

Ejercicio 1:

Dados los siguientes bloques de memoria:

- 1) 8Kbyte

2) 256 x 16bits

3) 2Kbits

4) 4K x 4bits
- 5) 16Knibble

6) 32Mbyte

7) 16K x 32bits

8) 1024Kbyte

Se pide:

- A. Ordenar los bloques de forma descendente según su capacidad total.
- B. Ordenar los bloques de forma ascendente según su cantidad de palabras.

A ① 8kbyte  $\Rightarrow$  8kword x 1byte  
 $2^{13}$  bytes de capacidad

② 256 x 16bits  $\Rightarrow$  256 words x 2byte  
 $256 = 2^8$ .  $2^9$  bytes de capaci

⑥ 32M byte  
 $1M = 2^{20}$   $2^{25}$  bytes  
 $32 = 2^5$

A

1	2	3	4	5	6	7	8
5	7	8	6	4	1	3	2

③ 2k bits  
 $2^{11}$  bits =  $2^8$  bytes

④ 4k x 4bits  
 $2^{14}$  bits  
 $2^{11}$  bytes

⑤ 16knibble  
 $2^{14}$  nibble  
1 nibble = 4bits  
 $2^{16}$  bits =  $2^{13}$  bytes

⑧  $2^{20}$  bytes

⑦ 16k x 32bits  
 $2^{14}$  x 4bytes  
 $2^{16}$  bytes

B cantidad de palabras: Palabras

~~①~~  $2^{13}$

~~②~~  $2^8$

~~③~~  $2^{11}$

~~④~~  $2^{12}$

~~⑤~~  $2^{14}$

~~⑥~~  $2^{25}$

⑦  $2^{14}$

⑧  $2^{20}$

1	2	3	4	5	6	7	8
4	1	2	3	5	8	6	7

Ejercicio 2:

Cuántos "chip" de memoria RAM de 2K palabras x 8 bits se necesitan para implementar un banco de memoria de:

- A. 2K palabras de 16 bits?
- B. 4K palabras de 8 bits?
- C. 4K palabras de 16 bits?

C)  $4kw \times 16bits$   
Se requieren 4 chips  
Dos bloques con 2 chips

A)  $2k palabras \times \underline{16bits}$

Se requieren 2 conectados en paralelo

B)  $\underline{4kw} \times 8bits$

Se requieren 2 conectados en serie

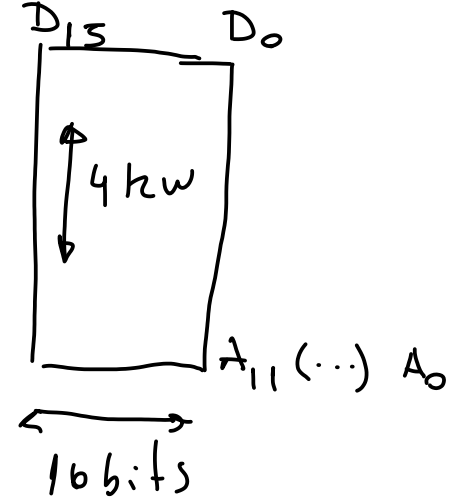
Ejercicio 3:

Construir un sistema de memoria RAM de 4K palabras de 16 bits mediante la utilización de "chips" de memoria de:

- A. 1K palabras de 16 bits.
- B. 4K palabras de 8 bits.
- C. 2K palabras de 8 bits.

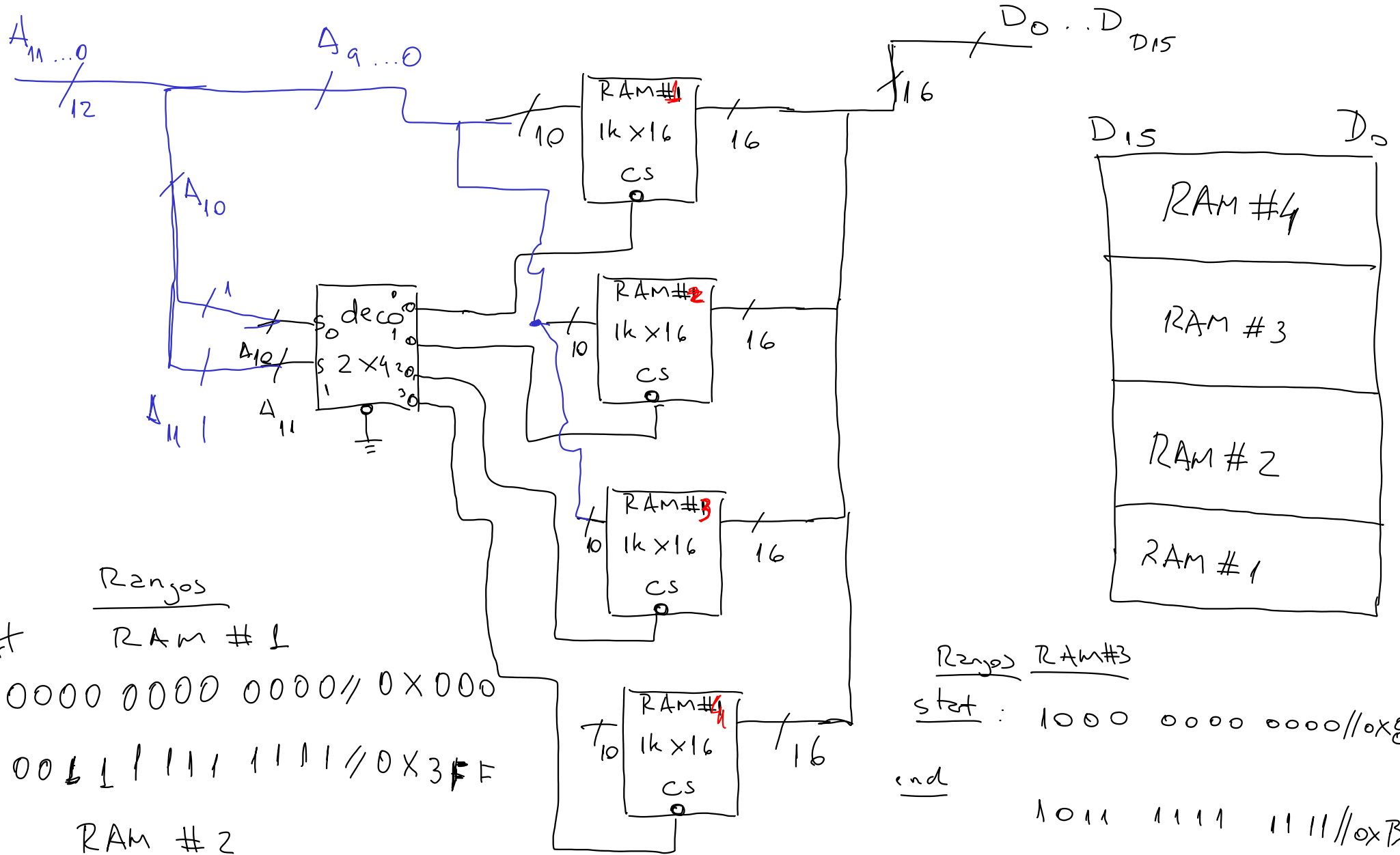
Ⓐ El ancho de palabra coincide  
no hay conexiones en paralelo  
- se requieren 4 chips en serie  
- los 2 bits más significativos  
de los señles de address codifican  
la selección de un decodificador  
 $2 \times 4$

Ⓐ Mapa de memoria



$$4kw = 2^{12}w$$

12 bits de codificación  
de direcciones



Rangos

RAM #1  
 start → 0000 0000 0000 // 0x000  
 end → 0011 1111 1111 // 0x3FF

RAM #2

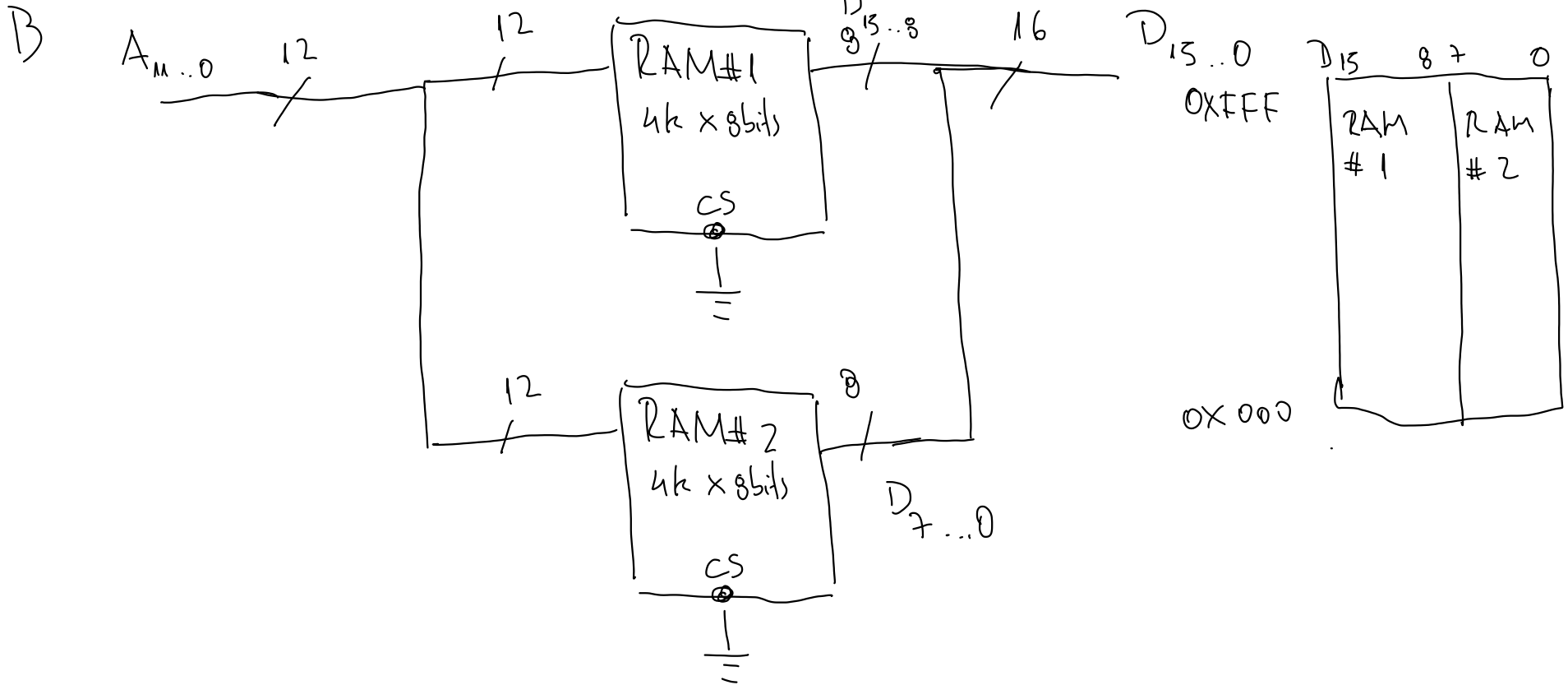
start → 0100 0000 0000 // 0x400  
 end → 0111 1111 1111 // 0x7FF

Rangos RAM #3

start : 1000 0000 0000 // 0x800  
 end : 1011 1111 1111 // 0xBFF

RAM #4

1100 0000 0000 // 0xC00  
 1111 1111 1111 // 0xFF



Ejercicio 4:

Construir un sistema de memoria como el que se muestra en el mapa de memoria de la figura. Se dispone para su implementación con los siguientes “chip” de memoria: EPROM de 2K x 8 bits y RAM de 2K x 4 bits.

- A. Realizar una implementación que NO genere posiciones imagen en el espacio no implementado.
- B. Realizar una implementación en la cual se generen posiciones imagen del contenido de la EPROM y la RAM a lo largo de todo el espacio direccionable. Analizar: ¿cuántas veces se replica el contenido de la RAM? y ¿cuántas veces se replica el contenido de la EPROM?, ¿por qué?

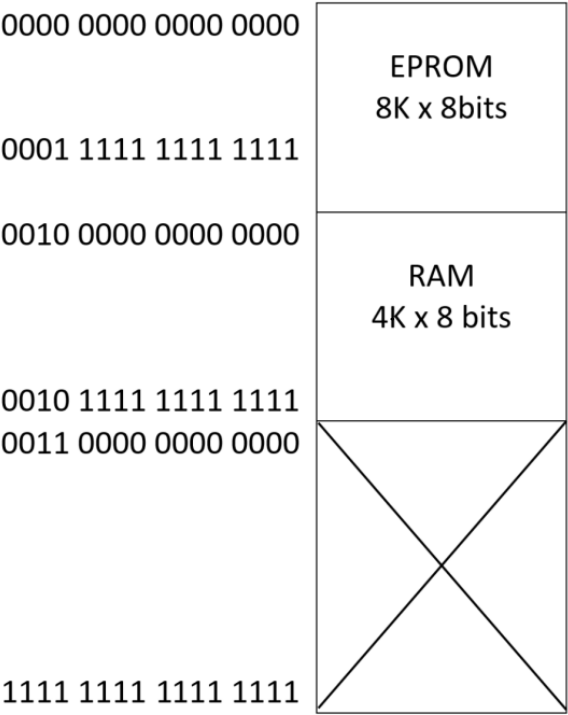
Ⓐ Parametros sistema: word = 8 bits

ⓧ El espacio implementado

↳ menor..

espacio direccionable = 16 bits de direcciones

64k palabras



### Ejercicio 5:

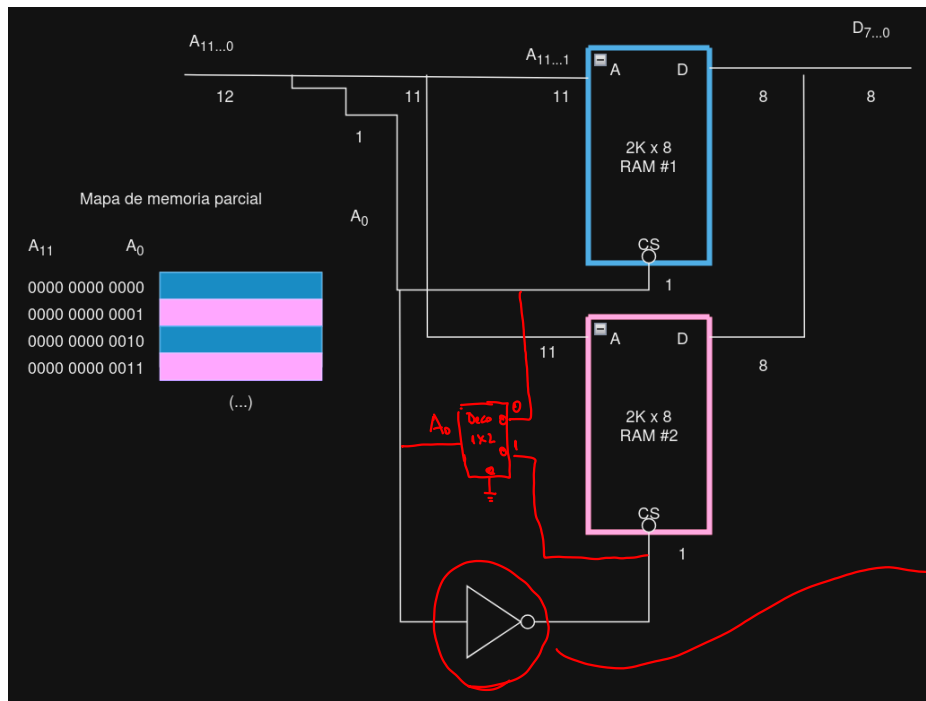
“Interleaved Memory” es una técnica utilizada para compensar la velocidad relativamente lenta de las memorias dinámicas de acceso aleatorio (DRAM) respecto al procesador. Esta técnica consiste en distribuir direcciones de memoria en forma uniforme a través de distintos bancos, y así evitar el tiempo de ciclo que se debería esperar entre dos accesos consecutivos a memoria. Sabiendo esto, se pide:

Implementar la sección de RAM 4Kx8bits como un sistema de memoria de dos bloques, de forma tal que las direcciones pares estén contenidas en un banco y las impares en otro.

### Solución

⊗ Señales de address total 12

⊗ Señales de address de selección 1



⊗ uno o otro, no ambos

⊗ Características del sistema  
4k x 8bits — Ancho de palabra: 8bits

— Cantidad de palabras:

$$4k = 2^{12} \text{ palabras}$$

⊗ El sistema debe ser de 2 bloques

1 bloque con direcciones impares

1 bloque con direcciones pares

impares: bit menos significativo en 1

pares: bit menos significativo en 0

Se van a requerir 2 bloques de 2k x 8bits

— la señal de address con el bit menos significativo selecciona que bloque está activo

→ se puede, pero mejor un decod