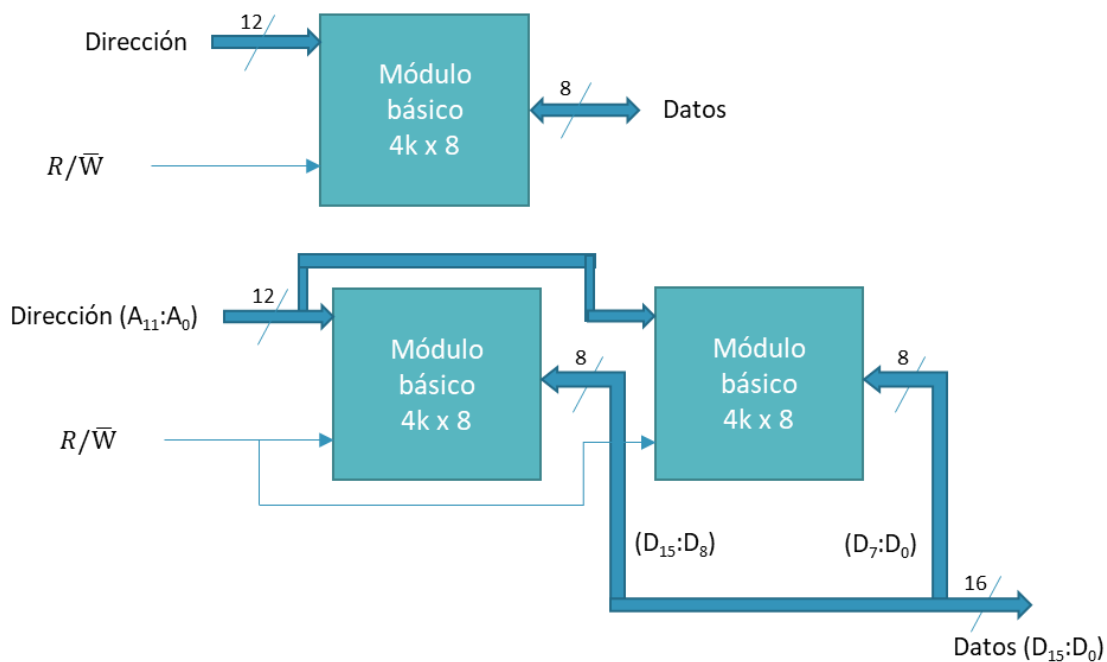


MÓDULO III. UNIDAD DE MEMORIA

MEMORIA A PARTIR DE MÓDULOS

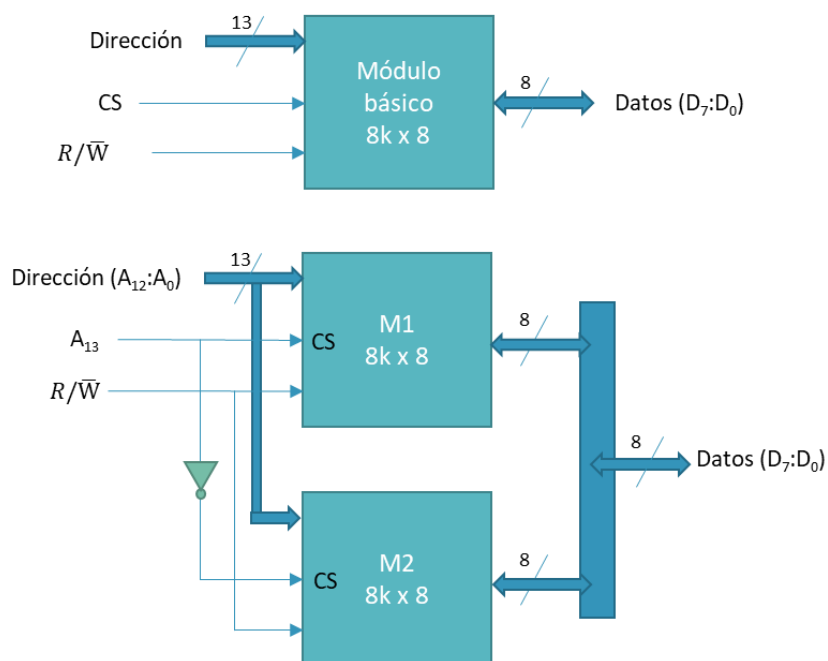
PROBLEMA 1. INCREMENTO DE LA LONGITUD DE LA PALABRA

Diseñar una memoria de 4K palabras X 16 bits a partir de módulos de 4 K palabras de 8 bits. (Ver página 264 de [1])



PROBLEMA 2. INCREMENTO DEL NÚMERO DE PALABRAS DE LA MEMORIA

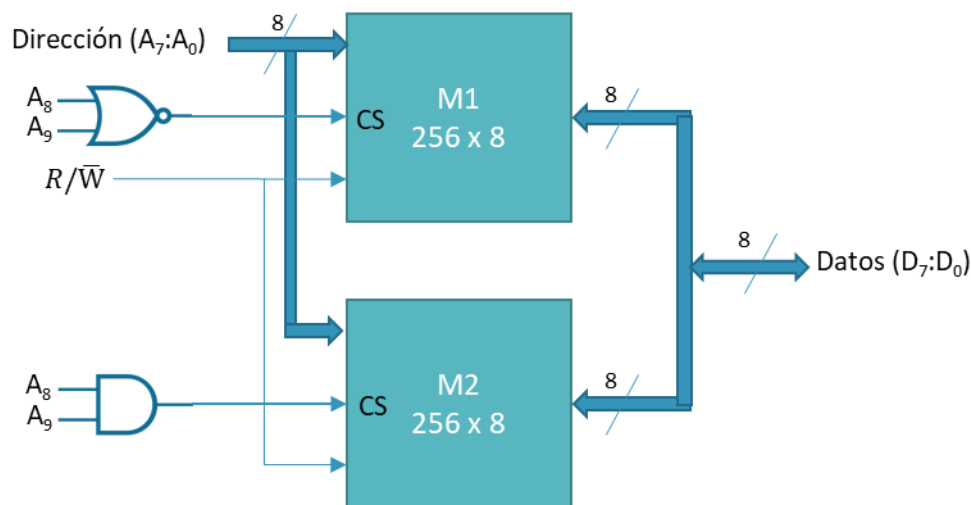
Diseñar una memoria de 16K palabras de 8 bits a partir de módulos de 8K palabras de 8 bits. (Ver página 265 de [1])



PROBLEMA 3. UBICAR MÓDULOS DE MEMORIA EN POSICIONES CONCRETAS DEL MAPA DE MEMORIA.

Se dispone de dos módulos de memoria M1 y M2 de 256 X 8. Se dispone de un procesador de bus de direcciones de 10 bits y bus de datos de 8 bits. M1 se quiere situar al comienzo del mapa de memoria y M2 al final (Ver páginas 267-8 de [1]).

Dirección ($A_9:A_0$)	Mapa de memoria
00 0000 0000 ... 00 1111 1111	M1
01 0000 0000 ... 01 1111 1111	Libre
10 0000 0000 ... 10 1111 1111	Libre
11 0000 0000 ... 11 1111 1111	M2


PROBLEMA 4. DISEÑAR UNA UNIDAD DE MEMORIA DE 8K X 8 CON CHIPS DE 1K X 1 (2115)

Se dispone de módulos de 1K palabras de 1 bit cada una. Se quieren conectar a una CPU que dispone de un bus de direcciones de 13 bits, un bus de datos de 8 bits y una línea de R/\overline{W} .

No disponible

PROBLEMA 5. DISEÑAR UN MAPA DE MEMORIA

Se dispone de 4 módulos SRAM de 2^{12} palabras de 4 bits. Estos módulos tienen una entrada CS para habilitarlos, y dos entradas separadas que se activan a baja, que indican si se quiere realizar una escritura \overline{WE} , o una lectura \overline{OE} . Se quieren conectar a una CPU con bus de direcciones de 16 bits, bus de datos de 8 bits, señal de R/\overline{W} , y una señal para controlar si se accede a memoria o a E/S (IO/\overline{M}).

No disponible

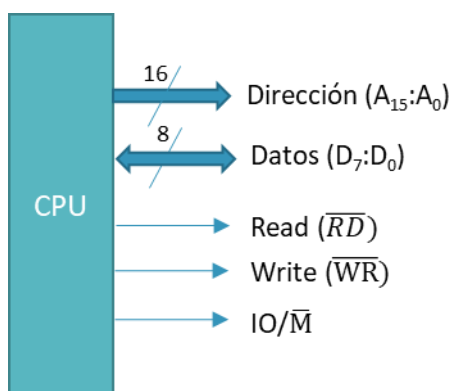
PROBLEMA 6. DISEÑAR UN MAPA DE MEMORIA

Un procesador dispone de un bus de direcciones de 16 bits, un bus de datos de 8 bits, una línea de IO/\overline{M} y líneas de lectura (\overline{RD}) y escritura (\overline{WR}). Se pretenden conectar al procesador cuatro ROMs de 2K×8 (con entradas \overline{OE} , output enable), seis RAMs de 2K×4 (con entradas \overline{CS} , chip select, y \overline{WE} , write enable) y cuatro controladores de periféricos con 8 puertos de 8 bits (con entradas \overline{RD} , read, \overline{WR} , write y \overline{CS}). Diseñar el esquema de conexión de estos elementos al procesador y construir una tabla (mapa de memoria) indicando el rango de direcciones que ocupan todos los dispositivos.

Solución:

En primer lugar, debemos ir analizando la información que se nos facilita.

Empezamos con la información del propio sistema. Llamaremos a los bits de nuestro bus de direcciones $A_{15} - A_0$; y a los del bus de datos $D_7 - D_0$, con lo que la CPU podría representarse como sigue:



A continuación, debemos ir completando el mapa de memoria a medida que vamos analizando cómo conectar cada tipo de módulo. La idea es ir presentando incrementalmente las conexiones para no saturar de líneas los dibujos y que quede claro cómo queremos conectarlos.

Para realizar un mapa de memoria debemos decidir qué configuración de bits de la dirección se emplean para cada tipo de módulo. Esto varía en función de si empleamos entrelazado o no. Si empleamos entrelazado, cada dirección de nuestro mapa de memoria tendrá típicamente la siguiente configuración:

Bits para tipo de módulo	Bits para dirección dentro del módulo	Bits para elegir módulo dentro de cada tipo
--------------------------	---------------------------------------	---

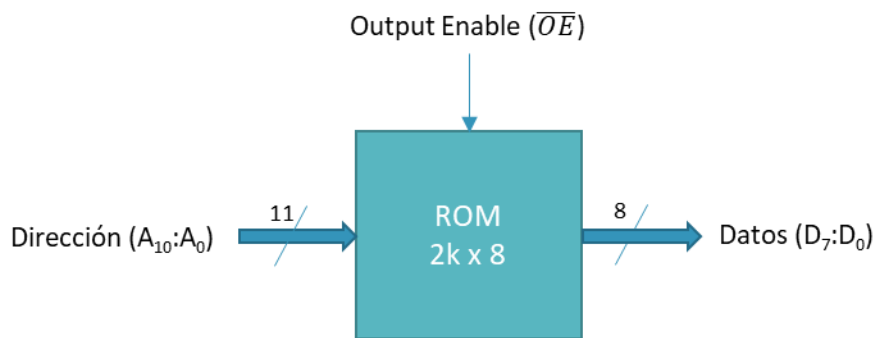
Si no empleamos entrelazado, la dirección tendrá típicamente la siguiente configuración:

Bits para tipo de módulo	Bits para elegir módulo dentro de cada tipo	Bits para dirección dentro del módulo
--------------------------	---	---------------------------------------

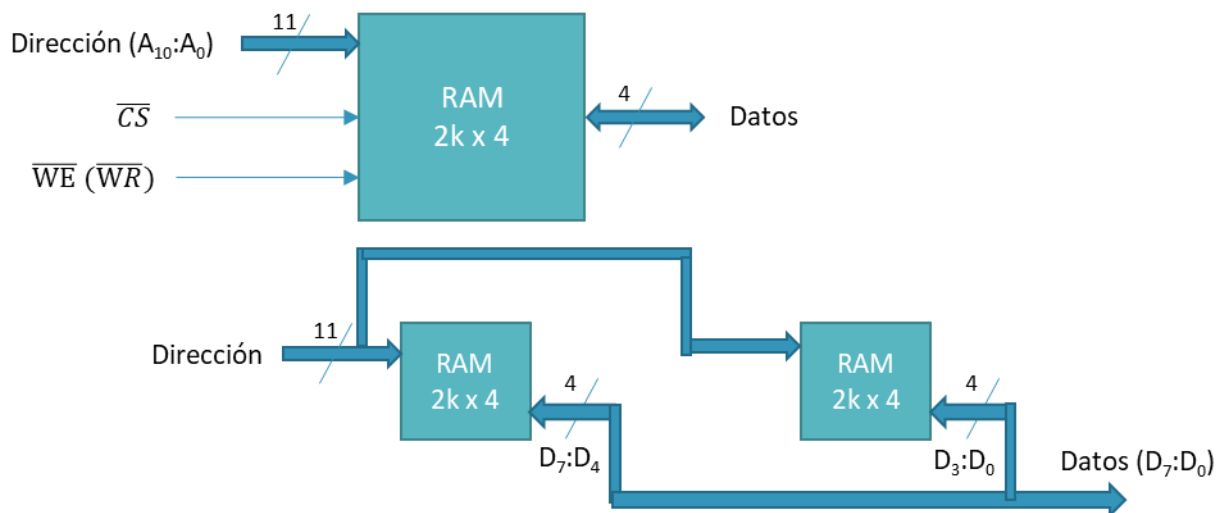
De momento, en este ejercicio, no haremos uso del entrelazado.

Pasemos a analizar cada tipo de módulo.

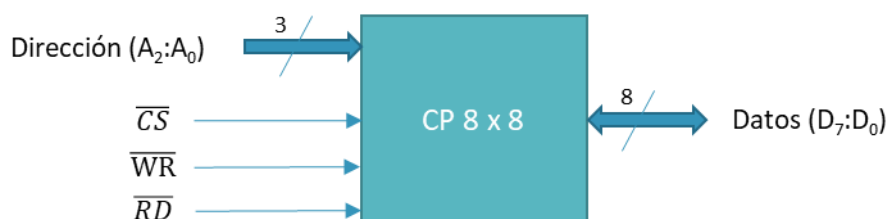
Los módulos ROM (a los que llamaremos ROM1, ROM2, ROM3 y ROM4) tienen un tamaño de palabra igual al de la CPU, por lo que la salida de datos del módulo podría conectarse directamente al bus de datos. Como los módulos ROM son de 2K palabras, necesitamos 11 bits para direccionar estas palabras dentro de cada módulo, que podemos tomar de la parte menos significativa del bus de direcciones ($A_{10} - A_0$).



Los módulos RAM los llamaremos RAM1 – RAM6. Al igual que con los módulos ROM tomaremos de la parte menos significativa del bus de direcciones ($A_{10} - A_0$) como entrada de dirección. Como tenemos la mitad de tamaño de palabra de lo que necesitamos para el bus de datos, tendremos que usar estos módulos de dos en dos: el módulo impar contendrá los bits más significativos ($D_7 - D_4$), mientras el módulo par contendrá los bits menos significativos ($D_3 - D_0$). La línea de WE podríamos conectarla directamente al WR de la CPU porque, además, ambas líneas se activan a baja.



Para los módulos controladores de periféricos (CP1 – CP4) podemos tomar cualquier convenio que queramos para conectar las direcciones. Necesitamos 3 bits para direccionar los 8 puertos. Podríamos usar, como en los otros módulos, los bits menos significativos del bus de direcciones ($A_2 - A_0$). Dado que no tenemos más controladores de periféricos, y si no nos preocupa no dejar más espacio para otros dispositivos, podemos usar los 2 bits más significativos del bus de direcciones ($A_{15} - A_{14}$) para escoger el controlador de entre los 4 disponibles, ignorando el resto de líneas del bus. Los datos pueden ir conectados directamente al bus de datos, al igual que con los módulos ROM.

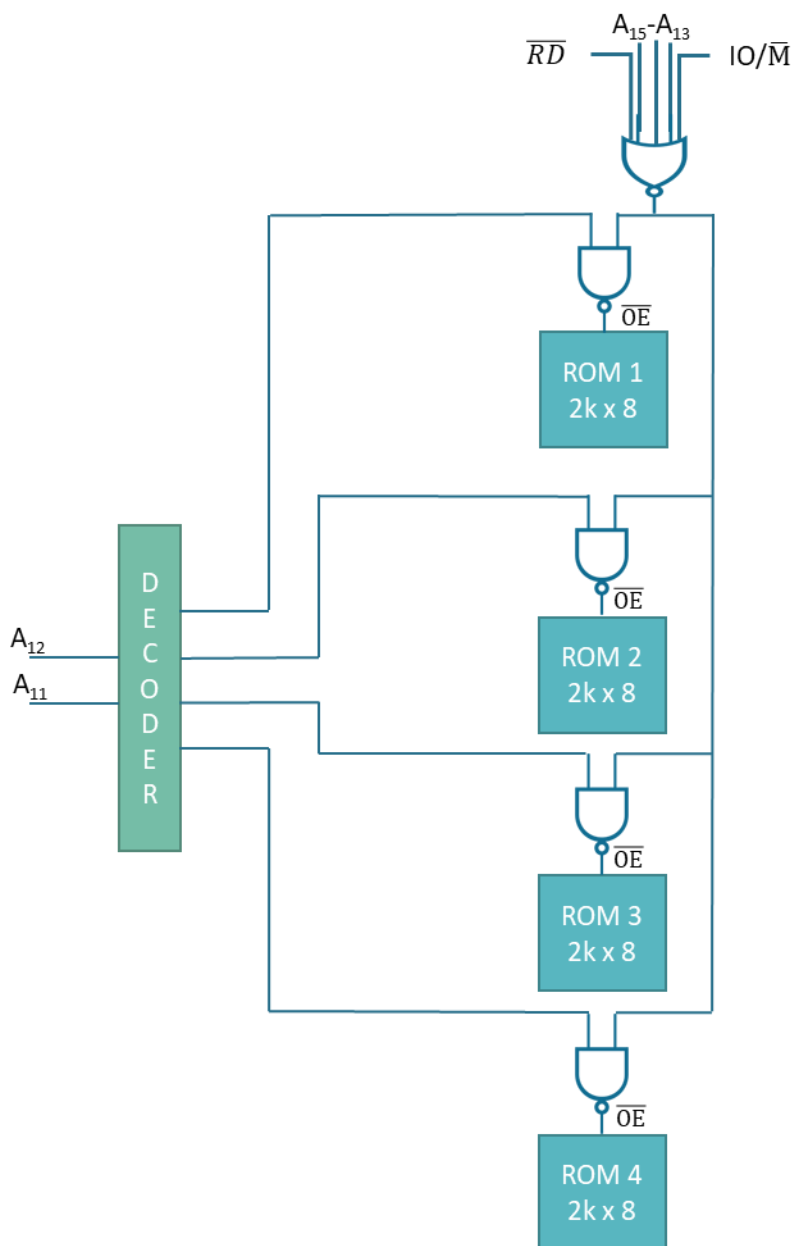


El mapa de memoria quedará de la siguiente manera (recuerden que deben incluir la línea IO/\overline{M}):

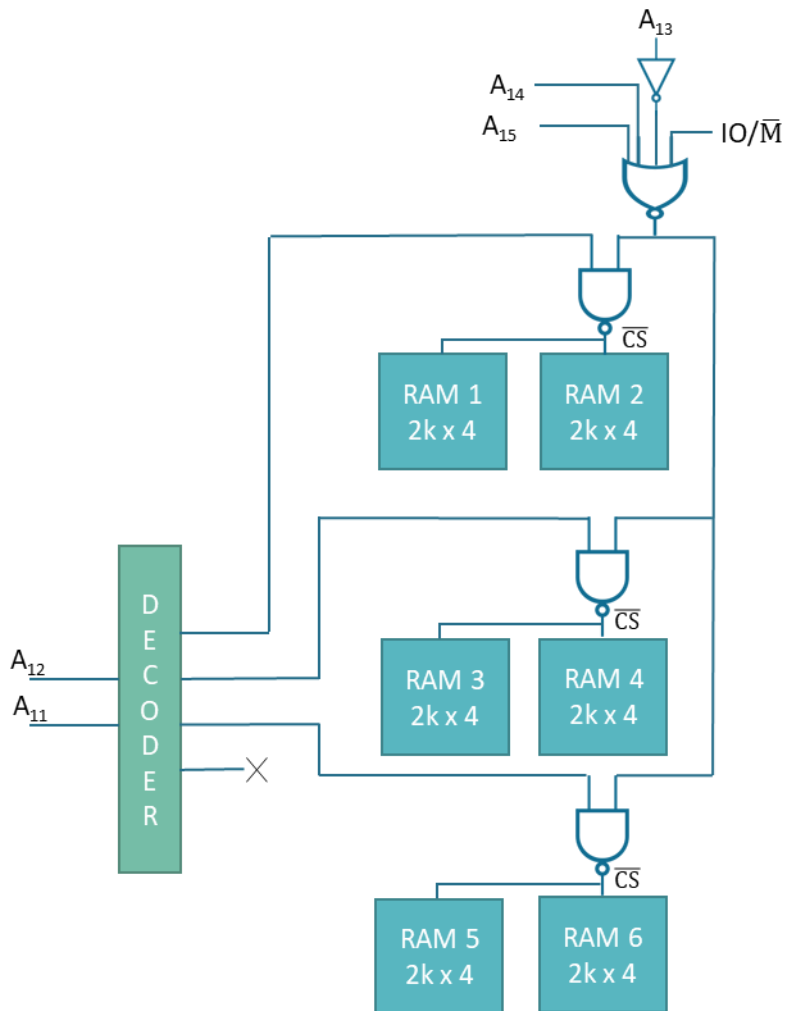
IO/\overline{M}	A15	A14	A13	A12	A11	A10	...	A2	A1	A0	Módulo
0	0	0	0	0	0	x	...	x	x	x	ROM1
0	0	0	0	0	1	x	...	x	x	x	ROM2
0	0	0	0	1	0	x	...	x	x	x	ROM3
0	0	0	0	1	1	x	...	x	x	x	ROM4
0	0	0	1	0	0	x	...	x	x	x	RAM1+RAM2
0	0	0	1	0	1	x	...	x	x	x	RAM3+RAM4
0	0	0	1	1	0	x	...	x	x	x	RAM5+RAM6
0	0	0	1	1	1	x	...	x	x	x	LIBRE
...
0	1	1	1	1	1	x	...	x	x	x	LIBRE
1	0	0	NA	NA	NA	NA	...	x	x	x	CP1
1	0	1	NA	NA	NA	NA	...	x	x	x	CP2
1	1	0	NA	NA	NA	NA	...	x	x	x	CP3
1	1	1	NA	NA	NA	NA	CP4

En la tabla están marcado en distintos tonos de azul los bits del A15 al A13, que me permitirían escoger el tipo de módulo (000 para ROM y 001 para RAM). El resto de configuraciones estarían libres. En el caso de las RAM, de hecho, sobra una configuración ($A_{12} - A_{11} = 11$), ya que sólo tenemos 3 parejas de módulos. Nos queda representar el conexionado final que nos permitirá cumplir con este mapa de memoria.

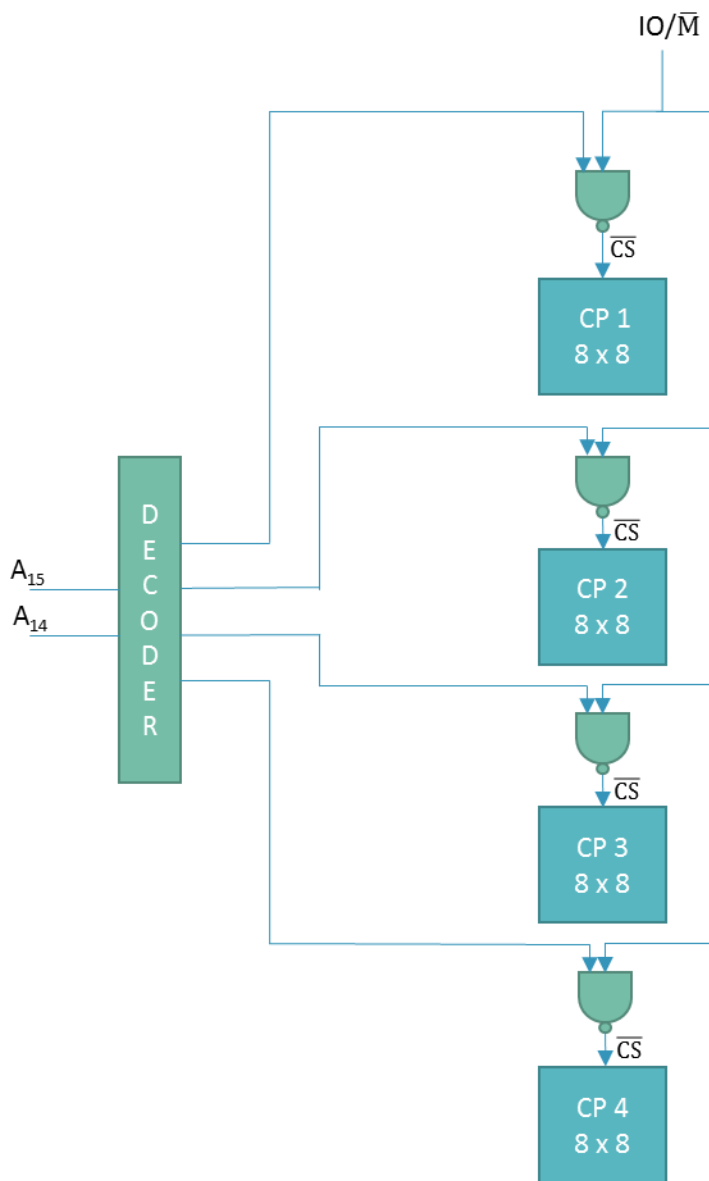
Los módulos ROM tienen la siguiente estructura de cableado (obviamos las líneas de datos y direcciones que ya hemos dibujado en los esquemas particulares de cada tipo de módulo):



El cambio fundamental en los módulos RAM es que tenemos que asegurarnos de activar solidariamente los dos módulos que funcionan como pareja para devolver una palabra de 8 bits. Como ahora se activan cuando $(A_{15} - A_{13} = 001)$, necesitamos invertir la entrada A_{13} en el esquema hardware para que funcione correctamente.



En los módulos controladores de periféricos, tal como los hemos configurado, la habilitación es mucho más sencilla, y sólo depende de un decodificador y la entrada IO/\overline{M} .



CACHE

PROBLEMA 1

En un sistema de cache el tamaño del bloque es de 4 palabras y el tamaño de la memoria cache es de 2 Kpalabras. Se pide:

a) Si la CPU posee un bus de direcciones de 20 bits, indicar el número de bits del campo Etiqueta (Tag) para el caso de que el cache sea de mapeado directo y para el caso de que sea completamente asociativo.

$$4 \text{ palabras/bloque} = 2^2$$

$$\text{Caché: } 2048 \text{ palabras} = 2^{11} \text{ palabras} \rightarrow 2^9 \text{ bloques}$$

$$\text{Memoria: } 2^{20} \text{ palabras} \rightarrow 2^{18} \text{ bloques}$$

$$\text{Mapeado directo: } 9 \times \text{ETIQUETA} + 9 \times \text{LÍNEA} + 2 \times \text{PALABRA}$$

$$20 - 9 - 2 = 9 \rightarrow \text{Etiqueta}$$

$$\text{Mapeado asociativo: } 18 \times \text{ETIQUETA} + 2 \times \text{PALABRA}$$

b) Si se emplea la misma memoria organizada como cache asociativo por conjuntos con cuatro bloques por conjunto y campos Etiqueta (Tag) de 12 bits conectada a otra CPU, ¿Cuál sería ahora la anchura del bus de direcciones de dicha CPU?

$$4 \text{ bloques/conjunto}$$

$$2^9 \text{ bloques} / (2^2 \text{ bloques/conjunto}) = 2^7 \text{ conjuntos}$$

$$\text{Etiqueta} = 12 \text{ bits} \rightarrow 12 + 7 + 2 = 21 \text{ bits de bus}$$

PROBLEMA 2

Un procesador con bus de direcciones de 32 bits dispone de un sistema de caché de 256 palabras y bloques de 16 palabras que emplea mapeado directo. El tiempo de acceso a la memoria caché (acierto) es de 20 ns; y el tiempo de penalización (fallo) es de 800 ns.

Se dispone de una traza de lectura representativa de la clase de programas que se ejecutan en esta CPU, que viene dada por los bloques de memoria principal accedidos siguientes (en hexadecimal)

0x00000000, 0x00000001, 0x00000000, 0x00000020, 0x00000000, 0x00000010, 0x00000081, 0x00000001, 0x00000101, 0x00000101, 0x00000000, 0x00000080

a) Determinar el tiempo total empleado en los accesos de la traza

$$16 \text{ palabras/bloque} = 2^4$$

$$\text{Memoria principal: } 2^{32} \text{ palabras} \rightarrow 2^{28} \text{ bloques}$$

$$\text{Caché: } 256 \text{ palabras} = 2^8 \rightarrow 2^4 \text{ líneas}$$

$$\text{Dirección: } 24 \times \text{ETIQUETA} + 4 \times \text{LÍNEA} + 4 \times \text{PALABRA}$$

Dirección (bloque)	Etiqueta	Línea	A/F	Bloque reemplazado	Tiempo
0x0000000	0	0	Fallo		800
0x0000001	0	1	Fallo		800
0x0000000	0	0	Acierto		20
0x0000020	2	0	Fallo	0x0000000	800
0x0000000	0	0	Fallo	0x0000020	800
0x0000010	1	0	Fallo	0x0000000	800
0x0000081	8	1	Fallo	0x0000001	800
0x0000001	0	1	Fallo	0x0000081	800
0x0000101	16	1	Fallo	0x0000101	800
0x0000101	16	1	Acierto		20
0x0000000	0	0	Fallo	0x0000010	800
0x0000080	8	0	Fallo	0x0000000	800
TOTAL					8400

Etiquetas en la caché:

Línea	Etiqueta
0	8
1	0
...	

- b) Determinar el tiempo total empleado si cambiamos a un sistema de caché asociativo por conjunto de 512 palabras, donde cada conjunto contiene 2 líneas y se emplea un algoritmo de reemplazamiento FIFO.

2 bloques/conjunto

Caché: 512 palabras = $2^9 \rightarrow 2^5$ bloques $\rightarrow 2^4$ conjuntos

Dirección: 24 x ETIQUETA + 4 x CONJUNTO + 4 x PALABRA

Dirección (bloque)	Etiqueta	Conjunto	Vía	A/F	Bloque reemplazado	Tiempo
0x0000000	0	0	0	Fallo		800
0x0000001	0	1	0	Fallo		800
0x0000000	0	0	0	Acierto		20
0x0000020	2	0	1	Fallo		800
0x0000000	0	0	0	Acierto		20
0x0000010	1	0	0	Fallo	0x0000000	800
0x0000081	8	1	1	Fallo		800
0x0000001	0	1	0	Acierto		20
0x0000101	16	1	0	Fallo	0x0000001	800
0x0000101	16	1	0	Acierto		20
0x0000000	0	0	1	Fallo	0x0000020	800
0x0000080	8	0	0	Fallo	0x0000001	800
TOTAL						6480

Etiquetas en la caché:

Conjuntos	Vía	
	0	1
0	8	0
1	16	8
...		

- c) Determinar el tiempo de acceso si usamos la misma estructura que en el apartado b, pero cambiando el algoritmo de reemplazamiento a LRU

Dirección (bloque)	Etiqueta	Conjunto	Vía	A/F	Bloque reemplazado	Tiempo
0x0000000	0	0	0	Fallo		800
0x0000001	0	1	0	Fallo		800
0x0000000	0	0	0	Acierto		20
0x0000020	2	0	1	Fallo		800
0x0000000	0	0	0	Acierto		20
0x0000010	1	0	1	Fallo	0x0000020	800
0x0000081	8	1	1	Fallo		800
0x0000001	0	1	0	Acierto		20
0x0000101	16	1	1	Fallo	0x0000081	800
0x0000101	16	1	1	Acierto		20
0x0000000	0	0	0	Acierto		20
0x0000080	8	0	1	Fallo	0x0000010	800
TOTAL						5700

Etiquetas en la caché:

Conjuntos	Vía	
	0	1
0	0	0
1	0	16
...		

- d) Determinar el tiempo de acceso si se emplea mapeado completamente asociativo

Dirección: $28 \times \text{ETIQUETA} + 4 \times \text{PALABRA}$

Dirección (bloque)	Etiqueta	Línea	A/F	Bloque reemplazado	Tiempo memoria
0x0000000	0	0	Fallo		800
0x0000001	1	1	Fallo		800
0x0000000	0	0	Acierto		20
0x0000020	32	2	Fallo		800
0x0000000	0	0	Acierto		20
0x0000010	16	3	Fallo		800
0x0000081	129	4	Fallo		800
0x0000001	1	1	Acierto		20
0x0000101	257	5	Fallo		800
0x0000101	257	5	Acierto		20
0x0000000	0	0	Acierto		20
0x0000080	128	6	Fallo		800
TOTAL					5700

- e) ¿Cuál sería el tiempo promedio de acceso a este sistema de memoria si la tasa de acierto es del 95%?

$$T_{\text{acceso}} = 20 \times 95\% + 800 \times 5\% = 59 \text{ ns}$$

PROBLEMA 3

Un microprocesador dispone de un cache de 32 Kbytes asociativo por conjuntos. Cada bloque tiene 16 bytes y un conjunto contiene 2 bloques. Sabiendo que el bus de direcciones es de 32 bits, se pide:

a) Indicar la estructura del cache, explicando el funcionamiento del mismo ante un acceso de lectura. Describir las ventajas del mapeado asociativo por conjuntos frente a otros tipos de mapeado.

Como no nos indican nada en el enunciado, supondremos que las palabras son de 1 byte.

16 palabras/bloque = 2^4

Memoria: 2^{32} palabras

2 bloques/conjunto

Caché: 2^{15} palabras = 2^{11} bloques = 2^{10} conjuntos

Dirección: 18 x ETIQUETA + 10 x CONJUNTO + 4 x PALABRA

b) Suponiendo que un programa realiza accesos a posiciones de memoria contenidas en los bloques que aparecen en la traza

0, 0, 1, 1, 128, 1024, 0, 2048, 0, 4096, 128, 1

determinar el contenido final de los dos primeros conjuntos del cache con los algoritmos de reemplazamiento FIFO y LRU. (Suponer que el cache inicialmente no contiene nada). Indicar la razón de fallos para cada algoritmo de reemplazamiento.

Con reemplazamiento FIFO:

Dirección (bloque)	Etiqueta	Conjunto	Vía	A/F	Bloque reemplazado
0	0	0	0	Fallo	
0	0	0	0	Acierto	
1	0	1	0	Fallo	
1	0	1	0	Acierto	
128	0	128	0	Fallo	
1024	1	0	1	Fallo	
0	0	0	0	Acierto	
2048	2	0	0	Fallo	0
0	0	0	1	Fallo	1024
4096	4	0	0	Fallo	2048
128	0	128	0	Acierto	
1	0	1	0	Acierto	

Aciertos = 5; Fallos = 7 → Razón fallos = 58.33 %

Etiquetas en la caché:

Conjuntos	Vía	
	0	1
0	4	0
1	0	
...		
128	0	

Con reemplazamiento LRU:

Dirección (bloque)	Etiqueta	Conjunto	Vía	Acción	Bloque reemplazado
0	0	0	0	Fallo	
0	0	0	0	Acierto	
1	0	1	0	Fallo	
1	0	1	0	Acierto	
128	0	128	0	Fallo	
1024	1	0	1	Fallo	
0	0	0	0	Acierto	
2048	2	0	1	Fallo	1024
0	0	0	0	Acierto	
4096	4	0	1	Fallo	2048
128	0	128	0	Acierto	
1	0	1	0	Acierto	

Aciertos = 6; Fallos = 6 → Razón fallos = 50 %

Etiquetas en la caché:

Conjuntos	Vía	
	0	1
0	0	4
1	0	
2		
...		
128	0	

c) ¿Cuáles son las razones de fallos si cambiamos el número de bloques por conjunto a cuatro manteniendo el tamaño del cache?

4 bloques/conjunto

Caché: 2^{15} palabras = 2^{11} bloques = 2^9 conjuntos

Dirección: 19 x ETIQUETA + 9 x CONJUNTO + 4 x PALABRA

Dirección (bloque)	Etiqueta	Conjunto	Vía	Acción	Bloque reemplazado
0	0	0	0	Fallo	
0	0	0	0	Acierto	
1	0	1	0	Fallo	
1	0	1	0	Acierto	
128	0	128	0	Fallo	
1024	2	0	1	Fallo	
0	0	0	0	Acierto	
2048	4	0	2	Fallo	
0	0	0	0	Acierto	
4096	8	0	3	Fallo	
128	0	128	0	Acierto	
1	0	1	0	Acierto	

Aciertos = 6; Fallos = 6 → Razón fallos = 50 %

Etiquetas en la caché:

Conjuntos	Vía			
	0	1	2	3
0	0	2	4	8
1	0			
2				
...				
128	0			

PROBLEMA 4

Un procesador tiene una caché de 2 KB y usa bloques de 16 B (palabras de 1 byte). La memoria total direccionable es de 64 KB. El tiempo de acceso a la caché es de 30 ns y el tiempo necesario para tratar un fallo es de 110 ns.

16 palabras/bloque = 2^4

Memoria: 2^{16} palabras

Caché: 2^{11} palabras = 2^7 bloques

1. Dibujar la estructura de la caché, e indicar cómo se descompone la dirección de memoria si:

a) El mapeado es directo

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	E	E	E	E	L	L	L	L	L	L	L	P	P	P	P

E: Etiqueta; L: Línea; P: Palabra

b) El mapeado es totalmente asociativo

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	E	E	E	E	E	E	E	E	E	E	E	P	P	P	P

E: Etiqueta; P: Palabra

c) El mapeado es asociativo por conjunto de 4 vías

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	E	E	E	E	E	E	C	C	C	C	C	P	P	P	P

E: Etiqueta; C: Conjunto; P: Palabra

2. Se quiere acceder a las palabras de memoria que ocupan las direcciones siguientes:

0x7001, 0x7002, 0x7010, 0x7010, 0x780A, 0x7002, 0x7811, 0x7011, 0x7811, 0x780A

Dirección	Bloque												Palabra			
0x7001	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
0x7002	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0
0x7010	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
0x7010	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
0x780A	0	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0
0x7002	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0
0x7811	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1
0x7011	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1
0x7811	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1
0x780A	0	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0

a) Hacer la traza del acceso a estas direcciones con cada esquema de caché asumiendo un algoritmo de reemplazo FIFO cuando sea necesario.

b) Calcular la tasa de fallos con cada esquema y el tiempo total requerido para acceder a todas las direcciones

Mapeado directo:

Dirección	Dirección (bloque)	Etiqueta	Línea	Acierto/Fallo	Bloque reemplazado	Tiempo (ns)
0x7001	0x700	01110	0	Fallo		110
0x7002	0x700	01110	0	Acierto		30
0x7010	0x701	01110	1	Fallo		110
0x7010	0x701	01110	1	Acierto		30
0x780A	0x780	01111	0	Fallo	0x700	110
0x7002	0x700	01110	0	Fallo	0x780	110
0x7811	0x781	01111	1	Fallo	0x701	110
0x7011	0x701	01110	1	Fallo	0x781	110
0x7811	0x781	01111	1	Fallo	0x701	110
0x780A	0x780	01111	0	Fallo	0x700	110

Tasa fallo = 80%, Tiempo total: 940 ns

Mapeado completamente asociativo:

Dirección	Etiqueta	Línea	Acierto/Fallo	Bloque reemplazado	Tiempo (ns)
0x7001	011100000000	0	Fallo		110
0x7002	011100000000	0	Acierto		30
0x7010	011100000001	1	Fallo		110
0x7010	011100000001	1	Acierto		30
0x780A	011110000000	2	Fallo		110
0x7002	011100000000	0	Acierto		30
0x7811	011110000001	3	Fallo		110
0x7011	011100000001	1	Acierto		30
0x7811	011110000001	3	Acierto		30
0x780A	011110000000	2	Acierto		30

Tasa fallo = 40%; Tiempo total: 620 ns

Mapeado asociativo por conjunto:

Dirección	Etiqueta	Conjunto	Vía	Acierto/Fallo	Bloque reemplazado	Tiempo (ns)
0x7001	0111000	0	0	Fallo		110
0x7002	0111000	0	0	Acierto		30
0x7010	0111000	1	0	Fallo		110
0x7010	0111000	1	0	Acierto		30
0x780A	0111100	0	1	Fallo		110
0x7002	0111000	0	0	Acierto		30
0x7811	0111100	1	1	Fallo		110
0x7011	0111000	1	0	Acierto		30
0x7811	0111100	1	1	Acierto		30
0x780A	0111100	0	1	Acierto		30

Tasa fallo = 40%; Tiempo total: 620 ns

PROBLEMA 5

Se dispone de un sistema con una memoria caché de dos niveles. En la ejecución de una determinada aplicación, la tasa de aciertos de la caché de nivel 1 (la más próxima al procesador) es del 90%; mientras la tasa de aciertos de la caché de nivel 2 es del 95%. Suponiendo que se han realizado un millón de accesos a memoria, determinar:

a) Número de accesos a la caché de nivel 1

1000000

b) Número de accesos a la caché de nivel 2

$1000000 * 0,1 = 100000$

c) Número de accesos a la memoria principal

$100000 * 0,05 = 5000$

d) Calcular el número de ciclos invertido en ejecutar esos accesos a memoria suponiendo que los accesos a la caché de nivel 1 tardan 1 ciclo; los que se realizan a la caché de nivel 2, 5 ciclos; y los accesos a memoria principal, 100 ciclos.

$1000000 * 1 + 100000 * 5 + 5000 * 100 = 2000000$ ciclos

PROBLEMA 6

Dispones de un sistema con bus de direcciones de 32 bits y bus de datos de 8 bits al que has incorporado una caché de datos de 64KB de capacidad y mapeado directo. El sistema de caché trabaja con bloques de 16 palabras

Considera el siguiente fragmento de código:

```
for (i = 0; i < 128; i++)
    for (j = 0; j < 128; j++)
        C[i][j] = A[i][j] + B[i][j];
```

donde A, B y C son matrices de 128x128 números de 4 bytes, almacenados por filas. La matriz A comienza en la dirección 0x00010000, y el resto de matrices están colocadas a continuación. Tanto i como j se operarán con registros.

a) Suponiendo que la caché está inicialmente vacía, indicar la tasa de fallos del código propuesto

2^4 palabras/bloque

Memoria: 2^{32} palabras de 32 bits $\rightarrow 2^{28}$ bloques

Caché: 2^{16} palabras $\rightarrow 2^{12}$ bloques

Dirección: 16 x ETIQUETA + 12 x LÍNEA + 4 x PALABRA

Matrices:

$128 \times 128 = 2^{14} \rightarrow (4 \text{ bytes/elemento}): 2^{14} \times 2^2 = 2^{16}$ palabras $\rightarrow 2^{12}$ bloques

A \rightarrow 0x00010000 : 0x0001FFFF

B \rightarrow 0x00020000 : 0x0002FFFF

C \rightarrow 0x00030000 : 0x0003FFFF

Hay que realizar 4 accesos por cada cifra de cada matriz. Por ejemplo, para acceder al elemento A[0][0], habrá que acceder a las direcciones 0x00010000, 0x00010001, 0x00010002 y 0x00010003. No obstante, como en una línea de la caché cabe un bloque de 16 palabras, con cada acceso me habré traído 4 elementos de la matriz.

Elemento	Dirección	Etiqueta	Línea	Acierto/fallo	Reemplaza?
A[0][0]	0x00010000	0001	000	Fallo	
A[0][0]	0x00010001	0001	000	Acierto	
A[0][0]	0x00010002	0001	000	Acierto	
A[0][0]	0x00010003	0001	000	Acierto	
B[0][0]	0x00020000	0002	000	Fallo	0x0001000
B[0][0]	0x00020001	0002	000	Acierto	
B[0][0]	0x00020002	0002	000	Acierto	
B[0][0]	0x00020003	0002	000	Acierto	
C[0][0]	0x00030000	0003	000	Fallo	0x0002000
...

Se va a producir un fallo cada 4 accesos: 25%

- b) Un esquema completamente asociativo sería demasiado caro y no puedes aumentar el tamaño de tu caché, ¿qué tipo de mejora plantearías en la estructura de la caché para reducir la tasa de fallos?**

Si planteamos una memoria asociativa de, por ejemplo, 4 vías, tendríamos:

2^2 bloques/conjunto \rightarrow Caché 2^{10} conjuntos

Dirección: 18 x ETIQUETA + 10 x CONJUNTO + 4 x PALABRA

Elemento	Dirección	Etiqueta	Conjunto	Vía	Acierto/fallo	Reemplaza?
A[0][0]	0x00010000	4	0	0	Fallo	
A[0][0]	0x00010001	4	0	0	Acierto	
A[0][0]	0x00010002	4	0	0	Acierto	
A[0][0]	0x00010003	4	0	0	Acierto	
B[0][0]	0x00020000	8	0	1	Fallo	
B[0][0]	0x00020001	8	0	1	Acierto	
B[0][0]	0x00020002	8	0	1	Acierto	
B[0][0]	0x00020003	8	0	1	Acierto	
C[0][0]	0x00030000	12	0	2	Fallo	
C[0][0]	0x00030001	12	0	2	Acierto	
C[0][0]	0x00030002	12	0	2	Acierto	
C[0][0]	0x00030003	12	0	2	Acierto	
A[0][1]	0x00010004	4	0	0	Acierto	
...

En cada línea cabe un bloque con 4 elementos de la matriz. Al acceder al primer byte del primer elemento de cada matriz habrá un fallo, pero después no habrá un nuevo fallo hasta que accedamos al quinto elemento de esa matriz.

Por lo tanto, para acceder a 4 elementos son 16 accesos \rightarrow 1 fallo por cada 16 accesos \rightarrow Tasa de fallo aproximada: $1/16 = 6,25\%$

MEMORIA VIRTUAL

PROBLEMA 1

Un pequeño ordenador dispone de un sistema de memoria virtual paginada con páginas de 4K palabras. La memoria principal está dividida en cuatro marcos, pero el marco cero no se puede reemplazar porque debe contener un sistema operativo básico con sus estructuras de datos asociadas (por ejemplo, la tabla de páginas). La dirección virtual es de 20 bits. Asumiendo que cada entrada de la tabla de páginas ocupa una palabra, justificar si la tabla de páginas cabe o no en el marco cero. Explicar el funcionamiento y estructura de la memoria virtual anterior, incluyendo las diferentes posibilidades que puedan darse en un acceso a memoria virtual.

Suponiendo que se realizan los accesos a las siguientes direcciones virtuales y reemplazamiento LRU, indicar los contenidos finales de las entradas de la tabla de páginas que hayan sido usadas (asumir que todos los marcos, salvo el cero, no contienen nada inicialmente):

Traza Dir. Virtuales (Hex): 02000, 05000, 02001, 0705C, 01A1B, 04016, 0705E, 01A1C

4K palabras/página = 2^{12} palabras/página

Memoria dividida en 4 marcos $\rightarrow 2^{12} \times 4 = 2^{14}$ palabras

Dirección física:

M	M	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Dirección virtual

P	P	P	P	P	P	P	P	P	O	O	O	O	O	O	O	O	O	O	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

P: Página; O: Offset

Hay 2^8 páginas virtuales < tamaño página 0 (2^{12}) \rightarrow Cabe en el marco 0.

Para realizar la traza, debemos quedarnos con la parte de la dirección que corresponde al número de página.

Página virtual (HEX)	Marco (BIN)	Acierto/Fallo	Reemplazo?
0x02	01	Fallo	
0x05	10	Fallo	
0x02	01	Acierto	
0x07	11	Fallo	
0x01	10	Fallo	0x05
0x04	01	Fallo	0x02
0x07	11	Acierto	
0x01	10	Acierto	

El contenido final de la memoria física sería:

Número de marco	Contenido
00	RESERVADO
01	0x04
10	0x01
11	0x07

PROBLEMA 2

Un procesador dispone de los siguientes elementos para gestionar la memoria:

- Emplea direcciones virtuales de 36 bits
- Tiene un TLB asociativo de 2 entradas por conjunto
- El TLB contiene 256 entradas
- La memoria se compone 1M de marcos
- Tiene una caché asociativa de 2 vías y 128KB
- Cada bloque es de 8 palabras
- La etiqueta de la caché es de 16 bits
- Cada palabra es de 1 byte

Indicar:

1. Tamaño de la dirección física
2. Estructura de la dirección física, desde la perspectiva de la caché (bits dedicados a conjuntos, etiqueta, palabras...)
3. Estructura de la dirección física, desde la perspectiva del paginado
4. Estructura de la dirección virtual, desde la perspectiva del paginado
5. Estructura de la dirección virtual, desde la perspectiva del TLB

Caché:

$$128 \text{ KB} = 2^{17} \text{ palabras}$$

$$8 \text{ palabras/bloque} = 2^3 \rightarrow 2^{14} \text{ bloques}$$

$$2 \text{ bloques/conjunto} = 2^1 \rightarrow 2^{13} \text{ conjuntos}$$

Dirección física (perspectiva de la caché): 32 bits

E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	C	C	C	C	C	C	C	C	C	C	C	C	C	P	P	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

P: Palabra (x3); C: Conjunto (x13); E: Etiqueta (x16)

Memoria: 1M páginas = 2^{20}

$$32 - 20 = 12 \rightarrow 2^{12} \text{ palabras/página}$$

Dirección física (perspectiva de la mem. virtual): 32 bits

M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	O	O	O	O	O	O	O	O	O	O	O	O	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

M: Marco (x20); O: Offset (x12)

Dirección virtual (perspectiva del paginado): 36 bits

P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	O	O	O	O	O	O	O	O	O	O	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

P: Página (x24); O: Offset (x12)

TLB:

$$256 \text{ entradas} = 2^8$$

$$2 \text{ entradas/conjunto} \rightarrow 2^7 \text{ conjuntos}$$

Dirección virtual (perspectiva de la TLB): 36 bits

E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	C	C	C	C	C	C	C	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

E: Etiqueta (x17); C: Conjunto (x7); O: Offset (x12)

PROBLEMA 3

Una CPU posee un sistema de gestión de memoria virtual paginada dotado de un TLB. Este último actúa de forma similar a un caché asociativo por conjunto para mantener las entradas más usadas de la tabla de páginas en el interior de la CPU. Suponiendo que

- Tanto el bus de direcciones como el bus de datos de la CPU son de 32 bits
 - El tamaño de la página es de 4K palabras
 - Cada entrada en la tabla de páginas ocupa 32 bits
 - El tamaño de la memoria del TLB (sin contar campos tag y bits dirty y valid) es de 1024 palabras
 - Un conjunto del TLB contiene a dos entradas de la tabla de páginas
 - El campo tag (etiqueta) asociado a cada entrada del TLB es de 15 bits
-
- a) Describir el proceso que tiene lugar a partir de que un programa referencia una dirección virtual hasta que se encuentra la correspondiente dirección física. ¿Cuántos bits componen una dirección virtual? ¿Qué tamaño tendrá la tabla de páginas?
 - b) Supuesta la traza de páginas siguiente: 1025, 0, 1536, 1, 0, 1025, 256, 512. Determinar el número total de accesos a memoria. Suponer que el TLB funciona con algoritmo LRU y que no contiene inicialmente ninguna entrada de las correspondientes a la traza. Asumir que todas las páginas de la traza residen en un marco en memoria principal.

Para resolver el primer apartado, debemos estudiar los datos que se nos proporcionan:

- La dirección física es de 32 bits $\rightarrow 2^{32}$ palabras
- Páginas de 4K $\rightarrow 4K=2^{12} \rightarrow 2^{20}$ marcos en la memoria física

[illegible]

M: Marco (x20); O: Offset (x12)

Sobre la TLB:

- 1024 palabras = 2^{10}
- Asociativa 2 vías = 2^9 conjuntos
- 15 bits de etiqueta (tag) $\rightarrow 15 + 9 = 24$ bits de la dirección para el número de página virtual

[illegible]

P: Página (x24); O: Offset (x12)

La dirección virtual es de $24 + 12 \text{ bits} = 36 \text{ bits}$

- Cada entrada en la tabla de páginas es de 32 bits = 1 palabra.

La tabla de páginas ocupará 2^{24} palabras.

Para el segundo apartado:

Página virtual	Etiqueta	Conjunto	Acierto/Fallo	Accesos	
1025	0000000000000010	000000001	Fallo	2	
0	0000000000000000	000000000	Fallo	2	
1536	0000000000000011	000000000	Fallo	2	
1	0000000000000000	000000001	Fallo	2	
0	0000000000000000	000000000	Acierto	1	
1025	0000000000000010	000000001	Acierto	1	
256	0000000000000000	100000000	Fallo	2	
512	0000000000000001	000000000	Fallo	2	Reemplaza a 1536
TOTAL				14	

La tasa de fallos sería del 75%