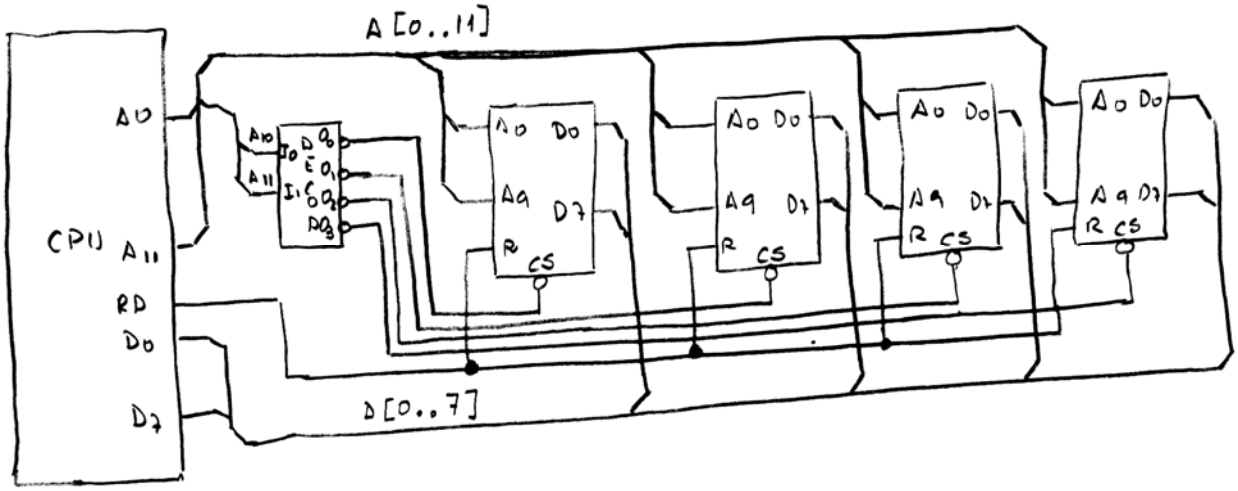
Problema 2.1

Circuito
- CPU
- 4 ROM 1K × 8
- Adress Bus 12 líneas
- Data Bus 8 "

1K × 8 ⇒ { D0 ÷ D7 ; A0 ÷ A9 }

1K = 1024 ⇒ 2^{10}



Problema 2.2

Se dispone de RAMs de 128 × 8

a) ¿Nº de módulos para hacer una RAM de 4096 Bytes?

$$4096\ B \Rightarrow 4096 \times 8 \Rightarrow \frac{4096 \times 8}{128 \times 8} = 32 \text{ módulos}$$

b) ¿Nº líneas bus direcciones para acceder a los 4096?

$$\begin{array}{r|l} 4096 & 16 \\ \hline 89 & 256 \quad 16 \\ 96 & 96 \quad 16 \\ 0 & 0 \quad 0 \quad 16 \\ & \quad\quad 1 \end{array} \Rightarrow 4096 = 1000_H = 1\ 0000\ 0000\ 0000$$

12 líneas { 0 0000 0000 0000 ; 1 1111 1111 1111 }

¿Nº líneas comunes? ⇒

$$\begin{array}{r|l} 128 & 16 \\ \hline 0 & 8 \end{array} \Rightarrow 128 = 80 = 1000\ 0000$$

7 líneas { 000 0000 ; 111 1111 }

c) ¿Líneas a decodificar para seleccionar cada módulo?

1 → 12 lin AB − 7 lin comunes = 5 líneas

2 → 32 módulos ⇒ 2^5 ⇒ 5 líneas

¿Tamaño de los decodificadores?

⇓

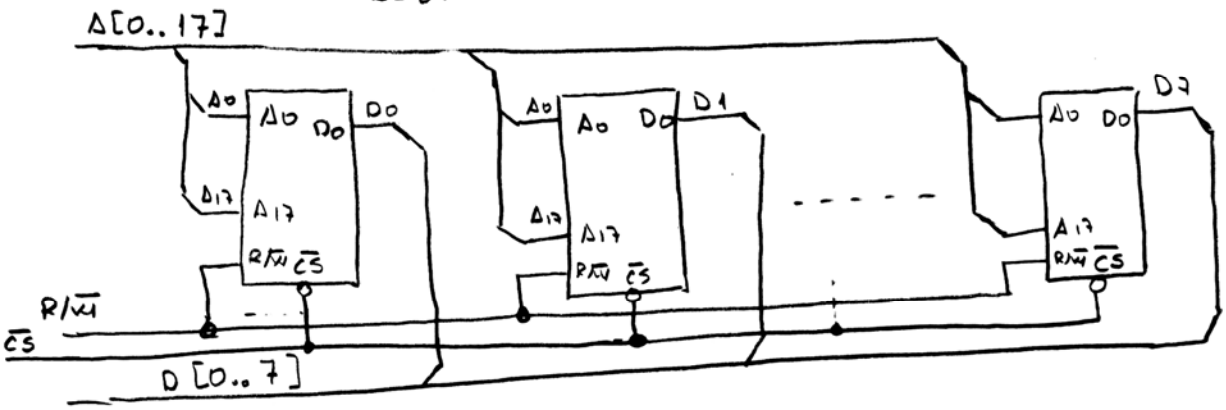
5 líneas entrada
32 módulos salida } 5 / 32

Problema 2.3

Con C.C.I.I. RAM 256K×1

a) Hacer mem 256 K×8 256 K ⇒ 2^{18} ⇒ 18 bit direcc

$$N^{\circ}\ \text{módulos} = \frac{256K \times 8}{256K \times 1} = 8 \text{ módulos}$$

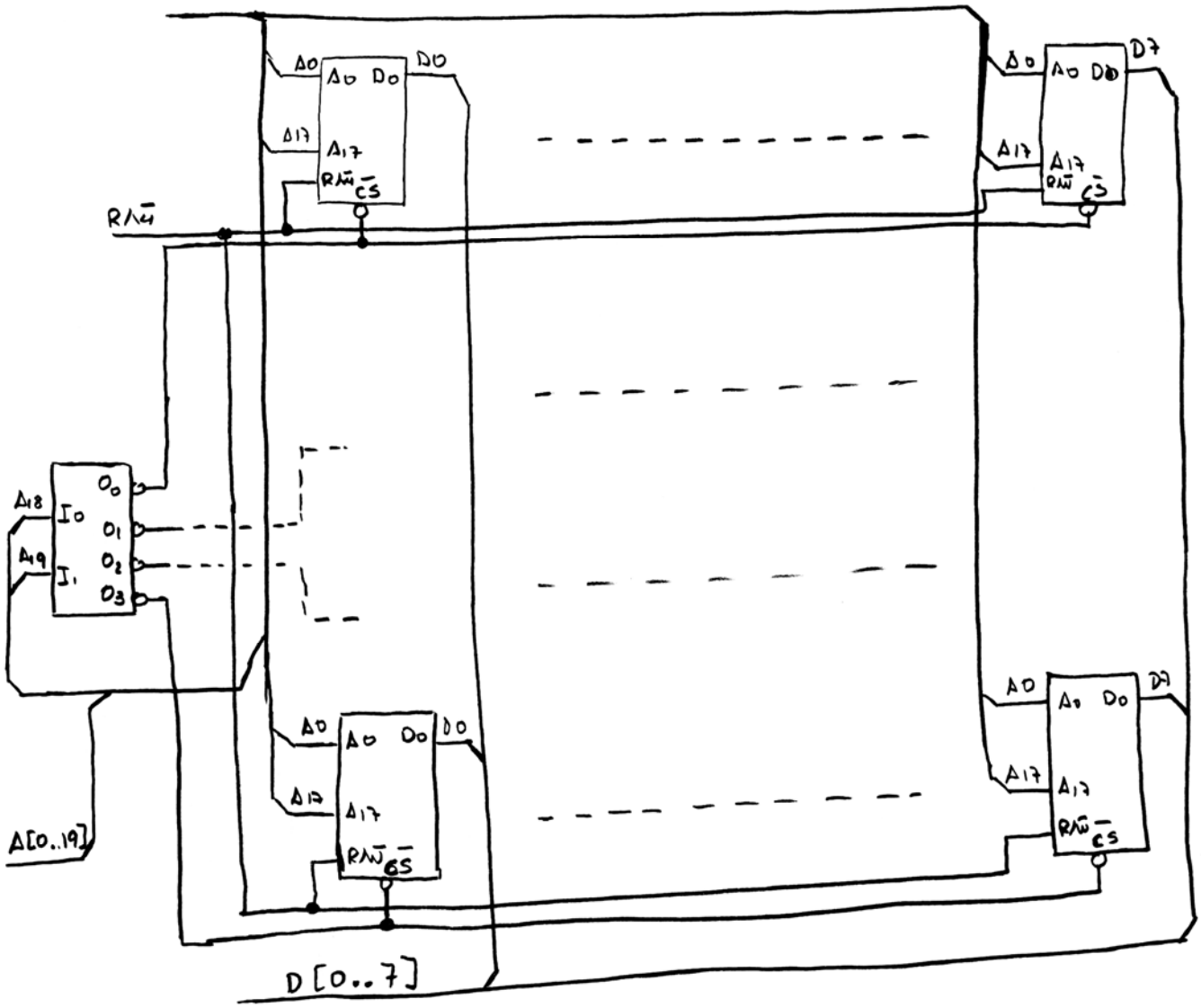


b) ¿Hacer mem 1M x 8 bits?

En apdo ⓐ se ha hecho 256K x 8 ⇒ para hacer

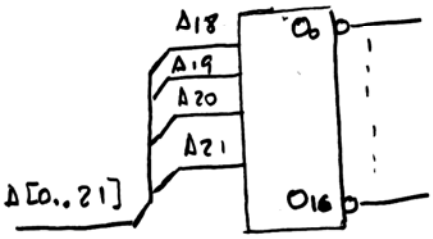
1M x 8 ⇒ $\frac{1M \times 8}{256 \times 8} = 4$ Módulos como el apdo ⓐ

Bus direcciones ⇒ 1M ⇒ 2^{20} ⇒ A0 ÷ A19



c) 4M x 16 bits

Igual al apdo ⓐ pero
- Cada línea 16 mod ⇒ D0 ÷ D15
- En lugar de 4 líneas 16 líneas
- " " decodificador 2 a 4 uno 4 a 16



Problema 2.4

Disponibilidad → RAM 1024 × 4
- → CS
- → R/$\overline{W}$

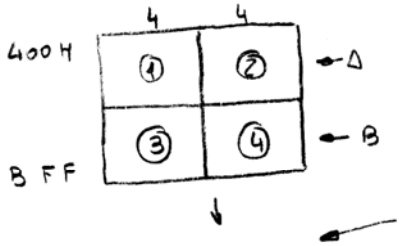
Diseño ⇒ 2K × 8

Mapa 0400H ÷ 0BFFH

Líneas de control de la CPU $\overline{MEML}$ y $\overline{MEME}$ → SELECCION PLA

Nº módulos ⇒ $\frac{2K \times 8}{1K \times 4}$ = 4 módulos

MAPA



400H = 0100 0000 0000

BFFH = 1011 1111 1111

1024 ⇒ 2^{10} ⇒ 10 bits = A[0..9]

2048 ⇒ 2^{11} ⇒ 11 bits = A[0..10]

A10 y A11 selecciona pg

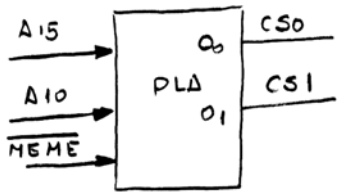
A o B ⇒ 01 → Pg A
10 → Pg B

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	Dir	CI
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0400	1+2
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FF	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800	3+4
0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	0BFF	

Control

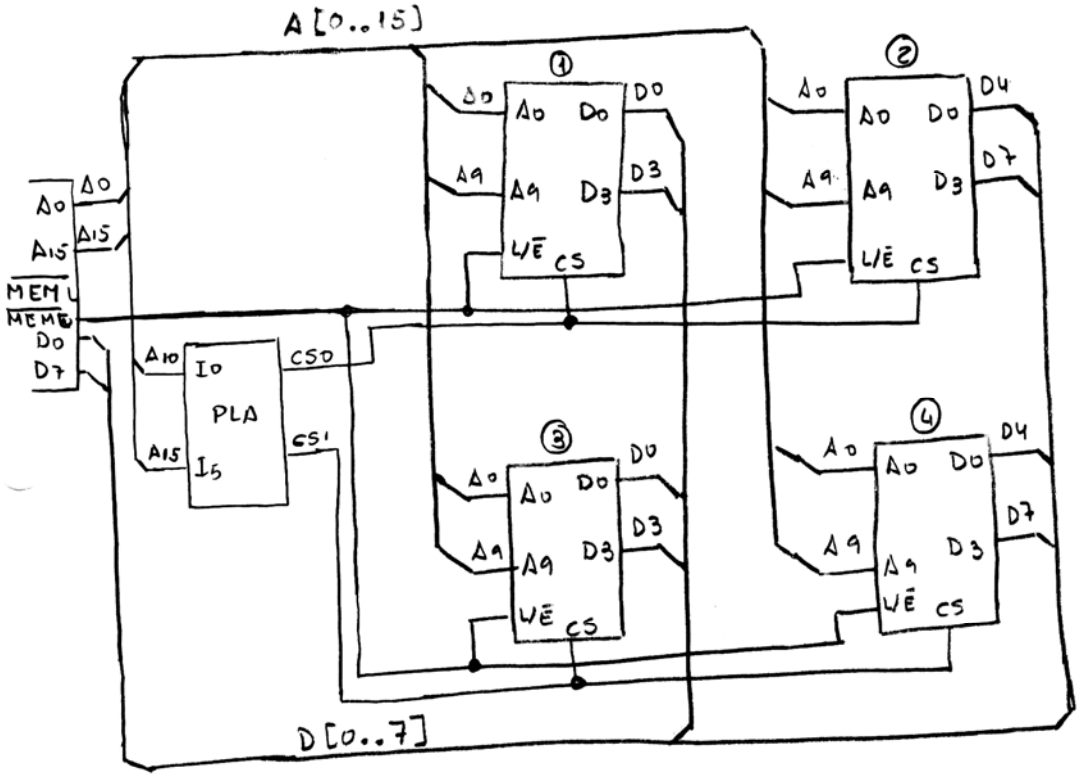
El control se efectuará con una PLA de forma que cuando le entre la combinación $A_{15}..A_{10}$ (000001) seleccionará los C.I. 1+2 y con la $A_{15}..A_{10}$ (000010) seleccionará 3+4

Al mismo tiempo como la mem tiene L/$\overline{E}$ y la CPU dos señales ($\overline{MEML}$ y $\overline{MEME}$) se transforman en una sola, de forma que nos olvidamos de $\overline{MEML}$ y lo que hacemos es que cuando $\overline{MEME}$ = 0 ⇒ escribir y si $\overline{MEME}$ = 1 ⇒ leer



$$CS0 = \overline{A_{15}}\ \overline{A_{14}}\ \overline{A_{13}}\ \overline{A_{12}}\ \overline{A_{11}}\ A_{10}$$

$$CS1 = \overline{A_{15}}\ \overline{A_{14}}\ \overline{A_{13}}\ \overline{A_{12}}\ A_{11}\ \overline{A_{10}}$$



- Para leer entre una dir 400 ÷ 7FF ⇒ se habilita CSØ y como no se escribe $\overline{MEME} = 1$ pq CPU pone $\overline{MEML} = 0$ y $\overline{MEME} = 1$ las mem 1 + 2 tienen CS = 1 y L/Ē = 1 y de A0 a A9 la dirección ⇒ sacan por bus datos el dato
- Para leer en una dir diferente a 400 ÷ 7FF ⇒ CSØ = Ø ⇒ No se selecciona CS de 1 y 2 ⇒ No habilitados y ni se lee ni escribe
- Para escribir entre dir 400 ÷ 7FF ⇒ se habilita CSØ y $\overline{MEME} = 0$ pq CPU pone $\overline{MEML} = 1$ y $\overline{MEME} = 0$ ⇒ los c.I. 1 + 2 están habilitados CSØ y L/Ē = 0 ⇒ escribe
- Al escribir en dir. diferentes a 400 ÷ 7FF ⇒ 1 y 2 están hibalitidos CSO = Ø ⇒ Ni se lee y se escribe en ellos
- Para 3 y 4 el razonamiento es el mismo

PRUM5

Problema 2.5

Computador → Bus datos 8 bits

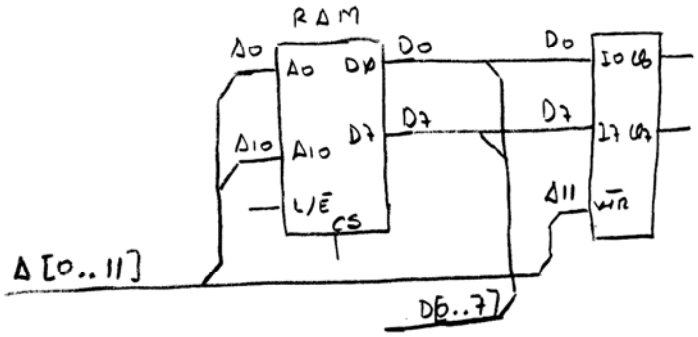
" dir. 12 bits

Con A[0..10] → Banco mem 2KB

Con A11 se habilita un reg que recibe el dato

→ Ampliación a 16KB

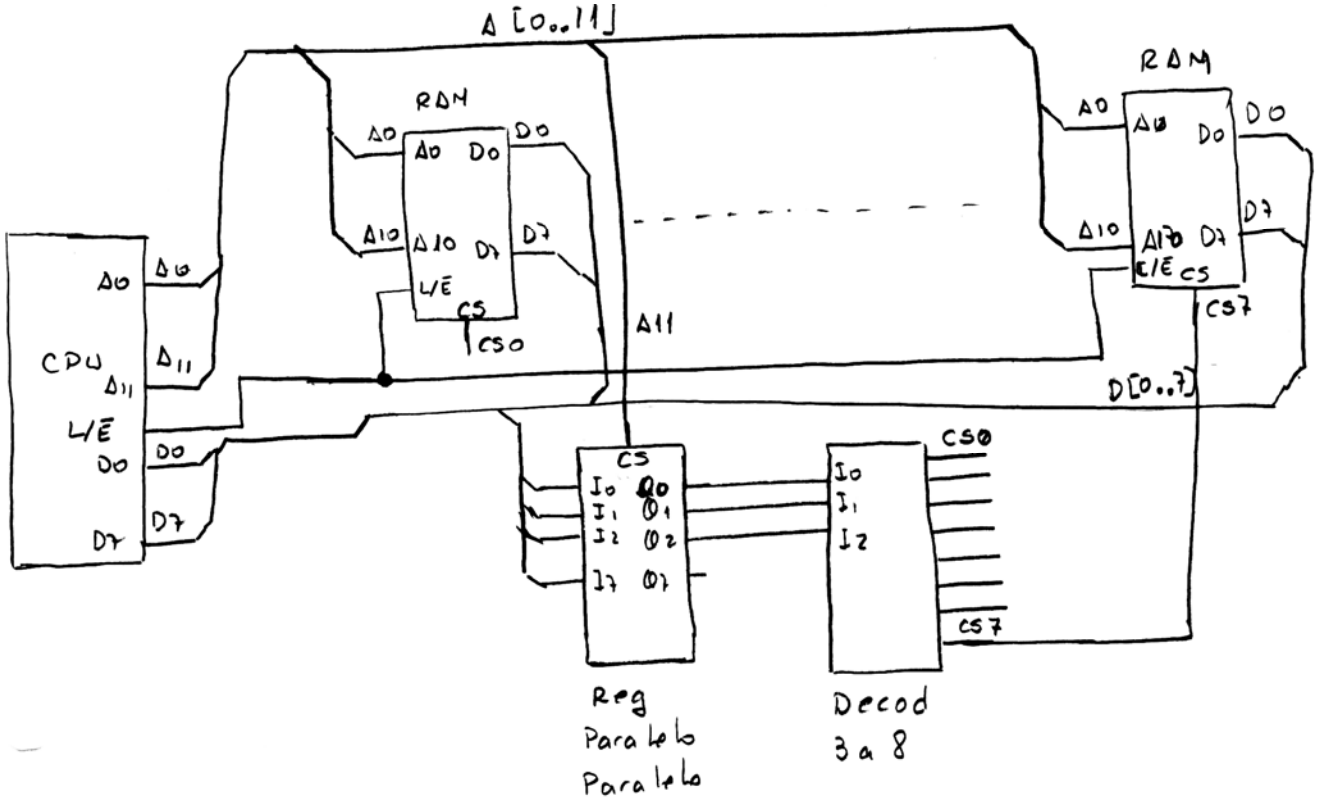
Actualidad (2K)



Solución

Hay que apuntar a 8 bloques de 2K ⇒ 16K ⇒ 2^{14} ⇒ bus direcciones 14 bits pero CPU solo 12 bits.

Solución: dividir la memoria en 8 bloques y realizar un direccionamiento indirecto, de forma que la dirección efectiva esté formada por los bits A[0..10] combinados con una selección de página que estará escrita en la memoria y a la que previamente se accederá para cargarla en el registro y posteriormente acceder a la página



Primero con A11 = 1 leemos memoria y guardamos en el registro y luego con A11 = Ø (ya está en el registro la página) se vuelve a sacar una dirección por A [10..0] que junto con la página apunta a la dirección final

Problema 2.6

Computador : - 8 bits bus datos

- 512 × 8 RAM
- 512 × 8 ROM

Circuitos integrados: RAM (128 × 8) ROM (512 × 8)

MAPA MEMORIA :

RAM 1	0000 ÷ 007F
RAM 2	0080 ÷ 00FF
RAM 3	0100 ÷ 017F
RAM 4	0180 ÷ 01FF
ROM	0200 ÷ 03FF

PR. UMY

RAM TOTAL = $512 \times 8 \Rightarrow 512 = 2^9 \Rightarrow$ bus dir = 9 bits

RAM C.I. = $128 \times 8 \Rightarrow 128 = 2^7 \Rightarrow$ " " = 7 bits $A[0..6]$

ROM TOTAL = ROM C.I. = $512 \times 8 \Rightarrow 512 = 2^9 \Rightarrow$ bus dir = 9 bits

MEMORIA TOTAL $\Rightarrow \underbrace{(512 \times 8)}_{RAM} + \underbrace{(512 \times 8)}_{ROM} = 1024 \times 8 \Rightarrow 1024 = 2^{10} \Rightarrow A[0..9]$

Nº C.C.I.I.:

RAM $\Rightarrow \dfrac{512 \times 8}{128 \times 8} = 4$ C.C. I.I.

ROM $\Rightarrow \dfrac{512 \times 8}{512 \times 8} = 1$ " "

MAPA

128 B de la RAM

De aquí deducimos que con A7 y A8 selecci. el C.I. RAM

Con A9 la ROM

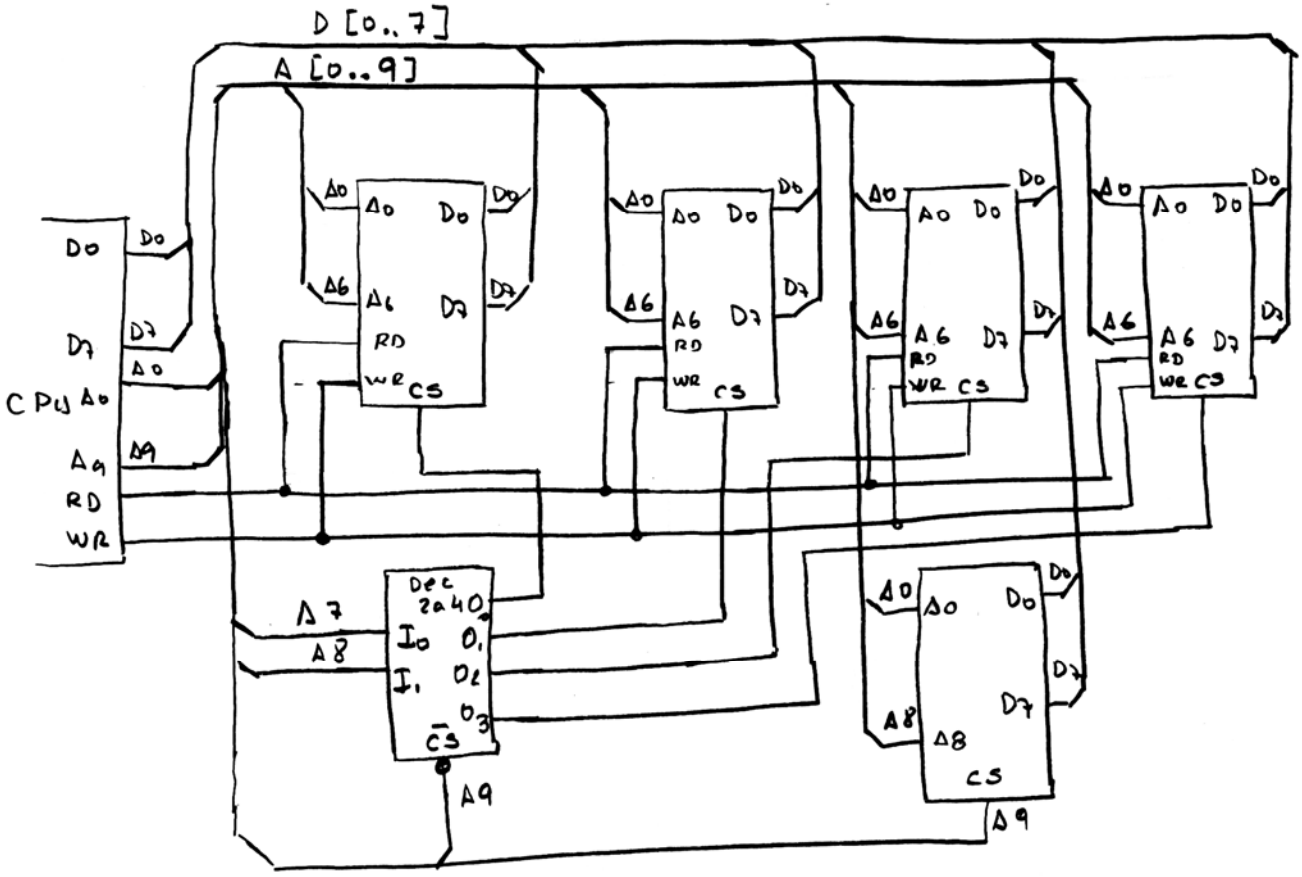
A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Dirección	C. I.
0	0	0	0	0	0	0	0	0	0	0000	RAM 1
0	0	0	1	1	1	1	1	1	1	007F	
0	0	1	0	0	0	0	0	0	0	0080	RAM 2
0	0	1	1	1	1	1	1	1	1	00FF	
0	1	0	0	0	0	0	0	0	0	0100	RAM 3
0	1	0	1	1	1	1	1	1	1	017F	
0	1	1	0	0	0	0	0	0	0	0180	RAM 4
0	1	1	1	1	1	1	1	1	1	01FF	
1	0	0	0	0	0	0	0	0	0	0200	ROM
1	1	1	1	1	1	1	1	1	1	03FF	

512 B de la ROM

Con lo que vemos que:

- A9 → 0 → Selección de mapa de RAMs
  - 1 → " " " " ROM
- A8, A7
  - 00 → " " RAM 1
  - 01 → " " RAM 2
  - 10 → " " RAM 3
  - 11 → " " RAM 4

⇒ Decodificador 2 a 4 con CS si A9 = 0 habilitado por UM8



PR.UM9

Problema 2.7

Mem. caché "asociativa por conjuntos" de 64 bloques y 4 bloques/conjunto.

Mem ppal 4 K bloques de 128 palabras/bloque

¿Formato de dirección de la mem. ppal?

- 4K × 128 pal/bloque = 512 K palabras ⇒ 2^{19} ⇒ 19 bits direcc
- 64 bloques a 4 bloq/conjunto ⇒ $\frac{64}{4} = 16$ conjun ⇒ 2^4 ⇒ 4 bits conjun
- 128 pal/bloque ⇒ 2^7 ⇒ 7 bits " dir palabra en bloque

dirección

b18 … b10	b7 … b6	b0
Etiqueta	Conjunto	Palabra
8	4	7

Problema 2.8

- Computadora: 32 K × 16 mem. ppal

  Caché: Asociativa por conjuntos
  4 K palabras
  4 bloq/conjunto
  64 pal/bloque

- CPU lee consecutivo de la dir 0 a la 4351 y luego otras 9 veces
- Cache 10 veces más rápida que la mem. ppal
- ¿Mejora en el η del sistema?
- Paso directo de mem ppal a caché de pala. que produce fallo.
- Algoritmo reemplazamiento = LRU

- 32 K mem ppal $\Rightarrow 2^{15} \Rightarrow$ 15 bits dir mem. ppal ($b_0 .. b_{14}$)

- 64 pal/bloque $\Rightarrow \frac{32K}{64} = 512$ bloques

- 4K caché $\Rightarrow$ 64 pal/bloque $\Rightarrow \frac{4096}{64} = 64$ bloques $\Rightarrow$ 4 blo/conjunto $\Rightarrow$

$\Rightarrow \frac{64}{4} = 16$ conjuntos $\Rightarrow 2^4 \Rightarrow$ 4 bits para conjunto

- 64 pal/bloque $\Rightarrow 2^6 \Rightarrow$ 6 bits dirección palabra

dirección

b_{14} — b_{10}	b_9 — b_6	b_5 — b_0
ETIQUETA	CONJUNTO	PALABRA
5	4	6

- Direcciones de la 0 a 4351 $\Rightarrow$ 4352 direcc $\Rightarrow \frac{4352 \text{ dir}}{64 \text{ pal/blo}} = 68$ bloq

SIN MEM CACHE

$T_{MP} = T_{total} = N^{\circ}$ pal. $\times$ 10 veces $\times$ $\underbrace{10\,T}_{\text{t. acceso respecto a caché}} = 4352 \cdot 10 \cdot 10\,T = 435200\,T$

CON CACHE

1ª lectura $\Rightarrow$ Todo fallo $\Rightarrow T_{fallo} = 4352 \cdot 10T = 43520\,T$ (ppal)

Caché $\Rightarrow$ Después de cargado el bloque leerlo todo menos 1ª palabra $\Rightarrow T_{caché} = 4352\,T - 68\,T = 4284\,T$

Además se carga la caché con los bloques 0 a 63

y luego los bloques 64, 65, 66, 67 producen fallo

y sustituyen a los bloques 0, 1, 2, 3. Los

bloques del 4 al 67 quedan en caché

$T_1 = T_{fallos} + T_{caché} = 43520T + 4284T = 47804T$

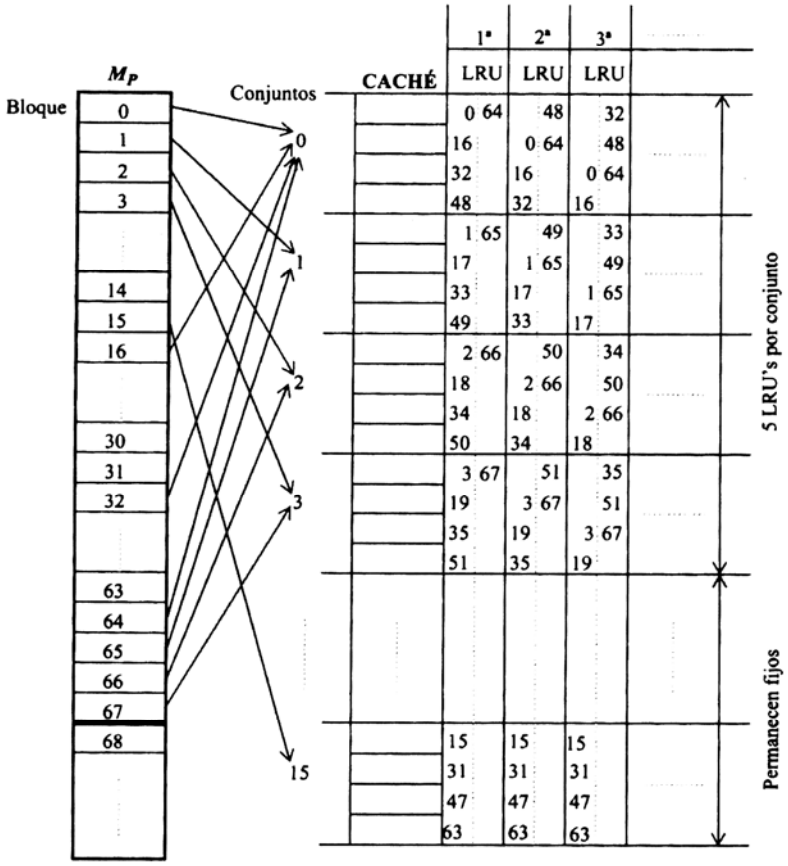
Lecturas posteriores

El fallo se va a producir en los bloques

0, 1, 2, 3 que pasarán a ocupar los bloques 16, 17, 18, 19

debido al algoritmo LRU y se producirá fallo al buscar

los 16 a 19 ⇒ produce 5 fallos por conjunto, en 4 conjuntos



$$T.\ fallos = \underset{\text{fallos/con}}{5} \times \underset{\text{conjun}}{4} \times \underset{\text{palabras}}{64} \times 10T = 12800T$$

$$T.\ caché = (4352 \times T) - \underbrace{(5 \times 4 \times T)}_{\text{Tiempo de no acceso a caché de 1ª pal. bloque}} = 4332T$$

$$T_2 = 12800T + 4332T = 17132T$$

$$T_n = 12800T + 4332T = 17132T$$

$$T_{caché} = 47804T + 9(17132T) = 201992T$$

$$Mejora = \frac{T_{MP} - T_{caché}}{T_{MP}} \times 100 = \frac{435200T - 201992T}{435200T} \times 100 = 53,6\%$$

PR.UMR

Problema 2.10

- Mem caché asociativa por conjuntos

  2 conjuntos

  4 palabr / bloque

  2048 palabras

  Mem ppal = 128 K × 32 bits

a) ¿Información para la caché?

2048 pal ⇒ 4 pal / bloque ⇒ 512 bloques

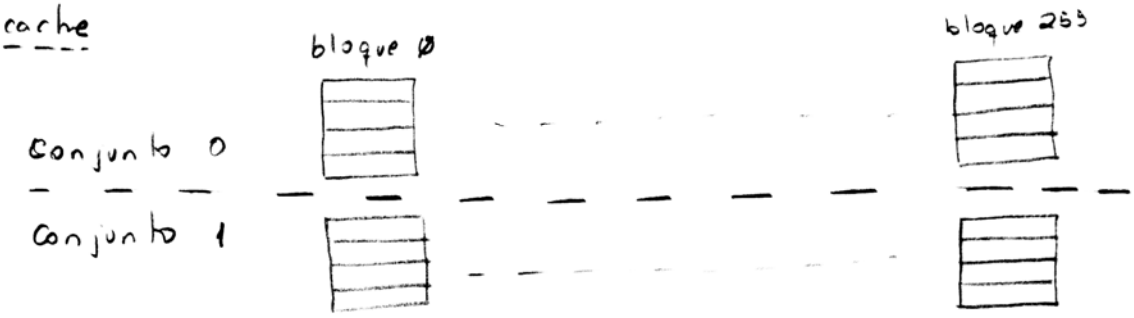
⇓

2^2 ⇒ 2 bits dir pal

2 conjuntos ⇒ 2^1 ⇒ 1 bit conjunto

$\frac{512}{2}$ = 256 blo/conjun

Mem cache



128 K mem ppal ⇒ 2^{17} ⇒ 17 bits bus dir

ANCHO MEM CACHE ⇒ ANCHO PAL + ANCHO ETIQUETA

⇓

ANCHO $_{CACHE}$ = 32 + 14 = 46 bits

Tamaño $_{CACHE}$ { 2048 × 32 = 65536 bits datos
2048 × 14 = 28672 " etiquetas

PR.UM13

Problema 2.11

Computador: mem ppal 64K × 16 bits

caché corresp. directa 1K × 16

↳ 4 pal / bloque

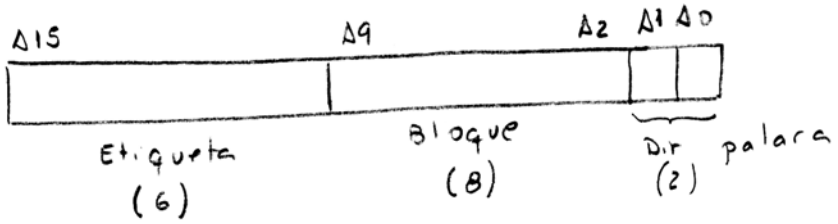
a) Formato dirección

64 K ⇒ 2^{16} ⇒ 16 bits

1K y 4 pal / bloque ⇒ 256 bloques ⇒ 2^8 ⇒ 8 bits bloque

⇓

2^2 ⇒ 2 bit dir palabra



b) ¿Bloques de la caché? → 256 bloques

c) ¿Bits / bloque en la caché?

bits / bloque = Nº pal / bloque × (bits datos + bits etiq) =

= 4 × (16 + 6) = 88 bits + auxiliares