

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 1 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

##### Ejercicio 1

Una computadora tiene una memoria principal de  $64K \times 16$  y una memoria caché, que utiliza la técnica de mapeo directo, capaz de almacenar 1K palabras de memoria principal. El tamaño de bloque de la caché es de 4 palabras. Determinar:

1. El número de bloques que puede alojar la caché.

La caché almacena 1K palabras de memoria principal. Teniendo en cuenta que cada bloque tiene 4 palabras, existirán  $1024/4 = 256$  bloques.

2. El número de bits en los campos de etiqueta, línea y palabra del formato de dirección.

Para direccionar las 64K palabras de la memoria principal se necesitan  $\log_2(64 \cdot 1024) = \log_2(2^6 \cdot 2^{10}) = 16$  bits.

Teniendo en cuenta que la caché tiene 256 bloques (256 líneas), se necesitarán  $\log_2(256) = \log_2(2^8) = 8$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits.

La longitud de la etiqueta será de 6 bits, ya que es la diferencia entre la longitud de la dirección de memoria física (16 bits) y la dirección de memoria caché ( $8+2 = 10$  bits).

Por tanto, el formato de la dirección de memoria principal, que indica el tamaño de cada campo (en bits), es el siguiente:

Etiqueta	Línea	Palabra
6	8	2

3. La cantidad de bits en cada línea de la caché y sus funciones, incluyendo un bit de validación.

En cada línea de caché habrá 1 bit de validación, 6 bits de etiqueta y 4 palabras de 16 bits cada una. En total,  $1+6+4 \cdot 16 = 71$  bits.

##### Ejercicio 2

Se dispone de un sistema de memoria en el que interviene una memoria caché de mapeo directo organizada en bloques de 16 palabras. Un ejemplo de dirección de memoria principal en este sistema es FF1DE (en hexadecimal), mientras que un ejemplo de dirección de memoria caché es 732 (en octal). Determinar:

1. Longitud de una dirección de memoria física. Longitud de una dirección de memoria caché. Cuántos bits de la dirección de caché se destinan a indicar la línea y cuántos a la palabra. Longitud de la etiqueta.

La dirección de memoria física de ejemplo es FF1DE (hex) = 1111 1111 0001 1101 1110. Tiene 20 bits, por lo que la longitud de una dirección de memoria física es de 20 bits.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 2 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

La dirección de memoria caché de ejemplo es 732 (oct) = 111 011 010. Tiene 9 bits, por lo que la longitud de una dirección de memoria caché es de 9 bits.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(16) = \log_2(2^4) = 4$  bits.

Teniendo en cuenta que la longitud de la dirección de memoria caché es de 9 bits y que 4 de ellos se destinan a indicar la palabra dentro del bloque, el resto  $9-4 = 5$  bits se emplearán en indicar la línea de caché.

La longitud de la etiqueta será de 11 bits, ya que es la diferencia entre la longitud de la dirección de memoria física (20 bits) y la dirección de memoria caché ( $5+4 = 9$  bits).

Por tanto, el formato de la dirección de memoria principal, que indica el tamaño de cada campo (en bits), es el siguiente:

Etiqueta	Línea	Palabra
11	5	4

2. Tamaño de la memoria principal si el ancho de palabra es de 16 bits. Tamaño de la memoria caché.

El tamaño de la memoria principal será  $2^{20}$  palabras  $\cdot (16 \text{ bits} / \text{palabra}) = 2^{24}$  bits.

El tamaño de la memoria caché será  $2^5 \cdot (1+11+16 \cdot 16) = 8576$  bits, donde  $2^5$  es el número de líneas de la caché, 1 es el bit de validez, 11 es la longitud de la etiqueta, y  $16 \cdot 16$  es el tamaño del bloque.

### Ejercicio 3

Una memoria caché asociativa por conjuntos de 2 vías utiliza bloques de 4 palabras, pudiendo almacenar 2048 palabras de la memoria principal. El tamaño de la memoria principal es de 128K $\times$ 32. Determinar:

1. Organización de la caché.

La caché almacena 2048 palabras de memoria principal. Como tiene 2 vías, almacena  $2048/2 = 1024$  palabras por cada vía. Teniendo en cuenta que cada bloque tiene 4 palabras, existirán  $1024/4 = 256$  bloques en cada vía, por lo que la caché tendrá 256 líneas, y necesitará  $\log_2(256) = \log_2(2^8) = 8$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits.

Para direccionar las 128K palabras de la memoria principal se necesitan  $\log_2(128 \cdot 1024) = \log_2(2^7 \cdot 2^{10}) = 17$  bits.

La longitud de la etiqueta será de 7 bits, ya que es la diferencia entre la longitud de la dirección de memoria física (17 bits) y la dirección de memoria caché ( $8+2 = 10$  bits).

Por tanto, el formato de la dirección de memoria principal, que indica el tamaño de cada campo (en bits), es el siguiente:

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

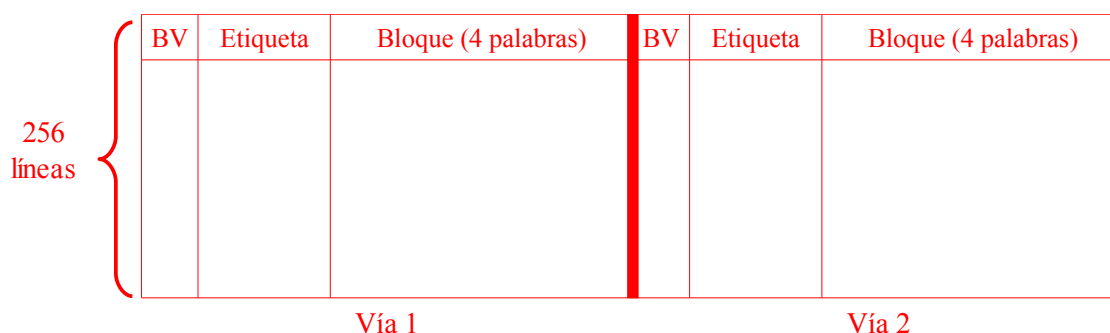
Página 3 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

Etiqueta	Línea	Palabra
7	8	2

La organización de la caché queda como sigue:



#### 2. Tamaño de la memoria caché.

El tamaño de la memoria caché será  $256 \cdot 2 \cdot (1+7+4 \cdot 32) = 69632$  bits, donde 256 es el número de líneas de la caché, 2 son las vías existentes, 1 es el bit de validez, 7 es la longitud de la etiqueta, y  $4 \cdot 32$  es el tamaño del bloque.

### Ejercicio 4

Se dispone de un sistema de memoria que utiliza direcciones de 32 bits. La unidad mínima direccionable es el byte. En memoria se encuentran tres arrays de elementos de 16 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A		Array B		Array C	
00101AF0	A[0]	3C509AF0	B[0]	27B10330	C[0]
00101AF2	A[1]	3C509AF2	B[1]	27B10332	C[1]
00101AF4	A[2]	3C509AF4	B[2]	27B10334	C[2]
00101AF6	A[3]	3C509AF6	B[3]	27B10336	C[3]
00101AF8	A[4]	3C509AF8	B[4]	27B10338	C[4]
	⋮		⋮		⋮

Se dispone también en memoria de dos programas que manipulan estos arrays y que se diferencian únicamente en la línea 5, siendo su código el siguiente:

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 4 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

Programa 1	Programa 2
1: int sp; 2: 3: sp = 0; 4: for (int i=0; i<n; ++i) 5:     sp += A[i]*B[i];	1: int sp; 2: 3: sp = 0; 4: for (int i=0; i<n; ++i) 5:     sp += A[i]*C[i];

1. Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo directo, que permite almacenar 32K palabras de memoria principal, y con un tamaño de bloque de 16 palabras, determinar qué programa se ejecutará más rápido.

En principio, teóricamente ambos programas deberían tardar el mismo tiempo en ejecutarse, puesto que se trata en ambos casos de una suma de productos que solo difieren en los datos. Sin embargo, la posición de los datos en la memoria es lo que marcará la diferencia.

Se calcula en primer lugar el formato de la dirección de memoria caché.

La caché almacena 32K palabras de memoria principal. Teniendo en cuenta que cada bloque tiene 16 palabras, existirán  $32 \cdot 1024 / 16 = 2048$  bloques, por lo que la caché tendrá 2048 líneas, y necesitará  $\log_2(2048) = \log_2(2^{11}) = 11$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(16) = \log_2(2^4) = 4$  bits.

Por tanto, el formato de la dirección de memoria caché, que indica el tamaño de cada campo (en bits), es el siguiente:

Línea	Palabra
11	4

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 5 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

##### Programa 1:

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

Elemento	Dirección	Etiqueta	Línea	Palabra	Datos en caché	Resultado del acceso
A[0]	00101AF0	0000 0000 0001 0000 0	001 1010 1111 (431)	0000	-----	Fallo
B[0]	3C509AF0	0011 1100 0101 0000 1	001 1010 1111 (431)	0000	A[0] ... A[7]	Fallo
A[1]	00101AF2	0000 0000 0001 0000 0	001 1010 1111 (431)	0010	B[0] ... B[7]	Fallo
B[1]	3C509AF2	0011 1100 0101 0000 1	001 1010 1111 (431)	0010	A[0] ... A[7]	Fallo
A[2]	00101AF4	0000 0000 0001 0000 0	001 1010 1111 (431)	0100	B[0] ... B[7]	Fallo
B[2]	3C509AF4	0011 1100 0101 0000 1	001 1010 1111 (431)	0100	A[0] ... A[7]	Fallo
....	....	....	....	....	....	....
A[7]	00101AFE	0000 0000 0001 0000 0	001 1010 1111 (431)	1110	B[0] ... B[7]	Fallo
B[7]	3C509AFE	0011 1100 0101 0000 1	001 1010 1111 (431)	1110	A[0] ... A[7]	Fallo
A[8]	00101B00	0000 0000 0001 0000 0	001 1011 0000 (432)	0000	-----	Fallo
B[8]	3C509B00	0011 1100 0101 0000 1	001 1011 0000 (432)	0000	A[8] ... A[15]	Fallo
A[9]	00101B02	0000 0000 0001 0000 0	001 1011 0000 (432)	0010	B[8] ... B[15]	Fallo
B[9]	3C509B02	0011 1100 0101 0000 1	001 1011 0000 (432)	0010	A[8] ... A[15]	Fallo
....	....	....	....	....	....	....

La sucesión de direcciones generadas por el procesador entrarán permanentemente en conflicto, ya que los arrays A y B, a pesar de estar ubicados en distintas zonas de memoria principal, comparten bloque de memoria caché. Se producirán continuos fallos de caché que obligarán a que todos los accesos a memoria tengan que ser servidos por la memoria principal, desaprovechando completamente la caché.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 6 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

##### Programa 2:

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

Elemento	Dirección	Etiqueta	Línea	Palabra	Datos en caché	Resultado del acceso
A[0]	00101AF0	0000 0000 0001 0000 0	001 1010 1111 (431)	0000	-----	Fallo
C[0]	27B10330	0010 0111 1011 0001 0	000 0011 0011 (51)	0000	-----	Fallo
A[1]	00101AF2	0000 0000 0001 0000 0	001 1010 1111 (431)	0010	A[0] ... A[7]	Acierto
C[1]	27B10332	0010 0111 1011 0001 0	000 0011 0011 (51)	0010	C[0] ... C[7]	Acierto
A[2]	00101AF4	0000 0000 0001 0000 0	001 1010 1111 (431)	0100	A[0] ... A[7]	Acierto
C[2]	27B10334	0010 0111 1011 0001 0	000 0011 0011 (51)	0100	C[0] ... C[7]	Acierto
....	....	....	....	....	....	....
A[7]	00101AFE	0000 0000 0001 0000 0	001 1010 1111 (431)	1110	A[0] ... A[7]	Acierto
C[7]	27B1033E	0010 0111 1011 0001 0	000 0011 0011 (51)	1110	C[0] ... C[7]	Acierto
A[8]	00101B00	0000 0000 0001 0000 0	001 1011 0000 (432)	0000	-----	Fallo
C[8]	27B10340	0010 0111 1011 0001 0	000 0011 0100 (52)	0000	-----	Fallo
A[9]	00101B02	0000 0000 0001 0000 0	001 1011 0000 (432)	0010	A[8] ... A[15]	Acierto
C[9]	27B10342	0010 0111 1011 0001 0	000 0011 0100 (52)	0010	C[8] ... C[15]	Acierto
....	....	....	....	....	....	....

La sucesión de direcciones generadas ahora por el procesador no entran en conflicto, a diferencia de lo que ocurría en el programa 1, ya que los bloques de memoria caché donde se almacenen los datos de ambos arrays son distintos. Teniendo en cuenta que se producirán 2 fallos de caché cada 16 accesos, el resto de accesos (14 de cada 16, un 87.5 %) provocarán un acierto de caché.

Por tanto, el programa 2 se ejecutará más rápido que el programa 1, ya que un 87.5 % de accesos serán servidos por la memoria caché, haciendo más rápido el acceso a memoria.

- Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo asociativo por conjunto de dos vías, que permite almacenar 32K palabras de memoria principal, y con un tamaño de bloque de 8 palabras, determinar qué programa se ejecutará más rápido.

Se calcula en primer lugar el formato de la dirección de memoria caché.

La caché almacena 32K palabras de memoria principal. Como tiene 2 vías, almacena  $32K/2 = 16K$  palabras por cada vía. Teniendo en cuenta que cada bloque tiene 8 palabras, existirán  $16 \cdot 1024/8 = 2048$  bloques en cada vía, por lo que la caché tendrá 2048 líneas, y necesitará  $\log_2(2048) = \log_2(2^{11}) = 11$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(8) = \log_2(2^3) = 3$  bits.

Por tanto, el formato de la dirección de memoria caché, que indica el tamaño de cada campo (en bits), es el siguiente:

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 7 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

Línea	Palabra
11	3

#### Programa 1:

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

Elemento	Dirección	Etiqueta	Línea	Palabra	Datos en caché (ambas vías)	Resultado del acceso
A[0]	00101AF0	0000 0000 0001 0000 00	01 1010 1111 0 (862)	000	----- -----	Fallo
B[0]	3C509AF0	0011 1100 0101 0000 10	01 1010 1111 0 (862)	000	A[0] ... A[3] -----	Fallo
A[1]	00101AF2	0000 0000 0001 0000 00	01 1010 1111 0 (862)	010	A[0] ... A[3] B[0] ... B[3]	Acierto
B[1]	3C509AF2	0011 1100 0101 0000 10	01 1010 1111 0 (862)	010	A[0] ... A[3] B[0] ... B[3]	Acierto
A[2]	00101AF4	0000 0000 0001 0000 00	01 1010 1111 0 (862)	100	A[0] ... A[3] B[0] ... B[3]	Acierto
B[2]	3C509AF4	0011 1100 0101 0000 10	01 1010 1111 0 (862)	100	A[0] ... A[3] B[0] ... B[3]	Acierto
A[3]	00101AF6	0000 0000 0001 0000 00	01 1010 1111 0 (862)	110	A[0] ... A[3] B[0] ... B[3]	Acierto
B[3]	3C509AF6	0011 1100 0101 0000 10	01 1010 1111 0 (862)	110	A[0] ... A[3] B[0] ... B[3]	Acierto
A[4]	00101AF8	0000 0000 0001 0000 00	01 1010 1111 1 (863)	000	----- -----	Fallo
B[4]	3C509AF8	0011 1100 0101 0000 10	01 1010 1111 1 (863)	000	A[4] ... A[7] -----	Fallo
A[5]	00101AFA	0000 0000 0001 0000 00	01 1010 1111 1 (863)	010	A[4] ... A[7] B[4] ... B[7]	Acierto
B[5]	3C509AFA	0011 1100 0101 0000 10	01 1010 1111 1 (863)	010	A[4] ... A[7] B[4] ... B[7]	Acierto
....	....	....	....	....	....	....

A diferencia de lo que ocurría con este mismo programa en el apartado anterior, y a pesar de que ambos arrays comparten bloque de memoria caché, ahora, al existir dos vías, cada bloque se ubica en una vía distinta, por lo que no existirá conflicto. Teniendo en cuenta que se producirán 2 fallos de caché cada 8 accesos, el resto de accesos (6 de cada 8, un 75 %) provocarán un acierto de caché.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 8 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

##### Programa 2:

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

Elemento	Dirección	Etiqueta	Línea	Palabra	Datos en caché (ambas vías)	Resultado del acceso
A[0]	00101AF0	0000 0000 0001 0000 00	01 1010 1111 0 (862)	000	----- -----	Fallo
C[0]	27B10330	0010 0111 1011 0001 00	00 0011 0011 0 (102)	000	----- -----	Fallo
A[1]	00101AF2	0000 0000 0001 0000 00	01 1010 1111 0 (862)	010	A[0] ... A[3] -----	Acierto
C[1]	27B10332	0010 0111 1011 0001 00	00 0011 0011 0 (102)	010	C[0] ... C[3] -----	Acierto
A[2]	00101AF4	0000 0000 0001 0000 00	01 1010 1111 0 (862)	100	A[0] ... A[3] -----	Acierto
C[2]	27B10334	0010 0111 1011 0001 00	00 0011 0011 0 (102)	100	C[0] ... C[3] -----	Acierto
A[3]	00101AF6	0000 0000 0001 0000 00	01 1010 1111 0 (862)	110	A[0] ... A[3] -----	Acierto
C[3]	27B10336	0010 0111 1011 0001 00	00 0011 0011 0 (102)	110	C[0] ... C[3] -----	Acierto
A[4]	00101AF8	0000 0000 0001 0000 00	01 1010 1111 1 (863)	000	----- -----	Fallo
C[4]	27B10338	0010 0111 1011 0001 00	00 0011 0011 1 (103)	000	----- -----	Fallo
A[5]	00101AFA	0000 0000 0001 0000 00	01 1010 1111 1 (863)	010	A[4] ... A[7] -----	Acierto
C[5]	27B1033A	0010 0111 1011 0001 00	00 0011 0011 1 (103)	010	C[4] ... C[7] -----	Acierto
....	....	....	....	....	....	....

Al igual que ocurría con este mismo programa en el apartado anterior, la sucesión de direcciones generadas por el procesador no entran en conflicto, ya que los bloques de memoria caché donde se almacenen los datos de ambos arrays son distintos. Teniendo en cuenta que se producirán 2 fallos de caché cada 8 accesos, el resto de accesos (6 de cada 8, un 75 %) provocarán un acierto de caché.

Por tanto, en este caso ambos programas se ejecutan igual de rápido, siendo el 75 % de las referencias a memoria servidas por la memoria caché.



# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 9 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

##### Ejercicio 5

Un sistema de memoria de  $256 \times 8$  incorpora una memoria caché de mapeo directo que puede almacenar 16 palabras de la memoria principal. El tamaño de bloque es de 4 palabras. En la memoria principal se encuentra una tabla A de 24 elementos, almacenada como indica la siguiente figura:

08	A[0,0]	A[0,1]	A[0,2]	A[0,3]
0C	A[1,0]	A[1,1]	A[1,2]	A[1,3]
	.....			
24	A[7,0]	A[7,1]	A[7,2]	A[7,3]

Se diseñan dos programas para sumar todos los elementos de la tabla, cuyo código es el siguiente:

Programa 1	Programa 2
<pre>1: int s; 2: 3: s = 0; 4: for (int i=0; i&lt;8; ++i) 5:     for (int j=0; j&lt;4; ++j) 6:         s += A[i][j];</pre>	<pre>1: int s; 2: 3: s = 0; 4: for (int j=0; j&lt;4; ++j) 5:     for (int i=0; i&lt;8; ++i) 6:         s += A[i][j];</pre>

Determinar qué programa se ejecutará más rápido.

##### Ejercicio 6

Se dispone de un sistema de memoria que utiliza direcciones de 32 bits. La unidad mínima direccionable es el byte. En memoria se encuentran dos arrays de elementos de 8 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A. Dirección de comienzo: 00101AF0												
Contenido	12	4	67	81	231	89	106	198	206	11	39	162

Array B. Dirección de comienzo: 3C50BAF0												
Contenido	102	44	54	241	163	25	175	102	39	89	156	250

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 10 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo directo que permite almacenar 32 K palabras de memoria principal y que el tamaño de bloque es de 8 palabras, representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```
1: int sp;  
2:  
3: sp = 0;  
4: for (int i=0; i<n; ++i)  
5:     sp += A[i]*B[i];
```

Nota: Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución.

#### **Ejercicio 7**

Se dispone de un sistema de memoria de 64K×8. La unidad mínima direccionable es el byte. En memoria se encuentran dos arrays de elementos de 8 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A. Dirección de comienzo: 18F0												
Contenido	12	4	67	81	231	89	106	198	206	11	39	162

Array B. Dirección de comienzo: B2D0												
Contenido	102	44	54	241	163	25	175	102	39	89	156	250

Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo directo que permite almacenar 1K palabras de memoria principal y que el tamaño de bloque es de 4 palabras, representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```
1: int sp;  
2:  
3: sp = 0;  
4: for (int i=0; i<n; i+=2)  
5:     sp += A[i]*B[i];
```

Nota: Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución.

Se calcula en primer lugar el formato de la dirección de memoria principal.

Para direccionar las 64K palabras de la memoria principal se necesitan  $\log_2(64 \cdot 1024) = \log_2(2^6 \cdot 2^{10}) = 16$  bits.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 11 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

La caché almacena 1K palabras de memoria principal. Teniendo en cuenta que cada bloque tiene 4 palabras, existirán  $1024/4 = 256$  bloques, por lo que la caché tendrá 256 líneas, y necesitará  $\log_2(256) = \log_2(2^8) = 8$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits.

La longitud de la etiqueta será de 6 bits, ya que es la diferencia entre la longitud de la dirección de memoria física (16 bits) y la dirección de memoria caché ( $8+2 = 10$  bits).

Por tanto, el formato de la dirección de memoria principal, que indica el tamaño de cada campo (en bits), es el siguiente:

Etiqueta	Línea	Palabra
6	8	2

#### Primera iteración:

Se accede a los elementos A[0] y B[0]. Estos elementos están situados en las siguientes direcciones:

Elemento	Dirección	Etiqueta	Línea	Palabra
A[0]	18F0	0001 10	00 1111 00 (60)	00
B[0]	B2D0	1011 00	10 1101 00 (180)	00

Se produce un fallo de caché ya que se encuentra inicialmente vacía. Hay que rellenar el contenido de las líneas 60 y 180. La caché queda como sigue:

	BV	Etiqueta	Bloque (4 palabras)			
60						
	1	0001 10	12	4	67	81
180						
	1	1011 00	102	44	54	241

#### Segunda iteración:

Se accede a los elementos A[2] y B[2]. Estos elementos están situados en las siguientes direcciones:

Elemento	Dirección	Etiqueta	Línea	Palabra
A[2]	18F2	0001 10	00 1111 00 (60)	10
B[2]	B2D2	1011 00	10 1101 00 (180)	10

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 12 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

Las líneas de caché accedidas son la 60 y la 180, igual que en la primera iteración. Teniendo en cuenta que el bit de validez de dichas líneas está a 1 y que la etiqueta coincide, se produce un acierto de caché en ambos casos. Los datos facilitados por la caché son  $A[2] = 67$  y  $B[2] = 54$ . La caché no se modifica.

#### Tercera iteración:

Se accede a los elementos  $A[4]$  y  $B[4]$ . Estos elementos están situados en las siguientes direcciones:

Elemento	Dirección	Etiqueta	Línea	Palabra
$A[4]$	18F4	0001 10	00 1111 01 (61)	00
$B[4]$	B2D4	1011 00	10 1101 01 (181)	00

Se produce un fallo de caché, ya que las líneas 61 y 181 están vacías. Hay que rellenar el contenido de dichas líneas. La caché queda finalmente como sigue:

	BV	Etiqueta	Bloque (4 palabras)			
60	1	0001 10	12	4	67	81
61	1	0001 10	231	89	106	198
180	1	1011 00	102	44	54	241
181	1	1011 00	163	25	175	102

#### **Ejercicio 8**

Se tiene el siguiente programa implementado en instrucciones MIPS. El array “data” se encuentra ubicado en memoria a partir de la posición 10010000 (en hexadecimal) y contiene 64 elementos de tipo palabra con el valor 1.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 13 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

```
1:      move $s0, $zero # índice X
2:      move $s1, $zero # índice Y
3:      move $s2, $zero # Acumulador
4:      la $s3, data
5:      li $s4, 8 # Número de filas y columnas del array "data"
6:
7: loop:  sll $t0, $s1, 3
8:      add $t0, $s0, $t0
9:      sll $t0, $t0, 2
10:     add $t0, $s3, $t0
11:     lw $t0, 0($t0)
12:     add $s2, $s2, $t0
13:     addi $s1, $s1, 1
14:     bne $s1, $s4, loop
15:     move $s1, $zero
16:     addi $s0, $s0, 1
17:     bne $s0, $s4, loop
```

Determinar el porcentaje de aciertos de caché del programa según las siguientes configuraciones de la memoria caché:

En primer lugar es necesario determinar el patrón de accesos a memoria que realiza el programa. Parece tratarse de un acceso a un array bidimensional, ya que se han definido registros para almacenar los índices X e Y. Las líneas 7–8 calculan  $8 \cdot Y + X$ , obteniendo así el desplazamiento del elemento (X,Y) dentro del array. La línea 9 multiplica por 4 dicho desplazamiento, ya que cada palabra en la arquitectura MIPS ocupa 4 bytes. La línea 10 añade la dirección base del array. Se realiza entonces la lectura del dato de memoria en la línea 11. La línea 13 incrementa el índice Y y la línea 14 permite iterar hasta que dicho índice tenga un valor de 8. En ese momento, el índice Y se vuelve a poner a valor 0 y se incrementa el índice X, comprobándose que no ha llegado a valor 8 y, en ese caso, iterando de nuevo.

Por tanto, el programa realiza un acceso por columnas a un array bidimensional de  $8 \times 8$  elementos de tipo palabra (32 bits). Las direcciones de todos los elementos accedidos se muestran en la tabla siguiente. Se muestra únicamente el valor del último byte de la dirección, ya que los demás permanecen invariables.

00	04	08	0C	10	14	18	1C
20	24	28	2C	30	34	38	3C
40	44	48	4C	50	54	58	5C
60	64	68	6C	70	74	78	7C
80	84	88	8C	90	94	98	9C
A0	A4	A8	AC	B0	B4	B8	BC
C0	C4	C8	CC	D0	D4	D8	DC
E0	E4	E8	EC	F0	F4	F8	FC

1. Mapeo directo con 16 bloques de 4 palabras.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 14 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

Se calcula en primer lugar el formato de la dirección de memoria caché.

Al existir 16 bloques la caché tendrá 16 líneas y necesitará  $\log_2(16) = \log_2(2^4) = 4$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits.

Hay que tener en cuenta que cada palabra está formada por 4 bytes, por lo que se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits para indicar el byte dentro de la palabra.

Por tanto, el formato de la dirección de memoria caché, que indica el tamaño de cada campo (en bits), es el siguiente:

Línea	Palabra	Byte
4	2	2

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

Elemento	Dirección	Etiqueta	Línea	Palabra	Byte	Datos en caché	Resultado del acceso
datos[0][0]	00	100100 (hex)	0000	00	00	-----	Fallo
datos[1][0]	20	100100 (hex)	0010	00	00	-----	Fallo
datos[2][0]	40	100100 (hex)	0100	00	00	-----	Fallo
datos[3][0]	60	100100 (hex)	0110	00	00	-----	Fallo
datos[4][0]	80	100100 (hex)	1000	00	00	-----	Fallo
datos[5][0]	A0	100100 (hex)	1010	00	00	-----	Fallo
datos[6][0]	C0	100100 (hex)	1100	00	00	-----	Fallo
datos[7][0]	E0	100100 (hex)	1110	00	00	-----	Fallo
datos[0][1]	04	100100 (hex)	0000	01	00	datos[0][0] ... datos[0][3]	Acierto
datos[1][1]	24	100100 (hex)	0010	01	00	datos[1][0] ... datos[1][3]	Acierto
datos[2][1]	44	100100 (hex)	0100	01	00	datos[2][0] ... datos[2][3]	Acierto
....	....	....	....	....		....	....

Puede advertirse que los primeros 8 accesos se realizan a las líneas pares de la caché. Estos accesos producen un fallo de caché al estar vacía. Pero a partir del noveno acceso, vuelve a repetirse el patrón de acceso a las líneas de caché, con el mismo valor de etiqueta, pero a palabras distintas. En concreto, antes se accedió a la palabra 0 de cada línea par de caché. Ahora se accede a la palabra 1 de las mismas líneas. Al estar esos datos ya en caché, se producirán aciertos en los 8 siguientes accesos. Igualmente ocurrirá en los 16 intentos siguientes, ya que se estará accediendo a las palabras 2 y 3 de cada uno de los bloques. Por tanto, en los primeros 32 accesos se producirán 24 aciertos de caché y 8 fallos.

Los siguientes accesos ocurrirán tal y como muestra la tabla a continuación.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 15 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

Elemento	Dirección	Etiqueta	Línea	Palabra	Byte	Datos en caché	Resultado del acceso
datos[0][4]	10	100100 (hex)	0001	00	00	-----	Fallo
datos[1][4]	30	100100 (hex)	0011	00	00	-----	Fallo
datos[2][4]	50	100100 (hex)	0101	00	00	-----	Fallo
datos[3][4]	70	100100 (hex)	0111	00	00	-----	Fallo
datos[4][4]	90	100100 (hex)	1001	00	00	-----	Fallo
datos[5][4]	B0	100100 (hex)	1011	00	00	-----	Fallo
datos[6][4]	D0	100100 (hex)	1101	00	00	-----	Fallo
datos[7][4]	F0	100100 (hex)	1111	00	00	-----	Fallo
datos[0][5]	14	100100 (hex)	0001	01	00	datos[0][4] ... datos[0][7]	Acierto
datos[1][5]	34	100100 (hex)	0011	01	00	datos[1][4] ... datos[1][7]	Acierto
datos[2][5]	54	100100 (hex)	0101	01	00	datos[2][4] ... datos[2][7]	Acierto
....	....	....	....	....		....	....

Nuevamente se repite el esquema de accesos, pero esta vez se accede a las líneas impares de la caché. De la misma manera que ocurría en los 32 primeros accesos, 24 de los 32 siguientes accesos producirán aciertos de caché y 8 de ellos producirán fallo.

Por lo tanto, de los 64 accesos a memoria que se realizan en total, se producirán  $24+24 = 48$  aciertos. Esto es un  $48/64 = 0.75 = 75\%$  de aciertos.

#### 2. Mapeo directo con 8 bloques de 8 palabras.

Se calcula en primer lugar el formato de la dirección de memoria caché.

Al existir 8 bloques la caché tendrá 8 líneas y necesitará  $\log_2(8) = \log_2(2^3) = 3$  bits para seleccionar la línea de caché.

Para seleccionar una palabra dentro del bloque se necesitarán  $\log_2(8) = \log_2(2^3) = 3$  bits.

Hay que tener en cuenta que cada palabra está formada por 4 bytes, por lo que se necesitarán  $\log_2(4) = \log_2(2^2) = 2$  bits para indicar el byte dentro de la palabra.

Por tanto, el formato de la dirección de memoria caché, que indica el tamaño de cada campo (en bits), es el siguiente:

Línea	Palabra	Byte
3	3	2

Seguidamente se muestra una relación de los primeros accesos a memoria que se realizan.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 16 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

Elemento	Dirección	Etiqueta	Línea	Palabra	Byte	Datos en caché	Resultado del acceso
datos[0][0]	00	100100 (hex)	000	000	00	-----	Fallo
datos[1][0]	20	100100 (hex)	001	000	00	-----	Fallo
datos[2][0]	40	100100 (hex)	010	000	00	-----	Fallo
datos[3][0]	60	100100 (hex)	011	000	00	-----	Fallo
datos[4][0]	80	100100 (hex)	100	000	00	-----	Fallo
datos[5][0]	A0	100100 (hex)	101	000	00	-----	Fallo
datos[6][0]	C0	100100 (hex)	110	000	00	-----	Fallo
datos[7][0]	E0	100100 (hex)	111	000	00	-----	Fallo
datos[0][1]	04	100100 (hex)	000	001	00	datos[0][0] ... datos[0][7]	Acierto
datos[1][1]	24	100100 (hex)	001	001	00	datos[1][0] ... datos[1][7]	Acierto
datos[2][1]	44	100100 (hex)	010	001	00	datos[2][0] ... datos[2][7]	Acierto
....	....	....	....	....		....	....

Puede advertirse que los primeros 8 accesos se realizan a líneas distintas pero consecutivas de la caché. De hecho, se direccionan todas las líneas de la misma. Estos accesos producen un fallo de caché al estar vacía. Pero a partir del noveno acceso, vuelve a repetirse el patrón de acceso a las líneas de caché, con el mismo valor de etiqueta, pero a palabras distintas. En concreto, antes se accedió a la palabra 0 de cada línea de caché, mientras que ahora se accede a la palabra 1 de las mismas líneas. Al estar esos datos ya en caché, se producirán aciertos en los 8 siguientes accesos. Lo mismo ocurrirá en los 48 intentos siguientes, ya que se estará accediendo a las palabras 2, 3, 4, 5, 6 y 7 de cada uno de los bloques.

Por tanto, en los 64 accesos a memoria que se realizan en total se producirán 56 aciertos de caché. Esto es un  $56/64 = 0.875 = 87.5\%$  de aciertos.

#### Ejercicio 9

Dado un espacio de direccionamiento virtual, especificado mediante una dirección de 24 bits, y un espacio de memoria física, especificado mediante una dirección de 16 bits, determinar:

1. El número de palabras que existen tanto en el espacio de direccionamiento virtual como en el de memoria física.

Ya que las direcciones de memoria virtual son de 24 bits, el número de palabras en memoria virtual serán  $2^{24}$ .

Ya que las direcciones de memoria física son de 16 bits, el número de palabras en memoria física serán  $2^{16}$ .

2. Si se utiliza la técnica de paginación con un tamaño de página de 2K palabras, indicar cuántas páginas y bloques hay en total.



# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 17 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

Existen  $2^{24}$  palabras en memoria virtual. Teniendo en cuenta que cada página tiene  $2K = 2^{11}$  palabras, existirán  $2^{24}/2^{11} = 2^{13}$  páginas.

Existen  $2^{16}$  palabras en memoria física. Teniendo en cuenta que cada bloque tiene  $2K = 2^{11}$  palabras, existirán  $2^{16}/2^{11} = 2^5$  bloques.

#### **Ejercicio 10**

Un sistema de memoria virtual paginado tiene un tamaño de página de 1K palabras. Pueden direccionarse 8 páginas. La memoria tiene capacidad para alojar 4 bloques. En un momento determinado, la tabla de páginas de un proceso contiene las siguientes entradas:

Página	Bloque
0	3
1	1
4	2
6	0

Elabórese una lista de los rangos de direcciones virtuales que producirán un fallo de página en caso de ser accedidas por la CPU.

Se calcula en primer lugar el formato de la dirección virtual.

Para direccionar las 8 páginas en memoria virtual se necesitan  $\log_2(8) = \log_2(2^3) = 3$  bits.

Para seleccionar una palabra dentro de una página se necesitan  $\log_2(1 \cdot 1024) = \log_2(2^{10}) = 10$  bits.

Por tanto, el formato de la dirección de memoria virtual, que indica el tamaño de cada campo (en bits), es el siguiente:

Página	Desplazamiento
3	10

Consultando la tabla de páginas del proceso se aprecia que las páginas 2, 3, 5 y 7 no se encuentran en memoria. Por tanto, cualquier dirección virtual dentro de estas páginas producirá un fallo de página.

Los siguientes rangos de direcciones producirán fallos de página:

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 18 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

Página	Direcciones de inicio y fin		Rango
2	010	000000000	0800 – 0BFF
	010	111111111	
3	011	000000000	0C00 – 0FFF
	011	111111111	
5	101	000000000	1400 – 17FF
	101	111111111	
7	111	000000000	1C00 – 1FFF
	111	111111111	

#### **Ejercicio 11**

Un sistema de memoria virtual paginado tiene un tamaño de página de 1K palabras. El espacio de direcciones es de 8K palabras y el de memoria de 4K palabras. Durante un cierto intervalo de tiempo se referencian las siguientes páginas en el orden indicado:

4, 0, 2, 0, 1, 2, 6, 1, 4, 0, 1, 1, 0, 2, 6, 3, 3, 5, 7

Determinar las 4 páginas que permanecen en memoria principal después de acceder a las páginas indicadas, según el algoritmo de sustitución utilizado (FIFO, LRU, LFU).

#### **Ejercicio 12**

En un sistema de memoria virtual paginado el espacio de direcciones es de 4 MB, mientras que el espacio de memoria es de 32 KB. El tamaño de página es de 512 bytes. Determinar:

1. Longitud en bits de una dirección virtual y de una dirección física.

Para direccionar los 4 MB de memoria virtual se necesitan  $\log_2(4 \cdot 1024 \cdot 1024) = \log_2(2^2 \cdot 2^{20}) = 22$  bits.

Para direccionar los 32 KB de memoria física se necesitan  $\log_2(32 \cdot 1024) = \log_2(2^5 \cdot 2^{10}) = 15$  bits.

2. Número de páginas y bloques del sistema de memoria.

Existen  $2^{22}$  palabras en memoria virtual. Teniendo en cuenta que cada página tiene  $512 = 2^9$  palabras, existirán  $2^{22}/2^9 = 2^{13}$  páginas.

Existen  $2^{15}$  palabras en memoria física. Teniendo en cuenta que cada bloque tiene  $512 = 2^9$  palabras, existirán  $2^{15}/2^9 = 2^6$  bloques.

3. Dimensiones de las tablas de páginas, según se implementen mediante una memoria de acceso aleatorio o a través de una memoria asociativa.

Ya que existen  $2^{13}$  páginas en memoria virtual, serán necesarios 13 bits para direccionarlas. Como el tamaño de página es de  $2^9$  palabras, se necesitarán 9 bits para direccionarlas. Por tanto, el formato de la dirección virtual, que indica el tamaño de cada campo (en bits), es el siguiente:

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 19 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

Página	Desplazamiento
13	9

Ya que existen  $2^6$  bloques en memoria física, serán necesarios 6 bits para direccionarlos. Por tanto, el formato de la dirección física, que indica el tamaño de cada campo (en bits), es el siguiente:

Bloque	Desplazamiento
6	9

La implementación de la tabla de páginas mediante una memoria de acceso aleatorio necesitaría situar en memoria una tabla que tuviera tantas filas como páginas existan en memoria virtual ( $2^{13}$ ). Cada fila debería contener un bit de validez y 6 bits para indicar el bloque en el que dicha página se encuentra mapeada en memoria física (si es el caso).

$2^{13}$ {	BV	Bloque (6 bits)

$2^{13} \cdot (1+6) = 57344$  bits.

La implementación de la tabla de páginas mediante una memoria asociativa necesitaría de una memoria que tuviera tantas filas como bloques existan en memoria física ( $2^6$ ). Cada fila debería contener un bit de validez, 13 bits para indicar la página y 6 bits para indicar el bloque en el que dicha página se encuentra mapeada en memoria física.

$2^6$ {	BV	Página (13 bits)	Bloque (6 bits)

$2^6 \cdot (1+13+6) = 1280$  bits.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 20 de 22

### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

4. Un proceso tiene una extensión de 32 páginas consecutivas en memoria virtual. El contenido de la tabla de páginas del proceso es el siguiente:

Página	Bloque
0	10
4	1
9	5

Página	Bloque
11	14
12	31
16	7

Página	Bloque
20	20
24	39
30	9

Realizar la traducción de las siguientes direcciones virtuales a direcciones físicas:

##### 4.1. 000020

El procedimiento para realizar la traducción de direcciones virtuales a direcciones físicas consiste en representar la dirección virtual según el formato de dirección virtual, identificar la página, consultar en la tabla de páginas el bloque correspondiente en memoria y calcular la dirección física sustituyendo el número de página por el de bloque.

Dirección virtual	Página	Desplazamiento
000020	00 0000 0000 000	0 0010 0000
↓		
Página 0		
↓		
Bloque 10		
1420	001 010	0 0010 0000
Dirección física	Bloque	Desplazamiento

##### 4.2. 0E8000

Dirección virtual	Página	Desplazamiento
0E8000	00 1110 1000 000	0 0000 0000
↓		
Página 1856		

El proceso tiene una extensión de 32 páginas, por lo que la página 1856 no pertenece al proceso. Por tanto, se trata de una dirección que no pertenece al proceso. Su traducción no es posible.

##### 4.3. 0011A4

Dirección virtual	Página	Desplazamiento
0011A4	00 0000 0001 000	1 1010 0100
↓		
Página 8		

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 21 de 22

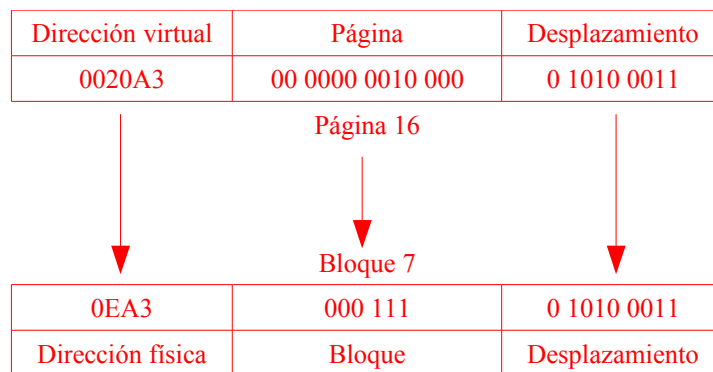
### Relación de ejercicios

#### **Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador**

---

La página 8 no se encuentra en memoria, por lo que se produce un fallo de página. La dirección se podrá traducir cuando la página sea ubicada en un bloque y se conozca el valor de este bloque.

4.4. 0020A3



#### **Ejercicio 13**

En un sistema de memoria virtual de segmentación paginado, el espacio de direccionamiento lógico consta de 128 segmentos. Cada segmento puede tener hasta 32 páginas de 4K palabras cada una. La memoria física consta de 4K bloques de 4K palabras cada una. Obtener los formatos de dirección lógica y física.

# ARQUITECTURA DE COMPUTADORES

## 2º Grado en Ingeniería Informática

Curso 2011 – 2012

Página 22 de 22

### Relación de ejercicios

#### Tema 5: Unidad de memoria. Jerarquía de la memoria de un computador

##### Ejercicio 14

En un sistema de memoria virtual de segmentación paginado, con un tamaño de página de 256 palabras, las tablas de segmento y página se encuentran en un instante determinado como indica la siguiente figura (todos los números están en hexadecimal):

Tabla de segmentos		Tabla de páginas	
Segmento	Dir. Tabla de páginas	Página	Bloque
0		0	012
1		1	000
2		2	019
3			
4		64	A61
5			
6		9A	8C7
7			
8			
9			
A			
B			
C			
D			
E			
F			

Tabla de páginas	
Página	Bloque
0	BA1
1F	6C2
A5	120
A6	FF0

Determinar:

1. Formato de dirección lógica y física.
2. Tamaño de la memoria lógica y física.
3. Rango de direcciones de memoria virtual que se encuentran en memoria física y rango de direcciones físicas que ocupan.
4. Contenido de la TLB suponiendo que todas las páginas indicadas en la figura han sido accedidas al menos una vez.