

ESTRUCTURA Y TECNOLOGIA DE COMPUTADORES II

TEMA 2: *Unidad de Memoria*

**SOLUCION A LOS PROBLEMAS
PROPUESTOS EN EXAMEN**

**Curso 2007-2008
Jose Manuel Díaz Martínez
Tutor de la asignatura ETC-II**

CONTENIDO

SOLUCION PROBLEMA 2.1	4
SOLUCION PROBLEMA 2.2	5
SOLUCION PROBLEMA 2.3	6
SOLUCION PROBLEMA 2.4	8
SOLUCION PROBLEMA 2.5	9
SOLUCION PROBLEMA 2.6	10
SOLUCION PROBLEMA 2.7	11
SOLUCION PROBLEMA 2.8	13
SOLUCION PROBLEMA 2.9	14
SOLUCION PROBLEMA 2.10	15
SOLUCION PROBLEMA 2.11	16
SOLUCION PROBLEMA 2.12	17
SOLUCION PROBLEMA 2.13	18
SOLUCION PROBLEMA 2.14	19
SOLUCION PROBLEMA 2.15	19
SOLUCION PROBLEMA 2.16	21
SOLUCION PROBLEMA 2.17	22
SOLUCION PROBLEMA 2.18	23
SOLUCION PROBLEMA 2.19	24
SOLUCION PROBLEMA 2.20	24
SOLUCION PROBLEMA 2.21	24
SOLUCION PROBLEMA 2.22	25
SOLUCION PROBLEMA 2.23	26
SOLUCION PROBLEMA 2.24	26
SOLUCION PROBLEMA 2.25	27
SOLUCION PROBLEMA 2.26	29
SOLUCION PROBLEMA 2.27	30
SOLUCION PROBLEMA 2.28	34
SOLUCION PROBLEMA 2.29	37
SOLUCION PROBLEMA 2.30	38
SOLUCION PROBLEMA 2.31	39
SOLUCION PROBLEMA 2.32	41
SOLUCION PROBLEMA 2.33	42
SOLUCION PROBLEMA 2.34	44
SOLUCION PROBLEMA 2.35	46
SOLUCION PROBLEMA 2.36	47
SOLUCION PROBLEMA 2.37	47
SOLUCION PROBLEMA 2.38	48
SOLUCION PROBLEMA 2.39	49
SOLUCION PROBLEMA 2.40	50
SOLUCION PROBLEMA 2.41	50
SOLUCION PROBLEMA 2.42	51
SOLUCION PROBLEMA 2.43	52
SOLUCION PROBLEMA 2.44	52
SOLUCION PROBLEMA 2.45	53
SOLUCION PROBLEMA 2.46	53
SOLUCION PROBLEMA 2.47	54
SOLUCION PROBLEMA 2.48	55
SOLUCION PROBLEMA 2.49	56
SOLUCION PROBLEMA 2.50	56
SOLUCION PROBLEMA 2.51	57
SOLUCION PROBLEMA 2.52	57
SOLUCION PROBLEMA 2.53	58
SOLUCION PROBLEMA 2.54	59
SOLUCION PROBLEMA 2.55	61

SOLUCION PROBLEMA 2.56	61
SOLUCION PROBLEMA 2.57	63
SOLUCION PROBLEMA 2.58	64
SOLUCION PROBLEMA 2.59	64
SOLUCION PROBLEMA 2.60	65
SOLUCION PROBLEMA 2.61	66
SOLUCION PROBLEMA 2.62	67
SOLUCION PROBLEMA 2.63	67
SOLUCION PROBLEMA 2.64	69
SOLUCION PROBLEMA 2.65	70
SOLUCION PROBLEMA 2.66	71
SOLUCION PROBLEMA 2.67	72
SOLUCION PROBLEMA 2.68	73
SOLUCION PROBLEMA 2.69	75
SOLUCION PROBLEMA 2.70	77
SOLUCION PROBLEMA 2.71	78
SOLUCION PROBLEMA 2.72	79
SOLUCION PROBLEMA 2.73	80
SOLUCION PROBLEMA 2.74	81
SOLUCION PROBLEMA 2.75	85
SOLUCION PROBLEMA 2.76	86
SOLUCION PROBLEMA 2.77	88
SOLUCION PROBLEMA 2.78	89
SOLUCION PROBLEMA 2.79	90
SOLUCION PROBLEMA 2.80	91
SOLUCION PROBLEMA 2.81	91
SOLUCION PROBLEMA 2.82	92
SOLUCION PROBLEMA 2.83	93

SOLUCION PROBLEMA 2.1**DATOS**

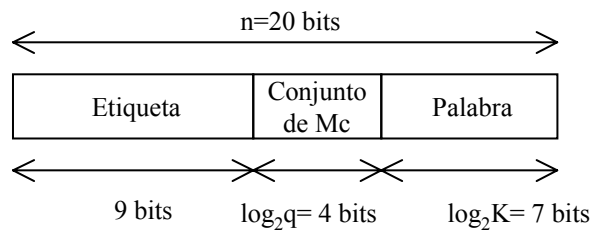
- Memoria caché (M_c) asociativa por conjuntos.
 - $q=16$ (2^4) conjuntos
 - $r=4$ particiones/ conjunto
- Memoria principal $C_{Mp} = 1\text{ M}$ (2^{20}) palabras
 - $K=128$ (2^7) palabras/bloque

Puesto que la M_p tiene 2^{20} palabras se requieren direcciones de longitud $n=\log_2 C_{Mp} = \log_2 2^{20} = 20$ bits.

En segundo lugar se va a calcular el número de bloques C de la M_c

$$C = q \cdot r = 16 \text{ conjuntos} \cdot 4 \frac{\text{bloques}}{\text{conjunto}} = 2^6 \text{ bloques en } M_c$$

El formato de una dirección de la M_c que utiliza una función de correspondencia asociativa por conjuntos es:



Luego la dirección que se nos plantea es:

Etiqueta	Conjunto de M_c	Palabra
101 000 011	0010	0110000

El conjunto de M_c al que hace referencia esta dirección es $0010 = 2_{10}$

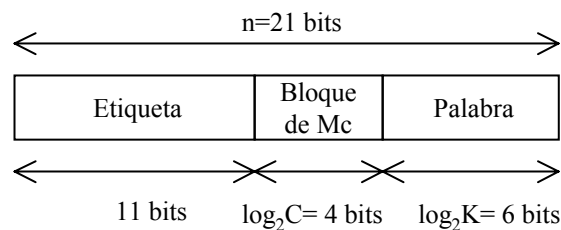
SOLUCION PROBLEMA 2.2**DATOS**

- Memoria principal $C_{Mp} = 2\text{ M } (2^{21})$ palabras
- Memoria caché $C_{Mc} = 1\text{ K } (2^{10})$ palabras
 - $K=64 (2^6)$ palabras/bloque

De los datos del enunciado se deduce que el número de bloques C de M_c es:

$$C = \frac{2^{10} \text{ pal}}{2^6 \text{ pal/bloque}} = 2^4 \text{ bloques}$$

- a) La memoria caché utiliza *función de correspondencia directa*. Por lo tanto el formato de una dirección desde el punto de vista de la memoria caché es :



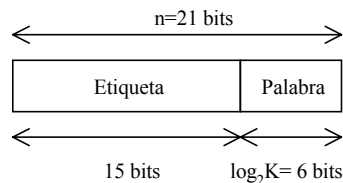
Luego dada la dirección en binario del enunciado,

000 000 001 10	0100	110000
----------------	------	--------

El bloque de la M_c al que hace referencia es el número 4_{10} (**0100**).

La etiqueta que habría que buscar es 6_{10} (**000 000 001 10**).

- b) La memoria caché utiliza *función de correspondencia totalmente asociativa*. Por lo tanto el formato de una dirección desde el punto de vista de la memoria caché es :



Luego dada la dirección en binario del enunciado,

000 000 001 100 100	110 000
---------------------	---------

La etiqueta que habría que buscar es 100_{10} (**000 000 001 100 100**).

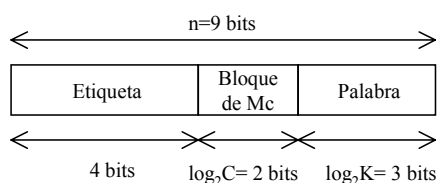
SOLUCION PROBLEMA 2.3**DATOS**

- $C_{Mp} = 512 (2^9)$ palabras
- $C_{Mc} = 32 (2^5)$ palabras
- El tamaño de partición es $K=8 (2^3)$ palabras/bloque
 - La caché está inicialmente vacía.

De los datos del enunciado se pueden calcular el número de bloques C de la M_c .

$$C = \frac{2^5}{2^3} = 2^2 = 4 \text{ bloques.}$$

a) La caché emplea *correspondencia directa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

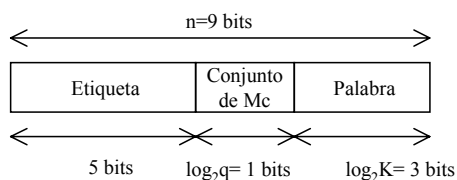
Direcciones leídas	Nº de bloque (j) de la M_p al que hace referencia la dirección.	Nº bloque (i) de la M_c al que hace referencia la dirección	Fallo o acierto
000 0 00 000	0	0	Fallo se carga j=0 en i=0
000 0 00 001	0	0	Acierto
000 0 00 011	0	0	Acierto
000 1 00 001	4	0	Fallo se carga j=4 en i=0
000 1 00 101	4	0	Acierto
000 0 10 000	2	2	Fallo se carga j=2 en i=2
000 0 10 010	2	2	Acierto
000 0 00 000	0	0	Fallo se carga j=0 en i=0

Nº Total de fallos=4

b) La caché emplea *correspondencia asociativa por conjuntos*, con $r=2$ bloques/conjunto. El algoritmo de reemplazamiento utilizado es FIFO (First-In First-Out). Se sustituye el bloque más antiguo en caché.

El número de conjuntos es $q=4/2=2$

La dirección desde el punto de vista de la M_c tendría los siguientes campos :



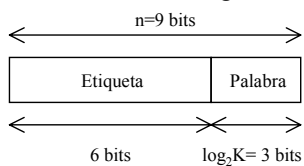
En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Nº conjunto (i) de la Mc al que hace referencia la dirección	Fallo o acierto
000 00 0 000	0	0	Fallo se carga el bloque j=0 en el bloque 0 del conjunto i=0
000 00 0 001	0	0	Acierto
000 00 0 011	0	0	Acierto
000 10 0 001	4	0	Fallo se carga el bloque j=4 en el bloque 1 del conjunto i=0
000 10 0 101	4	0	Acierto
000 01 0 000	2	0	Fallo se carga el bloque j=2 en el bloque 0 del conjunto i=0
000 01 0 010	2	0	Acierto
000 00 0 000	0	0	Fallo se carga el bloque j=0 en el bloque 1 del conjunto i=0

Nº Total de fallos=4

c) La caché emplea *correspondencia totalmente asociativa*. El algoritmo de reemplazamiento utilizado es LRU (Least Recently Used). Se sustituye el bloque utilizado menos recientemente.

La dirección desde el punto de vista de la M_c tendría los siguientes campos :



En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Fallo o Acierto
000 00 0 000	0	Fallo se carga el bloque j=0 en el bloque i=0
000 00 0 001	0	acierto
000 00 0 011	0	acierto
000 10 0 001	4	Fallo se carga el bloque j=4 en el bloque i=1
000 10 0 101	4	acierto
000 01 0 000	2	Fallo se carga el bloque j=2 en el bloque i=2
000 01 0 010	2	acierto
000 00 0 000	0	acierto

Nº Total de fallos = 3

SOLUCION PROBLEMA 2.4

DATOS

- Sistema jerárquico de memoria: Memoria Caché M_c y Memoria Principal M_p
- M_c : Capacidad 256 palabras y tiempo de acceso $t_{ac}=10$ nseg
- M_p : Capacidad 1024 Kpalabras y tiempo de acceso $t_{ap}= 100$ nseg
- Tasa de acierto $h=0.9$
- Cuando se produce un fallo en M_c :
 - El sistema tarda $t_{ex}=20$ nseg adicionales en tratar el fallo.
 - La palabra solicitada es enviada de M_p a M_c donde es leída por la CPU.

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que:

$$t_{ga} = t_{ac} = 10 \text{ nseg}$$

Por otra parte t_{gf} es el tiempo que tarda la CPU en gestionar un *fallo* en la M_c . De acuerdo con el enunciado cuando se produce un fallo:

- 1) El sistema utiliza un tiempo t_{ex} extra.
- 2) La palabra es enviada desde la M_p a la M_c , para lo que empleara un tiempo t_{ap} .
- 3) Finalmente dicha palabra es leída por la CPU en la M_c , por lo que empleará un tiempo t_{ac} .

$$t_{gf} = t_{ex} + t_{ap} + t_{ac} = 20 + 100 + 10 = 130 \text{ nseg}$$

Sustituyendo valores en la formula (1) se obtiene que :

$$t_{am} = 0.9 \cdot 10 + 0.1 \cdot 130 = 9 + 13 = \mathbf{22 \text{ nseg}}$$

SOLUCION PROBLEMA 2.5**DATOS**

- $C_{Mp} = 512 (2^9)$ palabras
- $C_{Mc} = 32 (2^5)$ palabras
- El tamaño de partición es $K=8 (2^3)$ palabras/bloque
- La caché está inicialmente vacía.
- Función de correspondencia totalmente asociativa

De los datos del enunciado se pueden calcular el número de bloques C de la M_c .

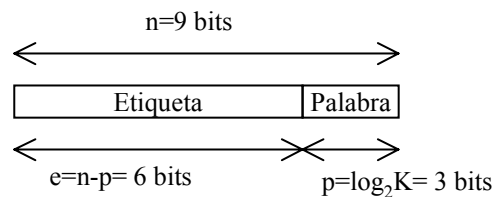
$$C = \frac{C_{Mc}}{K} = \frac{2^5}{2^3} = 2^2 = 4 \text{ bloques.}$$

Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^9}{2^3} = 2^6 = 64 \text{ bloques.}$$

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^9 = 9$ bits

Puesto que la caché emplea *correspondencia totalmente asociativa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



Obsérvese como en el caso de emplear *correspondencia totalmente asociativa* la etiqueta representa el número de bloque j de la M_p , con lo que el tamaño de la etiqueta también se podía haber calculado de la siguiente forma $e = \log_2 M = \log_2 2^6 = 6$ bits

En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c . Recuérdese que con la letra i se denota el número de bloque de memoria caché.

Direcciones leídas	Etiqueta= N° de bloque j de M_p	Fallo o acierto
000000 000	0	Fallo se carga $j=0$ en $i=0$
000000 001	0	Acierto
000000 011	0	Acierto
000100 001	4	Fallo se carga $j=4$ en $i=1$
000100 101	4	Acierto
000010 000	2	Fallo se carga $j=2$ en $i=2$
000010 010	2	Acierto
000000 000	0	Acierto

N° Total de fallos=3

SOLUCION PROBLEMA 2.6

El primer paso a realizar es tachar las columnas de la memoria asociativa cuyo bit del registro de MASCARA se encuentre a 0. Es decir el argumento que hay que comparar es 0X0X11X0 (donde la X significa que ese bit no se compara).

0	1	0	1	1	1	0	0	ARGUMENTO
1	0	1	0	1	1	0	1	MASCARA
1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	
								?
								?
								?
								?
								?

El segundo paso es ir comparando los bits no tachados del registro ARGUMENTO, 0X0X11X0, con los bits no tachados de cada una de las filas de la matriz asociativa. Si son iguales se coloca un 1 y en caso contrario se coloca un 0, en el bit de la fila correspondiente del registro de MARCA.

0	1	0	1	1	1	0	0	ARGUMENTO
1	0	1	0	1	1	0	1	MASCARA
1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	
								?
								?
								?
								?
								?

Sólo la fila 5 presenta el valor 0X0X11X0 igual que el registro ARGUMENTO. Luego el valor del registro de Marca resultante es:

0
0
0
0
1

SOLUCION PROBLEMA 2.7**DATOS**

- $C_{Mp} = 4096 (2^{12})$ palabras
- $C_{Mc} = 64 (2^6)$ palabras
- $K=16 (2^4)$ palabras/bloque
- Caché inicialmente vacía.
- Función de correspondencia directa

De los datos del enunciado se pueden calcular el número de bloques C de la M_c .

$$C = \frac{C_{Mc}}{K} = \frac{2^6}{2^4} = 2^2 = 4 \text{ bloques.}$$

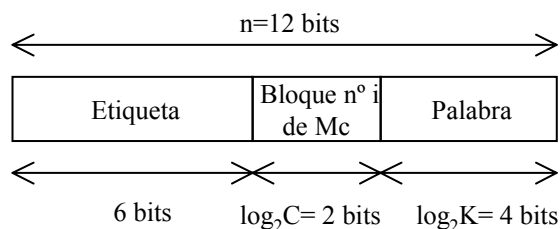
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^{12}}{2^4} = 2^8 = 256 \text{ bloques.}$$

Es decir, que los 8 bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^{12} = 12$ bits

Puesto que la caché emplea *correspondencia directa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la M_p al que hace referencia la dirección.	Nº bloque (i) de la M_c al que hace referencia la dirección	Fallo o acierto
000000 01 0000	1	1	Fallo se carga j=1 en i=1 que inicialmente esta vacío
000100 01 0100	17	1	Fallo se carga j=17 en i=1
000001 00 1000	4	0	Fallo se carga j=4 en i=0 que inicialmente esta vacío
000001 11 1000	7	3	Fallo se carga j=7 en i=3 que inicialmente esta vacío

Afirmación I:

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Nº bloque (i) de la Mc al que hace referencia la dirección	Fallo o acierto
000000 01 0100	1	1	Fallo se carga j=1 en i=1
010011 10 1100	78	2	Fallo se carga j=78 en i=2 que inicialmente esta vacío

La afirmación es **FALSA**, ya que aunque efectivamente se producen dos fallos, sólo hay que realizar un reemplazamiento de un bloque existente ya en caché.

Afirmación II:

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Nº bloque (i) de la Mc al que hace referencia la dirección	Fallo o acierto
000001 11 0001	7	3	Acierto

La afirmación es **FALSA**, ya que se produce un acierto y en consecuencia obviamente no hay que realizar ningún reemplazamiento.

SOLUCION PROBLEMA 2.8

Para resolver este problema los pasos a seguir son:

- 1) Tachar las filas de la memoria asociativa cuyo bit del registro de marca está a 0.
- 2) Tachar las columnas de la matriz asociativa cuyo bit del registro de máscara está a 0. Como se muestra en la figura :

?	?	?	?	?	?	?	?	?	ARGUMENTO
1	0	0	1	0	1	0	0	0	MASCARA
1	1	0	1	1	1	1	1	1	
1	1	0	1	1	0	1	0	0	
0	0	0	1	0	0	0	0	1	
1	0	1	0	0	1	1	0	0	
1	1	0	1	1	0	0	0	1	
									MARCA
									0
									1
									0
									0
									1

A la vista de la matriz asociativa el argumento que genera este registro de marca debe de tener la siguiente estructura: 1XX1X0XX donde la X representa que el valor de ese bit no importa

- 3) En la lista de argumentos dados se tachan las mismas columnas que se han tachado en la memoria asociativa. Como se muestra en la figura.

1	0	1	1	0	1	0	0	Argumento 1
0	1	0	1	1	1	1	1	Argumento 2
1	0	0	1	1	0	0	0	Argumento 3
1	1	1	1	1	0	1	0	Argumento 4
0	1	0	1	0	0	1	1	Argumento 5

- 5) Comparando cada uno de los argumentos con 1XX1X0XX. Se puede ver que sólo el argumento 3 y el 4 producirían el registro de marca dado.

1	0	1	1	0	1	0	0	Argumento 1
0	1	0	1	1	1	1	1	Argumento 2
1	0	0	1	1	0	0	0	Argumento 3
1	1	1	1	1	0	1	0	Argumento 4
0	1	0	1	0	0	1	1	Argumento 5

Luego la solución es: **2 argumentos.**

SOLUCION PROBLEMA 2.9**DATOS**

- $C_{Mp} = 4096 (2^{12})$ palabras
- $C_{Mc} = 128 (2^7)$ palabras
- $K=16 (2^4)$ palabras/bloque
- Caché inicialmente vacía.
- Función de correspondencia totalmente asociativa

De los datos del enunciado se pueden calcular el número de bloques C de la M_c .

$$C = \frac{C_{Mc}}{K} = \frac{2^7}{2^4} = 2^3 = 8 \text{ bloques.}$$

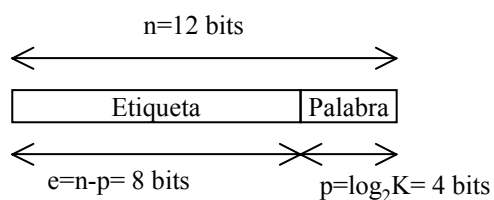
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^{12}}{2^4} = 2^8 = 256 \text{ bloques.}$$

Es decir, que los $\log_2 M = 8$ bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^{12} = 12$ bits

Puesto que la caché emplea *correspondencia totalmente asociativa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



Obsérvese como en el caso de emplear *correspondencia totalmente asociativa* la etiqueta representa el número de bloque j de la M_p , con lo que el tamaño de la etiqueta también se podía haber calculado de la siguiente forma $e = \log_2 M = \log_2 2^8 = 8$ bits

En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la M_p al que hace referencia la dirección.	Fallo o acierto
0000 0001 0000	1	Fallo se carga j=1 en i=0 que inicialmente esta vacío
0001 0001 0100	17	Fallo se carga j=17 en i=1 que inicialmente esta vacío
0000 0100 1000	4	Fallo se carga j=4 en i=2 que inicialmente esta vacío
0000 0111 1000	7	Fallo se carga j=7 en i=3 que inicialmente esta vacío

Afirmación I: Es **FALSA**, ya que se producen 4 fallos.

Afirmación II:

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Fallo o acierto
0110 0111 0100	103	Fallo se carga j=103 en i=4 que inicialmente está vacía

Es **FALSA**, ya que aunque efectivamente se produce un Fallo, no hay que reemplazar bloque de la caché ya que los bloques i=4, 5, 6, 7 se encuentran vacíos.

SOLUCION PROBLEMA 2.10**DATOS**

- $n=16$ bits
- $C_{Mc} = 2^{10}$ palabras
- $K=16$ (2^4) palabras/bloque
- Función de correspondencia asociativa por conjuntos.
- $r=4$ bloques/conjunto

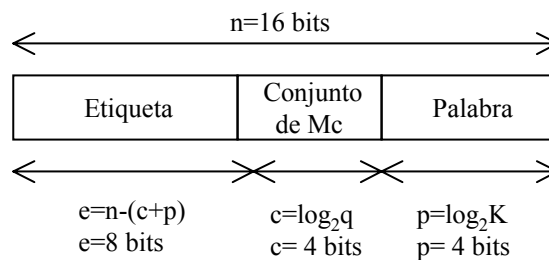
De los datos del enunciado se pueden calcular el número de bloques C de la M_C .

$$C = \frac{C_{Mc}}{K} = \frac{2^{10}}{2^4} = 2^6 = 64 \text{ bloques.}$$

Por otra parte, conocido C es posible calcular el número de conjuntos q de la M_C

$$q = \frac{C(\text{bloques})}{r(\text{bloques / conjunto})} = \frac{2^6}{2^2} = 2^4 = 16 \text{ conjuntos.}$$

Puesto que la caché emplea *correspondencia asociativa por conjuntos*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



Por lo tanto la dirección del enunciado se divide en tres campos:

0000 1101	1100	0011
-----------	------	------

El valor de sus campos en decimal es:

Etiqueta = $(0000 \ 1101)_2 = 13_{10}$
 Conjunto de M_c = $(1100)_2 = 12_{10}$
 Palabra = $(0011)_2 = 3_{10}$

13	12	3
----	----	---

SOLUCION PROBLEMA 2.11

Una posible solución a este problema consiste en ir probando uno a uno con los registros de máscara dados para verificar si producen el registro de Marca especificado, con lo que sería resolver 5 veces un problema como el problema 2.6

1) La Máscara 1 **si** que produce el registro de Marca especificado.

0	1	0	1	1	1	0	0	ARGUMENTO
0	0	1	0	1	0	0	1	MÁSCARA 1

1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	

0
1
0
0
1

MARCA

2) La Máscara 2 **no** produce el registro de Marca especificado

0	1	0	1	1	1	0	0	ARGUMENTO
1	0	1	0	0	0	0	1	MÁSCARA 2

1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	

0
1
1
0
1

MARCA

3) La Máscara 3 **si** produce el registro de Marca especificado

0	1	0	1	1	1	0	0	ARGUMENTO
1	0	1	0	1	0	0	1	MÁSCARA 3

1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	

0
1
0
0
1

MARCA

3) La Máscara 4 **si** produce el registro de Marca especificado

0	1	0	1	1	1	0	0	ARGUMENTO
1	0	1	1	1	0	0	1	MÁSCARA 4

1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	

0
1
0
0
1

MARCA

3) La Máscara 5 **si** produce el registro de Marca especificado

0	1	0	1	1	1	0	0	ARGUMENTO
0	0	1	0	1	1	0	1	MÁSCARA 5

1	0	0	1	1	1	1	1	
0	1	0	1	1	0	1	0	
0	1	0	1	0	0	1	0	MARCA
0	1	1	0	1	1	1	0	
0	0	0	1	1	1	1	0	

0
0
0
0
1

Luego la solución es **3 registros de máscara** producen el registro de Marca especificado.

SOLUCION PROBLEMA 2.12

DATOS

- Memoria de acceso aleatorio
- $t_A = 80 \text{ nseg.}$
- $t_C = 100 \text{ nseg}$

El tiempo de acceso t_A se define como el tiempo necesario para leer o escribir un determinado número de bits. O también como el tiempo transcurrido desde que se solicita una información hasta que se dispone de ella.

Por otro lado el tiempo de ciclo, t_C , se define como el tiempo transcurrido entre dos lecturas consecutivas en memoria. Y limitará la frecuencia de acceso en una memoria de acceso aleatorio.

Por definición, la frecuencia de acceso f_A o velocidad de transferencia v_T se define como el número de palabras que se pueden leer o escribir en memoria por unidad de tiempo.

$$f_A = \frac{1}{t_C}$$

Luego sustituyendo, se obtiene:

$$f_A = \frac{1}{100 \cdot 10^{-9}} = \frac{1}{10^{-7}} = 10^7 \text{ seg}^{-1}$$

SOLUCION PROBLEMA 2.13

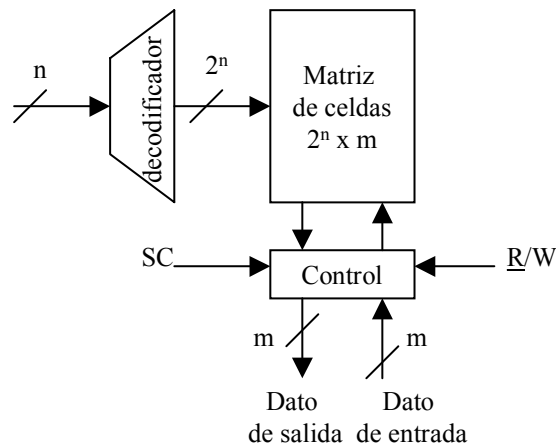
DATOS

- Se desea diseñar un módulo de memoria de capacidad 256 palabras de 4 bits

Con el dato que proporcionan en el enunciado es posible deducir los n bits necesarios para codificar todas las posiciones (palabras) de memoria, es decir, conocer cual es la anchura de una dirección de memoria. Puesto que $256=2^8 \rightarrow n=\log_2 2^8=8$ bits.

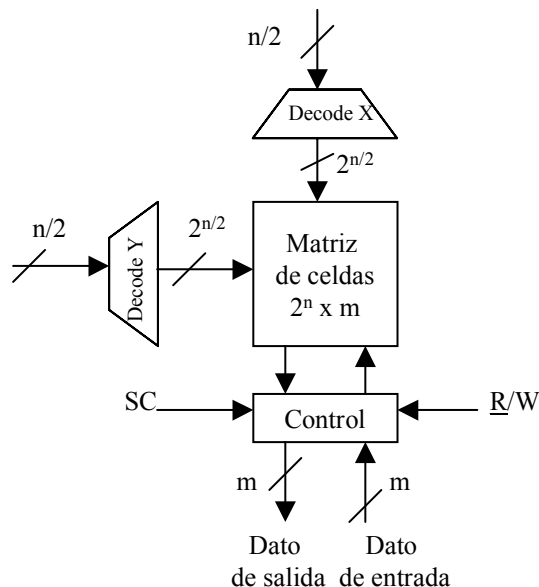
Análisis de la Afirmación I

Si se utiliza **organización 2D** (ver libro de teoría pags.62-64), se requiere un decodificar que reciba los $n=8$ bits del bus de dirección y genere las $2^n=2^8$ posiciones de memoria, es decir que posea 8 entradas y 2^8 salidas. Luego la afirmación I es **verdadera**.



Análisis de la Afirmación II

Si se utiliza **organización 21/2D** (ver libro de teoría pags.65-68), se requieren dos decodificares, cada uno con $n/2=4$ entradas y $2^{n/2}=2^4$ salidas. Luego la afirmación II es **verdadera**.



SOLUCION PROBLEMA 2.14**DATOS**

- Número total de accesos $N_T=2^{30}$
- Tasa de aciertos $h=75\%$

La definición de tasa de fallos ($1-h$) es:

$$1-h = \frac{N_F}{N_T}$$

Despejando el número total de Fallos:

$$N_F = (1-h)N_T$$

Sustituyendo valores se obtiene:

$$N_F = (1-0.75)2^{30} = 0.25 \cdot 2^{30} = \frac{2^{30}}{2^2} = 2^{28}$$

Luego el **número total de fallos** es $N_F = 2^{28}$.

SOLUCION PROBLEMA 2.15**DATOS**

- $C_{Mp} = 256 (2^8)$ palabras
- $C_{Mc} = 32 (2^5)$ palabras
- $K=8 (2^3)$ palabras/bloque
- Caché inicialmente vacía.
- Reemplazamiento FIFO
- Función de correspondencia totalmente asociativa

De los datos del enunciado se pueden calcular el número de bloques C de la M_C .

$$C = \frac{C_{Mc}}{K} = \frac{2^5}{2^3} = 2^2 = 4 \text{ bloques.}$$

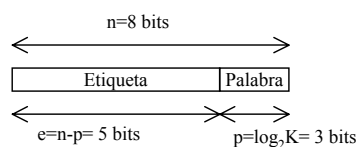
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^8}{2^3} = 2^5 = 32 \text{ bloques.}$$

Es decir, que los $\log_2 M = 5$ bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^8 = 8$ bits

Puesto que la caché emplea *correspondencia totalmente asociativa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



Obsérvese como en el caso de emplear *correspondencia totalmente asociativa* la etiqueta representa el número de bloque j de la M_p , con lo que el tamaño de la etiqueta también se podía haber calculado de la siguiente forma $e = \log_2 M = \log_2 2^5 = 5$ bits

En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la M_p al que hace referencia la dirección.	Fallo o acierto
00000 000	0	Fallo se carga $j=0$ en $i=0$ que inicialmente esta vacío
00000 001	0	Acierto
00000 011	0	Acierto
00100 001	4	Fallo se carga $j=4$ en $i=1$ que inicialmente esta vacío
00100 101	4	Acierto
00010 000	2	Fallo se carga $j=2$ en $i=2$ que inicialmente esta vacío
00010 010	2	Acierto
00000 000	0	Acierto

Número total de fallos: **3 Fallos**

SOLUCION PROBLEMA 2.16

A la hora de calcular el valor del registro de marca sólo hay que tener en cuenta las palabras activas, es decir, aquellas cuyo bit de etiqueta está a 1. Luego a la vista del registro de etiqueta las palabras 4 y 5 no se encuentran activas y los bits correspondientes del registro de marca deben ser 0.

		<table><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>								1	0	0	1	1	0	1	0	1	0	0	1	0	1	0	1	ARGUMENTO																																			
1	0	0	1	1	0	1	0																																																						
1	0	0	1	0	1	0	1																																																						
		<table><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>								1	0	0	1	0	1	0	1	MASCARA																																											
1	0	0	1	0	1	0	1																																																						
ETIQUETA	<table><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	1	1	1	0	0	<table><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>								1	1	0	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	0	0	<table><tr><td>?</td></tr><tr><td>?</td></tr><tr><td>?</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	?	?	?	0	0	MARCA
	1																																																												
	1																																																												
	1																																																												
	0																																																												
0																																																													
1	1	0	1	1	1	1	1																																																						
1	1	0	1	1	0	1	0																																																						
1	0	0	1	0	0	0	0																																																						
1	0	1	0	0	1	1	0																																																						
1	1	0	1	1	0	0	0																																																						
?																																																													
?																																																													
?																																																													
0																																																													
0																																																													

Por otro lado y de acuerdo al funcionamiento estándar de una memoria asociativa, en las palabras activas sólo se consideran los bits del argumento cuya correspondiente bit de máscara está a 1. Por lo tanto las columnas 2, 3, 5 y 7 de la memoria asociativas pueden ser tachadas ya que no van a ser comparadas.

		<table><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>								1	0	0	1	1	0	1	0	1	0	0	1	0	1	0	1	ARGUMENTO																																			
1	0	0	1	1	0	1	0																																																						
1	0	0	1	0	1	0	1																																																						
		<table><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>								1	0	0	1	0	1	0	1	MASCARA																																											
1	0	0	1	0	1	0	1																																																						
ETIQUETA	<table><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	1	1	1	0	0	<table><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>								1	1	0	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	0	0	<table><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	0	1	1	0	0	MARCA
	1																																																												
	1																																																												
	1																																																												
	0																																																												
0																																																													
1	1	0	1	1	1	1	1																																																						
1	1	0	1	1	0	1	0																																																						
1	0	0	1	0	0	0	0																																																						
1	0	1	0	0	1	1	0																																																						
1	1	0	1	1	0	0	0																																																						
0																																																													
1																																																													
1																																																													
0																																																													
0																																																													

El argumento a comparar de acuerdo con el registro de máscara es: $1 \times 1 \times 0 \times 0$. Luego la fila 1 de la memoria asociativa al tener $1 \times 1 \times 1 \times 1$ dará un 0 en su bit del registro de marca. Mientras que las filas 2 y 3 de la memoria asociativa al tener el mismo valor que el argumento $1 \times 1 \times 0 \times 0$ darán un 1 en su bit del registro de marca.

La marca correcta es la que corresponde a la respuesta C :

0
1
1
0
0

SOLUCION PROBLEMA 2.17**DATOS**

- $C_{Mp} = 256 (2^8)$ palabras
- $C_{Mc} = 32 (2^5)$ palabras
- $K=8 (2^3)$ palabras/bloque
- Caché inicialmente vacía.
- Reemplazamiento FIFO
- Función de correspondencia totalmente asociativa

El desarrollo de este problema es exactamente igual que el del Problema 2.15, lo único que varía es que ahora se considera que se lee además la dirección 00100 111, que se ha añadido al final de la tabla construida en el Problema 2.15.

Direcciones leídas	Nº de bloque (j) de la Mp al que hace referencia la dirección.	Fallo o acierto
00000 000	0	Fallo se carga j=0 en i=0 que inicialmente esta vacío
00000 001	0	Acierto
00000 011	0	Acierto
00100 001	4	Fallo se carga j=4 en i=1 que inicialmente esta vacío
00100 101	4	Acierto
00010 000	2	Fallo se carga j=2 en i=2 que inicialmente esta vacío
00010 010	2	Acierto
00000 000	0	Acierto
00100 111	4	Acierto

Afirmación I: Es **VERDADERA**, al leer dicha dirección se produce un acierto ya que el bloque j=4 ya se encuentra cargado en la M_C , en concreto en el bloque i=1.

Afirmación II: Es **FALSA**.

SOLUCION PROBLEMA 2.18**DATOS**

- Sistema jerárquico de memoria: Memoria Caché M_c y Memoria Principal M_p
- M_c : Capacidad 256 palabras, $K=8$ pal/ bloque y tiempo de acceso $t_{ac}=10$ nseg
- M_p : Capacidad 1024 Kpalabras y tiempo de acceso $t_p=100$ nseg
- Tasa de acierto $h=0.9$
- Cuando se produce un fallo en M_c :
 - Se mueve el dato a la CPU.
 - Simultáneamente se mueve el bloque desde la M_p a la M_c

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que :

$$t_{ga} = t_{ca} = 10 \text{ nseg}$$

Por otra parte t_{gf} es el tiempo que tarda la CPU en gestionar un fallo en M_c . De acuerdo con el enunciado cuando se produce un fallo, el sistema entonces simultáneamente mueve el dato a la CPU empleando un tiempo t_{ap} y mueve el bloque que consta de 8 palabras desde la M_p a la M_c , este movimiento emplea un tiempo de $8 \cdot t_{ap}$. Luego t_{gf} es:

$$t_{gf} = \max\{8 \cdot t_{ap}, t_{ap}\} = 8 \cdot t_{ap} = 8 \cdot 100 = 800 \text{ nseg}$$

Sustituyendo valores en la formula (1) se obtiene que:

$$t_{am} = 0.9 \cdot 10 + 0.1 \cdot 800 = 9 + 80 = \mathbf{89 \text{ nseg}}$$

SOLUCION PROBLEMA 2.19

DATOS

- Sistema jerárquico de memoria: Memoria Caché M_c y Memoria Principal M_p
- M_c : Capacidad 256 palabras, $K=8$ pal/ bloque y tiempo de acceso $t_{ca}=10$ nseg
- M_p : Capacidad 1024 Kpalabras y tiempo de acceso $t_{ap}=100$ nseg
- Tasa de acierto $h=0.9$
- Cuando se produce un fallo en M_c :
 - Primero se mueve el bloque completo desde M_p a M_c
 - Después la CPU lee el dato en la M_c

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que:

$$t_{ga} = t_{ac} = 10 \text{ nseg}$$

Por otra parte t_{gf} es el tiempo que tarda la CPU en gestionar un fallo en M_c . De acuerdo con el enunciado cuando se produce un Fallo, el sistema primero mueve el bloque que consta de 8 palabras desde la M_p a la M_c , este movimiento emplea un tiempo de $8 \cdot t_{ap}$, y a continuación lee el dato desde la caché. Luego t_{gf} es:

$$t_{gf} = 8 \cdot t_{ap} + t_{ac} = 8 \cdot 100 + 10 = 810 \text{ nseg}$$

Sustituyendo valores en la formula (1) se obtiene que:

$$t_{am} = 0.9 \cdot 10 + 0.1 \cdot 810 = 9 + 81 = 90 \text{ nseg}$$

SOLUCION PROBLEMA 2.20

En las memorias asociativas llenas, la escritura de nuevos datos se realiza reemplazando aquellas palabras que no se encuentren activas, es decir, cuyo bit correspondiente en el *registro de etiqueta* sea 0. En relación al problema, el nuevo dato se escribirá en la primera palabra cuyo bit correspondiente en el *registro de etiqueta* sea 0: la **palabra 3**.

SOLUCION PROBLEMA 2.21

DATOS

- $C_{Mp} = 4096$ (2^{12}) palabras
- $C_{Mc} = 64$ (2^6) palabras
- $K=16$ (2^4) palabras/bloque
- Caché inicialmente vacía.
- Función de correspondencia directa.

Afirmación I y Afirmación II: FALSAS. Cuando se utiliza la *función de correspondencia directa* no tiene sentido hablar de conjuntos.

SOLUCION PROBLEMA 2.22**DATOS**

- Procesador con 12 líneas de dirección $A_{11}A_{10}...A_0$
- Para construir su unidad de memoria se dispone de módulos de $1K=(2^{10})$ palabras
- Las líneas más significativas ($A_{11}-A_{10}$) se utilizan para la selección de cada módulo

De los datos se deduce que las direcciones para acceder a posiciones de memoria tienen 12 bits, uno por cada línea. Por otro lado puesto que se tienen dos líneas A_{11} y A_{10} para seleccionar módulos de memoria de 1K, el número total de módulos para implementar el mapa de memoria es $2^2=4$ módulos. En la siguiente tabla se muestra el módulo de memoria al que se hace referencia en función del valor de estas dos líneas.

A_{11}	A_{10}	Módulo de memoria
0	0	Módulo 0
0	1	Módulo 1
1	0	Módulo 2
1	1	Módulo 3

Así las direcciones que hacen referencia al primer módulo de memoria (módulo nº 0) son:

00XX XXXX XXXX

donde X puede valer 0 o 1.

Las direcciones que hacen referencia al segundo módulo de memoria (módulo nº 1) son:

01XX XXXX XXXX

Las direcciones que hacen referencia al tercer módulo de memoria (módulo nº 2) son:

10XX XXXX XXXX

Y las direcciones que hacen referencia al cuarto módulo de memoria (módulo nº 3) son:

11XX XXXX XXXX

En la siguiente tabla aparece la primera dirección (dirección base) y la última de cada módulo de memoria, tanto en binario como en hexadecimal.

Módulo de memoria	Dirección de inicio (Dirección Base)	Dirección final
módulo nº 0	0000 0000 0000 000 (hexadecimal)	0011 1111 1111 3FF (hexadecimal)
módulo nº 1	0100 0000 0000 400 (hexadecimal)	0111 1111 1111 7FF (hexadecimal)
módulo nº 2	1000 0000 0000 800 (hexadecimal)	1011 1111 1111 BFF (hexadecimal)
módulo nº 3	1100 0000 0000 C00 (hexadecimal)	1111 1111 1111 FFF (hexadecimal)

La solución al problema (las direcciones base de cada módulo) está dada por lo tanto en la tabla anterior.

SOLUCION PROBLEMA 2.23

DATOS

- Computador con un sistema jerárquico de memoria.
- El orden de los niveles superiores a los inferiores es:
 - registros de la CPU -----> Nivel Superior
 - caché primaria
 - caché secundaria
 - memoria principal
 - discos magnéticos -----> Nivel Inferior

Cuando se dispone de un **sistema jerárquico de memoria** se cumplen los siguientes principios:

- 1) El coste por palabra es mayor en los niveles superiores que en los inferiores.
- 2) La capacidad de memoria es menor en los niveles superiores que en los inferiores.
- 3) La velocidad de transferencia o frecuencia de acceso es mayor en los niveles superiores que en los inferiores.

Considerando estos tres principios, es posible verificar si las afirmaciones son ciertas o no.

Afirmación I : Es **Falsa**, puesto que la caché primaria está en un nivel superior a la caché secundaria, el coste por palabra es mayor en la caché primaria que en la secundaria.

Afirmación II: Es **Verdadera**, puesto que la caché primaria está en un nivel superior a la caché secundaria, la capacidad es menor en la caché primaria que en la secundaria.

SOLUCION PROBLEMA 2.24

DATOS

- Sistema de memoria caché con correspondencia totalmente asociativa

Afirmación I: Esta afirmación es **Falsa**, por el propio funcionamiento de este tipo de función de correspondencia, cuando la memoria Caché está llena hay que definir una política de reemplazamiento de bloques.

Afirmación II: Esta afirmación es verdadera

SOLUCION PROBLEMA 2.25**DATOS**

- Procesador con 12 líneas de dirección $A_{11}A_{10}...A_0$
- Para construir su unidad de memoria se dispone de módulos de $1K=(2^{10})$ palabras
- Las líneas más significativas ($A_{11}-A_0$) se utilizan para la selección de cada módulo

De los datos se deduce que las direcciones para acceder a posiciones de memoria tienen 12 bits, uno por cada línea. Por otro lado puesto que se tienen dos líneas A_{11} y A_{10} para seleccionar módulos de memoria de 1K, el número total de módulos para implementar el mapa de memoria es $2^2=4$ módulos. En la siguiente tabla se muestra el módulo de memoria al que se hace referencia en función del valor de estas dos líneas.

A_{11}	A_{10}	Módulo de memoria
0	0	Módulo 0
0	1	Módulo 1
1	0	Módulo 2
1	1	Módulo 3

Módulo 0

Así las direcciones que hacen referencia al primer módulo de memoria (módulo nº 0) son:

XXXX XXXX XX00

donde X puede valer 0 o 1.

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 0

Y Y 4

Y Y 8

Y Y C

Módulo 1

Las direcciones que hacen referencia al segundo módulo de memoria (módulo nº 1) son:

XXXX XXXX XX01

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 1

Y Y 5

Y Y 9

Y Y D

Módulo 2

Las direcciones que hacen referencia al tercer módulo de memoria (módulo nº 2) son:

XXXX XXXX XX10

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 2

Y Y 6

Y Y A

Y Y E

Módulo 3

Y las direcciones que hacen referencia al cuarto módulo de memoria (módulo nº 3) son:

XXXX XXXX XX11

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 3

Y Y 7

Y Y B

Y Y F

Conocidas las direcciones que hacen referencia a cada módulo de memoria, examinemos las afirmaciones:

Afirmación I: Es **FALSA**, la dirección C89 se refiere al módulo 1 y la dirección C8A se refiere al módulo 2.

Afirmación II: Es **VERDADERA**, por que efectivamente C89 y C8A se refieren a diferentes módulos, la dirección C89 se refiere al módulo 1 y la dirección C8A se refiere al módulo 2.

SOLUCION PROBLEMA 2.26**DATOS**

- $C_{Mp} = 4096 (2^{12})$ palabras
- $C_{Mc} = 64 (2^6)$ palabras
- $K=16 (2^4)$ palabras/bloque
- Caché inicialmente vacía.
- Función de correspondencia *directa*

De los datos del enunciado se pueden calcular el número de bloques C de la M_c .

$$C = \frac{C_{Mc}}{K} = \frac{2^6}{2^4} = 2^2 = 4 \text{ bloques.}$$

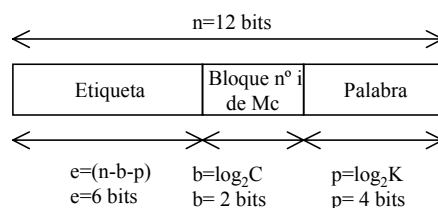
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^{12}}{2^4} = 2^8 = 256 \text{ bloques.}$$

Es decir, que los $\log_2 M = 8$ bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^{12} = 12$ bits

Puesto que la caché emplea *correspondencia directa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c .

Direcciones leídas	Nº de bloque (j) de la M_p	Nº de bloque (i) de la M_c	Fallo o acierto
000000 01 0000	1	1	Fallo se carga j=1 en i=1 que inicialmente esta vacío
000100 01 0100	17	1	Fallo se carga j=17 en i=1
000001 00 1000	4	0	Fallo se carga j=4 en i=0 que inicialmente esta vacío
000001 11 1000	7	3	Fallo se carga j=7 en i=3 que inicialmente esta vacío

Afirmación I: De acuerdo con la tabla construida anteriormente, la afirmación es **VERDADERA**.

Afirmación II: De acuerdo con la tabla construida anteriormente, el bloque i=1 de la M_c tiene almacenado el bloque j=17 de la M_p , por lo tanto su etiqueta es 000100. Luego la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 2.27**DATOS**

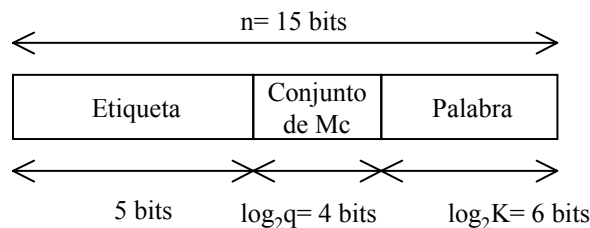
- M_p con 32k palabras de 16 bits/palabra
- M_c asociativa por conjuntos de 4k palabras
 - $r=4$ bloques/conjunto
 - $K=64$ palabras/bloque
- La M_c está inicialmente vacía y utiliza el algoritmo FIFO para el reemplazo de bloques.
- La CPU accede a 4352 palabras de las direcciones 0, 1, 2,...,4351, en orden.
- Después repite esta secuencia 9 veces.
- La M_c es 10 veces más veloz que la $M_p \rightarrow t_{ap}=10t_{ac}$

a) Como se trata de una caché asociativa por conjuntos el formato de dirección de la memoria principal se divide en tres campos: Etiqueta, Conjunto y Palabra.

Para direccionar las 32k palabras de la memoria principal hacen falta 15 bits ($32k = 2^5 \cdot 2^{10} = 2^{15}$). 15 bits es, por tanto, el ancho de una dirección de memoria principal.

La memoria caché tendrá $(4 \text{ bloques/conjunto}) \cdot (64 \text{ palabras/bloque}) = 256 (2^8)$ palabras/conjunto. La caché tendrá, por tanto, un total de $q=(4096 \text{ palabras}) / (256 \text{ palabras/conjunto}) = 16$ conjuntos.

Así, la caché tendrá 2^6 palabras/bloque y 2^4 conjuntos. Por tanto, los campos Conjunto y Palabra tendrán 4 y 6 bits, respectivamente. Esto hace un total de 10 bits, de los 15 de una dirección de memoria principal. Los 5 bits restantes constituyen el campo Etiqueta.



b) Si el computador no tiene memoria caché todas las palabras se han de leer de la memoria principal. El tiempo necesario para hacerlo será el tiempo necesario para leer una palabra multiplicado por el número total de palabras a las que se accede.

Como las 4352 palabras se leen un total de 10 veces, el número total de palabras que se leen es $4352 \cdot 10$. Si el tiempo necesario para acceder a una palabra de memoria principal es $10t_{ac}$ (frente a t_{ac} que sería el tiempo necesario para acceder a una palabra de caché), el tiempo total es

$$t_a = 4352 \cdot 10 \cdot 10t_{ac} = 435200t_{ac}.$$

c) Para resolver esta cuestión conviene distinguir entre la primera lectura del bucle y las nueve restantes:

- **Primera lectura:**

Corresponde a la primera lectura secuencial de las palabras 0 a 4351. Se considera que la caché está inicialmente vacía. Dado que el tamaño de la memoria caché es de 4096 palabras y el número de palabras a las que se accede es de 4352, es claro que la caché no puede almacenar simultáneamente todas estas palabras.

Como se ha visto en a), la caché está organizada en 16 conjuntos de 4 particiones por conjunto, lo que hace un total de 64 particiones. Cada una de estas particiones tiene 64 palabras, lo que hace el total de las 4096 palabras de la caché.

Las 4352 palabras *consecutivas* que hay que leer corresponden a $(4352 \text{ palabras}) / (64 \text{ palabras/partición}) = 68$ particiones, 4 más que las 64 de la caché.

Cada vez que se produzca un fallo se traerá un bloque completo desde la memoria principal, es decir, 64 palabras.

Se lee la palabra de dirección 0: Como la caché está vacía se producirá un fallo y se traerán las palabras 0 a 63 que, según el formato de dirección de la memoria principal corresponden al conjunto 0. Como las cuatro particiones de este conjunto se encuentran vacías se puede suponer que se almacenarán en la partición 0 del conjunto 0. La lectura de las palabras 1 a 63 no origina fallos, pues estas palabras ya se encuentran en la caché.

Se lee la palabra 64: Un nuevo fallo. Se traen las direcciones 64 a 127, que corresponden al conjunto 1. Se supone que se utiliza la partición 0 de este conjunto.

Así, los primeros 16 bloques de la memoria principal irán a la partición 0 de sus respectivos conjuntos. Los 16 bloques siguientes irán a la partición 1 de sus conjuntos. Se suceden las lecturas, hasta haber leído los primeros 64 (0 a 63) bloques de la caché, es decir, las primeras 4096 (0-4095) palabras: en este momento la caché se llena, pero quedan todavía 4 bloques por leer.

Conjunto i de M_c	Contenido particiones (nº bloque j de M_p)
0	64
	16
	32
	48
1	65
	17
	33
	49
2	66
	18
	34
	50
3	67
	19
	35
	51
4	68
	20
	36
	52
.	
.	
.	
15	15
	31
	47
	63

Tabla 1. Contenido de la memoria caché después del primer ciclo de lecturas

Conjunto i de M_c	Contenido particiones (nº bloque j de M_p)
0	48 64 16 32
1	49 65 17 33
2	50 66 18 34
3	51 67 19 35
4	4 20 36 52
.	.
.	.
.	.
15	15 31 47 63

Tabla 2. Contenido de la memoria caché después del segundo ciclo de lecturas

Dado que la política de reemplazamiento es FIFO, se reemplazarán las particiones de cada conjunto que lleven más tiempo en la caché. En este caso, la partición 0 de cada conjunto.

Se lee la palabra 4096: Produce un nuevo fallo. Corresponde nuevamente al conjunto 0, pero éste se encuentra completo. Hay que reemplazar la partición más antigua *de este conjunto*, que como se acaba de ver es la 0.

Las palabras 4160, 4224 y 4288 producirán tres fallos más, y corresponden a los conjuntos 1, 2 y 3, respectivamente. En cada uno de estos conjuntos se reemplaza la partición más antigua, que es la 0 en todos los casos.

De este modo termina la primera lectura de las direcciones de memoria. El contenido de la caché queda como se muestra en la Tabla 1.

• **Siguientes lecturas (2^a a 10^a) :**

Al realizar el segundo ciclo de lecturas a cada dirección le corresponde el mismo conjunto que en la primera lectura, según el mecanismo de la caché asociativa por conjuntos. El que una palabra produzca un fallo o no dependerá de si todavía se encuentra en la caché. En la Tabla 1 se observa que los conjuntos 4 a 15 no van a producir fallos, pues su contenido no ha variado. Pero sí va a haber problemas con los conjuntos 0 a 3:

La dirección 0, que correspondía al bloque 0, ya no está en la caché, pues fue sustituido por el bloque 64. La partición más antigua en esta ocasión es la 1, que contenía el bloque 16 de memoria principal. Luego habrá que escribir en la partición 1 del conjunto 0, por lo que el bloque 16 es reemplazado por el 0.

Cuando más adelante se lea el bloque 16 éste ya no estará (ha sido reemplazado por el 0) y habrá que hacer un reemplazamiento en la partición más antigua de este conjunto, que ahora es la 2: el bloque 32 es

reemplazado por el 16. Análogamente, el bloque 32 reemplazará al 48, el 48 al 64 y el 64 al 0. En la segunda lectura se producen 5 fallos en el conjunto 0.

Análogamente, en los conjuntos 1, 2 y 3 se producirán otros 5 fallos en cada una de ellos durante la segunda lectura. Al finalizar este segundo ciclo de lecturas el contenido de la caché quedará como se muestra en la Tabla 2.

La situación en el tercer ciclo es análoga a lo sucedido en el segundo: se producirán 4 fallos en cinco grupos, es decir, 20 fallos en total. Lo mismo sucede en los ciclos siguientes.

Cómputo del número total de fallos:

En el primer ciclo de lecturas se producía un fallo en cada uno de los 68 bloques a leer: 68 fallos en total. En los 9 ciclos siguientes, 20 fallos en cada uno de ellos. Luego

$$N_F = 68 + 9 \cdot 20 = 248 \text{ fallos}$$

D) En caso de fallo se especifica que las acciones que tienen lugar son :

1. Se mueve el bloque completo de M_p a M_c .
2. La CPU lee el dato desde la caché.

Es decir, que el tiempo de gestión de fallos t_{gf} es:

$$t_{gf} = K \cdot t_{ap} + t_{ac} = K \cdot 10 \cdot t_{ac} + t_{ac} = (10 \cdot K + 1)t_{ac}$$

Por otro lado el tiempo de gestión del acierto, es el tiempo de acceso a la memoria caché $t_{ga} = t_{ac}$

Luego el tiempo total que se emplea es:

$$t_a = N_A \cdot t_{ac} + N_F \cdot (10 \cdot K + 1)t_{ac}$$

Como $N_A = N_T - N_F$, entonces:

$$t_a = (N_T - N_F) \cdot t_{ac} + N_F \cdot (10 \cdot K + 1)t_{ac} = (N_T + N_F \cdot 10 \cdot K) \cdot t_{ac}$$

Sustituyendo valores se obtiene

$$t_a = (43520 + 248 \cdot 10 \cdot 64) \cdot t_{ac} = 202240 \cdot t_{ac} \text{ seg}$$

SOLUCION PROBLEMA 2.28**DATOS**

- $C_{Mp}=64k \text{ palabras} = 2^{16} \text{ palabras}$
- $C_{Mc}=1k \text{ palabras} = 2^{10} \text{ palabras}$
- Memoria caché con función de correspondencia directa
- $K=128 \text{ palabras/bloque}$
- $t_{ap}=10 \cdot t$
- $t_{ac}=1 \cdot t$

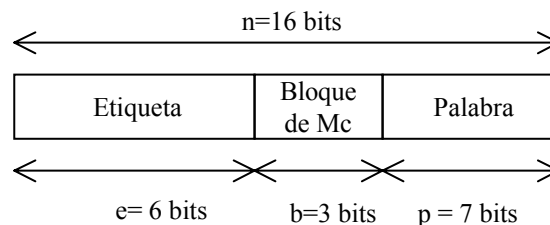
a) La anchura de una dirección de memoria es $n=\log_2 C_{Mp}=\log_2 2^{16}=16 \text{ bits}$.

Por otro lado la anchura del campo PALABRA que especifica la posición de una palabra en el interior de un bloque viene dada $p=\log_2 K=\log_2 128=7 \text{ bits}$.

La anchura del campo BLOQUE que especifica el i bloque de M_c donde se busca la palabra viene dado por: $b=\log_2 (C_{Mc}/K)=\log_2 (2^{10}/2^7)=\log_2 (2^3)=3 \text{ bits}$.

Por ultimo la anchura del campo ETIQUETA, es $e=n-(b+p)=16-(7+3)=6 \text{ bits}$.

Es decir, desde el punto de vista de la memoria caché, una dirección de memoria principal se considera compuesta de la forma indicada en la figura.



b) En este apartado se pide calcular el tiempo que tardaría en ejecutarse el programa si el computador no tuviera caché. Ya que no se indica el tiempo que tardan en ejecutarse las instrucciones, se ha de suponer que es despreciable frente al tiempo de acceso a memoria (la memoria es muy lenta comparada con la velocidad de la UCP). Por tanto, el tiempo de ejecución total será el tiempo de acceso a memoria, que será el número total de accesos a memoria multiplicado por el tiempo que lleva cada acceso. En este caso, el tiempo de acceso a tener en cuenta es el de la memoria principal ($10t$), ya que se supone que no hay caché.

Para saber cuántas veces se accede a la memoria principal basta con contar cuántas instrucciones hay en cada sección del programa, y tener en cuenta cuántas veces se ejecuta cada una de estas secciones, tal como se muestra en la Tabla 1.

Sección	Nº Instrucciones	Veces que se ejecuta	Total instrucciones
17-22	6	1	6
23-164	142	10	1420
165-239	75	20 x 10	15000
240-1200	961	10	9610
1201-1500	300	1	300
Nº Total de accesos a memoria principal			26336

Tabla 1. Cálculo del número total de accesos a memoria principal

El tiempo total de ejecución del programa es por tanto:

$$t_a = N_T \cdot t_{ap} = 26.336 \cdot 10t = \mathbf{263.360t}.$$

c) Dado que el computador utiliza una caché con correspondencia directa, a cada palabra de la memoria principal le corresponde un único bloque de la memoria caché, según el formato de los campos que se dedujo en el apartado a). En la Tabla 2 se muestra la correspondencia entre las distintas direcciones de la memoria principal y sus campos etiqueta, partición y palabra. Estos campos se obtienen haciendo grupos de 6, 3 y 7 bits, respectivamente, con la representación en binario puro de la dirección. La representación binaria se ha obtenido, a su vez, fácilmente a partir de la representación hexadecimal.

Dirección	Formato dirección memoria principal		
	Etiqueta (6 bits)	Bloque i de M_c (3 bits)	Palabra (7 bits)
17	000000	000	0010001
22	000000	000	0010110
23	000000	000	0010111
164	000000	001	0100100
165	000000	001	0100101
239	000000	001	1101111
240	000000	001	1110000
1200	000001	001	0110000
1201	000001	001	0110001
1500	000001	001	1011100

Tabla 2. Formato de las direcciones de memoria principal

A partir de la Tabla 2 es fácil calcular el número de fallos que se producen en el acceso a la memoria caché. Para ello, basta con ir ejecutando el programa, tal como se muestra en la figura del enunciado, “con lápiz y papel” y llevar en cuenta cómo se va ocupando la memoria caché.

1) Direcciones 17-22:

Estas direcciones corresponden, según la Tabla 2, a la partición 000 (etiqueta 000000). La caché inicialmente está vacía, por lo que se produce el primer fallo. Esto hace que se traiga la partición entera desde la memoria principal.

2) Primera ejecución del bucle exterior:

2.1) Direcciones 23-164:

La dirección 23 corresponde a la partición 000 (etiqueta 000000); la 164, a la partición 001 (etiqueta 000000). La partición 000 ya se encontraba en la caché. En la dirección 128 se produjo el segundo fallo, por lo que se trajo completa la partición 001 (etiqueta 000000) desde la memoria principal.

2.2) Direcciones 165-239:

Estas direcciones corresponden a la partición 001 (etiqueta 000000), que ya se encuentra en la caché. El bucle interior se ejecuta 20 veces sin que se produzca ningún fallo.

2.3) Direcciones 240-1200:

Estas direcciones abarcan desde la partición 001 (etiqueta 000000) hasta la partición 001 (etiqueta 000001). Se han producido 8 fallos, que corresponden desde la partición 010 (etiqueta 000000) hasta la partición 111. (etiqueta 000000), y las particiones 000 y 001 (etiqueta 000001). Obsérvese que estas dos últimas particiones han reemplazado a las particiones 000 y 001 (etiqueta 000000).

Con esto finaliza la primera ejecución del bucle exterior. Se ha producido un total de 10 fallos (1 + 1 + 8). El estado de la caché ha quedado como se muestra en la Tabla 3.

Bloque i de M_c	Etiqueta
0	000001
1	000001
2	000000
3	000000
4	000000
5	000000
6	000000
7	000000

Tabla 3. Estado de la memoria caché al finalizar la primera ejecución del bucle exterior

3) Segunda a décima ejecución del bucle exterior:

Como se ha visto anteriormente, el bucle exterior comienza referenciando las particiones 000 y 001 correspondientes a la etiqueta 000000. Estas particiones han sido reemplazadas por las correspondientes a la etiqueta 000001. Se producen por tanto dos fallos y se reemplazan las particiones.

El bucle sigue referenciando las particiones 010 a 111 (etiqueta 000000), que como se comprueba en la Tabla 3 se encuentran en la caché. Estas particiones no producen fallos.

El bucle termina referenciando las particiones 000 y 001 con etiqueta 000001, pero éstas han sido sustituidas al comienzo del bucle por las correspondientes a la etiqueta 000000. Se producen, por tanto, otros dos fallos, y se reemplazan las particiones.

La segunda ejecución del bucle ha terminado con cuatro fallos y el contenido de la caché es el mismo que había al comenzar, es decir, el mostrado en la Tabla 3.

Cada subsiguiente ejecución del bloque exterior producirá otros cuatro fallos y dejará la caché con el mismo contenido que había al comenzar. En total, en las nueve ejecuciones se producirán $9 \cdot 4 = 36$ fallos, y el contenido de la caché al finalizar será el mostrado en la Tabla 3.

4) Direcciones 1201-1500:

La salida del bucle exterior hace que se referencien las particiones 1, 2 y 3 correspondientes a la etiqueta 000001. La 1 ya se encuentra cargada en la caché, pero la 2 y la 3 no; por tanto se producen otros 2 fallos.

Cómputo del número total de fallos:

El número total de fallos es $N_F = 10 + 36 + 2 = 48$ fallos.

d) Para calcular el tiempo total que se tarda en ejecutar el programa con la memoria caché hay que contabilizar el número total de accesos N_T realizan a la memoria caché, que es igual al número total de instrucciones que se ejecutan en el programa: 26.336. Es decir $N_T = 26336$

En caso de fallo se especifica que las acciones que tienen lugar son :

1. Se mueve el bloque completo de M_p a M_c .
2. La CPU lee el dato desde la caché.

Es decir, que el tiempo de gestión de fallos t_{gf} es:

$$t_{gf} = K \cdot t_{ap} + t_{ac} = K \cdot 10 \cdot \tau + \tau = (10 \cdot K + 1)\tau$$

Por otro lado el tiempo de gestión del acierto, es el tiempo de acceso a la memoria caché $t_{ga} = t_{ac} = \tau$

Luego el tiempo total que se emplea es:

$$t_a = N_A \cdot \tau + N_F \cdot (10 \cdot K + 1)\tau$$

Como $N_A = N_T - N_F$, entonces:

$$t_a = (N_T - N_F) \cdot \tau + N_F \cdot (10 \cdot K + 1)\tau = (N_T + N_F \cdot 10 \cdot K) \cdot \tau$$

Sustituyendo valores se obtiene

$$t_a = (26336 + 48 \cdot 10 \cdot 128) \cdot \tau = 87776 \cdot \tau \text{ seg}$$

Aunque no lo solicita el enunciado del problema, a modo de curiosidad se puede calcular la tasa de acierto y la razón entre los tiempos de acceso t_a con caché y sin caché (t_a apartado (b)):

$$h = \frac{N_A}{N_T} = \frac{N_T - N_F}{N_T} = \frac{26336 - 48}{26336} = 99.8\%$$

$$\frac{t_a(M_c + M_p)}{t_a(M_p)} = \frac{87776 \cdot \tau}{263360 \cdot \tau} = 33.3\%$$

Obsérvese que a pesar de la alta tasa de acierto y de que la caché es 10 veces más rápida que la memoria principal la velocidad sólo mejora en un factor 3, debido a la carga que supone mover bloques completos desde la memoria principal a la memoria caché.

SOLUCION PROBLEMA 2.29

DATOS

- Número total de accesos $N_T = 2^{20}$
- Número de fallos $N_F = 2^{17}$
- Tasa de fallos?

La definición de tasa de fallos es:

$$\text{Tasa de fallos} = \frac{N_F}{N_T} = \frac{2^{17}}{2^{20}} = \frac{1}{2^3} = \frac{1}{8} = 0.125$$

Luego la Tasa de fallos es **12.5 %**.

SOLUCION PROBLEMA 2.30**DATOS**

- Procesador con 12 líneas de dirección $A_{11}A_{10}...A_0$
- Para construir su unidad de memoria se dispone de módulos de $2K = (2^{11})$ palabras
- La línea A_1 se utilizan para la selección de cada módulo

De los datos se deduce que las direcciones para acceder a posiciones de memoria tienen 12 bits, uno por cada línea. Por otro lado puesto que se tienen una única línea A_1 para seleccionar módulos de memoria de $2K$, el número total de módulos para implementar el mapa de memoria es 2 módulos. En la siguiente tabla se muestra el módulo de memoria al que se hace referencia en función del valor de A_1 .

A_1	Módulo de memoria
0	Módulo 0
1	Módulo 1

Módulo 0

Así las direcciones que hacen referencia al primer módulo de memoria (módulo nº 0) son:

XXXX XXXX XX0X

donde X puede valer 0 o 1.

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 0	Y Y 8
Y Y 1	Y Y 9
Y Y 4	Y Y C
Y Y 5	Y Y D

Módulo 1

Las direcciones que hacen referencia al segundo módulo de memoria (módulo nº 1) son:

XXXX XXXX XX1X

Lo que en hexadecimal equivale a las siguientes direcciones, supuesto que Y puede tomar cualquier valor hexadecimal (0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F):

Y Y 2	Y Y A
Y Y 3	Y Y B
Y Y 6	Y Y E
Y Y 7	Y Y F

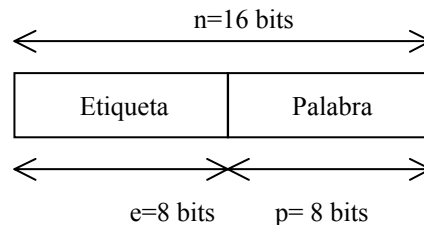
De acuerdo con el análisis realizado las afirmaciones son:

- A) **FALSA** Ya que C89 se encuentra en el módulo 0 y C8A en el módulo 1.
 B) **VERDADERA**. Ya que C89 se encuentra en el módulo 0 y C8A en el módulo 1.
 C) **FALSA**. La línea de selección es A_1 .
 D) **FALSA**. La afirmación B es verdadera.

SOLUCION PROBLEMA 2.31**DATOS**

- Memoria principal de $C_{Mp} = 64k$ palabras $= 2^{16}$
- Memoria caché de $C_{Mc} = 2k$ palabras $= 2^{11}$
- Utiliza correspondencia totalmente asociativa
- $K = 256$ palabras/bloque $= 2^8$
- La memoria caché está inicialmente vacía.
- Cuando la memoria caché está llena, se reemplaza el bloque que se ha utilizado más recientemente.
- La CPU ejecuta un bucle 5 veces, en el que accede a 2301 palabras, desde la dirección 0 a la 2300.
- $t_{ap} = 7 \cdot t_{ac}$

- A) Cuando se utiliza correspondencia totalmente asociativa la dirección de memoria principal se descompone en dos campos: etiqueta y palabra. La anchura de una dirección de memoria principal es $n = \log_2 C_{Mp} = \log_2 2^{16} = 16$ bits. Por otro lado los bits necesarios para codificar la posición de una palabra en un bloque es $p = \log_2 K = \log_2 2^8 = 8$ bits. Por lo tanto los bits del campo etiqueta son $e = n - p = 16 - 8 = 8$ bits. Es decir el formato de una dirección queda de la siguiente forma:



- B) Puesto que se supone que no hay memoria caché, el número total de accesos que realiza la CPU a la memoria principal en la ejecución 5 veces de ese bucle de direcciones es $N_T = 5 \cdot 2301 = 11505$ accesos. El tiempo de acceso medio a la memoria principal es $t_{ap} = 7 \cdot t_{ac}$. Luego el tiempo que se tardaría en realizar estos N_T accesos es:

$$tiempo = N_T \cdot 7 \cdot t_{ac} = 11505 \cdot 7 \cdot t_{ac} = 80535 t_{ac} \text{ seg}$$

- C) El número de bloques de la memoria caché es:

$$C = \frac{C_{Mc}}{K} = \frac{2^{11}}{2^8} = 2^3 = 8 \text{ bloques}$$

Por lo tanto la tabla que nos piden tendrá 8 filas y 5 columnas una por cada vez que se ejecuta el bucle.

Por otra parte el número de bloques de la memoria principal es:

$$M = \frac{C_{Mp}}{K} = \frac{2^{16}}{2^8} = 2^8 = 256 \text{ bloques}$$

En cada bloque hay $K = 256$ palabras, por lo tanto las 2301 palabras a las que hace referencia la CPU, caben en $2301/256 \approx 9$ bloques de memoria principal. Puesto que la M_c tiene sólo ocho en la ejecución de cada ciclo habrá que realizar un único reemplazamiento.

En la siguiente Tabla se muestra los rangos de direcciones de M_p correspondientes a los 9 primeros bloques, los que intervienen en el problema, con las etiquetas que tendrían al ser cargados en memoria caché. En el apartado A) se vio que los 8 bits más significativos de la dirección se corresponden con el

campo etiqueta, esto equivale a decir que los 2 dígitos más significativos en hexadecimal son la etiqueta.

Rango de direcciones (Decimal)	Rango de direcciones (Hexadecimal)	Etiqueta en la M _c
0 - 255	0000 - 00FF	00
256 - 511	0100 - 01FF	01
512 - 767	0200 - 02FF	02
768 - 1023	0300 - 03FF	03
1024 - 1279	0400 - 04FF	04
1280 - 1535	0500 - 05FF	05
1536 - 1791	0600 - 06FF	06
1792 - 2027	0700 - 07FF	07
2048 - 2303	0800 - 08FF	08

A la hora de explicar que ocurre cuando se acceden a las direcciones del enunciado del problema, se va a distinguir el primer ciclo del resto de ciclos.

- **Primera ejecución del ciclo**

La caché se va llenando con bloques de memoria principal que son copiados cuando se produce la primera referencia a una dirección del bloque. Por ejemplo, cuando se referencia la dirección 1536, se mueve de memoria principal a memoria caché el bloque completo que contiene las direcciones de la 1536 a la 1791. Al utilizar la técnica de correspondencia totalmente asociativa, el bloque se mueve a cualquiera de las particiones vacías de la caché, en el caso del ejemplo anterior a la partición 6.

El problema se plantea cuando la memoria ya está llena y se referencia la dirección 2048, perteneciente al bloque con etiqueta 08. La técnica de reemplazamiento propuesta en el problema es: “reemplazar el bloque que se haya utilizado más recientemente”. En este caso, es el bloque almacenado en la partición 7 y con etiqueta 07.

Por todo lo visto, al finalizar el primer ciclo, en la caché se encuentran almacenadas todas las direcciones propuestas excepto las pertenecientes al bloque con etiqueta 07.

- **Segunda ejecución del ciclo**

En esta segunda ejecución del ciclo se producen aciertos hasta referenciar la primera dirección del bloque 07, sustituido en el ciclo anterior. Utilizando la técnica de reemplazamiento propuesta, el último bloque referenciado es el 06, que es sustituido por el 07. Posteriormente las direcciones del bloque 08 dan aciertos.

- **Resto de ciclos**

Se producen situaciones similares a la del segundo ciclo, con una sustitución en cada uno de ellos.

Luego la tabla que nos piden es la siguiente:

Bloque de M _c	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez
i=0	00	00	00	00	00
i=1	01	01	01	01	01
i=2	02	02	02	02	02
i=3	03	03	03	03	04
i=4	04	04	04	05	05
i=5	05	05	06	06	06
i=6	06	07	07	07	07
i=7	08	08	08	08	08

C) Para contestar este apartado se va a utilizar la Tabla creada en el apartado anterior. Allí se ha visto que la ejecución del ciclo la primera vez sigue un patrón distinto a la ejecución del ciclo las 4 veces siguientes. Estudiando por separado ambos casos:

- **Primera ejecución del ciclo.** Se producen un fallo en cada uno de los nueve bloques de los consta el programa, produciéndose una sustitución del bloque 07 cuando se accede al bloque 08, tal y como se ha mencionado en el apartado anterior.
- **Segunda ejecución del ciclo y sucesivas.** Se produce un único fallo en cada ciclo.

El número total de fallos será: $N_F = 9 + 4 = 13$ fallos.

D) En caso de fallo se especifica que las acciones que tienen lugar son :

1. Se mueve el bloque completo de M_p a M_c .
2. La CPU lee el dato desde la caché.

Es decir, que el tiempo de gestión de fallos t_{gf} es:

$$t_{gf} = K \cdot t_{ap} + t_{ac} = K \cdot 7 \cdot t_{ac} + t_{ac} = (7 \cdot K + 1)t_{ac}$$

Por otro lado el tiempo de gestión del acierto, es el tiempo de acceso a la memoria caché $t_{ga} = t_{ac}$

Luego el tiempo total que se emplea es:

$$t_a = N_A \cdot t_{ac} + N_F \cdot (7 \cdot K + 1)t_{ac}$$

Como $N_A = N_T - N_F$, entonces:

$$t_a = (N_T - N_F) \cdot t_{ac} + N_F \cdot (7 \cdot K + 1)t_{ac} = (N_T + N_F \cdot 7 \cdot K) \cdot t_{ac}$$

Sustituyendo valores se obtiene

$$t_a = (11505 + 13 \cdot 7 \cdot 256) \cdot t_{ac} = 34801 \cdot t_{ac} \text{ seg}$$

SOLUCION PROBLEMA 2.32

DATOS

- Número total de accesos $N_T = 2^{20}$
- Número de fallos $N_F = 2^{17}$

La definición de tasa de fallos es:

$$\text{Tasa de fallos} = (1 - h) = \frac{N_F}{N_T}$$

Despejando la tasa de aciertos h:

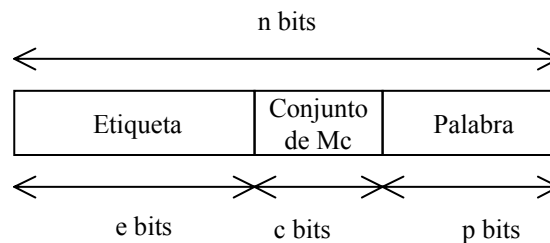
$$h = 1 - \frac{N_F}{N_T} = 1 - \frac{2^{17}}{2^{20}} = 1 - 0.125 = 0.875$$

Luego la tasa de aciertos $h = 87.5 \%$.

SOLUCION PROBLEMA 2.33**DATOS**

- Memoria principal de $C_{Mp} = 8192$ Kpalabras.
- Memoria caché de $C_{Mc} = 2$ Kpalabras = 2^{11} palabras
- La memoria caché utiliza correspondencia asociativa por conjuntos
 - $K = 64$ palabras/ bloque (2^6)
 - $r = 4$ bloques/conjunto
- Inicialmente la memoria caché está **llena** con la ejecución de un programa **Prog1**
- Se carga en memoria principal un nuevo programa **Prog2**
- **Prog2** efectúa la siguiente secuencia de referencias en la ejecución de su código :
 - Lee secuencialmente las direcciones 128,129,130,....., 143
 - Repite 20 veces un bucle de lectura de las direcciones 131, 132, 133
 - Finalmente lee secuencialmente las direcciones 134,135,.....,168

El formato de dirección para una memoria caché con correspondencia asociativa por conjuntos es, el siguiente:



Del dato de la capacidad de la memoria principal obtenemos el número de bits n que tienen las direcciones. Puesto que $8192 \text{ Kpalabras} = 8 \cdot 1024 \cdot \text{Kpalabras} = 2^3 \cdot 2^{10} \cdot 2^{10} \text{ palabras} = 2^{23} \text{ palabras} = 2^n$. Con lo que $n=23$ bits.

El número de bits que codifican a las palabras que tiene un bloque es $p = \log_2 K = \log_2 64 = 6$ bits

Por otro lado, el número de bloques C en M_c , es:

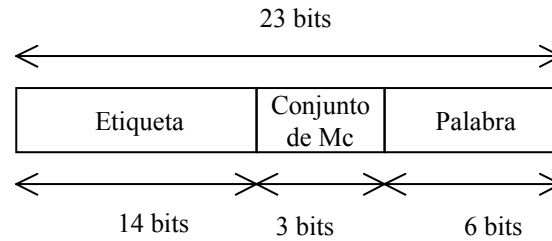
$$C = \frac{C_{Mc}}{K} = \frac{2^{11}}{2^6} = 2^5 \text{ bloques}$$

El número total de conjuntos q viene dado por:

$$q = \frac{C}{r} = \frac{2^5}{4} = \frac{2^5}{2^2} = 2^3 \text{ conjuntos}$$

Con lo que el número de bits que codifican a los conjuntos de la memoria caché es $c = \log_2 q = \log_2 2^3 = 3$ bits.

Por lo tanto el número de bits que definen la etiqueta $e = n - (c + p) = 23 - (3 + 6) = 14$ bits y el formato de dirección es el siguiente:



La tasa de acierto h , se define como:

$$h = \frac{\text{Numero de aciertos}}{\text{Numero total de accesos}} = \frac{N_A}{N_T}$$

Del enunciado se puede calcular el número total de accesos N_T .

- Lee secuencialmente las direcciones 128,129,130,..., 143 $\rightarrow (143-128)+1= 16$ accesos
- Repite 20 veces un bucle de lectura de las direcciones 131, 132, 133 $\rightarrow 20 \cdot 3= 60$ accesos
- Finalmente lee secuencialmente las direcciones 134,135,...,168 $\rightarrow (168-134)+1=35$ accesos

Por lo tanto $N_T= 16+60+35= 111$ accesos

Queda por calcular el número de aciertos.

Para su cálculo hay que tener en cuenta, que cuando se carga en memoria principal **Prog2**, en la memoria caché no hay ninguna copia del código de este nuevo programa por lo que primera referencia a su código va a producir necesariamente un fallo en la memoria caché.

En la función de correspondencia asociativa por conjuntos, cada bloque j de palabras de memoria principal tiene asignado un conjunto i de memoria caché mediante la relación:

$$i = j \text{ modulo } q$$

De acuerdo a esta relación y sabido que el número de palabras que tiene un bloque de memoria principal es $K= 64$. Es posible conocer el conjunto de M_c asociado a cada bloque de la M_p .

- El bloque $j=0$ de M_p (direcciones 0 a 63) \rightarrow Conjunto $i=0$ de M_c
- El bloque $j=1$ de M_p (direcciones 64 a 127) \rightarrow Conjunto $i=1$ de M_c
- El bloque $j=2$ de M_p (direcciones 128 a 191) \rightarrow Conjunto $i=2$ de M_c

El programa **Prog2** utiliza las direcciones (128 a 168) que se encuentran exclusivamente dentro del bloque $j=2$ de M_p , por lo tanto sólo se producirá **1 Fallo**, cuando se lee la primera dirección del mismo, la 128, cargándose entonces el bloque $j=2$ de M_p en el conjunto $i=2$ de M_c . Las restantes accesos a M_c producirán aciertos.

Es decir, el número de aciertos es $N_A=N_T-N_F$, siendo N_F el número de fallos $N_A=111-1=110$ aciertos.

Luego la tasa de acierto es:

$$h = \frac{110}{111} = 0.99$$

Es decir **99%**

SOLUCION PROBLEMA 2.34

Una posible forma de resolver el problema consiste en ir probando uno a uno con los registros de máscara dados para verificar si producen el registro de Marca especificado.

1) Máscara 1

		<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>								0	1	0	1	1	1	0	0	0	1	0	0	1	0	0	1	ARGUMENTO																																																																																																																								
0	1	0	1	1	1	0	0																																																																																																																																											
0	1	0	0	1	0	0	1																																																																																																																																											
		<table><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>								0	1	0	0	1	0	0	1	MASCARA																																																																																																																																
0	1	0	0	1	0	0	1																																																																																																																																											
		<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																																																																																
ETIQUETA	<table><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr></table>	0	1	1	1	1	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																														<table><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	1	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	0	0	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																														<table><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr></table>	0	1	0	0	1	MARCA
	0																																																																																																																																																	
	1																																																																																																																																																	
	1																																																																																																																																																	
	1																																																																																																																																																	
1																																																																																																																																																		
1	1	0	1	1	1	1	0																																																																																																																																											
1	1	0	1	1	0	1	0																																																																																																																																											
1	0	0	1	0	0	0	0																																																																																																																																											
1	0	1	0	0	1	1	0																																																																																																																																											
1	1	0	1	1	0	0	0																																																																																																																																											
0																																																																																																																																																		
1																																																																																																																																																		
0																																																																																																																																																		
0																																																																																																																																																		
1																																																																																																																																																		

La máscara 1 **si** genera este registro de marca. Hay que recordar que si el bit de etiqueta está a 0, el bit correspondiente del registro de marca se pone a 0, por eso se ha tachado la primera fila de la matriz de celdas asociativa.

2) Máscara 2

		<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>								0	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	ARGUMENTO
0	1	0	1	1	1	0	0																			
0	1	0	0	0	0	0	0																			
		<table><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>								0	1	0	0	0	0	0	0	MASCARA								
0	1	0	0	0	0	0	0																			
ETIQUETA	0									0																
	1									1																
	1									0																
	1									0																
	1									1																

La máscara 2 **si** genera este registro de marca.

3) Máscara 3

		<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>								0	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	ARGUMENTO																											
0	1	0	1	1	1	0	0																																														
0	0	0	1	1	0	0	0																																														
										MASCARA																																											
ETIQUETA	0		<table><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>								1	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	0	0		0	MARCA
	1	1	0	1	1	1	1	0																																													
	1	1	0	1	1	0	1	0																																													
	1	0	0	1	0	0	0	0																																													
	1	0	1	0	0	1	1	0																																													
1	1	0	1	1	0	0	0																																														
1										1																																											
1										0																																											
1										0																																											
1										1																																											

La mascara 3 **si** genera este registro de marca.

4) Máscara 4

		<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>								0	1	0	1	1	1	0	0	0	0	0	1	1	0	0	1	ARGUMENTO	
0	1	0	1	1	1	0	0																				
0	0	0	1	1	0	0	1																				
										MASCARA																	
ETIQUETA	0		1	1	0	1	1	1	1	0		0															
	1		1	1	0	1	1	0	1	0		1															
	1		1	0	0	1	0	0	0	0		0															
	1		1	0	1	0	0	1	1	0		0															
	1		1	1	0	1	1	0	0	0		1															
										MARCA																	

La mascara 4 **si** genera este registro de marca.

5) Máscara 5

		<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>								0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	1	ARGUMENTO	
0	1	0	1	1	1	0	0																				
0	0	0	0	1	0	0	1																				
										MASCARA																	
ETIQUETA	0		1	1	0	1	1	1	1	0		0															
	1		1	1	0	1	1	0	1	0		1															
	1		1	0	0	1	0	0	0	0		0															
	1		1	0	1	0	0	1	1	0		0															
	1		1	1	0	1	1	0	0	0		1															
										MARCA																	

La mascara 5 **si** genera este registro de marca.

SOLUCION : Los **5 registros de mascara** propuesto generan el mismo registro de marca.

SOLUCION PROBLEMA 2.35**DATOS**

- $C_{Mp} = 2^{20}$ palabras
- $C_{Mc} = 1024 (2^{10})$ palabras
- $K=64 (2^6)$ palabras/bloque
- Función de correspondencia directa

De los datos del enunciado se pueden calcular el número de bloques C de la M_C .

$$C = \frac{C_{Mc}}{K} = \frac{2^{10}}{2^6} = 2^4 = 16 \text{ bloques.}$$

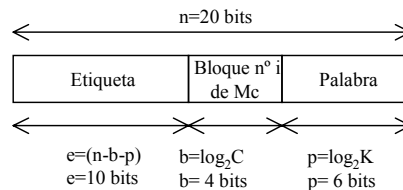
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^{20}}{2^6} = 2^{14} \text{ bloques}$$

Es decir, que los $\log_2 M = 14$ bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^{20} = 20$ bits

Puesto que la caché emplea *correspondencia directa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:

**Afirmación I:**

La dirección **1D45F** se expresa en binario de la siguiente forma: 0001 1101 0100 0101 1111

Desde el punto de vista de la M_c se descompone en los siguientes campos:

Etiqueta	Bloque n° i de M_C	Palabra
0001110101	0001	011111

Luego esta dirección hace referencia al bloque $i=0001_2=1_{10}$ de M_C . Luego la afirmación es **FALSA**. No hace falta analizar la dirección **02075**.

Afirmación II:

La dirección **2A23B** se expresa en binario de la siguiente forma: 0010 1010 0010 0011 1011

Desde el punto de vista de la M_c se descompone en los siguientes campos:

Etiqueta	Bloque n° i de M_C	Palabra
0010101000	1000	111011

Luego esta dirección hace referencia al bloque $i=1000_2=8_{10}$ de M_C . Luego la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 2.36**DATOS**

- Memoria RAM de capacidad total $C_T =$ de 1024 palabras x 16 bits/palabra.
- Construida con módulos de memoria RAM de capacidad $C_0 =$ 128 palabras x 8 bits/palabra

El número de líneas del bus de direcciones n se calcula a partir del número de palabras que posee la memoria RAM construida. Es decir:

$$n = \log_2 1024 = \log_2 2^{10} = 10 \text{ líneas}$$

Puesto que el número de palabras que posee el modulo RAM unidad es de 128, para conseguir 1024 palabras de capacidad total se necesitarán:

$$\frac{1024}{128} = \frac{2^{10}}{2^7} = 2^3 = 8 \text{ módulos}$$

Por otra parte, el número de líneas del bus de direcciones que se utilizarán para seleccionar entre estos ocho módulos n_s es:

$$n_s = \log_2 2^3 = 3 \text{ líneas}$$

Luego, el número de líneas del bus de direcciones comunes n_c a todos los módulos serán:

$$n_c = n - n_s = 10 - 3 = 7 \text{ líneas}$$

Hasta aquí se han indicado todos los cálculos necesarios para resolver el problema. De forma adicional, vamos a calcular el número de módulos de memoria RAM de capacidad C_0 necesarios para construir una memoria RAM de capacidad C_T . Pues bien, obsérvese que C_T se descompone de la siguiente forma:

$$C_T = 2^{10} \text{ pal} \times 2^4 \text{ bits/pal} = 2^3 \cdot 2^7 \text{ pal} \times 2 \cdot 2^3 \text{ bits/pal} = (2^3 \cdot 2) \cdot 2^7 \text{ pal} \times 2^3 \text{ bits/pal}$$

$$C_T = 16 \cdot (2^7 \text{ pal} \times 2^3 \text{ bits/pal}) = 16 \cdot C_0$$

Es decir se necesita un total de **16 módulos**.

SOLUCION PROBLEMA 2.37**DATOS**

- Memoria de acceso aleatorio.
- $t_a = 100 \text{ nseg.}$
- $t_c = 200 \text{ nseg.}$

En una memoria de acceso aleatorio la frecuencia de acceso f_a se calcula de la siguiente forma:

$$f_a = \frac{1}{t_c} = \frac{1}{200 \text{ nseg}} = \frac{1}{200 \cdot 10^{-9} \text{ seg}} = \frac{10^9}{200 \cdot \text{seg}} = \frac{10^7}{2} \cdot \text{seg}^{-1} = 5 \cdot 10^6 \cdot \text{seg}^{-1}$$

Luego la frecuencia de acceso es $f_a = 5 \cdot 10^6 \text{ seg}^{-1}$.

SOLUCION PROBLEMA 2.38**DATOS**

- Sistema jerárquico de memoria.
- $C_{Mc}=128$ palabras, $K=16$ pal/ bloque, $t_{ca}=10 \mu\text{seg}$
- $C_{Mp}=1024$ Kpalabras, $t_p=200 \mu\text{seg}$
- Tasa de acierto $h=0.99$
- Cuando se produce un fallo en M_c :
 - Se mueve el bloque completo a la M_c .
 - Se lee el dato desde M_c .
 - $t_{extra}=20 \mu\text{seg}$

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo empleado en gestionar un acierto. Normalmente se cumple que :

$$t_{acierto} = t_{ca} = 10 \mu\text{seg}$$

De acuerdo con el enunciado el tiempo de gestión de un fallo será la suma de tres componentes, ya que cuando se produce un fallo:

- 1) El sistema mueve el bloque completo a la caché, es decir hay que realizar K accesos a la memoria principal, lo que supone un tiempo de $K \cdot t_p$.
- 2) Se lee el dato desde la caché, lo que supone un tiempo de t_{ca} .
- 3) Además se emplea un tiempo extra $t_{extra} = 20 \mu\text{seg}$ debido a la política de ubicación y reemplazamiento en la caché.

Luego:

$$t_{gf} = K \cdot t_p + t_{ca} + t_{extra} = 16 \cdot 200 + 10 + 20 = 3230 \mu\text{seg}$$

Sustituyendo valores en la formula (1) se obtiene que:

$$t_{am} = 0.99 \cdot 10 + 0.01 \cdot 3230 = 9.9 + 32.3 = 42.2 \mu\text{seg}$$

Por otra si sólo se utilizara la memoria principal el tiempo de acceso medio sería:

$$t_{am}' = t_p = 200 \mu\text{seg}$$

Se observa que:

$$\frac{t_{am}'}{t_{am}} = 4.7$$

Luego las afirmaciones son:

Afirmación I: **VERDADERA**.

Afirmación II: **VERDADERA**

SOLUCION PROBLEMA 2.39**DATOS**

- $C_{Mp} = 64 \text{ Kpalabras} = (2^{16}) \text{ palabras}$
- $C_{Mc} = 1 \text{ K palabras} = (2^{10}) \text{ palabras}$
- $K=8 \text{ palabras/bloque} = (2^3) \text{ palabras/bloque}$
- Función de correspondencia directa.
- En un determinado instante la dirección 6B59, de la memoria principal está en la memoria caché

De los datos del enunciado se pueden calcular el número de bloques C de la M_C .

$$C = \frac{C_{Mc}}{K} = \frac{2^{10}}{2^3} = 2^7 = 128 \text{ bloques.}$$

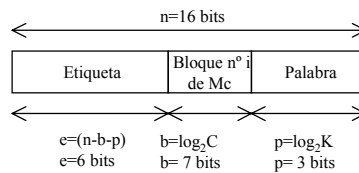
Por otra parte, el número de bloques M de la M_p .

$$M = \frac{C_{Mp}}{K} = \frac{2^{16}}{2^3} = 2^{13} = 8 \text{ Kbloques.}$$

Es decir, que los $\log_2 M = 13$ bits más significativos de una dirección de memoria hacen referencia al bloque j de M_p

Además el tamaño de una dirección viene dado por $n = \log_2 C_{Mp} = \log_2 2^{16} = 16$ bits

Puesto que la caché emplea *correspondencia directa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:

**AFIRMACION I:**

Para comprobar la validez de esta afirmación hay que pasar la dirección 6B59 a código binario.

$$6B59_{16} = 0110\ 1011\ 0101\ 1001$$

La dirección desde el punto de vista de la M_c tendría los siguientes campos:

011010	1101011	001
--------	---------	-----

Luego el bloque i de M_C al que hace referencia es: $i = 1101011_2 = 107_{10}$. Luego la afirmación es **VERDADERA**.

AFIRMACION II:

Para comprobar la validez de esta afirmación hay que pasar la dirección 675E a código binario.

$$675E_{16} = 0110\ 0111\ 0101\ 1111$$

La dirección desde el punto de vista de la M_c tendría los siguientes campos:

011001	1101011	111
--------	---------	-----

La etiqueta de esta dirección es 011001, distinta de la etiqueta 011010 de la dirección 6B59 que sí que estaba en la M_c . Luego 675E no está en la M_C . La afirmación es **VERDADERA**.

SOLUCION PROBLEMA 2.40

DATOS

- Sistema jerárquico de memoria con memoria caché de dos niveles.
- La caché de nivel 1, es la más cercana a la CPU
- La caché de nivel 2, es la más cercana a la memoria principal.

AFIRMACION I:

En un sistema jerárquico de memoria con dos niveles de memoria caché, la caché 1 (M_{C1}), más cercana a la CPU, posee una capacidad inferior a la caché de nivel 2 (M_{C2}), más cercana a la memoria principal. En consecuencia nunca la M_{C1} puede contener una copia de todos los bloques de la M_{C2} , sólo de algunos de ellos. Luego, la afirmación es **FALSA**.

AFIRMACION II:

Con la política de post-escritura, las modificaciones que se realicen en la caché 1 no se reflejarán inmediatamente en la caché 2. Por tanto, el contenido de un bloque de caché de nivel 1 no siempre será igual al contenido de un bloque correspondiente en la caché de nivel 2. Luego, la afirmación es **FALSA**.

SOLUCION PROBLEMA 2.41

DATOS

- 16 líneas de dirección $A_{15}-A_0$.
- Memoria compuesta de módulos ROM y módulos RAM.
- La ROM consta de un único módulo y ocupa las direcciones más bajas de memoria, comenzando por la dirección 0.
- La RAM con una capacidad de $C_{RAM}=48Kbytes$ ocupa las direcciones restantes.

Puesto que el sistema consta de $n=16$ líneas de dirección, podrá direccionar una capacidad de memoria de

$$C_T = 2^{16} \text{ bytes} = 64 \text{ Kbytes}$$

Dicha capacidad es la suma de la capacidad de la memoria ROM y de la memoria RAM.

$$C_T = C_{ROM} + C_{RAM}$$

Puesto que se conocen C_T y C_{RAM} es posible calcular C_{ROM}

$$C_{ROM} = C_T - C_{RAM} = 64Kbytes - 48Kbytes = 16Kbytes$$

Luego se tiene un sistema de $C_T=64$ Kbytes que se distribuye de la siguiente forma:

- Un único módulo de ROM de capacidad $C_{ROM}=16$ Kbytes, que contiene las direcciones bajas de memoria, comenzando desde la 0.
- Varios módulos de memoria RAM (en el enunciado no concreta el número, suponemos como hipótesis de trabajo que hay tres) que suman entre todos ellos una capacidad $C_{RAM}=48$ Kbytes.

Si se supone que el sistema posee al menos tres módulos de memoria RAM y uno de ROM, se necesitarán dos bits para seleccionar uno de los cuatro módulos de memoria. Esos bits de selección deben ser $A_{15}A_{14}$ ya que la ROM ocupa las direcciones más bajas de memoria, comenzando por la dirección 0. Es decir, las direcciones que acceden a la ROM son de la forma:

00XX XXXX XXXX XXXX

Se tiene por tanto la siguiente tabla de selección de módulos:

$A_{15}A_{14}$	Función lógica de selección	Modulo seleccionado
00	$CS_{ROM} = \overline{A_{15}}\overline{A_{14}}$	ROM
01	$CS_{RAM0} = \overline{A_{15}}A_{14}$	RAM nº 0
10	$CS_{RAM1} = A_{15}\overline{A_{14}}$	RAM nº 1
11	$CS_{RAM2} = A_{15}A_{14}$	RAM nº 2

De acuerdo con el razonamiento realizado, se tiene:

Afirmación I: **VERDADERA.**

Afirmación II: **FALSA.**

Nota: El enunciado de este problema es bastante puñetero ya que el alumno debe de suponer dos hipótesis, nada evidentes, para poder resolverlo: 1) Cada palabra de memoria ocupa un byte. 2) Hay dos o tres módulos de memoria RAM.

SOLUCION PROBLEMA 2.42

DATOS

- Memoria de acceso NO aleatorio.
- $f_a = 2 \cdot 10^6$ bits/seg..
- $t_a = 2$ mseg
- $N = 10^3$ bytes

En una memoria de acceso NO aleatorio la frecuencia de acceso f_a o velocidad de transferencia v_T se calcula de la siguiente forma:

$$v_T = f_a = \frac{N}{t_n - t_a}$$

Despejando t_n :

$$t_n = \frac{N}{v_T} + t_a$$

Sustituyendo valores:

$$t_n = \frac{10^3(\text{bytes}) \cdot 2^3(\text{bits/byte})}{2 \cdot 10^6(\text{bits/seg})} + 2 \cdot 10^{-3}(\text{seg}) = 4 \cdot 10^{-3} + 2 \cdot 10^{-3} = 6 \cdot 10^{-3} \text{ seg} = 6 \text{ mseg}$$

Luego el tiempo medio en leer o escribir 10^3 bytes es $t_n = 6$ mseg.

SOLUCION PROBLEMA 2.43**DATOS**

- Sistema jerárquico de memoria.
- $C_{Mc}=256$ palabras, $K=8$ pal/ bloque, $t_{ac}=20$ nseg
- $C_{Mp}=1024$ Kpalabras, $t_{ap}=200$ nseg
- Tasa de acierto $h=0.90$
- Cuando se produce un fallo en M_c :
 - Se mueve el bloque completo a la M_C .
 - Se lee el dato desde M_C .

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que:

$$t_{ga} = t_{ca} = 20 \text{ nseg}$$

De acuerdo con el enunciado el tiempo de gestión de un fallo será la suma de dos componentes, ya que cuando se produce un fallo:

- 1) El sistema mueve el bloque completo a la caché, es decir hay que realizar K accesos a la memoria principal, lo que supone un tiempo de $K \cdot t_{ap}$.
- 2) Se lee el dato desde la caché, lo que supone un tiempo de t_{ac} .

Luego:

$$t_{gf} = K \cdot t_{ap} + t_{ac} = 8 \cdot 200 + 20 = 1620 \text{ nseg}$$

Sustituyendo valores en la formula (1) se obtiene que :

$$t_{am} = 0.90 \cdot 20 + 0.10 \cdot 1620 = 18 + 162 = 180 \text{ nseg}$$

Luego la solución es $t_{am}=180$ nseg.

SOLUCION PROBLEMA 2.44**DATOS**

- Número total de accesos $N_T=10^{20}$
- Número de fallos $N_F=10^{18}$

La definición de tasa de fallos es:

$$\text{Tasa de fallos} = (1 - h) = \frac{N_F}{N_T}$$

Despejando la tasa de aciertos h:

$$h = 1 - \frac{N_F}{N_T} = 1 - \frac{10^{18}}{10^{20}} = 1 - \frac{1}{10^2} = 1 - 0.01 = 0.99$$

Luego la tasa de aciertos $h=99\%$.

SOLUCION PROBLEMA 2.45**DATOS**

- Función de correspondencia asociativa por conjuntos:
- ETIQUETA: 20 bits, CONJUNTO: 7 bits, PALABRA: 5 bits.
- $r=2$ bloques/conjunto.
- L =Longitud de palabra es 1 byte.

Puesto que se conoce el número de bits $c=7$ bits del campo CONJUNTO con el que se codifica el número de conjunto de la M_C , es posible obtener el número de conjuntos total q de la M_C :

$$q = 2^c = 2^7 = 128 \text{ conjuntos}$$

El número de bloques C de la memoria caché se calcula de la siguiente forma:

$$C = q(\text{conjuntos}) \cdot r(\text{bloques / conjunto}) = 2^7 \cdot 2 = 2^8 \text{ bloques}$$

Por otra parte el número de bits $p=5$ utilizados para el campo PALABRA con el que se codifica la posición de una palabra dentro de un bloque, es posible obtener el número de palabras K que tiene un bloque:

$$K = 2^p = 2^5 = 32 (\text{palabras / bloque})$$

Finalmente la capacidad C_{Mc} de la memoria caché viene dada por:

$$C_{Mc} = K \cdot C \cdot L = 2^5 (\text{palabras / bloque}) \cdot 2^8 (\text{bloques}) \cdot 1 (\text{byte / palabra})$$

$$C_{Mc} = 2^{13} (\text{palabras}) \cdot 1 (\text{byte / palabra}) = 8 \text{ Kbytes}$$

Luego $C_{Mc}=8 \text{ Kbytes}$

SOLUCION PROBLEMA 2.46**DATOS**

- Se desea acercar la velocidad de la unidad de memoria de un computador a la velocidad de los registros de su CPU, a un coste razonable.

Para resolver este problema hay que recordar las propiedades de un sistema jerárquico de memoria.

Afirmación A: Es **falsa** ya que aumentar el número de registros de la CPU no influye en la velocidad de la unidad de memoria principal, además es muy costoso.

Afirmación B: Es **verdadera**, ya que incluir una memoria caché en el sistema disminuye el tiempo medio de acceso a la memoria y por lo tanto aumenta la velocidad. Además a un coste razonable.

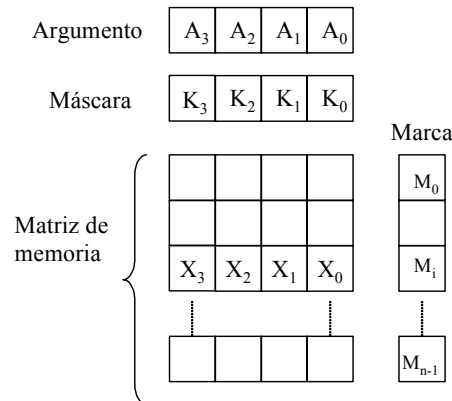
Afirmación C: Es **falsa**, aumentar la capacidad de los discos magnéticos no influye para nada en la velocidad de la unidad de memoria.

Afirmación D: Es **falsa**, la afirmación B es verdadera.

SOLUCION PROBLEMA 2.47**DATOS**

- Memoria asociativa de n palabras y 4 bits/palabra.
- El registro de marca debe estar a 1 si la celda de memoria asociativa contiene un número PAR, sino estará a 0.

De acuerdo con el enunciado se tendría un sistema de memoria asociativa como el que se ilustra en la Figura:



Para que el bit M_i del registro de marca valga 1 la palabra i-esima $X_3 \ X_2 \ X_1 \ X_0$ de la matriz de memoria debe ser un número PAR, eso implica que el bit menos significativo $X_0 = 0$. Por lo tanto el registro argumento con el que se va a comparar debe ser de la forma

$$A = [A_3 \ A_2 \ A_1 \ 0]$$

Donde $A_j \ j=1,2,3$ pueden valer 0 o 1.

Y el registro de máscara K sólo debe tener activo el bit menos significativo, es decir, la comparación entre A y una palabra de la matriz de celdas se debe restringir al bit menos significativo.

$$K = [0 \ 0 \ 0 \ 1]$$

De las cuatro posibles respuestas la única que cumple estos dos requisitos para A y para K es la respuesta **C**.

SOLUCION PROBLEMA 2.48**DATOS**

- Disco magnético de 1024 pistas, numeradas del 0 al 1023.
- La cola de peticiones de acceso es: [850, 25, 308, 400, 632, 168, 720, 302].
- Se usa planificación LOOK.
- La cabeza se halla en la pista 500.

La planificación LOOK se caracteriza por ir recorriendo todas las pistas en una dirección y atendiendo todas las peticiones que se encuentre en el camino, hasta que alcanza la última pista o no hay más peticiones en esa dirección. En este punto se invierte el sentido del recorrido y la búsqueda prosigue de la misma forma.

En el enunciado se indica que la cabeza se halla en la pista 500 pero no se dice el sentido del desplazamiento, luego se pueden dar dos posibles soluciones:

Caso A: Se halla en la pista 500 y se desplaza hacia las pistas cuyo número de pista va aumentando.

- 1) De la cola hay que fijarse en aquellas peticiones cuyo número de pista es mayor que 500, este el caso de [850, 632, 720].
- 2) Se colocan en orden creciente las peticiones del paso 1: [632, 720, 850].
- 3) Ahora la cabeza lectora invierte el sentido de desplazamiento e iría hacia las pistas cuyo número de pista va decreciendo. Luego las peticiones pendientes son [25, 308, 400, 168, 302].
- 4) Ordenando en orden decreciente las peticiones del paso 3: [400, 308, 302, 168, 25].
- 5) Luego el orden de atención de las peticiones es la unión del resultado del paso 2 y del paso 4:

[632, 720, 850, 400, 308, 302, 168, 25]

Caso B: Se halla en la pista 500 y se desplaza hacia las pistas cuyo número de pista va decreciendo.

- 1) De la cola hay que fijarse en aquellas peticiones cuyo número de pista es menor que 500, este el caso de [25, 308, 400, 168, 302].
- 2) Se colocan en orden decreciente las peticiones del paso 1: [400, 308, 302, 168, 25].
- 3) Ahora la cabeza lectora invierte el sentido de desplazamiento e iría hacia las pistas cuyo número de pista va aumentando. Luego las peticiones pendientes son [850, 632, 720].
- 4) Ordenando en orden creciente las peticiones del paso 3: [632, 720, 850].
- 5) Luego el orden de atención de las peticiones es la unión del resultado del paso 2 y del paso 4:

[400, 308, 302, 168, 25, 632, 720, 850]

SOLUCION PROBLEMA 2.49**DATOS**

- Disco magnético de 256 pistas, numeradas del 0 al 255.
- La cola de peticiones de acceso es: [80,130,151,31,20,200].
- Se usa planificación LOOK.
- La cabeza se halla en la pista 150 y en dirección de las pistas crecientes.

El razonamiento que hay que seguir es el mismo que en el problema anterior con la diferencia de que ahora sí que indica el sentido del desplazamiento inicial.

Por el enunciado se sabe que la cabeza se halla en la pista 150 y se está moviendo en la dirección de las pistas crecientes. Para resolver el problema hay que seguir los siguientes pasos:

- 1) De la cola hay que fijarse en aquellas peticiones cuyo número de pista es mayor que 150, este el caso de [151, 200].
- 2) Se colocan en orden creciente las peticiones del paso 1: [151, 200].
- 3) Ahora la cabeza lectora invierte el sentido de desplazamiento e irá hacia las pistas cuyo número de pista va decreciendo. Luego las peticiones pendientes son [80, 130, 31, 20].
- 4) Se ordenan en orden decreciente las peticiones del paso 3: [130, 80, 31, 20].
- 5) Luego el orden de atención de las peticiones es la unión del resultado del paso 2 y del paso 4:

[151, 200, 130, 80, 31, 20]

SOLUCION PROBLEMA 2.50**DATOS**

- Memoria caché con $C=32$ bloques.
- $K=256$ pal/bloque.
- Función de correspondencia totalmente asociativa.
- $n=24$ bits.
- En que bloque i de M_c se ubicará la dirección de memoria 001FFF

Para resolver este ejercicio no hace falta hacer ningún cálculo, únicamente hay que tener muy claro el funcionamiento de una memoria caché que utiliza función de correspondencia totalmente asociativa. De acuerdo con el libro de texto (página 84) cuando se utiliza este tipo de función de correspondencia se permite que un bloque j de la memoria principal se cargue en cualquier bloque i de la memoria caché. Luego la respuesta correcta es la **C**.

SOLUCION PROBLEMA 2.51**DATOS**

- Memoria principal formada por módulos RAM de $C_0=64$ Kpalabras x 16 bits/palabra

Vamos a analizar si las afirmaciones que se proponen son verdaderas o falsas.

La capacidad C_0 se puede expresar de forma equivalente como $C_0=2^6$ Kpalabras x 2^4 bits/palabra

AFIRMACION I:

Se desea una memoria de capacidad total $C_T=2^8$ Kpalabras x 2^5 bits/palabras.

Forma 1 de verificar la afirmación: Se parte de C_T y se descompone en $N \cdot C_0$
Se pueden hacer las siguientes manipulaciones sobre C_T

$$C_T = 2^2 \cdot 2^6 \text{ Kpalabras} \times 2 \cdot 2^4 \text{ bits/palabras} = 2^3 \cdot [2^6 \text{ Kpalabras} \times 2^4 \text{ bits/palabras}] = 8 \cdot C_0$$

Es decir, se necesitan 8 módulos de capacidad C_0 para proporcionar la memoria C_T pedida. Luego la afirmación es **VERDADERA**

Forma 2 de verificar la afirmación: Demostrar que $8 \cdot C_0 \geq C_T$ (donde aquí el signo \geq se debe considerar tanto al número de palabras de la memoria como al número de bits que posee cada palabra) . [Se recomienda al alumno/a que pruebe a verificar la afirmación de esta forma]

AFIRMACION II:

Se desea una memoria de capacidad total $C_T=2^7$ Kpalabras x 8 bits/palabras.

Forma 2 de verificar la afirmación: Demostrar que $2 \cdot C_0 \geq C_T$ (donde aquí el signo \geq se debe considerar tanto al número de palabras de la memoria como al número de bits que posee cada palabra). Se tiene que:

$$2 \cdot C_0 = 2 \cdot [2^8 \text{ Kpalabras} \times 2^6 \text{ bits/palabras}]$$

Con $2C_0$ Se podría formar dos posibles configuraciones:

Configuración 1: $C_{T1}=2^9$ Kpalabras x 2^6 bits/palabras

Configuración 2: $C_{T2}=2^8$ Kpalabras x 2^7 bits/palabras

Se verifican que:

$$C_{T1} \geq C_T$$

$$C_{T2} \geq C_T$$

Luego la afirmación es **VERDADERA**.

Comentario 1: Puesto que en la afirmación no se indica lo contrario, esta afirmación se consideraría verdadera con que una de las dos configuraciones C_{T1} o C_{T2} hubiese sido mayor o igual a C_T .
Recuérdese que el signo \geq se debe considerar tanto al número de palabras de la memoria como al número de bits que posee cada palabra.

Comentario 2: En ambas configuraciones se esta desaprovechando memoria, ya que la memoria necesaria C_T es menor que la disponible C_{T1} o C_{T2} .

SOLUCION PROBLEMA 2.52

La solución a este problema se encuentra en el archivo `probT2_52.pdf`

SOLUCION PROBLEMA 2.53**DATOS**

- Disco formateado con entrelazado doble.
- $S=16$ sectores de $C_S=4$ Kbytes.
- Velocidad de rotación de $f=6000$ rpm $=100$ (rev/s)
- La cabeza lectora se encuentra en la pista correcta y sobre el punto de comienzo del sector 0.

El tiempo t_p que se tarda en leer todos los sectores de una pista es la suma de tres contribuciones:

$$t_p = t_b + t_r + t_t$$

donde:

- t_b es el *tiempo medio de búsqueda de la pista*. En este caso puesto que la cabeza lectora está posicionada ya en la pista adecuada $t_b=0$ s.
- t_r es el *tiempo de latencia rotacional* que en este caso vale $t_r=0$ s, puesto que la cabeza lectora está posicionada al comienzo del sector adecuado.
- t_t es el *tiempo de transferencia de todos los sectores de una pista*. Al existir entrelazado doble para leer la pista se comienza leyendo el sector 0 y después, para el resto de sectores, primero la cabeza pasa por encima de dos sectores que ignora y lee el tercero. Por tanto, la cabeza lectora pasa por un total de

$$N_s = 1 + (f_e + 1) \cdot (S - 1) = 1 + (2 + 1) \cdot (16 - 1) = 46 \text{ sectores}$$

Por lo tanto

$$t_t = \frac{N_s}{S \cdot f} = \frac{46}{16 \cdot 100} = 0.02875 \text{ s} = 28.75 \text{ ms}$$

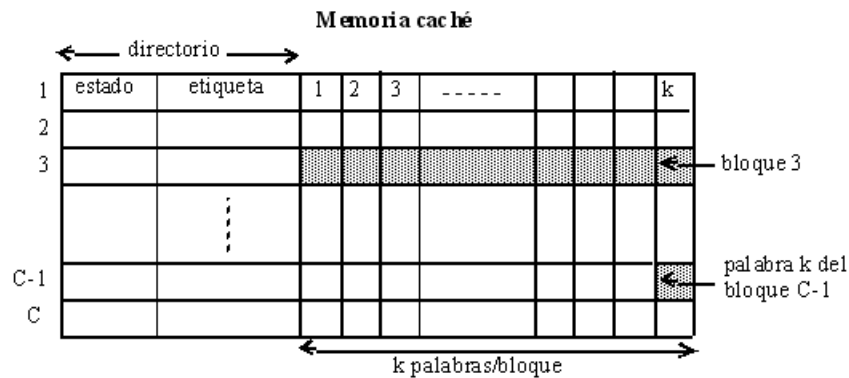
Luego:

$$t_p = t_b + t_r + t_t = 0 + 0 + 28.75 = 28.75 \text{ ms}$$

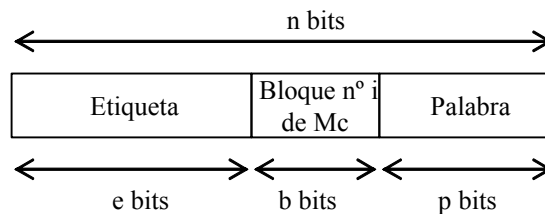
SOLUCION PROBLEMA 2.54**DATOS**

- $C_{Mp} = 2^{32}$ palabras x 8 bits/palabra
- Memoria caché con correspondencia directa
- $C_{Mc} = 64$ Kpalabras.
- $K = 4$ palabras/bloque
- Estrategia de escritura write-back

La estructura típica de una memoria caché es la indicada en la figura



Para resolver este problema en primer lugar hay que obtener el formato de una dirección de memoria desde el punto de vista de la memoria caché, como se utiliza función de correspondencia directa, el formato es:



Del dato $C_{Mp} = 2^{32}$ pal se obtiene n :

$$n = \log_2 2^{32} = 32 \text{ bits}$$

Del dato K se obtiene p :

$$p = \log_2 K = \log_2 4 = 2 \text{ bits}$$

Por otro lado el número de bloques C de M_c es:

$$C = \frac{C_{Mc}}{K} = \frac{64 \text{ Kpal}}{4 \text{ pal/bloque}} = \frac{2^{16} \text{ pal}}{2^2 \text{ pal/bloque}} = 2^{14} \text{ bloques}$$

con lo b es:

$$b = \log_2 C = \log_2 2^{14} = 14 \text{ bits}$$

Y e es:

$$e = n - (b + p) = 32 - (14 + 2) = 16 \text{ bits}$$

En segundo lugar, puesto que se usa estrategia de escritura *write back* en el directorio se necesita un bits de estado por bloque para indicar si dicho bloque se debe escribir en la memoria principal cuando sea reemplazado.

Por lo tanto la estructura de la memoria caché es:

Directorio					
	1 bit Estado	16 bits Etiqueta	8 bits Palabra	8 bits Palabra	8 bits Palabra
.1	.				
.2	.				
∴	.				
⋮					
2^{14}					

Luego para cada uno de los 2^{14} bloques, se almacenan 49 bits, que se distribuyen de la siguiente forma:

- 1 del bit de estado.
- 16 bits de etiqueta
- 32 bits para almacenar las 4 palabras de un bloque (8 bits/palabra).

En conclusión el número mínimo de bits que se requiere para diseñar esta memoria caché es: $2^{14} \times 49$

SOLUCION PROBLEMA 2.55

Este problema es similar al Problema 2.16, por lo que se van a omitir las explicaciones pormenorizadas.

AFIRMACION I:

		<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>																1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	ArgumentoA
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0																																				
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1																																				
		<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>																0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	Máscara K																	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1																																				
ETIQUETA	0		0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	Palabra 0	0	MARCA																															
	1		1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	1	Palabra 1	0																																
	1		0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	Palabra 2	1																																	
	1		1	1	1	1	0	0	0	0	0	0	0	0	1	0	1	1	Palabra 3	0																																
	0		1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	Palabra 4	0																																
	1		1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	Palabra 5	1																																

La afirmación es **FALSA** ya que también valdría 1 el bit de marca de la palabra 5.

AFIRMACION II:

		<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>																1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	ArgumentoA
1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1																																				
1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1																																				
		<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>																1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	Máscara K																	
1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1																																				
ETIQUETA	0		0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	Palabra 0	0	MARCA																															
	1		1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	1	Palabra 1	0																																
	1		0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	Palabra 2	0																																
	1		1	1	1	1	0	0	0	0	0	0	0	0	1	0	1	1	Palabra 3	0																																
	0		1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	Palabra 4	0																																
	1		1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	Palabra 5	0																																

La afirmación es **FALSA** ya que todos los bits de marca valen 0.

SOLUCION PROBLEMA 2.56

DATOS

- 16 bits de longitud de palabra
- Bus de direcciones de 24 bits ($A_{23}...A_0$)
- $C_T = 2^5$ Mpalabras x 16 bits/palabra.
- Módulos RAM disponibles de $C_0 = 1$ Mpalabra x 1 bits/palabra.
- Direcciones de memoria consecutivas empezando en la 0

Puesto que se conoce la capacidad total C_T de la memoria que se desea diseñar y la capacidad de los módulos disponibles C_0 es posible calcular el número de módulos de estas características necesarios para implementar esta memoria:

$$N = \frac{C_T}{C_0} = \frac{2^5 \text{ Mpalabras} \times 16 \text{ bits / palabra}}{1 \text{ Mpalabras} \times 1 \text{ bits / palabra}} = 2.5 \times 16 \approx 3 \times 16 = 48 \text{ módulos}$$

Es decir, para implementar esta memoria se necesitan tres filas de 16 módulos (de capacidad C_0) cada una. Con este diseño se tiene una capacidad total de 3 Mpalabras, por tanto se tendrá que prohibir el

acceso a la mitad de las palabras de la última fila de módulos. Para codificar $3 \cdot M = 3 \cdot 2^{20} \leq 4 \cdot 2^{20}$ palabras se requieren 22 bits. Una posible utilización de las 24 líneas del bus de direcciones sería:

- Líneas A_{23} y A_{22} no tienen ningún uso, por lo que se pueden hacer iguales a 0.
- Líneas A_{21} y A_{20} permiten seleccionar la fila de módulos donde se va a leer la palabra.

$(A_{21}=0, A_{20}=0) \rightarrow$ Primera línea de módulos.

$(A_{21}=0, A_{20}=1) \rightarrow$ Segunda línea de módulos.

$(A_{21}=1, A_{20}=0) \rightarrow$ Tercera línea de módulos.

- Líneas A_{19} y A_0 permiten seleccionar una palabra ubicada en una línea de módulos.

Se tiene en consecuencia el siguiente mapa de direcciones:

	$A_{23} \dots \dots \dots A_0$
1ª Fila	0000 0000 0000 0000 0000 0000
	0000 1111 1111 1111 1111 1111
2ª Fila	0001 0000 0000 0000 0000 0000
	0001 1111 1111 1111 1111 1111
3ª Fila (uso)	0010 0000 0000 0000 0000 0000
	0010 0111 1111 1111 1111 1111
3ª Fila (prohibido)	0010 1000 0000 0000 0000 0000
	0010 1111 1111 1111 1111 1111
Resto de mapa de memoria	0011 0000 0000 0000 0000 0000
	1111 1111 1111 1111 1111 1111

En consecuencia las direcciones prohibidas (superiores a $2^{25} M$) se pueden producir en los siguientes casos:

Caso 1) Si $A_{23}=1$

Caso 2) Si $A_{22}=1$

Caso 3) Si $A_{21}=1$ y $A_{20}=1$

Caso 4) Si $A_{21}=1$ y $A_{19}=1$

La expresión lógica asociada a las direcciones prohibidas es:

$$A_{23} + A_{22} + A_{21} \cdot A_{20} + A_{21} \cdot A_{19} = A_{23} + A_{22} + A_{21} \cdot (A_{20} + A_{19})$$

Se tiene por tanto que:

AFIRMACION I: Es **verdadera**, la expresión lógica indicada en la afirmación coincide con la obtenida.

AFIRMACION II: Es **falsa**, se requieren 48 módulos de RAM no 32.

SOLUCION PROBLEMA 2.57**DATOS**

- Disco magnético con 1024 pistas {0,1,...,1023}
- Cola de peticiones: 2, 35,46,23, 90,102,10, 34
- Planificación LOOK
- Inicialmente la cabeza lectora se encuentra en la pista 29

La planificación LOOK se caracteriza por ir recorriendo todas las pistas en una dirección y atendiendo todas las peticiones que se encuentre en el camino, hasta que alcanza la última pista o no hay más peticiones en esa dirección. En este punto se invierte el sentido del recorrido y la búsqueda prosigue de la misma forma.

En el enunciado se indica que la cabeza se halla en la pista 29 pero no se dice el sentido del desplazamiento, luego se pueden dar dos posibles soluciones:

Caso A: Se halla en la pista $p_0=29$ y se desplaza hacia las pistas cuyo número de pista va aumentando.

- 1) De la cola hay que fijarse en aquellas peticiones cuyo número de pista es mayor que 29, este el caso de [35, 46, 90, 102, 34].
- 2) Se colocan en orden creciente las peticiones del paso 1: [34, 35, 46, 90, 102].
- 3) Ahora la cabeza lectora invierte el sentido de desplazamiento e iría hacia las pistas cuyo número de pista va decreciendo. Luego las peticiones pendientes son [2, 23, 10].
- 4) Ordenando en orden decreciente las peticiones del paso 3: [23, 10, 2].
- 5) Luego el orden de atención de las peticiones es la unión del resultado del paso 2 y del paso 4:

[34, 35, 46, 90, 102, 23, 10, 2]

Se puede construir la siguiente tabla

Próxima pista a la que se accede	(p ₁)	(p ₂)	(p ₃)	(p ₄)	(p ₅)	(p ₆)	(p ₇)	(p ₈)
	34	35	46	90	102	23	10	2
Número de pistas que se atraviesan	p ₁ -p ₀	p ₂ -p ₁	p ₃ -p ₂	p ₄ -p ₃	p ₅ -p ₄	p ₆ -p ₅	p ₇ -p ₆	p ₈ -p ₇
	5	1	11	44	12	79	13	8

Si se denota por N al número de peticiones en la cola., entonces la longitud media de búsqueda se calcula mediante la siguiente expresión:

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N}$$

$$LMB = \frac{5+1+11+44+12+79+13+8}{8} = 21.625$$

Caso B: Se halla en la pista $p_0=29$ y se desplaza hacia las pistas cuyo número de pista va decreciendo.

- 1) De la cola hay que fijarse en aquellas peticiones cuyo número de pista es menor que 29, este el caso de [23, 10, 2].
- 2) Se colocan en orden decreciente las peticiones del paso 1: [23, 10, 2].
- 3) Ahora la cabeza lectora invierte el sentido de desplazamiento e iría hacia las pistas cuyo número de pista va aumentando. Luego las peticiones pendientes son [35, 46, 90, 102, 34].
- 4) Ordenando en orden creciente las peticiones del paso 3: [34, 35, 46, 90, 102].
- 5) Luego el orden de atención de las peticiones es la unión del resultado del paso 2 y del paso 4:

[23, 10, 2, 34, 35, 46, 90, 102,]

Se puede construir la siguiente tabla

Próxima pista a la que se accede	(p ₁) 23	(p ₂) 10	(p ₃) 2	(p ₄) 34	(p ₅) 35	(p ₆) 46	(p ₇) 90	(p ₈) 102
Número de pistas que se atraviesan	p ₁ -p ₀ 6	p ₂ -p ₁ 13	p ₃ -p ₂ 8	p ₄ -p ₃ 32	p ₅ -p ₄ 1	p ₆ -p ₅ 11	p ₇ -p ₆ 44	p ₈ -p ₇ 12

Si se denota por N al número de peticiones en la cola., entonces la longitud media de búsqueda se calcula mediante la siguiente expresión:

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N}$$

$$LMB = \frac{6 + 13 + 8 + 32 + 1 + 11 + 44 + 12}{8} = 15.875$$

Luego ya se posee la información necesaria para analizar la veracidad de las afirmaciones:

Afirmación I: Es **FALSA** ya que en un caso es LMB=21.625 y en el otro es LMB=15.875.

Afirmación II: Es **VERDADERA** se corresponde con el resultado expuesto en el caso A

SOLUCION PROBLEMA 2.58

DATOS

- C_T=2048 palabras x 8 bits/palabra
- C₀= 128 palabras x 8 bits/palabra

Se desea tener una memoria de capacidad total C_T= 2¹¹ palabras x 8 bits/palabras usando módulos de capacidad C₀=2⁷ palabras x 8 bits/palabra

El número de líneas totales del bus de dirección se obtiene a partir del dato del número de palabras de C_T

$$n^{\circ} \text{ líneas del Bus} = \log_2 2^{11} = 11 \text{ líneas}$$

Se pueden hacer las siguientes manipulaciones sobre C_T para expresarla en función de C₀:

$$C_T = 2^4 \cdot 2^7 \text{ palabras x 8 bits/palabras} = 2^4 \cdot [2^7 \text{ palabras x 8 bits/palabras}] = 16 \cdot C_0$$

Es decir, se necesitan 16 módulos de capacidad C₀ para proporcionar la memoria C_T pedida. El número de líneas para seleccionar uno de estos 16 módulos es:

$$n^{\circ} \text{ líneas del Bus} = \log_2 16 = \log_2 2^4 = 4 \text{ líneas}$$

Por lo tanto de las 11 líneas del bus, 4 líneas se utilizan para seleccionar entre uno de los 16 módulos de capacidad C₀. Las 7 restantes serán comunes a todos los módulos y permiten seleccionar una de las 128 palabras que contiene cada módulo.

En conclusión la respuesta correcta es la A.

SOLUCION PROBLEMA 2.59

La solución a este problema se encuentra en el archivo probT2_59.pdf

SOLUCION PROBLEMA 2.60**DATOS**

- Disco formateado con entrelazado simple $f_e=1$
- $S=4$ superficies
- $N_C=128$ cilindros
- $S=16$ sectores/pista
- $C_S=4$ Kbytes/sector
- $f=6000$ rev/min= $6000/60=100$ rev/s

Puesto que no se dice lo contrario en el enunciado para resolver este problema se va a suponer que la cabeza lectora se encuentra posicionada en la pista correcta al comienzo del sector 0.

El tiempo t_p que se tarda en leer todos los sectores de una pista es la suma de tres contribuciones:

$$t_p = t_b + t_r + t_t$$

Donde

- t_b es el tiempo medio de búsqueda de una pista. Se ha supuesto que la cabeza lectora ya se encuentra sobre la pista correcta luego $t_b=0$ ms.

- t_r es el tiempo de latencia rotacional o retardo rotacional. Es decir el tiempo medio que tarda el sector en estar debajo de la cabeza de lectura/escritura. Se ha supuesto que la cabeza se encuentra al comienzo del sector 0 luego $t_r=0$ ms.

- t_t es el tiempo de transferencia de todos los sectores de una pista. Se calcula mediante la siguiente expresión

$$t_t = \frac{N_s}{S \cdot f}$$

Donde N_s es el número de sectores por los que pasa la cabeza lectora. Al existir entrelazado simple se comienza leyendo el sector 0 y después, para el resto de sectores, primero la cabeza pasa por encima de un sector que ignora y lee el siguiente. Por tanto, la cabeza lectora pasa por un total de

$$N_s = (f_e + 1) \cdot S = (1 + 1) \cdot 16 = 32 \text{ sectores}$$

Por lo tanto

$$t_t = \frac{N_s}{S \cdot f} = \frac{32}{16 \cdot 100} = 0.02 \text{ s} = 20 \text{ ms}$$

Luego:

$$t_p = t_b + t_r + t_t = 0 + 0 + 20 = 20 \text{ ms}$$

La velocidad de transferencia v_T será el cociente entre los P bytes que hay en una pista y el tiempo t_p que se tarda en transmitir dichos bytes.

$$v_T = \frac{P}{t_p}$$

Con los datos que se dan en el enunciado es posible calcular P:

$$P = C_s \cdot S = 4(Kbytes / sector) \cdot 16(sector / pista) = 64(Kbytes / pista)$$

Luego

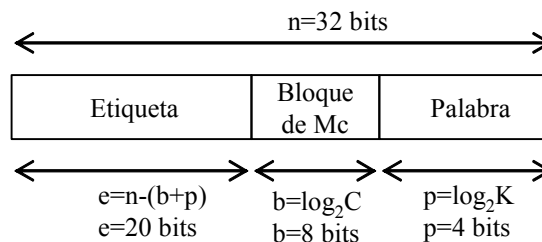
$$v_t = \frac{64(Kbytes / pista)}{0.02(s / pista)} = 3200(Kbytes / s)$$

SOLUCION PROBLEMA 2.61

DATOS

- $C=256$ bloques de M_c
- $K=16$ palabras/bloque
- Función de correspondencia directa.
- $n=32$ bits

Con los datos del enunciado es posible obtener el formato de una dirección desde el punto de vista de la memoria caché:



Si se considera que la dirección viene expresada en hexadecimal puesto que una dirección tiene 20 bits se requerirían 8 cifras hexadecimales $(D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0)_{16}$, cuyo significado sería el siguiente:

$(D_7 D_6 D_5 D_4 D_3)_{16}$	→ Etiqueta
$(D_2 D_1)_{16}$	→ Bloque de Mc
$(D_0)_{16}$	→ Palabra

Con estas consideraciones previas se está en disposición de poder analizar las afirmaciones propuestas:

Afirmación I:

La dirección A3271 13 3₁₆ se ubica en el bloque 13₁₆ de memoria caché

La dirección A3261 13 7₁₆ se ubica en el bloque 13₁₆ de memoria caché

Luego ambas direcciones se ubicarán en el mismo bloque de memoria caché (aunque no simultáneamente). La afirmación es **VERDADERA**.

Afirmación II:

La dirección A3502 12 7₁₆ se ubica en el bloque 12₁₆ que expresado en binario toma la forma 00010010₂. Y pasando este número a decimal se obtiene que esta dirección se ubicaría en el bloque $2^4 + 2^1 = 18_{10}$ de la memoria caché. En conclusión la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 2.62

En una operación de escritura sobre una memoria asociativa se selecciona para escribir la primera palabra que tenga su bit de etiqueta a 0. En este caso la **Palabra 1**.

SOLUCION PROBLEMA 2.63

DATOS

- Memoria caché (M_c) con función de correspondencia directa
- $C=128$ bloques en la caché.
- $M=16K$ bloques en la memoria principal (M_p).
- $K=16$ palabras/bloque
- $t_{ac}=10$ ns
- El tiempo en cargar un bloque en la M_c es 200 ns.
- La caché está inicialmente vacía.
- En caso de fallo primero se trae todo el bloque desde la M_p a M_c y luego se lee la palabra en M_c .
- Se ejecuta un programa que ejecuta un bucle 10 veces que va desde las posiciones 15 a 200.

a) La capacidad de la memoria principal C_{Mp} es

$$C_{Mp} = M \cdot K = 16K(\text{bloques}) \cdot 16(\text{palabras / bloque}) = 2^4 \cdot 2^{10} \cdot 2^4 = 2^{18} (\text{palabras})$$

Luego el número de bits n de una dirección de memoria es:

$$C_{Mp} = 2^{18} = 2^n \Rightarrow n = 18 \text{ bits}$$

El número de bits p que se necesitan para codificar la posición de una palabra en el interior de un bloque es:

$$K = 16 = 2^4 = 2^p \Rightarrow p = 4 \text{ bits}$$

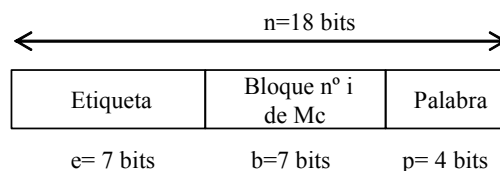
El número de bits b que se necesitan para codificar un bloque i de la memoria caché es:

$$C = 128 = 2^7 = 2^b \Rightarrow b = 7 \text{ bits}$$

Finalmente el número de bits e del campo etiqueta es:

$$e = n - (b + p) = 18 - (7 + 4) = 7 \text{ bits}$$

Luego el formato de una dirección de memoria desde el punto de vista de la memoria caché es



b) Se va a estudiar el número de fallos o de aciertos que se producen en la caché al ejecutarse por primera vez el bucle del programa que se extiende desde la dirección 15 hasta la 200.

En primer lugar se busca se busca en la caché la dirección 15. Como la caché está inicialmente vacía, se produce un fallo y se almacena el bloque $j=0$ de la Mp, asociado a las direcciones 0 a 15, en el bloque $i=0$ de la Mc.

A continuación se busca en la caché la dirección 16 como la caché está inicialmente vacía se produce un fallo y se almacena el bloque $j=0$ de la Mp asociado a las direcciones 16 a 31, en el bloque $i=1$ de la Mc.

Cuando se vaya a buscar en la caché las direcciones 17 a 31, se producirá aciertos ya que todas ellas pertenecen al bloque $j=1$ de Mp que ya está cargado en la caché, en concreto en su bloque $i=1$ de la Mc.

Luego se observa que puesto que la memoria caché está inicialmente vacía cuando se accede a la primera dirección contenida en un bloque j de Mp que no esté en la caché se producirá un fallo, lo que produce que dicho bloque sea copiado en un bloque i de Mc. En consecuencia al acceder a las restantes 15 direcciones contenidas en el bloque j como ya está copiado en Mc se producirán 15 aciertos.

En la Tabla 1 se resumen los fallos o aciertos que se producen en la caché al ejecutarse por primera vez el bucle.

Direcciones	Bloque j de Mp	Fallo a aciertos
15	0	1 fallo se copia el bloque $j=0$ en el bloque $i=0$
16	1	1 fallo se copia el bloque $j=1$ en el bloque $i=1$
17-31	1	15 aciertos
32	2	1 fallo se copia el bloque $j=2$ en el bloque $i=2$
33-47	2	1 fallo y 15 aciertos
48	3	1 fallo se copia el bloque $j=3$ en el bloque $i=3$
49-63	3	1 fallo y 15 aciertos
64	4	1 fallo se copia el bloque $j=4$ en el bloque $i=4$
65-79	4	1 fallo y 15 aciertos
80	5	1 fallo se copia el bloque $j=5$ en el bloque $i=5$
81-95	5	1 fallo y 15 aciertos
96	6	1 fallo se copia el bloque $j=6$ en el bloque $i=6$
97-111	6	1 fallo y 15 aciertos
112	7	1 fallo se copia el bloque $j=7$ en el bloque $i=7$
113-127	7	1 fallo y 15 aciertos
128	8	1 fallo se copia el bloque $j=8$ en el bloque $i=8$
129-143	8	1 fallo y 15 aciertos
144	9	1 fallo se copia el bloque $j=9$ en el bloque $i=9$
145-159	9	1 fallo y 15 aciertos
160	10	1 fallo se copia el bloque $j=10$ en el bloque $i=10$
161-175	10	1 fallo y 15 aciertos
176	11	1 fallo se copia el bloque $j=11$ en el bloque $i=11$
177-191	11	1 fallo y 15 aciertos
192	12	1 fallo se copia el bloque $j=12$ en el bloque $i=12$
193-200	12	1 fallo y 8 aciertos

Tabla 1

Luego en esta primera ejecución del bucle se producen:

Número de accesos $= (200-15)+1 = 186$ accesos.

Número de fallos $= 1(\text{fallo/bloque}) \times 13 \text{ bloques} = 13$ fallos.

Número de aciertos $= 15 (\text{aciertos/bloque}) \times 11 (\text{bloques}) + 8 (\text{aciertos/bloque}) \times 1 \text{ bloque}$
 $= 173$ aciertos.

En las siguientes 9 veces que se ejecuta el bucle del programa como los bloques ya están cargados en Mc se producen aciertos.

Luego el número total de accesos asociados a este programa es

$$N_T = 10 \text{ (bucles)} \cdot 186 \text{ (accesos/bucle)} = 1860 \text{ accesos.}$$

Mientras que el número total de aciertos es

$$N_A = 173 + 186 \times 9 = 1847$$

Luego la tasa de acierto del programa es:

$$h = \frac{N_A}{N_T} = \frac{1847}{1860} = 0.993 \Rightarrow 99.3 \%$$

c) El tiempo de acceso medio t_{am} al sistema $M_C - M_p$ viene dada por la expresión:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf}$$

donde t_{ga} es el tiempo de gestión de un acierto y t_{gf} es el tiempo de gestión de un fallo. En este caso, según el enunciado, se tiene $t_{ga} = 10 \text{ ns}$ y $t_{gf} = 200 + 10 = 210 \text{ ns}$. Luego sustituyendo valores se obtiene:

$$t_{am} = 0.993 \cdot 10 + 0.007 \cdot 210 = 9.93 + 1.47 = 11.4 \text{ ns}$$

SOLUCION PROBLEMA 2.64

DATOS

- 4 módulos de memoria RAM y 1 módulo de memoria ROM.
- Longitud de palabra $L = 8 \text{ bits}$.
- Bus de direcciones de 16 bits

Este problema se puede resolver de dos formas:

- FORMA 1

Si se escriben las direcciones de acceso a cada módulo en binario

Componente	Dirección hexadecimal	Dirección binaria
RAM 1	0000 – 007F	[0000 0000 0000 0000] – [0000 0000 0111 1111]
RAM 2	0080 – 00FF	[0000 0000 1110 0000] – [0000 0000 1111 1111]
RAM 3	0100 – 017F	[0000 0001 0000 0000] – [0000 0001 0111 1111]
RAM 4	0180 – 01FF	[0000 0001 1000 0000] – [0000 0001 1111 1111]
ROM	0200 – 03FF	[0000 0010 0000 0000] – [0000 0011 1111 1111]

Se observa que el formato general de las direcciones de cada módulo, supuesto que la variable X representa a un 1 o a un 0, es:

Componente	Formato general de una dirección [A ₁₅ A ₁₄ A ₁₃ A ₁₂ A ₁₁ A ₁₀ A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀]
RAM 1	[0000 0000 0XXX XXXX]
RAM 2	[0000 0000 1XXX XXXX]
RAM 3	[0000 0001 0XXX XXXX]
RAM 4	[0000 0001 1XXX XXXX]
ROM	[0000 001X XXXX XXXX]

Luego todas las direcciones que tienen $A_9=0$ son direcciones RAM, mientras que cuando $A_9=1$ son de ROM. Asimismo se observa que en el caso de los módulos RAM, A_8 y A_7 permiten seleccionar uno de los cuatro módulos de la RAM, y A_6-A_0 permiten seleccionar una palabra dentro de cada módulo RAM. Por lo tanto, en cada módulo RAM hay $2^7=128$ palabras.

En el caso de la ROM, A_8-A_0 permiten seleccionar una palabra dentro de este módulo, luego contiene $2^9=512$ palabras.

En conclusión tanto la afirmación I como la II son **verdaderas**.

- FORMA 2

Si se escriben las direcciones de acceso a cada módulo en decimal

Componente	Dirección hexadecimal	Dirección decimal
RAM 1	0000 – 007F	0-127
RAM 2	0080 – 00FF	128-255
RAM 3	0100 – 017F	256-383
RAM 4	0180 – 01FF	384-511
ROM	0200 – 03FF	512-1023

se observa fácilmente que cada módulo RAM contiene 128 palabras y que el módulo ROM contiene 512 palabras. Luego ambas afirmaciones son **Verdaderas**.

SOLUCION PROBLEMA 2.65

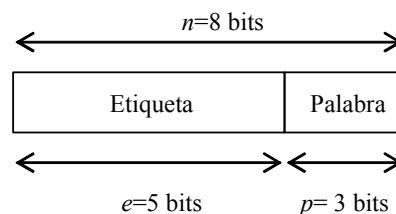
DATOS

- $C_{Mp} = 256$ palabras
- $C_{Mc} = 32$ palabras
- El tamaño de partición es $K=8$ palabras/bloque
- La caché está inicialmente vacía.
- Función de correspondencia totalmente asociativa

El número de bits n de una dirección de memoria se obtiene del dato de la capacidad de la memoria principal: $C_{Mp}=256=2^8=2^n$. Luego $n=8$ bits.

Por otra parte del dato del tamaño de un bloque se obtiene el número de bits p que se necesitan para codificar la posición de una palabra dentro de un bloque: $K=8=2^3=2^p$. Luego $p=3$ bits.

Puesto que la caché emplea *correspondencia totalmente asociativa*. La dirección desde el punto de vista de la M_c tendría los siguientes campos:



Por otra parte la caché puede almacenar $C=C_{Mc}/K=32/8=4$ bloques ($i=0,1,2$ y 3).

En la siguiente tabla se recoge la secuencia de direcciones leídas y los resultados que se producen al ir a buscarlas a M_c . Recuerdese que con la letra i se denota el número de bloque de memoria caché.

Direcciones leídas	Etiqueta= N° de bloque j de M _p (en decimal)	Fallo o acierto
00000 000	0	Fallo se carga j=0 en i=0
00000 001	0	Acierto
00000 011	0	Acierto
00100 001	4	Fallo se carga j=4 en i=1
00100 101	4	Acierto
00010 000	2	Fallo se carga j=2 en i=2
00010 010	2	Acierto
00000 000	0	Acierto

Por lo tanto si se leyera la dirección 00100 111 se produciría un acierto ya que se encuentra almacenada en el bloque i=1 de la caché. En conclusión la afirmación I es **verdadera**.

Por otra parte como la caché no se ha llenado no ha sido necesario realizar ninguna operación de reemplazamiento. En consecuencia la afirmación II es **falsa**.

SOLUCION PROBLEMA 2.66

DATOS

- Disco magnético con dos superficies. $N_s=2$ superficies.
- Radio interior de cada superficie $R_1=1$ cm.
- Radio exterior de cada superficie $R_2=5$ cm.
- El tamaño de cada pista es diferente pero almacena el mismo número de bits.
- La densidad de almacenamiento máxima es $\rho=10000$ bits/cm.
- Espaciado entre pistas adyacentes es $\varepsilon=0.1$ mm
- Se supone despreciable el espaciado entre sectores.
- Velocidad de rotación $f=3600$ rpm.
- El disco no tiene entrelazado.
- En un determinado instante sólo se puede acceder a una de las superficies del disco.

a) Puesto que se supone despreciable el espaciado entre sectores, entonces se va a considerar que la información se distribuye regularmente a lo largo de cada pista con la densidad lineal ρ dada en el enunciado.

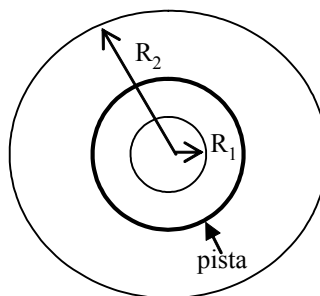


Figura 2.66.1

Para calcular el número máximo de bits que se puede almacenar en el disco, en primer lugar hay que calcular el número de pistas que contiene cada superficie N_{ps} , que es el cociente entre la anchura efectiva de almacenamiento R_2-R_1 (ver Figura 2.66.1) y la distancia entre pistas adyacentes ε :

$$N_{PS} = \frac{R_2 - R_1}{\varepsilon}$$

Sustituyendo valores y operando se obtiene

$$N_{PS} = \frac{5 - 1(\text{cm} / \text{superficie})}{0.1(\text{mm} / \text{pista})} = \frac{40(\text{mm} / \text{superficie})}{0.1(\text{mm} / \text{pista})} = 400(\text{pistas} / \text{superficie})$$

Por otra parte la capacidad de una pista j viene dada por la expresión

$$C_{Pj} = 2 \cdot \pi \cdot R_j \cdot \rho$$

Se observa que es función del radio de la pista R_j . Según el enunciado todas las pistas almacenan la misma cantidad de bits, luego cualquier pista no puede tener una capacidad mayor que la capacidad de la pista más interna, ya que es la que posee el menor radio de todas. Luego la capacidad de una pista de la superficie del disco es:

$$C_P = 2 \cdot \pi \cdot R_1 \cdot \rho = 2 \cdot \pi \cdot 1(\text{cm} / \text{pista}) \cdot 10000(\text{bits} / \text{cm}) = 62832(\text{bits} / \text{pista})$$

La capacidad de una superficie C_S vendrá dada por la expresión:

$$C_S = N_{PS} \cdot C_P = 400(\text{pistas} / \text{superficie}) \cdot 62832(\text{bits} / \text{pista}) = 25.13 \cdot 10^6 (\text{bits} / \text{superficie})$$

Luego la capacidad del disco será:

$$C_T = 2 \cdot C_S = 2 \cdot 25.13 \cdot 10^6 = 50.26 \cdot 10^6 \text{ bits} = 50.26 \cdot 10^6 \text{ Mbits}$$

b) La velocidad de transferencia de datos desde el disco a la cabeza de lectura/escritura en bits/s si no existe entrelazado en el disco y si sólo se puede acceder en un determinado a una cierta superficie, será el producto de la velocidad de rotación del disco por la capacidad de una pista.

$$\begin{aligned} v_T &= f \cdot C_P = 3600(\text{rev} / \text{min}) \cdot 62832(\text{bits}) = 60(\text{rev} / \text{s}) \cdot 62832(\text{bits}) \\ &= 3.77 \cdot 10^6 \text{ bits} / \text{s} = 3.77 \text{ Mbits} / \text{s} \end{aligned}$$

SOLUCION PROBLEMA 2.67

DATOS

- Memoria caché asociativa por conjuntos, con 4 bloques por conjuntos.
- Algoritmo de reemplazamiento LRU, modificado para evitar el rebosamiento de los contadores.
- A cada bloque se le asocia un contador de 2 bits.
- C_0 , C_1 , C_2 y C_3 son los contadores asociados a los bloques de la caché 0, 1, 2 y 3, respectivamente.

En la implementación del algoritmo de reemplazamiento LRU modificado para evitar el rebosamiento de los contadores C_i asociados a los bloques i de la caché, estos contadores siguen las siguientes reglas:

■ En caso de acierto:

- 1) El contador C_i asociado con el bloque i que contiene la palabra buscada se inicializa a 0, lo que indica que dicho bloque se ha utilizado recientemente.
- 2) Todos los contadores que tienen un valor más pequeño que el que tenía el contador del bloque accedido se incrementan en 1.
- 3) Todos los contadores que tienen un valor mayor que el que tenía el contador del bloque accedido no se modifican.

■ En caso de fallo:

- 4) Si la caché no está llena, el contador asociado con el bloque de la caché donde se carga el

bloque que se lee de la memoria principal se inicializa a 0, los restantes contadores se incrementan en 1.

- 5) Si la caché está llena, se sustituye el bloque de la caché que tiene el contador con el valor más elevado y se inicializa a 0 su contador, todos los demás contadores se incrementan en 1.

De acuerdo con estas reglas, cuando se accede al bloque 0 (se ha producido un acierto), el contador de ese bloque se pone a cero: $X_0 = 0$. Todos los contadores que tenían un valor más pequeño que el que tenía el contador del bloque accedido se incrementan en uno: $X_1 = 1$. Todos los contadores que tienen un valor mayor que el que tenía el contador del bloque accedido no se modifican: $X_2 = X_3 = 2$.

Luego, $(X_0, X_1, X_2, X_3) = (0, 1, 2, 2)$.

SOLUCION PROBLEMA 2.68

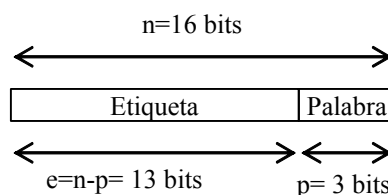
DATOS

- Memoria caché (M_c) con función de correspondencia totalmente asociativa.
- $C=16$ bloques en la caché.
- $K=8$ palabras/bloque.
- $C_{Mp}=2^{16}$ palabras.
- La caché está inicialmente vacía.
- $t_{ac}=40$ ns
- El tiempo en cargar un bloque en la M_c es $1\ \mu s$.
- En caso de fallo primero se trae todo el bloque desde la M_p a M_c y luego se lee la palabra en M_c .
- Se ejecuta un programa que ejecuta en primer lugar las instrucciones ubicadas en las direcciones de memoria 20 a 45, y luego efectúa 4 veces un bucle entre las posiciones de memoria 28 a 45.

a) El número n de bits de una dirección de memoria se obtiene del dato de la capacidad de la memoria principal $C_{Mp}=2^{16}=2^n \rightarrow n=16$ bits

El número p de bits que se necesitan para codificar la posición de una palabra en el interior de un bloque se obtiene del dato del número de palabras que almacena un bloque: $K=2^3=2^p \rightarrow p=3$ bits.

Luego el formato de una dirección de memoria desde el punto de vista de una memoria caché con correspondencia totalmente asociativa es:



b) Se va a estudiar el número de fallos o de aciertos que se producen en la caché al ejecutarse las instrucciones localizadas desde las posiciones de memoria 20 a 45

En primer lugar se busca se busca en la caché la dirección 15. Como la caché está inicialmente vacía, se produce un fallo y se almacena el bloque $j=0$ de la M_p , asociado a las direcciones 0 a 15, en el bloque $i=0$ de la M_c .

A continuación se busca en la caché la dirección 16 como la caché está inicialmente vacía se produce un fallo y se almacena el bloque $j=0$ de la M_p asociado a las direcciones 16 a 31, en el bloque $i=1$ de la M_c .

Cuando se vaya a buscar en la caché las direcciones 17 a 31, se producirá aciertos ya que todas ellas pertenecen al bloque $j=1$ de M_p que ya está cargado en la caché, en concreto en su bloque $i=1$ de la M_c .

Luego se observa que puesto que la memoria caché está inicialmente vacía cuando se accede a la primera dirección contenida en un bloque j de M_p que no esté en la caché se producirá un fallo, lo que produce que dicho bloque sea copiado en un bloque i de M_c . En consecuencia al acceder a las restantes 15 direcciones contenidas en el bloque j como ya está copiado en M_c se producirán 15 aciertos.

En la Tabla 1 se resumen los fallos o aciertos que se producen en la caché al ejecutarse por primera vez el bucle.

Direcciones	Bloque j de M_p	Resultado
20	2	1 fallo se copia el bloque $j=2$ en el bloque $i=0$ de M_c
21-23	2	3 aciertos
24	3	1 fallo se copia el bloque $j=3$ en el bloque $i=1$ de M_c
25-31	3	7 aciertos
32	4	1 fallo se copia el bloque $j=4$ en el bloque $i=2$ de M_c
33-39	4	7 aciertos
40	5	1 fallo se copia el bloque $j=5$ en el bloque $i=3$ de M_c
41-45	5	5 aciertos

Tabla 1

Luego en esta primera parte de la ejecución programa se producen:

Número de accesos $= (45-20)+1 = 26$ accesos.

Número de fallos = 4 fallos.

Número de aciertos = $3+2\cdot 7+5=22$ aciertos

En la segunda parte del programa se ejecuta 4 veces un bucle ubicado en las direcciones 28-45, al tratarse de direcciones que ya se encuentran en la caché únicamente se producirán aciertos. El número de accesos en esta segunda parte del programa es $4\cdot[(45-28)+1]=72$ accesos.

Luego el número total de accesos que se producen al ejecutar este programa sería:

$$N_T = 26 + 72 = 98 \text{ accesos}$$

Y el número total de aciertos que se producen al ejecutar este programa sería:

$$N_A = 22 + 72 = 94 \text{ aciertos}$$

Luego la tasa de acierto del programa es:

$$h = \frac{N_A}{N_T} = \frac{94}{98} = 0.959 \Rightarrow 95.9 \%$$

c) El tiempo de acceso medio t_{am} al sistema $M_c - M_p$ viene dada por la expresión:

$$t_{am} = h \cdot t_{ga} + (1-h) \cdot t_{gf}$$

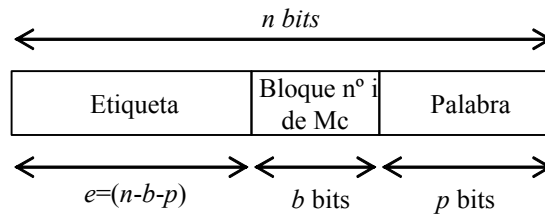
donde t_{ga} es el tiempo de gestión de un acierto y t_{gf} es el tiempo de gestión de un fallo. En este caso, según el enunciado, se tiene $t_{ga}=40$ ns y $t_{gf}=1 \mu s+40=1040$ ns. Luego sustituyendo valores se obtiene:

$$t_{am} = 0.959 \cdot 40 + 0.041 \cdot 1040 = 38.36 + 42.64 = 81 \text{ ns}$$

SOLUCION PROBLEMA 2.69**DATOS**

- $C_{Mp} = 256 \text{ Kpalabras} = 2^{18} \text{ palabras}$
- $C_{Mc} = 32 \text{ Kpalabras} = 2^{15} \text{ palabras}$
- $K = 16 \text{ palabras/bloque} = 2^4 \text{ palabras/bloque}$
- Función de correspondencia directa.
- $L = 2 \text{ bytes} = 16 \text{ bits}$
-

En primer lugar hay que determinar el formato de una dirección de memoria desde el punto de vista de una memoria caché con función de correspondencia directa es:



Del dato de la capacidad de la memoria principal se obtiene el tamaño n de una dirección de memoria:

$$C_{Mp} = 2^{18} \leq 2^n \rightarrow n = 18 \text{ bits}$$

Del dato del número de palabras que tiene un bloque se obtiene el tamaño p del campo *palabra*.

$$K = 2^4 \leq 2^p \rightarrow p = 4 \text{ bits}$$

Para obtener el tamaño b del campo *Bloque n° i de M_c* hay que calcular el número de bloques que pueden almacenarse en la memoria caché

$$C = \frac{C_{Mc}}{K} = \frac{2^{15}}{2^4} = 2^{11}$$

Como

$$C \leq 2^b$$

entonces

$$b = 11 \text{ bits}$$

Conocidos n , p y b , es inmediato determinar e

$$e = n - (b + p) = 18 - (11 + 4) = 3 \text{ bits}$$

En segundo lugar hay que recordar que en la función de correspondencia directa un bloque j de M_p tiene asignado de antemano el bloque i de M_c donde se va alojar su copia. Por lo tanto, dada una dirección de memoria DIR se producirá el reemplazamiento del contenido de un bloque de la caché, si DIR hace referencia a un bloque i de M_c que contiene una copia de un bloque j de M_p distinto del bloque j' al que hace referencia DIR.

Según el enunciado existe almacenado en un bloque i de la memoria caché una copia del bloque j de memoria principal que contiene la palabra cuya dirección asociada es $DIR=8560_{10}$. Para determinar el valor de i y de j se debe pasar DIR de decimal a binario:

$$\begin{aligned} 8560_{10} &= 1024 * 8 + 368 = 2^{13} + 256 + 112 = 2^{13} + 2^8 + 64 + 48 = \\ &= 2^{13} + 2^8 + 2^6 + 32 + 16 = 2^{13} + 2^8 + 2^6 + 2^5 + 2^4 = \\ &= 00001000101110000_2 \end{aligned}$$

Separando los bits según el tamaño de cada campo se obtiene:

$$DIR_0 = 000_2 \ 0100010111_2 \ 0000_2$$

Luego el campo *etiqueta* es 000_2 el campo *bloque de M_c* es $n^\circ i = 0100010111_2$ y el campo *palabra* es 0000_2 . Asimismo la unión del campo *etiqueta* y del campo *bloque i de M_c* permiten conocer el bloque j de M_p al que hace referencia una dirección de memoria en este caso $j=000 \ 010010111_2$.

Dada otra dirección de memoria se producirá un reemplazo en la caché, si y solo si, hace referencia al bloque $i = 0100010111_2$ y a un bloque $j \neq 000 \ 010010111_2$.

Opción A:

$$\begin{aligned} 8575_{10} &= 8560 + 15 = 8560 + 8 + 4 + 2 + 1 = \\ &= 2^{13} + 2^8 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = \\ &= 00001000101111111_2 = 000_2 \ 0100010111_2 \ 1111_2 \end{aligned}$$

Hace referencia al *bloque $i = 0100010111_2$* de M_c y al bloque $j=000 \ 010010111_2$ de M_p . Luego al hacer referencia a un bloque j cuya copia ya está almacenada en la caché no se produce reemplazamiento.

Opción B:

$$\begin{aligned} 8688_{10} &= 8560 + 128 \\ &= [2^{13} + 2^8 + 2^6 + 2^5 + 2^4] + 2^7 = \\ &= 00001000111110000_2 = 000_2 \ 0100011111_2 \ 1111_2 \end{aligned}$$

Hace referencia al *bloque $i = 0100011111_2$* de M_c no al $i = 0100010111_2$ luego la dirección 8688 nunca producirá un reemplazamiento del contenido del bloque $i = 0100010111_2$.

Opción C:

$$\begin{aligned} 41334_{10} &= 32768 + 8566 = 2^{15} + 8560 + 6 = 2^{15} + 2^{13} + 2^8 + 2^6 + 2^5 + 2^4 + 2^2 + 2 \\ &= 00101000111110110_2 = 001_2 \ 0100011111_2 \ 0110_2 \end{aligned}$$

Hace referencia al *bloque $i = 0100010111_2$* de M_c y al bloque $j=001 \ 010010111_2$ de M_p . Luego al hacer referencia a un bloque j cuya copia no está almacenada en la caché y al tener que ser almacenado ésta en el bloque $i=0100010111_2$ que contiene actualmente una copia del bloque $j=000 \ 010010111_2$ de M_p se hace necesario un reemplazamiento.

Luego la respuesta correcta es la C.

SOLUCION PROBLEMA 2.70**DATOS**

- Disco magnético con 64 pistas $\{0,1,\dots,63\}$
- Cola de peticiones: 1, 3, 63, 47, 2, 46

Afirmación I:

La planificación SCAN consiste en ir recorriendo todas las pistas en una dirección y satisfaciendo todas las peticiones que se encuentra en el camino, hasta que alcanza la última pista. En este punto se invierte el sentido del recorrido y la búsqueda prosigue de la misma forma

La cabeza lectora se halla en la pista $p_0=0$ y se desplaza hacia las pistas cuyo número de pista va aumentando. Luego atiende las peticiones de la cola en el siguiente:

[1, 2, 3, 46, 47, 63]

Para calcular la longitud media de búsqueda (LMB)

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N}$$

es recomendable construirse la siguiente tabla

Próxima pista a la que se accede	(p_1) 1	(p_2) 2	(p_3) 3	(p_4) 46	(p_5) 47	(p_6) 63
Número de pistas que se atraviesan	$ p_1 - p_0 $ 1	$ p_2 - p_1 $ 1	$ p_3 - p_2 $ 1	$ p_4 - p_3 $ 43	$ p_4 - 0 + p_5 - 0 $ 1	$ p_6 - p_5 $ 16

Por lo tanto:

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N} = \frac{1 + 1 + 1 + 43 + 1 + 16}{6} = 10.5$$

En conclusión la afirmación es **VERDADERA**.

Afirmación II:

La planificación FCFS consiste en ir desplazando la cabeza lectora atendiendo las peticiones en el orden en que van llegando. Luego atiende las peticiones de la cola en el mismo orden en que se encuentran en la cola:

[1, 3, 63, 47, 2, 46]

Para calcular la longitud media de búsqueda (LMB)

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N}$$

es recomendable construirse la siguiente tabla

Próxima pista a la que se accede	(p ₁) 1	(p ₂) 3	(p ₃) 63	(p ₄) 47	(p ₅) 2	(p ₆) 46
Número de pistas que se atraviesan	p ₁ -p ₀ 1	p ₂ -p ₁ 2	p ₃ -p ₂ 60	p ₄ -p ₃ 16	p ₄ -0 + p ₅ -0 45	p ₆ -p ₅ 44

Por lo tanto:

$$LMB = \frac{\sum_{k=1}^N |p_k - p_{k-1}|}{N} = \frac{1 + 2 + 60 + 16 + 45 + 44}{6} = 28$$

Luego la planificación FCFS empeora en más del 50 % la LMB que se obtiene con la planificación SCAN luego la afirmación es **FALSA**.

SOLUCION PROBLEMA 2.71

DATOS

- Disco formateado con entrelazado cuádruple.
- A=40 superficies
- N_C= 1024 cilindros=> N_P= 1024 pistas/superficie
- S=128 sectores/pista
- C_S 32 Kbytes/sector
- f=7200 rpm=120 (rev/s)

Afirmación I:

La Capacidad de la unidad de disco es:

$$\begin{aligned} C &= A \cdot N_P \cdot S \cdot C_S = 40 (\text{superficies}) \cdot 1024 (\text{pistas/superficie}) \cdot 128 (\text{sectores/pista}) \cdot 32 (\text{Kbytes/sector}) \\ &= 40 (\text{superficies}) \cdot 210 (\text{pistas/superficie}) \cdot 27 (\text{sectores/pista}) \cdot 215 (\text{bytes/sector}) \\ &= 40 \cdot 232 \text{ bytes} \\ &= 40 \cdot 4 \text{ GB} = 160 \text{ GB} \end{aligned}$$

Luego la afirmación I es **VERDADERA**.

Afirmación II:

El tiempo t_p que se tarda en leer todos los sectores de una pista es la suma de tres contribuciones:

$$t_p = t_b + t_r + t_t$$

donde:

- t_b es el *tiempo medio de búsqueda de la pista*. En este caso puesto que la cabeza lectora está posicionada ya en la pista adecuada $t_b=0$ s.
- t_r es el tiempo de latencia rotacional Es decir el tiempo medio que tarda el sector en estar debajo de la cabeza de lectura/escritura. Viene dado por

$$t_r = \frac{1}{2 \cdot f} = \frac{1}{2 \cdot 120} = 4.17 \text{ ms}$$

- t_t es el *tiempo de transferencia de todos los sectores de una pista*. Al existir entrelazado doble para leer la pista se comienza leyendo el sector 0 y después, para el resto de sectores, primero la cabeza pasa por encima de cuatro sectores que ignora y lee el quinto. Por tanto, la cabeza lectora pasa por un total de

$$N_s = 1 + (f_e + 1) \cdot (S - 1) = 1 + (4 + 1) \cdot (128 - 1) = 636 \text{ sectores}$$

Por lo tanto

$$t_t = \frac{N_s}{S \cdot f} = \frac{636}{128 \cdot 120} = 0.04140 \text{ s} = 41.40 \text{ ms}$$

Luego:

$$t_p = t_b + t_r + t_t = 0 + 4.17 + 41.40 = 45.57 \text{ ms}$$

La velocidad de transferencia v_T será el cociente entre los P bytes que hay en una pista y el tiempo t_p que se tarda en transmitir dichos bytes.

$$v_T = \frac{P}{t_p}$$

Con los datos que se dan en el enunciado es posible calcular P:

$$P = C_s \cdot S = 32 \text{ (Kbytes / sector)} \cdot 128 \text{ (sector / pista)} = 2^{12} \text{ (Kbytes / pista)} = 4 \text{ (Mbytes / pista)}$$

Luego

$$v_t = \frac{4 \text{ (Mbytes / pista)}}{45.57 \cdot 10^{-3} \text{ (s / pista)}} = 87.7 \text{ (Mbytes / s)}$$

En conclusión la afirmación II es **FALSA**.

SOLUCION PROBLEMA 2.72

DATOS

- Dispositivo de E/S con dirección base 400_{16}
- Las líneas A_0 y A_1 del bus de direcciones se utilizan para seleccionar registros internos del dispositivo.
- La línea CS de selección del circuito está conectada a un circuito que implementa la lógica de selección y cuyas entradas son A_2 , A_5 , A_6 y A_7

La dirección base del dispositivo de E/S es 400_{16} o equivalentemente en binario 0100 0000 0000. Luego el bus de direcciones tiene 12 líneas.

$$A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$$

El circuito de selección del dispositivo de E/S considerado tiene como entradas a CS, A_2 , A_5 , A_6 y A_7 . De acuerdo con la dirección base para seleccionar el dispositivo A_2 , A_5 , A_6 y A_7 deben valer 0.

Por otra lado puesto que sólo se emplean dos líneas (A_1A_0) para seleccionar los registros internos del dispositivo de E/S, eso indica que cómo máximo dicho dispositivo contiene $2^2=4$ registros internos.

La dirección propuesta como referencia 403_{16} en binario es 0101 0000 0011, luego hace referencia al registro (A_1A_0)= 11_2 del dispositivo de E/S.

El formato de una dirección que acceda al registro $(A_1A_0)=11_2$ del dispositivo de E/S, debe cumplir las siguientes condiciones:

- 1) Referenciar al registro $(A_1A_0)=11_2$ del dispositivo
- 2) No referenciar a una posición inferior a la dirección base del dispositivo, es decir, $(A_{11}A_{10})$ no pueden valer 00_2 .
- 3) A_2, A_5, A_6 y A_7 deben valer 0

En la siguiente tabla se compara la dirección de referencia 403_{16} con las de las soluciones A ($41B_{16}$), B (483_{16}) y C (407_{16}). Se han sombreado las columnas que deben ser comparadas

Dirección	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
403_{16}	0	1	0	0	0	0	0	0	0	0	1	1
$41B_{16}$	0	1	0	0	0	0	0	1	1	0	1	1
483_{16}	0	1	0	0	1	0	0	0	0	0	1	1
407_{16}	0	1	0	0	0	0	0	0	0	1	1	1

Se observa que de las tres direcciones propuestas la única cuyos bits a comparar coinciden con la dirección de referencia 403_{16} es la dirección $41B_{16}$. Nótese que la dirección 483_{16} queda descartada por que $A_7=1$ luego no cumple la condición 3. Por su parte la dirección 407_{16} queda descartada porque $A_2=1$ luego tampoco cumple la condición 3. Luego la solución correcta es la A.

SOLUCION PROBLEMA 2.73

DATOS

- $N_T=2^{12}$ accesos.
- $N_F=2^{10}$ fallos.
- $t_{ac}=5$ ns
- $t_{ap}=100$ ns

Afirmación I:

La tasa de aciertos h es:

$$h = \frac{N_A}{N_T} = \frac{N_T - N_F}{N_T} = \frac{2^{12} - 2^{10}}{2^{12}} = \frac{2^{10} \cdot (2^2 - 1)}{2^{12}} = \frac{3}{4} = 0.75 \equiv 75\%$$

Luego la afirmación es **FALSA**.

Afirmación II:

El tiempo de acceso medio t_{am} al conjunto memoria principal - memoria caché se define como:

$$t_{am} = h \cdot t_{ga} + (1-h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que:

$$t_{ga} = t_{ac} = 5nseg$$

Por otra parte t_{gf} es el tiempo que tarda la CPU en gestionar un *fallo* en la M_c . Puesto que no se dice lo contrario se supone que:

$$t_{ga} = t_{ap} = 100nseg$$

Sustituyendo valores en la fórmula (1) se obtiene que:

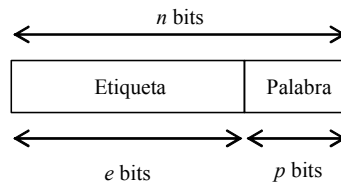
$$t_{am} = 0.75 \cdot 5 + 0.25 \cdot 100 = 3.75 + 25 = \mathbf{28.75 \text{ nseg}}$$

Luego la afirmación es **FALSA**.

SOLUCION PROBLEMA 2.74**DATOS**

- $C_{Mp}=64$ Kpalabras
- $L=16$ bits/pal
- t_{ap} tiempo de acceso a memoria principal.
- Función de correspondencia totalmente asociativa.
- $C_{Mc}=4$ Kpal
- $K= 512$ palabras/bloque.
- M_c inicialmente vacía.
- Algoritmo de reemplazamiento del tipo LRU.
- CPU ejecuta un bucle 5 veces.
- Cada bucle consta de 4101 palabras. $Dir=0,\dots,4100$.
- M_c es ocho veces más rápida que la M_p . $\Rightarrow t_{ac}=t_{ap}/8$.

A) El formato de una dirección de memoria desde el punto de vista de una memoria caché con función de correspondencia totalmente asociativa es el siguiente:



Del dato de la capacidad de la memoria principal se puede obtener el número n de bits que se necesitan para codificar todas las direcciones posibles.

$$C_{Mp} = 64Kpalabras = 2^{16} palabras \leq 2^n \Rightarrow n = 16bits$$

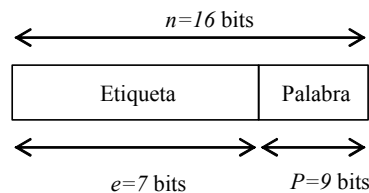
Del dato del tamaño de un bloque de palabras se puede obtener el número p de bits que se necesitan para codificar las posiciones de una palabra en el interior de un bloque.

$$K = 512 palabras = 2^9 palabras \leq 2^p \Rightarrow p = 9bits$$

El tamaño del campo etiqueta sería:

$$e = n - p = 16 - 9 = 7bits$$

Luego se tendría el siguiente formato para las direcciones:



Para responder a la segunda parte de este apartado, en primer lugar hay que calcular el número total de accesos a M_p que supone la ejecución cinco veces de un bucle de 4101 palabras. Se supone que cada palabra requiere de un acceso a M_p :

$$N_T = 5 \cdot 4101 = 20505 \text{ accesos}$$

Puesto que cada acceso a memoria requiere un tiempo t_{ap} , entonces el tiempo total empleado en realizar estos N_T accesos sería:

$$T_T = 20505 \cdot N_T \text{ accesos}$$

B) Antes de contestar a este apartado conviene calcular el número de bloques que se pueden contener en la memoria caché:

$$C = \frac{C_{Mc}}{K} = \frac{2^{12}}{2^9} = 2^3 = 8 \text{ bloques}$$

Como en la tabla de evolución del contenido de la Mc se pide poner el valor de la etiqueta que al finalizar la ejecución de cada uno de los cinco ciclos, en la Tabla 1 se han calculado dichas etiquetas.

La primera ejecución del ciclo se muestra en la Tabla 2:

- **Primera ejecución del ciclo** (ver Tabla 2). Los bloques se van llenando de uno en uno, hasta llenar la memoria caché. Cuando se accede a la dirección 4096 se vuelve a producir un fallo pero la memoria caché ya está llena. El enunciado propone como técnica de reemplazamiento sustituir el bloque que se haya utilizado menos recientemente, que es el 0.
- **Segunda ejecución del ciclo** (ver Tabla 3). Siguiendo el mismo razonamiento de antes, la Tabla 3 muestra qué va ocurriendo en la caché según se va ejecutando por segunda vez el ciclo. En esta Tabla se puede observar que siempre se producen fallos. Al finalizar el ciclo anterior hubo que sustituir el bloque 0 con etiqueta 0 que es el primero en ser referenciado al comenzar la ejecución de este segundo ciclo. Para resolver este fallo sustituye el contenido del bloque 1 de la caché, que será el siguiente en ser referenciado. Y así sucesivamente con todos.
- **Resto de ciclos.** La situación es parecida a la segunda ejecución del ciclo, produciéndose continuos fallos.

En la Tabla 4 se muestra el valor de la etiqueta existente en cada bloque de la caché al finalizar cada uno de los 5 ciclos.

Rango direcciones (Decimal)	Rango direcciones (Binario)	Rango direcciones (Hexadecimal)	Etiqueta en caché (Binario)	Etiqueta en caché (Hexadecimal)
0 - 511	0000 0000 0000 0000 0000 0001 1111 1111	0000 - 01FF	000 0000	00
512 - 1023	0000 0010 0000 0000 0000 0011 1111 1111	0200 - 03FF	000 0001	01
1024 - 1535	0000 0100 0000 0000 0000 0101 1111 1111	0400 - 05FF	000 0010	02
1536 - 2047	0000 0110 0000 0000 0000 0111 1111 1111	0600 - 07FF	000 0011	03
2048 - 2559	0000 1000 0000 0000 0000 1001 1111 1111	0800 - 09FF	000 0100	04
2560 - 3071	0000 1010 0000 0000 0000 1011 1111 1111	0A00 - 0BFF	000 0101	05
3072 - 3583	0000 1100 0000 0000 0000 1101 1111 1111	0C00 - 0DFF	000 0110	06
3584 - 4095	0000 1110 0000 0000 0000 1111 1111 1111	0E00 - 0FFF	000 0111	07
4096 - 4607	0001 0000 0000 0000 0001 0001 1111 1111	1000 - 11FF	000 1000	08

0

Tabla 1: Rango de direcciones y etiqueta correspondiente en la caché.

Acceso a dirección	Etiqueta en caché (Hexadecimal)	Comentarios
0	00	Fallo - Etiqueta 00 en el bloque 0
512	01	Fallo - Etiqueta 01 en el bloque 1
1024	02	Fallo - Etiqueta 02 en el bloque 2
1536	03	Fallo - Etiqueta 03 en el bloque 3
2028	04	Fallo - Etiqueta 04 en el bloque 4
2560	05	Fallo - Etiqueta 05 en el bloque 5
3072	06	Fallo - Etiqueta 06 en el bloque 6
3584	07	Fallo - Etiqueta 07 en el bloque 7
4096	08	Fallo - Reemplazar - Etiqueta 08 en el bloque 0

Tabla 2: Primera ejecución del bucle.

Acceso a dirección	Etiqueta en caché (Hexadecimal)	Comentarios
0	00	Fallo - Reemplazar - Etiqueta 00 en el bloque 1
512	01	Fallo - Reemplazar - Etiqueta 01 en el bloque 2
1024	02	Fallo - Reemplazar - Etiqueta 02 en el bloque 3
1536	03	Fallo - Reemplazar - Etiqueta 03 en el bloque 4
2028	04	Fallo - Reemplazar - Etiqueta 04 en el bloque 5
2560	05	Fallo - Reemplazar - Etiqueta 05 en el bloque 6
3072	06	Fallo - Reemplazar - Etiqueta 06 en el bloque 7
3584	07	Fallo - Reemplazar - Etiqueta 07 en el bloque 0
4096	08	Fallo - Reemplazar - Etiqueta 08 en el bloque 1

Tabla 3: Segunda ejecución del bucle.

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Bloque 0	8	7	6	5	4
Bloque 1	1	8	7	6	5
Bloque 2	2	1	8	7	6
Bloque 3	3	2	1	8	7
Bloque 4	4	3	2	1	8
Bloque 5	5	4	3	2	1
Bloque 6	6	5	4	3	2
Bloque 7	7	6	5	4	3

Tabla 4: Etiquetas de cada bloque de la caché al finalizar cada ciclo.

C) Para calcular el número de fallos hay que utilizar los resultados del apartado anterior y, especialmente, la Tabla 4.

- *Primer ciclo*: 9 fallos, uno por cada bloque del ciclo.
- *Resto de ciclos*: 9 fallos, uno por cada bloque del ciclo.

Por tanto, el número total de fallos es:

$$N_F = 45 \text{ fallos}$$

D) El tiempo total que se emplea en realizar los N_T accesos a memoria vendría dado por la siguiente expresión:

$$T = T_A + T_F$$

donde T_A es el tiempo total en gestionar los aciertos en la memoria caché y T_F es el tiempo total en gestionar los fallos.

T_A se puede calcular mediante la siguientes expresión:

$$T_A = N_A \cdot t_{ac}$$

donde N_A es el número total de aciertos y t_{ac} es el tiempo medio de acceso a M_c . N_A se obtiene como la diferencia entre el número total de accesos N_T y el número total de fallos, luego:

$$N_A = N_T - N_F = 20505 - 45 = 20460$$

Luego

$$T_A = 20460 \cdot t_{ac} = 20460 \cdot (t_{ap} / 8) = 2557.5 \cdot t_{ap}$$

Por otra parte del enunciado:

$$T_F = N_F \cdot t_{gf}$$

Donde t_{gf} es el tiempo medio de gestión de un fallo, que de acuerdo al enunciado consiste en mover el bloque completo de memoria principal a memoria caché y después leer el dato de la caché. Luego

$$t_{gf} = K \cdot t_{ap} + t_{ac} = K \cdot t_{ap} + (t_{ap} / 8) = (512 + 1 / 8) \cdot t_{ap} = 512.125 \cdot t_{ap}$$

Por lo tanto:

$$T_F = 45 \cdot 512.125 \cdot t_{ap} = 23045.625 \cdot t_{ap}$$

Con lo que finalmente se tendría:

$$T = T_A + T_F = 2557.5 \cdot t_{ap} + 23045.625 \cdot t_{ap} = 25603.125 \cdot t_{ap}$$

SOLUCION PROBLEMA 2.75**DATOS**

- $C_{Mp}=128$ Kpalabras
- $C_{Mc}=2$ Kpalabras
- M_c con función de correspondencia directa.
- $K=512$ palabras/bloque
- Se ejecuta 1000 veces un bucle que accede a 2200 posiciones de memoria consecutivas.

Se va analizar en cuantos bloques de M_p cabrían 2200 palabras consecutivas. El resultado de la división entera $C_{Mp}/K=2200/512$ es 4 y su resto es 152. Luego 2200 se puede expresar equivalentemente de la siguiente forma:

$$2200=512*4+152=2048+152$$

Se pueden presentar tres posibles casos en función de como se distribuyan las 2200 palabras consecutivas:

- *Caso 1:* 2048 palabras se almacenan en 4 bloques y 152 palabras al principio de otro bloque. En total 5 bloques de M_p .
- *Caso 2:* 152 palabras al final de un bloque y 2048 palabras se almacenan en 4 bloques. En total 5 bloques de M_p .
- *Caso 3:* C palabras se almacenan al final de un bloque, 2048 palabras se almacenan en 4 bloques y $152-C$ se almacenan al principio de otro bloque. En total 6 bloques de M_p .

A continuación se va a determinar el número de bloques que se pueden almacenar en la memoria caché:

$$C = \frac{C_{Mc}}{K} = \frac{2 \text{ Kpalabras}}{512 \text{ palabras / bloque}} = \frac{2^{11}}{2^9} = 2^2 = 4 \text{ bloques}$$

A estos bloques los vamos a denotar como $i=0,1,2$ y 3.

Conviene recordar que en una memoria caché con función de correspondencia directa cada bloque j de M_p tenía asignado a priori el bloque i de M_c donde se debe almacenar. A través de la fórmula $i=j \bmod C$

Supóngase que las 2200 palabras se encuentran almacenadas en 5 bloques de M_p , por ejemplo $j=16, 17, 18, 19$ y 20. Estos bloques tendrían asignados (de acuerdo a la función de correspondencia) los siguientes bloques de M_c : $i=0, 1, 2, 3$ y 0. Se observa que los bloques $j=16$ y $j=20$ tienen asignado el mismo bloque de M_c , el $i=0$. De esta forma, cuando se ejecuta por novena vez el bucle de 2200 palabras se producen 2 fallos:

- El primer fallo se produce cuando se direcciona una palabra contenida en el bloque $j=16$. Dicho bloque tiene asignado el bloque $i=0$ pero allí debido a la octava ejecución del bucle se encuentra almacenada una copia del bloque $j=20$. En la gestión del fallo el bloque $j=16$ se copia en $i=0$.
- El segundo fallo se produce cuando se direcciona una palabra contenida en el bloque $j=20$. Dicho bloque tiene asignado el bloque $i=0$ pero allí debido al fallo anterior se encuentra almacenada una copia del bloque $j=16$. En la gestión del fallo el bloque $j=20$ se copia en $i=0$.

Supóngase ahora que las 2200 palabras se encuentran almacenadas en 6 bloques de M_p , por ejemplo $j=16, 17, 18, 19, 20$ y 21. Estos bloques tendrían asignados (de acuerdo a la función de correspondencia) los siguientes bloques de M_c : $i=0, 1, 2, 3, 0$ y 1. Se observa que los bloques $j=16$ y $j=20$ tienen asignado el mismo bloque de M_c , el $i=0$. Asimismo, los bloques $j=17$ y $j=21$ tienen asignado el bloque $i=1$. De esta forma, cuando se ejecuta por novena vez el bucle de 2200 palabras se producen 4 fallos:

- El primer fallo se produce cuando se direcciona una palabra contenida en el bloque $j=16$. Dicho bloque tiene asignado el bloque $i=0$ pero allí debido a la octava ejecución del bucle se encuentra almacenada una copia del bloque $j=20$. En la gestión del fallo el bloque $j=16$ se copia en $i=0$.

- El segundo fallo se produce cuando se direcciona una palabra contenida en el bloque $j=17$. Dicho bloque tiene asignado el bloque $i=1$ pero allí debido a la octava ejecución del bucle se encuentra almacenada una copia del bloque $j=21$. En la gestión del fallo el bloque $j=17$ se copia en $i=1$.
- El tercer fallo se produce cuando se direcciona una palabra contenida en el bloque $j=20$. Dicho bloque tiene asignado el bloque $i=0$ pero allí debido al fallo primero se encuentra almacenada una copia del bloque $j=16$. En la gestión del fallo el bloque $j=20$ se copia en $i=0$.
- El cuarto fallo se produce cuando se direcciona una palabra contenida en el bloque $j=21$. Dicho bloque tiene asignado el bloque $i=1$ pero allí debido al fallo segundo se encuentra almacenada una copia del bloque $j=17$. En la gestión del fallo el bloque $j=21$ se copia en $i=1$.

En resumen en la novena ejecución del bucle, se pueden producir como mínimo 2 fallos (si las 2200 palabras se almacenan en 5 bloques de M_p) y como máximo 4 fallos (si las 2200 palabras se almacenan en 6 bloques de M_p). Luego las dos afirmaciones son **VERDADERAS**.

SOLUCION PROBLEMA 2.76

DATOS

- $L=16$ bits/palabra
- Bus de direcciones de 24 líneas $[A_{23}, \dots, A_0]$
- Bloques base de memoria RAM de capacidad $C_0=2$ Mpalabras x 4 bits/palabra
- Capacidad total memoria RAM $C_{Mp}=9$ Mpalabras x 16 bits/palabra
- Direcciones ROM: 0 a 1 Mpalabras -1
- Las direcciones RAM son consecutivas a las direcciones

En primer lugar se va a determinar el número de módulos base de capacidad C_0 necesarios para construir una memoria RAM de capacidad C_T , para ello se deben dividir ambas cantidades y aproximar por el primer número entero que sea mayor que el número decimal que se obtenga:

$$\frac{C_T}{C_0} = \frac{9 \text{ Mpalabras}}{2 \text{ Mpalabras}} \times \frac{16 \text{ bits/palabra}}{4 \text{ bits/palabra}} \leq (5 \times 4) = 20$$

Luego se necesitan 20 módulos RAM de capacidad C_0 .

La disposición de los 20 módulos de capacidad C_0 sería la siguiente: 5 filas de 4 módulos cada una. En cada fila se direccionan 2 Mpalabras x 16 bits/palabra. Nótese que en la quinta fila de módulos RAM solamente estará ocupado o disponible el primer mega de palabras con objeto de obtener la capacidad de memoria RAM total deseada: 9 Mpalabras = 2 Mpalabras (1 Fila) + 2 Mpalabras (2 Fila) + 2 Mpalabras (3 Fila) + 2 Mpalabras (4 Fila) + 1 Mpalabras (5 Fila).

A_{23}	A_{22}	A_{21}	A_{20}	$A_{19} \dots A_0$	Direcciones contenidas en:
0	0	0	0	$X \dots X$	Módulo ROM
0	0	0	1	$X \dots X$	Fila 1 de módulos RAM
0	0	1	0	$X \dots X$	Fila 1 de módulos RAM
0	0	1	1	$X \dots X$	Fila 2 de módulos RAM
0	1	0	0	$X \dots X$	Fila 2 de módulos RAM
0	1	0	1	$X \dots X$	Fila 3 de módulos RAM
0	1	1	0	$X \dots X$	Fila 3 de módulos RAM
0	1	1	1	$X \dots X$	Fila 4 de módulos RAM
1	0	0	0	$X \dots X$	Fila 4 de módulos RAM
1	0	0	1	$X \dots X$	Fila 5 de módulos RAM
1	0	1	0	$X \dots X$	Direcciones no asignadas
1	0	1	1	$X \dots X$	Direcciones no asignadas
1	1	0	0	$X \dots X$	Direcciones no asignadas
1	1	0	1	$X \dots X$	Direcciones no asignadas
1	1	1	0	$X \dots X$	Direcciones no asignadas
1	1	1	1	$X \dots X$	Direcciones no asignadas

Tabla 1: Mapa de direcciones de memoria

En Tabla 1 se muestra el mapa de direcciones de memoria del conjunto: ROM + RAM (5 filas x 4 módulos /fila) que se obtiene considerando los datos del enunciado y la información obtenida anteriormente. Para reducir el número de filas de la Tabla se ha denotado con X...X al conjunto posible de direcciones comprendidas entre 0000 0000 0000 0000 0000 - 1111 1111 1111 1111 1111. Asimismo se han sombreado el conjunto de direcciones válidas que permiten acceder a la RAM diseñada

Con la información que se ha obtenido hasta ahora ya se está en condiciones de poder responder a las afirmaciones propuestas.

Afirmación I:

Hay que demostrar que la función lógica propuesta no permite detectar direcciones RAM válidas (es decir que permiten acceder a la RAM diseñada), ya que devuelve el mismo valor tanto para una dirección de RAM válida como para una no válida..

Por ejemplo, para una dirección de ROM ($A_{23}=A_{22}=A_{21}=A_{20}=0$) la función devuelve el siguiente valor:

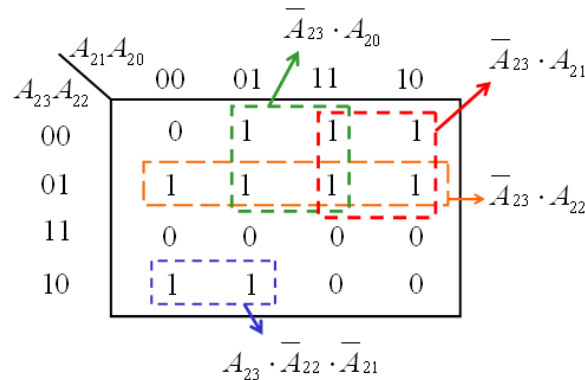
$$\bar{A}_{23} \times (\bar{A}_{22} + \bar{A}_{21} + \bar{A}_{20}) + A_{23} \times \bar{A}_{22} \times \bar{A}_{21} = 1x(1+1+1) + 0x1x1 = 1$$

Mientras que para una dirección de RAM válida como ($A_{23}=A_{22}=A_{21}=0, A_{20}=1$) la función también devuelve el valor

$$\bar{A}_{23} \times (\bar{A}_{22} + \bar{A}_{21} + \bar{A}_{20}) + A_{23} \times \bar{A}_{22} \times \bar{A}_{21} = 1x(1+1+0) + 0x1x1 = 1$$

En consecuencia, la afirmación I es **FALSA**.

Aunque no es necesario para responder a la afirmación propuesta se va a obtener una función lógica F que devuelve 0 si se trata de una dirección RAM no válida y devuelve 1 si es válida. Nótese que para especificar la validez de una dirección simplemente es necesario fijarse en los bits A_{23} , A_{22} , A_{21} y A_{20} . Tomando como referencia la Tabla 1, es posible construir el siguiente mapa de Karnaugh:



Con lo que la función F para la detección de direcciones RAM válidas tomaría la forma:

$$F = \bar{A}_{23} \times (\bar{A}_{22} + \bar{A}_{21} + \bar{A}_{20}) + A_{23} \times \bar{A}_{22} \times \bar{A}_{21}$$

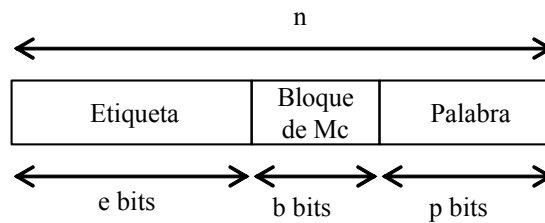
Afirmación II:

De acuerdo al cálculo realizado al principio de este problema el número de módulos de capacidad C_0 necesarios es 20, no 19 como indica la afirmación. Luego, es **FALSA**.

SOLUCION PROBLEMA 2.77**DATOS**

- $C_{Mp} = 4096 \text{ Mpalabras} = (2^{32}) \text{ palabras}$
- $C_{Mc} = 64K \text{ palabras} = (2^{16}) \text{ palabras}$
- $K = 2048 \text{ palabras/bloque} = (2^{11}) \text{ palabras/bloque}$
- Función de correspondencia directa.
- En un determinado instante la dirección B3A972E0, de la memoria principal está en la memoria caché

Para una memoria caché M_c con función de correspondencia directa el formato de una dirección de memoria desde el punto de vista de la M_c tiene la siguiente estructura:



Del dato de la capacidad de la memoria principal se puede obtener el número de bits n de una dirección de memoria:

$$C_{Mp} \leq 2^n \rightarrow 2^{32} \leq 2^n \rightarrow n = 32 \text{ bits}$$

Del dato del tamaño de un bloque se puede obtener el número de bits p que se necesitan para direccionar una palabra dentro de un bloque

$$K \leq 2^p \rightarrow 2^{11} \leq 2^p \rightarrow p = 11 \text{ bits}$$

Para obtener el número de bits b que se necesitan para direccionar un bloque de la memoria caché, primero hay que calcular el número de bloques C de la M_c .

$$C = \frac{C_{Mc}}{K} = \frac{2^{16}}{2^{11}} = 2^5 = 32 \text{ bloques.}$$

Se tiene por tanto que

$$C \leq 2^b \rightarrow 2^5 \leq 2^b \rightarrow b = 5 \text{ bits}$$

Finalmente el número de bits del campo etiqueta son:

$$e = n - (b + p) = 32 - (11 + 5) = 16 \text{ bits}$$

Afirmación I:

Para comprobar la validez de esta afirmación hay que determinar el número de bloque i de memoria caché al que hace referencia la dirección B3A972E0. Únicamente se necesita pasar a binario los 16 bits menos significativos, que son los asociados a los campos bloque de M_c y palabra:

$$72E0_{16} = 0111\ 0010\ 1110\ 0000$$

Agrupando por campos se tiene

01110	010 1110 0000
-------	---------------

Luego el bloque i de M_C al que hace referencia esta dirección es: $i=01110_2=14_{10}$. Luego la afirmación es **VERDADERA**.

Afirmación II:

Para comprobar la validez de esta afirmación hay que determinar el número de bloque i de memoria caché al que hace referencia la dirección 629E72E9. Únicamente se necesita pasar a binario los 16 bits menos significativos, que son los asociados a los campos bloque de M_C y palabra:

$$72E9_{16} = 0111\ 0010\ 1110\ 1001$$

Agrupando por campos se tiene

01110	010 1110 1110
-------	---------------

Luego el bloque i de M_C al que hace referencia esta dirección es: $i=01110_2=14_{10}$.

Luego esta dirección hace referencia también a una palabra que se ubicaría en el bloque $i=14$ de la caché. En este caso, para saber si la dirección está en la caché, es necesario comparar las etiquetas de ambas direcciones. Como la etiqueta de esta dirección es 629E₁₆ distinta de la etiqueta B3A9₁₆ de la dirección que si que estaba en la M_C , entonces eso significa que 629E72E9₁₆ no está en la M_C . Luego la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 2.78

DATOS

- Sistema jerárquico de memoria.
- $C_{Mc} = 4K = 2^{12}$ palabras, $K = 128 = 2^7$ pal/ bloque, $t_{ac} = 15$ nseg
- $C_{Mp} = 128$ Kpalabras = 2^{17} palabras, $t_{ap} = 150$ nseg
- Tasa de acierto $h = 0.95$
- Cuando se produce un fallo en M_c :
 - Se mueve el bloque completo a la M_C .
 - Se lee el dato desde M_C .

El tiempo de acceso medio t_{am} se define como:

$$t_{am} = h \cdot t_{ga} + (1 - h) \cdot t_{gf} \quad (1)$$

Donde t_{ga} es el tiempo de gestión del acierto. Normalmente se cumple que:

$$t_{ga} = t_{ca} = 15 \text{ nseg}$$

De acuerdo con el enunciado el tiempo de gestión de un fallo será la suma de dos componentes, ya que cuando se produce un fallo:

- 1) El sistema mueve el bloque completo a la caché, es decir hay que realizar K accesos a la memoria principal, lo que supone un tiempo de $K \cdot t_{ap}$.
- 2) Se lee el dato desde la caché, lo que supone un tiempo de t_{ac} .

Luego:

$$t_{gf} = K \cdot t_{ap} + t_{ac} = 128 \cdot 150 + 15 = 19215 \text{ nseg}$$

Sustituyendo valores en la formula (1) se obtiene que:

$$t_{am} = 0.95 \cdot 15 + 0.05 \cdot 19215 = 14.25 + 960.75 = 975 \text{ n seg}$$

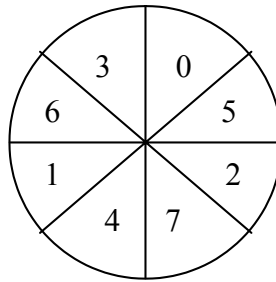
Luego la solución es $t_{am}=975 \text{ nseg}$.

SOLUCION PROBLEMA 2.79

DATOS

- Disco formateado con entrelazado cuádruple.
- $S=8$ sectores/pista
- Velocidad de rotación de $f=6000 \text{ rpm}=100 \text{ (rev/s)}$
- La cabeza lectora se encuentra en la pista correcta y sobre el punto de comienzo del sector 4.

En la figura se muestra la numeración de los sectores de un disco con 8 sectores por pista que posee entrelazado cuádruple:



El tiempo t_p que se tarda en leer todos los sectores de una pista es la suma de tres contribuciones:

$$t_p = t_b + t_r + t_t$$

donde:

- t_b es el *tiempo medio de búsqueda de la pista*. En este caso puesto que la cabeza lectora está posicionada ya en la pista adecuada $t_b=0$ s.
- t_r es el *tiempo de latencia rotacional* que es el tiempo medio que tarda el sector en estar debajo de la cabeza de lectura/escritura. Se calcula, de acuerdo con lo estudiado en teoría, a partir de la siguiente expresión:

$$t_r = \frac{1}{2 \cdot f} = \frac{1}{2 \cdot 100} = 5 \text{ ms}$$

- t_t es el *tiempo de transferencia de todos los sectores de una pista*. Al existir entrelazado cuádruple ($f_e=4$), para leer la pista se comienza leyendo el sector 0 y después, para el resto de sectores, primero la cabeza pasa por encima de cuatro sectores que ignora y lee el quinto. Por tanto, la cabeza lectora pasa por un total de

$$N_s = 1 + (f_e + 1) \cdot (S - 1) = 1 + (4 + 1) \cdot (8 - 1) = 36 \text{ sectores}$$

Por lo tanto

$$t_t = \frac{N_s}{S \cdot f} = \frac{36}{8 \cdot 100} = 0.045 \text{ s} = 45 \text{ ms}$$

Luego:

$$t_p = t_b + t_r + t_t = 0 + 5 + 45 = 50 \text{ ms}$$

Otra forma de llegar a este mismo resultado es calcular el número de revoluciones R que da el disco en la lectura de todos los sectores de una pista. Puesto que existe entrelazado cuádruple y la cabeza de lectura/escritura se encuentra situada al comienzo del sector 4, en la primera rotación la cabeza lee el sector 0. En la segunda rotación lee los sectores 1 y 2. En la tercera rotación lee el sector 3. En la cuarta rotación lee los sectores 4 y 5. Finalmente en la quinta rotación lee los sectores 6 y 7. Luego $R=5$.

Nótese que puesto $f=100 \text{ rev/s}$, el tiempo que tarda el disco en dar una rotación es $1/100=0.01 \text{ s}$. Luego el tiempo que se emplearía en dar las 5 rotaciones necesarias para leer en su orden todos los sectores de una pista sería

$$t_p = R \cdot \frac{1}{f} = 5 \cdot \frac{1}{100} = 50 \text{ ms}$$

SOLUCION PROBLEMA 2.80

DATOS

- Memoria asociativa de n palabras y 8 bits/palabra
- Se desea saber el contenido del registro argumento (A) y del registro de máscara (K) que hace que se activen los bits del registro de marca (M) cuando las celdas de memoria a los que están asociados contengan un número impar.

Un número binario es impar si su bit menos significativo es 1 luego el contenido del registro argumento, que se utiliza para comparar con el contenido de las palabras de la memoria asociativa, tendrá el siguiente formato:

$$A = \text{XXXXXXXX}1$$

donde X puede tomar el valor 0 o 1. Es decir obligatoriamente el bit menos significativo debe ser 1 el valor del resto de los bits resulta indiferente para deducir si el número es impar.

Por otra parte, el registro de máscara K, el cual se utiliza para indicar los bits que se desean comparar, debe tener el siguiente contenido:

$$K = 00000001$$

es decir, únicamente se va a comparar el bit menos significativo.

De acuerdo con este análisis, la única solución propuesta que tiene un registro argumento y un registro de máscara con un contenido adecuado es la **B**.

SOLUCION PROBLEMA 2.81

DATOS

- Disco formateado con entrelazado doble.
- $S=8$ sectores/pista.
- $C_S=16 \text{ Kbytes/sector}$
- Velocidad de rotación de $f=4500 \text{ rpm}=75 \text{ (rev/s)}$

La velocidad de transferencia máxima se consigue cuando se lee un único sector del disco, se define como el cociente entre la capacidad de un sector C_S y el tiempo t_S que se tarda en leer un sector.

$$v_{T \max} = \frac{C_S}{t_S}$$

La velocidad de rotación del disco es $f=75$ rev/s luego el tiempo que tarda en dar un giro completo será la inversa, es decir, $1/75=0.0133$ s. Puesto que el disco tiene 8 sectores por pista, para leer un único sector le basta con dar $1/8$ de vuelta, luego el tiempo de lectura de un sector es:

$$t_s = \frac{1/f}{S} = \frac{1}{S \cdot f} = \frac{1}{8 \cdot 75} = \frac{1}{600} (s/\text{sector})$$

Por lo tanto

$$v_{T\max} = \frac{16(Kbytes/\text{sector})}{(1/600)(s/\text{sector})} = 9600(Kbytes/s)$$

SOLUCION PROBLEMA 2.82

DATOS

- Función de correspondencia asociativa por conjuntos
- Dirección con los siguientes tamaños de sus campos:
Etiqueta=24 bits, Conjunto=8 bits, Palabra=8 bits
- $r=4$ bloques/conjunto
- $L=2$ bytes/palabra

El campo CONJUNTO de una dirección de memoria tiene 8 bits luego el número Q de conjuntos de la memoria caché es:

$$Q = 2^8 \text{ conjuntos}$$

El número de bloques C que se pueden almacenar en la memoria caché se determina multiplicando el número de conjuntos Q por el número r de bloques que hay en un conjunto

$$C = Q \cdot r = 2^8 \cdot 4 = 2^{10} \text{ bloques}$$

El campo PALABRA de una dirección de memoria tiene 8 bits luego tamaño de un bloque de palabras es:

$$B = 2^8 (\text{palabras/bloque})$$

Ya se tiene por tanto, toda la información necesaria para calcular la capacidad de la memoria caché:

$$C_{Mc} = C \cdot B \cdot L = 2^{10} \cdot 2^8 \cdot 2 = 2^9 \cdot 2^{10} = 512 \text{ Kbytes}$$

SOLUCION PROBLEMA 2.83

DATOS

- Diseñar memoria con:
 - Palabras de memoria de 16 bits
 - 512K de espacio direccionable (64K ROM en direcciones bajas y el resto RAM)
- Tipos de módulos disponibles: RAM de 64 K x 8 bits, RAM 128K x 16 bits, ROM 16 K x 16 bits, decodificadores.
- Módulo de memoria y decodificadores disponen de una entrada de activación SC, que se activa cuando SC=1.

A) La Figura 1 muestra el número de líneas de datos y de dirección que pueden entrar a cada uno de los chips de memoria del problema. Dado que las primeras 64K palabras deberán ser de ROM, parece claro que ese espacio de memoria deberá construirse utilizando cuatro módulos de ROM de 16K x 16 bits. Las 448K palabras restantes deberán construirse utilizando módulos de RAM. Aunque existen diferentes posibilidades, la más razonable parece colocar dos módulos de RAM de 64K x 8 bits a continuación de los módulos de ROM. Entre ambos tipos de memoria se han completado 128K palabras de 16 bits. Una vez hecho esto, se colocan tres módulos de RAM de 128K palabras x 16 bits para completar el resto del espacio hasta las 512K palabras.

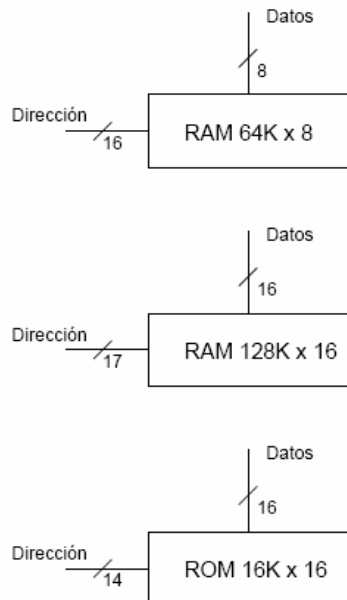


Figura 1: Módulos de memoria del problema

La Figura 2 muestra la distribución de los módulos de memoria, la conexión de las líneas del bus de datos y de direcciones, así como la lógica de selección de dichos módulos. Se utiliza un primer decodificador de 2 a 4 para seleccionar uno de entre cuatro módulos de 128K palabras, haciendo uso de las líneas A_{18} y A_{17} . Las tres salidas de más peso se conectan a las entradas de selección SC de los tres módulos de RAM de 128K palabras. La salida de menos peso se usa para habilitar un segundo decodificador, de 1 a 2 (también podrían usarse puertas lógicas), que permite seleccionar entre las 64K palabras superiores (las de RAM) o las inferiores (de ROM), haciendo uso de la línea A_{16} . Si se activa la salida de menos peso del decodificador, dicha salida habilita a su vez a un tercer decodificador, de 2 a 4, que permite seleccionar uno de los cuatro módulos ROM de 16K palabras haciendo uso de las señales de control A_{15} y A_{14} .

Las líneas de dirección de menos peso entran directamente a cada uno de los chips correspondientes. Puede observarse que todas las líneas de dirección intervienen de una u otra manera en la selección de datos dentro de cada chip.

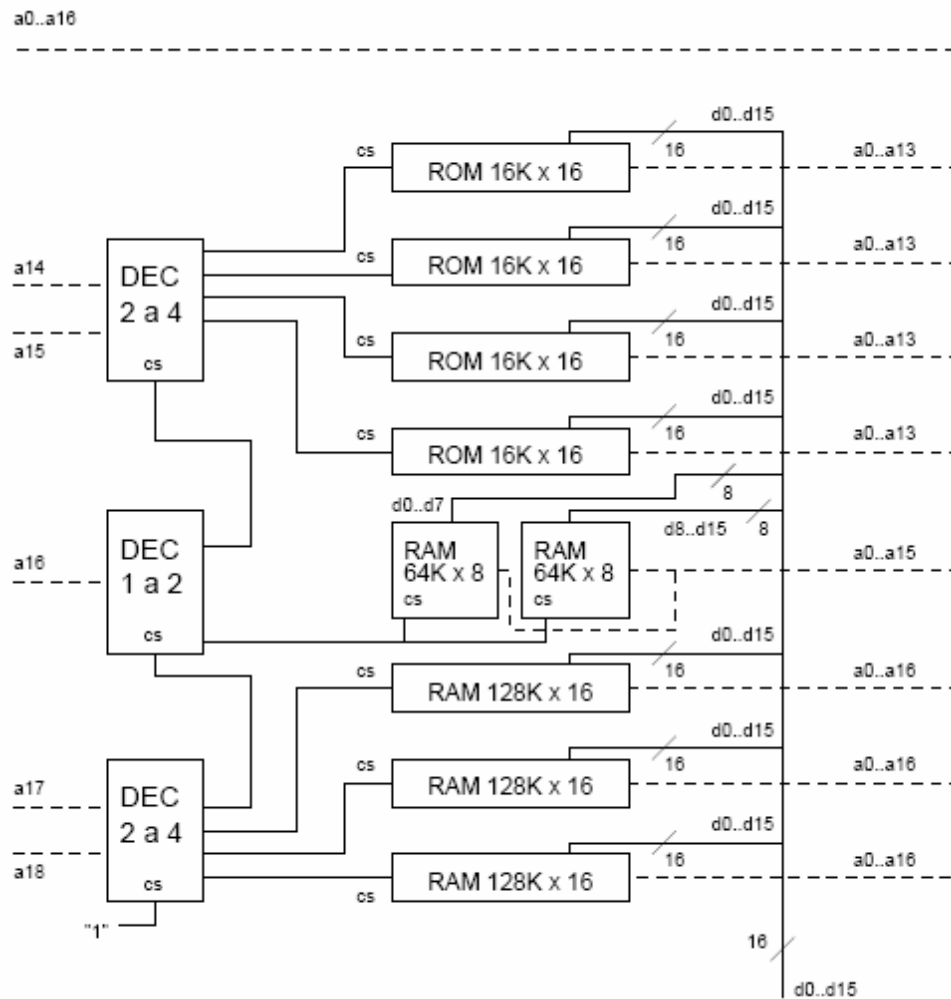


Figura 2: Esquema de la unidad de memoria y su lógica de selección

B) La palabra con dirección hexadecimal 186A0 es, en binario, la dirección siguiente (representándola en 19 bits, ya que hay 19 líneas de direcciones):

001 1000 0110 1010 0000

Puede verse que esta dirección habilita la salida 0 del decodificador inferior, con lo que se activa el decodificador intermedio. A su vez, la línea A_{16} está a 1, por lo que se habilita la pareja de módulos RAM de 64K x 8. En consecuencia, el dato está en estos módulos. Puede verse también que la dirección hexadecimal 186A0 es en decimal la dirección 100000, lo que se corresponde con el rango de direcciones almacenado en dicha pareja de módulos, que va desde 64K palabras a (128K palabras-1).