r10,#4320
movi r6,#2
buc: load r2,B[r1-16]
      load r3,B[r1]
      load r4,B[r1+16]
      add r2,r2,r3
      add r2,r2,r4
      load r5,[r10]
      divi r2,r2,r5
      store r2,64[r1]
      addi r1,r1,#8
      addi r6,r6,#1
      subi r7,r6,#511
      bnz r7,buc
for (i=2; i<52; i++)
    A[i] = (B[i-2]+B[i]+B[i+2]) / C[0];

```

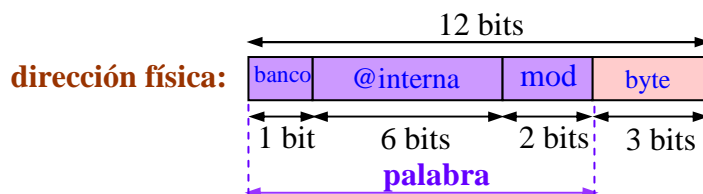
## Solución

En este ejercicio podemos deducir las características del sistema de memoria a partir de los esquemas de direccionamiento lógico y físico que proporciona el enunciado. En primer lugar, se indica que el **direccionamiento** del sistema de memoria se realiza a nivel de **byte**.

En lo que respecta a la **memoria virtual**, se indica la división en bits de la dirección lógica. También se indica que el sistema es **segmentado**. Por ello, los dos campos presentes en la dirección lógica son los siguientes: los bits que indican el **segmento**, y el **desplazamiento** dentro del segmento del byte al que se quiere acceder. Por lo tanto, la dirección lógica se interpreta de la siguiente forma:



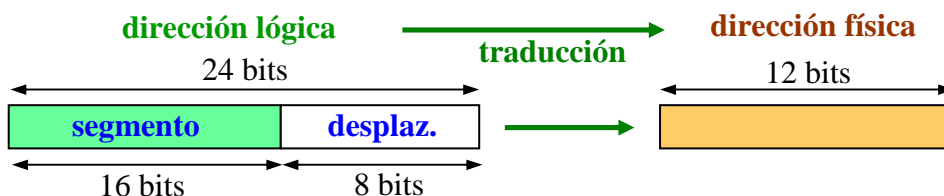
En lo que respecta a la **memoria principal**, el enunciado indica la división en bits de la dirección física. Dado que esta dirección se divide en cuatro campos, existe una única posibilidad para la organización de la memoria principal: se trata de **bancos consecutivos de módulos entrelazados**. La razón es sencilla: en los otros dos esquemas posibles (módulos consecutivos o módulos entrelazados), la dirección de memoria presenta sólo tres campos. Una vez determinada la organización de la memoria principal, podemos decir que el significado de los cuatro campos es el siguiente, de derecha a izquierda: los bits que indican el **byte** dentro de la palabra, los bits que indican el **módulo** entrelazado, los bits que indican la **dirección interna** en el módulo a nivel de palabra y, por último, los bits que indican el **banco** consecutivo. En definitiva, la interpretación de la dirección física es la siguiente:



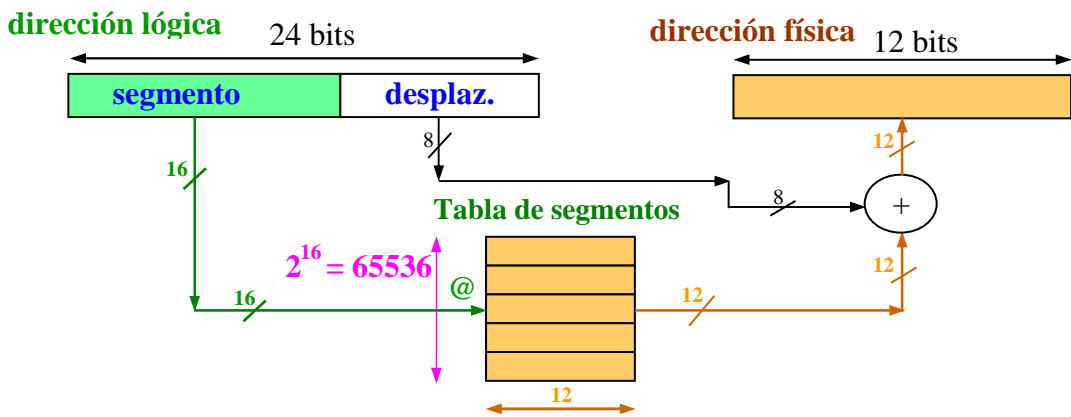
Esta información es suficiente para contestar las preguntas del enunciado.

- (a) Hay que dibujar el esquema de traducción de direcciones, indicando cómo se utilizan los bits de los distintos campos de la dirección.

En lo que se refiere al proceso de traducción, sabemos que los bits que indican el segmento son los que se utilizan para acceder al TLB o a la tabla de segmentos y obtener la dirección base de segmento, esto es, una dirección de memoria principal: un segmento se carga en memoria a partir de una dirección cualquiera de memoria principal. A esta dirección base hay que sumar el desplazamiento para poder obtener la dirección del byte de memoria principal al que se quiere acceder. En definitiva, éste es el proceso de traducción:

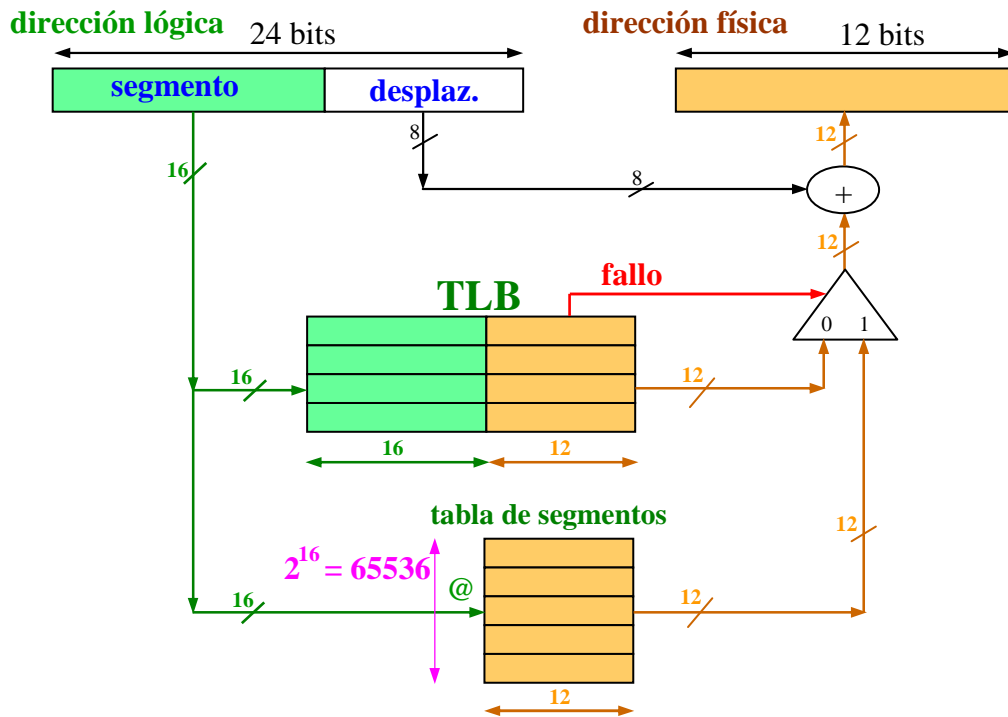


A continuación, se presenta el esquema del hardware encargado de realizar la traducción “dirección lógica → dirección física”:



La tabla de segmentos es una memoria RAM a la que se accederá utilizando los 16 bits que indican el segmento en la dirección lógica. Por ello, la **tabla de segmentos** tendrá  $2^{16} = 65536$  posiciones o **entradas** (es decir, una por cada segmento, dado a que a cada segmento le corresponde una entrada en la tabla de segmentos). Por otra parte, cada entrada o posición de la tabla de segmentos debe almacenar la dirección base del segmento (una dirección de memoria física), por lo que el **tamaño de cada entrada** será de **12 bits**.

En caso de la existencia de un TLB para la traducción de direcciones, el esquema hardware completo es el siguiente:



El TLB es una memoria asociativa a la que se accede por contenido. Este contenido está formado por dos partes: (a) los 16 bits que indican el segmento que hay que buscar, y (b) los 12 bits que indican la dirección base en la que se ha cargado ese segmento en memoria principal. Esta segunda parte, que es que la que obtiene tras la búsqueda en el TLB, se incorpora al sumador para sumarla con el desplazamiento y obtener la dirección física del byte al que se quiere acceder. Por ello, **el tamaño una cada entrada en el TLB** es de **28 bits** (16 bits + 12 bits).

A partir de información obtenida de los campos de la dirección lógica, podemos contestar las preguntas de este apartado. Por una parte, dado que se utilizan 24 bits para indicar una dirección lógica, podemos decir que la **memoria virtual es de 8 MB** ( $2^{24}$ ). Por otra parte, dado que se utilizan 16 bits para indicar un segmento, podemos deducir que un **programa** puede tener un máximo de **65536 segmentos** ( $2^{16}$ ). Por último, dado que para indicar un desplazamiento se utilizan 8 bits, podemos decir que el **tamaño máximo de segmento** es de **256 bytes** ( $2^8$ ).

- (b) En este apartado hay que dibujar la estructura de la memoria principal. Analizando los campos de la dirección física podemos deducir las características de la memoria principal. En primer lugar, sabemos que el direccionamiento se realiza a nivel de byte y, tal y como se ha comentado, la memoria principal está organizada por **bancos consecutivos de módulos entrelazados**. Dado que se utilizan 3 bits para indicar el byte dentro de la palabra, podemos decir que las **palabras** son de **8 bytes** ( $2^3$ ). Por otra parte, dado que se utilizan 2 bits para indicar el módulo, podemos decir que cada banco de memoria está formado por **4 módulos** ( $2^2$ ). Para indicar una dirección a nivel de palabra dentro del módulo se utilizan 6 bits, por lo que podemos deducir que cada módulo tiene un tamaño de 64 palabras ( $2^6$ ) o 512 bytes ( $2^9$ ). Finalmente, para indicar el número de bancos se utiliza 1 bit en la dirección física, por lo que podemos deducir que la memoria principal está formada por **2 bancos consecutivos**. En resumen, la memoria principal se estructura en 2 bancos consecutivos, cada uno de ellos con 4 módulos entrelazados.

Dado que la memoria principal es de 4 kB ( $2^{12}$  byte) y la **palabra es de 8 bytes** ( $2^3$ ), podemos decir que la memoria principal tiene una capacidad para **512 palabras** ( $2^9$ ):

$$\frac{4 \text{ kB}}{8 \text{ bytes/pal}} = \frac{2^{12} \text{ bytes}}{2^3 \text{ bytes/pal}} = 2^9 \text{ palabras} = 512 \text{ palabras}$$

De la misma forma, dado que la memoria principal está formada por 2 bancos consecutivos, cada **banco** tiene una capacidad de **2 kB** ( $2^{11}$ ), o **256 palabras** ( $2^8$ ):

$$\frac{4 \text{ kB}}{2 \text{ bancos}} = 2 \text{ kB/banco}$$

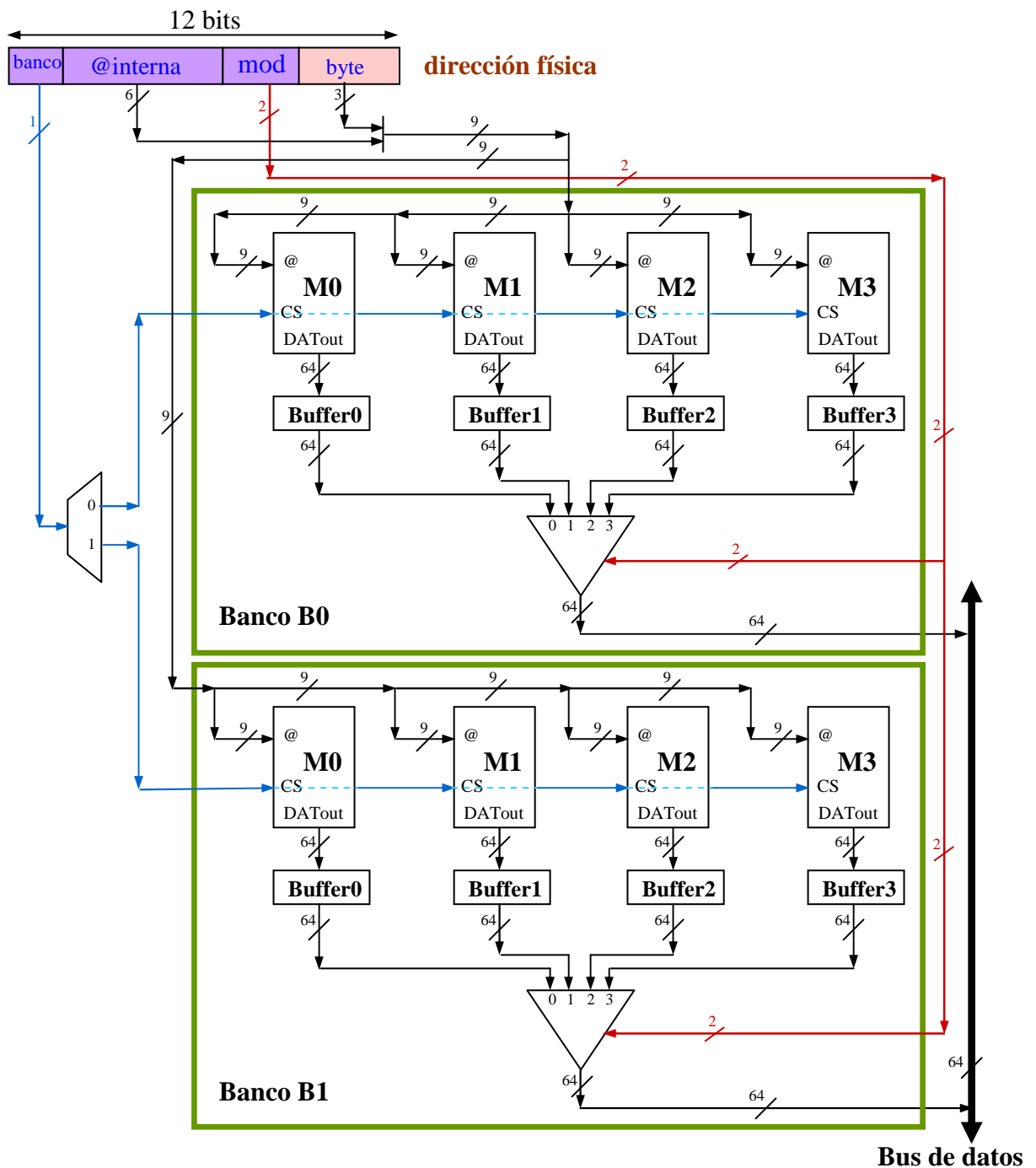
$$\frac{2^9 \text{ palabras}}{2^1 \text{ bancos}} = 2^8 \text{ palabras/banco} = 256 \text{ palabras/banco}$$

Finalmente, como cada banco está formado por 4 módulos ( $2^2$ ), cada **módulo** tendrá una capacidad de **512 bytes** ( $2^9$ ), o **64 palabras** ( $2^6$ ):

$$\frac{2 \text{ kB/banco}}{4 \text{ módulos/banco}} = 512 \text{ B/módulos}$$

$$\frac{2^8 \text{ pal/banco}}{2^2 \text{ módulos/banco}} = 2^6 \text{ palabras/módulo} = 64 \text{ palabras/módulo}$$

En definitiva, el esquema hardware de la memoria principal es el siguiente:



- (c) Tenemos que calcular el tiempo de traducción y de acceso a memoria principal para la secuencia de referencias dada. Para ello, recogeremos los resultados en una tabla.

Para resolver este apartado, hay que recordar las ecuaciones vistas en clase para conseguir los resultados parciales. Veamos a continuación estas ecuaciones:

Para calcular el segmento y desplazamiento a partir de la dirección lógica:

$$\text{segmento} = @\text{lógica} \text{ div tamaño de segmento (bytes)} = @\text{lógica} \text{ div } 256$$

$$\text{desplazamiento} = @\text{lógica} \text{ mod tamaño de segmento (bytes)} = @\text{lógica} \text{ mod } 256$$

Para calcular la dirección física, hay que sumar a la dirección base de segmento, obtenida a través de la tabla de segmentos, el desplazamiento correspondiente:

$$\text{@física} = \text{dirección base segmento} + \text{desplazamiento}$$

Para calcular la dirección física a nivel de palabra a partir de la dirección física a nivel de byte:

$$\text{palabra} = \text{@física} \text{ div tamaño palabra (bytes)} = \text{@física} \text{ div } 8$$

Para calcular el banco a partir de la dirección física a nivel de palabra:

$$\text{banco} = \text{palabra} \text{ div tamaño banco (palabras)} = \text{palabra} \text{ div } 2^8 = \text{palabra} \text{ div } 256$$

$$\text{resto} = \text{palabra} \text{ mod tamaño banco (palabras)} = \text{palabra} \text{ mod } 2^8 = \text{palabra} \text{ mod } 256$$

A partir del resto anterior se obtiene el valor de dos campos, el módulo y la dirección interna a nivel de palabra dentro del módulo:

$$\text{@interna} = \text{resto} \text{ div número módulos banco} = \text{resto} \text{ div } 4$$

$$\text{mod} = \text{resto} \text{ mod número módulos banco} = \text{hondarra} \text{ mod } 4$$

También podemos agrupar estas cuatro últimas ecuaciones para calcular los diferentes campos sin necesidad de calcular el resto intermedio:

$$\text{banco} = (\text{palabra} \text{ div número módulos banco}) \text{ div tamaño módulo palabras} =$$

$$= (\text{palabra} \text{ div } 2^2) \text{ div } 2^6 = (\text{palabra} \text{ div } 4) \text{ div } 64$$

$$\text{@interna} = (\text{palabra} \text{ div número módulos banco}) \text{ mod tamaño módulo palabras} =$$

$$= (\text{palabra} \text{ div } 2^2) \text{ mod } 2^6 = (\text{palabra} \text{ div } 4) \text{ mod } 64$$

$$\text{mod} = \text{palabra} \text{ mod número módulos banco} = \text{palabra} \text{ mod } 2^2 = \text{palabra} \text{ mod } 4$$

Teniendo en cuenta todas esas ecuaciones, la siguiente tabla resume los resultados que se obtienen para la secuencia de referencias dada:

@ℓ	segmento	d	T <sub>trad.</sub>	@base	d	@f	pal	banco	resto	@interna	mod	T <sub>acceso</sub>
1480	5	200	30	600	200	800	100	0	100	25	0	10
412	1	156	30	2700	156	2856	357	1	101	25	1	10
1496	5	216	1	600	216	816	102	0	102	25	2	1
428	1	172	1	2700	172	2872	359	1	103	25	3	1

En lo que respecta al tiempo de traducción, en la tabla se ve claramente que se accede a dos segmentos, concretamente el 5 y el 1. El primer acceso a dichos segmentos será fallo y serán necesarios 30 ciclos para realizar la traducción de esas direcciones, esto es, para acceder a la tabla de segmentos y completar las entradas correspondientes en el TLB. Una vez completado el TLB, el resto de los accesos serán acierto y su tiempo de traducción será, por tanto, de 1 ciclo.

En lo que se refiere al tiempo de acceso a memoria principal, podemos ver que en todos los casos la dirección interna es la dirección 25 alternando los diferentes módulos de memoria. Podríamos pensar que para el primer acceso serían

necesarios 10 ciclos y que el resto, que se obtendrían del buffer de entrelazado, sería 1 ciclo para cada referencia. Este supuesto es erróneo: las direcciones hacen referencia a módulos que se encuentran en distinto banco.

En el primer acceso se lee la dirección interna 25 del banco 0 y se cargan en los buffers de entrelazado de ese banco todas las palabras que se corresponden con la dirección interna 25 en todos los módulos de ese banco, pero no sabemos qué información existe en los buffers de entrelazado del banco 1. Por ello, en la segunda referencia se accede al banco 1 y se lee la dirección interna 25 en todos los módulos de ese banco para cargar los buffers de entrelazado del banco. En estos dos casos, el tiempo de acceso es de 10 ciclos. La tercera y cuarta referencia se corresponden con accesos entrelazados dentro de su respectivo banco, por lo que su tiempo de acceso es de 1 ciclo.

- (d) En este apartado hay que calcular el tiempo necesario para la traducción de todas las referencias lógicas que genera el programa, sabiendo que el programa está formado por 4 segmentos: las instrucciones por un lado, y un segmento para cada uno de los vectores A, B y C.

El tiempo de traducción dependerá del número de aciertos y fallos que se produzcan en el TLB en la traducción de todas las direcciones lógicas. En el caso de la segmentación, el número de fallos coincide con el número de segmentos distintos utilizados, ya que se producirá fallo en el TLB en el primer acceso a un segmento. Por tanto, si el programa está formado por 4 segmentos, tendremos 4 fallos de TLB, el resto de los accesos al TLB serán aciertos. Para determinar el número de aciertos, basta con calcular cuántos accesos a memoria principal genera el programa.

Existen 3 instrucciones fuera del bucle, en las que no se accede a ningún dato. Por tanto, tendremos 3 accesos a memoria principal fuera del bucle.

Dentro del bucle, además de las 12 instrucciones, se accede 3 veces a elementos del vector B, 1 vez a elementos del vector C y 1 vez a elementos del vector A (en este caso en escritura). Por tanto, tenemos  $12 + 3 + 1 + 1 = 17$  accesos a memoria principal dentro del bucle. Dado que el bucle se ejecuta 50 veces (desde  $i = 2$  hasta  $i = 51$ ), el número total de accesos a memoria será el siguiente:

$$\text{número de accesos} = 3 + (17 \times 50) = 853$$

De todos ellos, 4 supondrán fallo en el TLB y el resto serán aciertos, es decir, se producirán  $853 - 4 = 849$  aciertos.

Por tanto, el tiempo de traducción para todas las referencias del programa será:

$$T_{\text{traducción}} = 4 \times 30 + 849 \times 1 = 969 \text{ ciclos}$$