

Arquitectura de Computadoras

TP4 - Decodificación de Memoria

Ejercicios	1
Ejercicio 1	1
Ejercicio 2	1
Ejercicio 3	1
Ejercicio 4	2
Ejercicio 5	2
Ejercicio 6	2
Ejercicio 7	2
Bus de Control	2
Ejercicio 8	3
Ejercicio 9	4
Ejercicio 10	4
Ejercicio 11	4
Ejercicio 12	5
Ejercicio 13	5

Ejercicios

Ejercicio 1

- ¿Cuántas direcciones de memoria puede acceder un procesador genérico que tiene 16 líneas de bus de address y 8 de bus de datos? ¿Cuál es el ancho de la palabra?
- ¿Cuántas direcciones de memoria puede acceder un procesador genérico que tiene 16 líneas de bus de address y 16 de bus de datos? ¿Cuál es el ancho de la palabra?
- ¿Cuántas direcciones de memoria ocupa un dispositivo que tiene 10 líneas de bus de address y 8 de bus de datos?
- ¿Cuántas direcciones de memoria ocupa un dispositivo que tiene 10 líneas de bus de address y 16 de bus de datos?
- ¿Si un dispositivo de 1Kx8 debe mapearse a partir de la dirección 03FF, cómo se debe activar su Chip Select?

Ejercicio 2

Diseñar un circuito con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con **una** memoria ROM de 32K x 16. Conecte el procesador y el integrado de ROM y explique las decisiones tomadas en cuanto a la ubicación en el mapa de memoria y que sucede con las posiciones de memoria no ocupadas por la ROM.

Ejercicio 3

Diseñar un circuito con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con **dos** integrados de memoria ROM de 16K x 16. Se desea

correr en este sistema un programa de 32 KBytes de tamaño que comience en la dirección de memoria 0000h. Tenga presente las compuertas para resolver este ejercicio.

Ejercicio 4

Diseñar un circuito de una computadora, con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con dos tipos de integrados para las memorias RAM y ROM. Los chips son de 32K x 16 para ambas. Armar el sistema de tal manera que el sistema cuente con 32K x 16 de RAM y 32K x 16 de ROM.

Ejercicio 5

Igual que el ejercicio anterior, pero ahora las memorias son todas de 16K x 16, y se quieren mapear 32K x 16 de RAM y 16K x 16 de ROM.

Ejercicio 6

Igual que el ejercicio anterior, pero ahora las memorias son de 16K x 8. ¿Cómo conectaría las memorias para que para el procesador sea transparente?

Ejercicio 7

Al circuito del Ejercicio 4 se le desea adicionar un periférico que esté mapeado en la memoria. Posee 16 registros de 16 bits cada uno. ¿Qué es lo único que interpreta el procesador?

Bus de Control

Además del Bus de Direcciones y el Bus de Datos, hay un tercer Bus, llamado Bus de Control. Este Bus lleva la información digital de lo que deben hacer los distintos periféricos. Por ejemplo:

- R/^W: Esta línea del bus indica si se va a leer un dato o se va a escribir. Es decir, el dispositivo que esté activado (por ejemplo la Memoria) debe saber si debe tomar o escribir un dato en el bus de datos. Cuando esta línea está en 0, significa que el dispositivo activo debe escribir un dato en el bus de datos. Si está en 1, significa que debe guardar el dato que esté presente en el bus de datos.
- IO/^M: Existe el mapa de memoria, que es al que accede el procesador cada vez que ejecuta una instrucción de acceso a memoria con la instrucción **mov**. Pero también está el mapa de E/S. A este mapa se accede con las instrucciones IN/OUT. Esta línea está en 0 si se quiere acceder al mapa de memoria y 1 si se accede al mapa de entrada y salida.

Por ejemplo, cuando el procesador ejecuta la instrucción:

```
mov eax, [0x1000]
```

Está realizando la siguiente configuración en los buses del sistema:

- Bus de Control:
 - IO/^M: 0

- R/^W: 1
 - Bus de Direcciones: 0x1000
 - Bus de Datos: ?? lo que escriba el dispositivo activado por el circuito decodificador.
- Caso contrario, cuando se está guardando un dato en memoria:

```
mov eax, 0x1234 ;dentro del procesador, no afecta a los buses
mov [0x1100], eax ;afecta los buses
```

La configuración de los buses queda de la siguiente forma:

- Bus de Control:
 - IO/^M: 0
 - R/^W: 0
- Bus de Direcciones: 0x1100
- Bus de Datos: 0x1234

Cuando se desea leer un dato de un dispositivo mapeado en E/S. Y por ejemplo se ejecuta la instrucción

```
in ax, 70h
```

Se accede al dispositivo mapeado en la dirección 70h y se escribe el dato en el registro ax.

La configuración de los buses quedan de la siguiente forma:

- Bus de Control:
 - IO/^M: 1
 - R/^W: 1
- Bus de Direcciones: 0x70
- Bus de Datos: ?? Lo que se esté presentando por el dispositivo activado.

Se puede observar que para que un dispositivo esté en Memoria o en Entrada/Salida, basta decodificar correctamente haciendo uso de la línea IO/^M dentro del circuito decodificador. Recordar que antes de poder ejecutar una instrucción, ésta también debe traerse de memoria, realizando una instrucción equivalente a:

```
mov instruction, [eip]
```

Ejercicio 8

Se dispone de un microprocesador con 20 líneas de bus de address y 16 líneas de bus de datos.

Se desea utilizarlo para enviar y recibir datos a través de una red. Para esto se cuenta con una placa de RED de red con las siguientes características:

- 1 registro ST0, cuyo valor es 1 cuando hay datos esperando ser leídos.
- 1 registro ST1, del cual se pueden obtener los datos recibidos.

- 1 registro CR0, que se setea en 1 para enviar datos.
- 1 registro CR1, en el cual se colocan los datos a enviar.

Realice la decodificación completa de éste sistema. Mapee la placa de Red en el Mapa de Entrada y Salida.

Ejercicio 9

Explique de qué manera se debe modificar el sistema del ejercicio 8, si en lugar de tener un registro de envío y otro para recepción, la placa tuviese un buffer de 64 bytes mapeado en memoria

Realice la decodificación completa de este nuevo sistema y muestre el mapa de memoria.

Ejercicio 10

Únicamente se cuenta con Chips de 8K x 8, ¿Cómo se logran las siguientes configuraciones de memoria?

- 16K x 8
- 8K x 16
- 64K x 16
- 128K x 32
- 4K x 8
- 4K x 16

Ejercicio 11

Se tiene un procesador genérico (no es Intel) con 32 líneas de BA y 16 líneas de BD. Un programador necesita memoria suficiente para poder ubicar un programa de 1Gb pero también quiere que la RAM o las RAMs en la que esté ubicado dicho programa tengan la menor cantidad de direcciones posibles para que quepa dicho programa. Es decir que al ubicar el programa deben sobrar la menor cantidad de direcciones posibles en la RAM o las RAMs. Además nos dice que su programa comienza en la mitad del mapa de memoria. Se cuenta con las siguientes RAMs:

- 1Gx16
- 512Mx8
- 2Gx8

Se tiene libertad para elegir entre cualquiera de ellas y cualquier cantidad. Por último, se solicita ubicar una ROM de 1Gx16 al comienzo del mapa.

Indicar la RAM o las RAMs que habría que elegir y dibujar el mapa de memoria, diagrama y decodificación. Se puede elegir un único decodificador de entre los siguientes:

- 2 a 4

- 3 a 8
- 4 a 16

Ejercicio 12

Se está desarrollando un pequeño sistema para encriptar mensajes. El sistema debe soportar 9Kbytes para datos volátiles y 20Kbytes para el programa prefijado de fábrica. Además se desea mapear como dispositivo un integrado que tiene 2 registros de 16 bits en la dirección 0xC0DE. En el primer registro se encuentra la configuración y en la segunda se encuentran los datos pseudoaleatorios para encriptar.

Se cuenta con un microprocesador genérico de 16 bits de bus de datos y 16 de bus de address, RAM de 16K x 16 y ROM 8K x 8

- Dibuje el Mapa de Memoria y de Entrada /Salida
- Haga esquema de todo el circuito con todas las memorias y periféricos
- Haga esquema del circuito Decodificador

Ejercicio 13

Se dispone de un microprocesador del tipo Intel (tiene salida IO/M) de 8 bits de bus de datos de 32 bits de bus de direcciones. El dueño del proyecto nos brinda 2 (dos) módulos de memoria RAM, cada uno de ellos de 1G x 8 y nos pide que los ubiquemos a partir de la dirección 0h en forma contigua para lograr 2Gbytes de RAM.

Además tenemos que decodificar un modulo de ROM de 256M x 8 para el código que corre al encender el sistema, pero el único dato que tenemos es que el registro EIP al iniciarse el procesador toma el valor 0xFFFFFFF00. El dueño del proyecto nos avisa que su código ocupa 200Mbytes.

Por una cuestión de espacio y economía usted solo puede usar **hasta 3 decodificadores como máximo** para resolver la decodificación completa de RAM y ROM. Pueden ser decodificadores de 2 a 4, 3 a 8 y 4 a 16. No dispone de compuertas para la decodificación.

Se pide **en este orden** que:

1. Dibuje el mapa de memoria.
2. Dibuje un diagrama esquemático de los integrados utilizados para la RAM y ROM (líneas de datos, direcciones, etc.) Debe colocar nombres a todas las líneas de circuito utilizadas.
3. Resuelva la decodificación con estas condiciones y dibuje el circuito obtenido.
4. ¿Qué pedido le tiene que hacer usted al programador del código que irá en la ROM para que todo esto funcione correctamente al iniciarse el sistema.

Ejercicio 1

- ¿Cuántas direcciones de memoria puede acceder un procesador genérico que tiene 16 líneas de bus de address y 8 de bus de datos? ¿Cuál es el ancho de la palabra?
- ¿Cuántas direcciones de memoria puede acceder un procesador genérico que tiene 16 líneas de bus de address y 16 de bus de datos? ¿Cuál es el ancho de la palabra?
- ¿Cuántas direcciones de memoria ocupa un dispositivo que tiene 10 líneas de bus de address y 8 de bus de datos?
- ¿Cuántas direcciones de memoria ocupa un dispositivo que tiene 10 líneas de bus de address y 16 de bus de datos?
- ¿Si un dispositivo de 1Kx8 debe mapearse a partir de la dirección 03FF, cómo se debe activar su Chip Select?

16 BA: $2^{16} = \underbrace{2^{10}}_K \cdot \underbrace{2^6}_{64} = 64\text{K}$ con 8 de BD \rightarrow 1 byte a la vez // ancho palabra: 1 byte
// espacio de 64KB

16 BA: $2^{16} = \underbrace{2^{10}}_K \cdot \underbrace{2^6}_{64} = 64\text{K}$ con 16 de BD \rightarrow 2 bytes a la vez // ancho palabra: 2 bytes
// espacio de 128KB

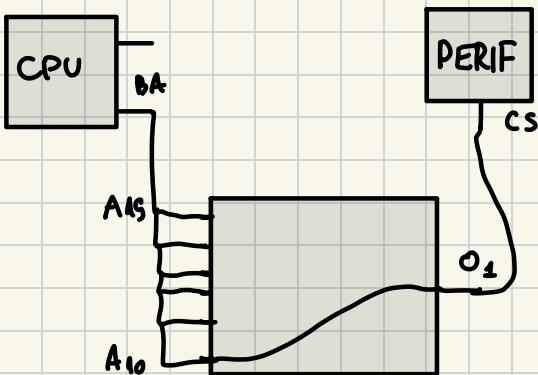
10 BA: $2^{10} = 1\text{K}$ y 8BD: \rightarrow 1 byte a la vez / ancho palabra: 1 byte / espacio de 1KB

10 BA: $2^{10} = 1\text{K}$ y 16BD: 2 bytes a la vez / ancho de palabra 2 bytes / espacio de 2KB

DISPOSITIVOS

$1\text{K} \times 8 \rightarrow 10\text{BA}$ y 8BD $\rightarrow 10\text{BA}: 2^{10} : 00\ 11\ 1111\ 1111 = 03FF$

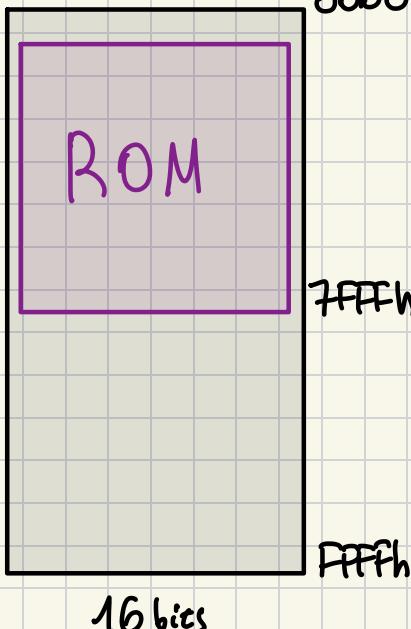
$03FF :$ <table border="0"> <tr> <td>0000</td> <td>0100</td> <td>0000</td> <td>0000</td> <td>b</td> </tr> <tr> <td colspan="4" style="background-color: #f0f0f0;"></td> <td>A₀</td> </tr> </table>	0000	0100	0000	0000	b					A ₀	$+ 0400$ $03FF$ <hr/> $07FF$
0000	0100	0000	0000	b							
				A ₀							
$0800 :$ <table border="0"> <tr> <td>0000</td> <td>0111</td> <td>1111</td> <td>1111</td> <td>b</td> </tr> <tr> <td colspan="4" style="background-color: #f0f0f0;"></td> <td>A₉</td> </tr> </table>	0000	0111	1111	1111	b					A ₉	A_9 A_0
0000	0111	1111	1111	b							
				A ₉							



Ejercicio 2

$$2^{16} = \frac{K}{2^{10}} \cdot \frac{64}{2^6} = 64K$$

Diseñar un circuito con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con una memoria ROM de 32K x 16. Conecte el procesador y el integrado de ROM y explique las decisiones tomadas en cuanto a la ubicación en el mapa de memoria y que sucede con las posiciones de memoria no ocupadas por la ROM.



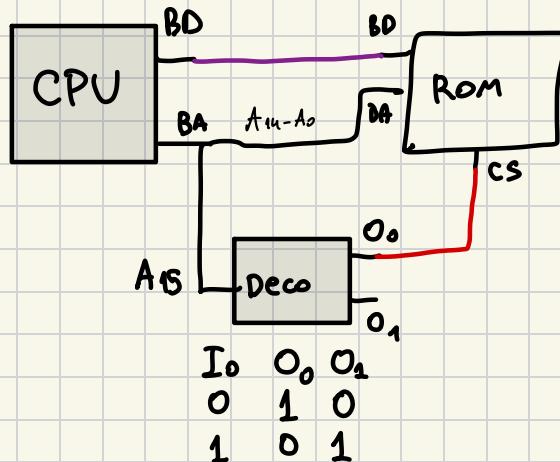
$$32K = 2^{10} \cdot 2^5 = 2^{15}$$

Mapa de memoria:

0111	1111	1111	1111
F	F	F	F

Start 0000 0000 0000 0000
end 0111 1111 1111 1111
next 1000 0000 0000 0000

A₁₅ A₁₄
A₁₅ = 0 → ROM ON
A₁₅ = 1 → ROM OFF



Ejercicio 3

Diseñar un circuito con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con dos integrados de memoria ROM de 16K x 16. Se desea

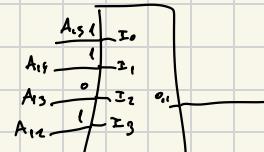
1/5

V20240416 2320

16BA 16BD

ROM 16K x 16 (x2)

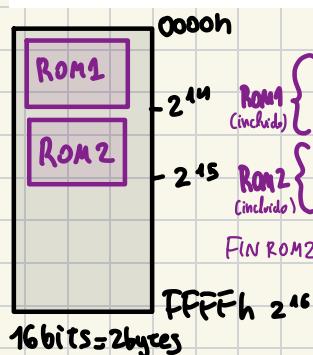
$$16K = 2^{10} \cdot 2^4 = 2^{14}$$



I ₃	I ₂	I ₁	I ₀
1	0	1	1

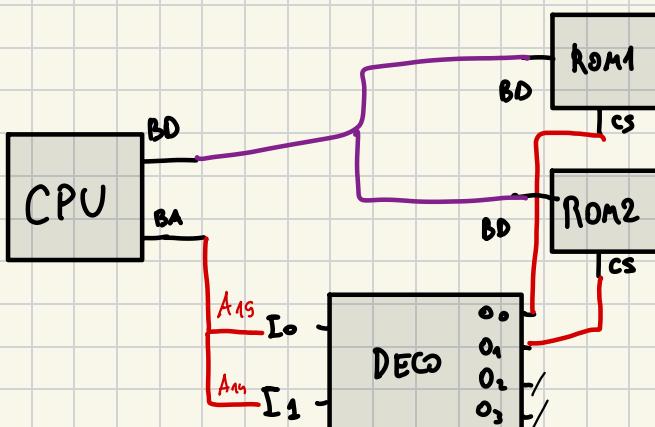
$2^0 \cdot 1 + 2^1 \cdot 1 + 2^2 \cdot 0 + 2^3 \cdot 1 = 11 \rightarrow O_{11}$

$$O: 2^0 \cdot 1 + 2^1 \cdot 1 + 2^2 \cdot 0 + 2^3 \cdot 1 = 11 \rightarrow O_{11}$$



necesito A₁₅ = 0 → ROM1
A₁₄ = 0 → ROM2

$$\begin{aligned} & A_{14} \quad A_{13} \\ & I_1 \quad I_0 \quad \text{ROM1 Output} = 0 \cdot 2^0 + 0 \cdot 2^1 = 0 (0_0) \\ & 2^1 \quad 2^0 \\ & I_3 \quad I_2 \quad I_1 \quad I_0 \quad \text{ROM2 Output} = 1 \cdot 2^0 + 0 \cdot 2^1 = 1 (0_1) \end{aligned}$$

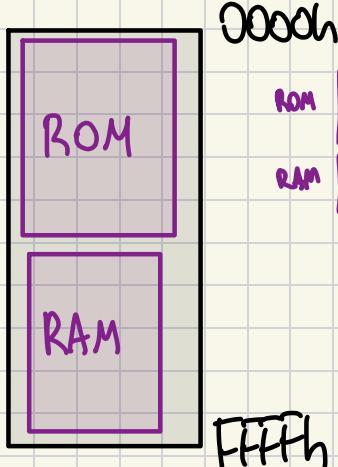


Ejercicio 4

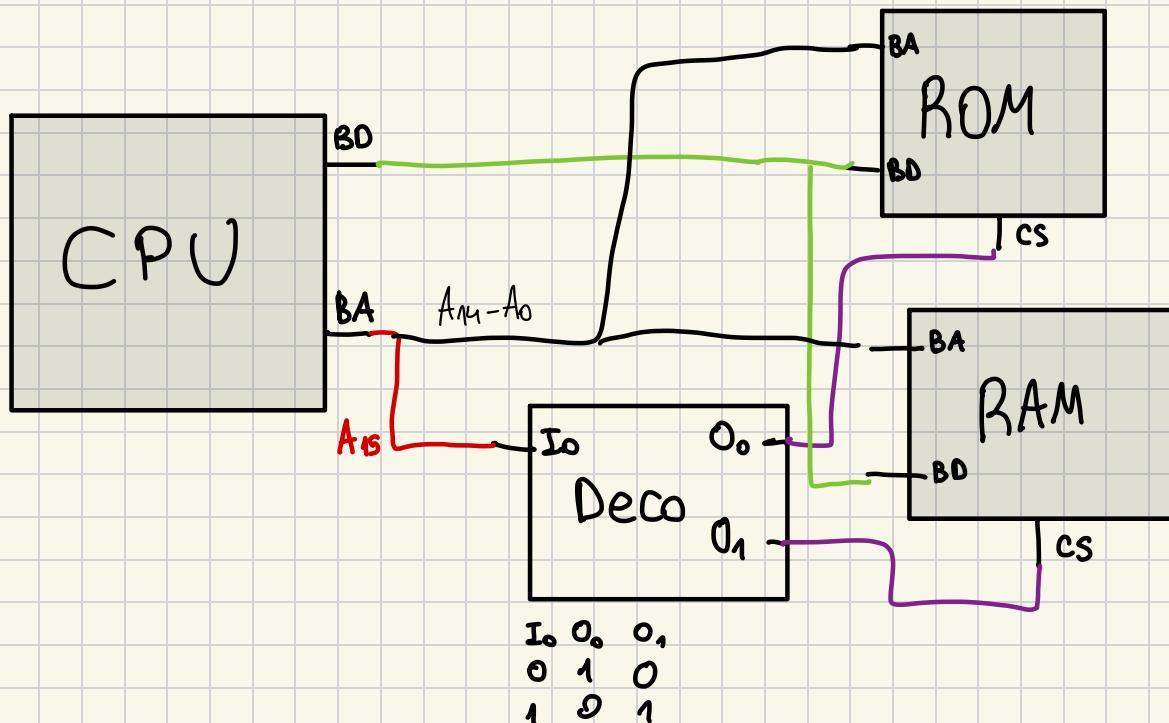
Diseñar un circuito de una computadora, con un procesador de 16 líneas de bus de address y 16 líneas de bus de datos. Se cuenta con dos tipos de integrados para las memorias RAM y ROM. Los chips son de 32K x 16 para ambas. Armar el sistema de tal manera que el sistema cuente con 32K x 16 de RAM y 32K x 16 de ROM.

$$\begin{array}{l} BA \text{ 16 } BD \text{ 16 } \\ \text{RAM } 32K \times 16 \quad 32K = 2^{10} \quad 32 = 2^5 \\ \text{RAM } 32K \times 16 \end{array}$$

$$k \quad 32 \\ 2^{10} \quad 2^5 \\ 32K = 2^{10} \cdot 2^5 = 2^{15}$$



FFFFh



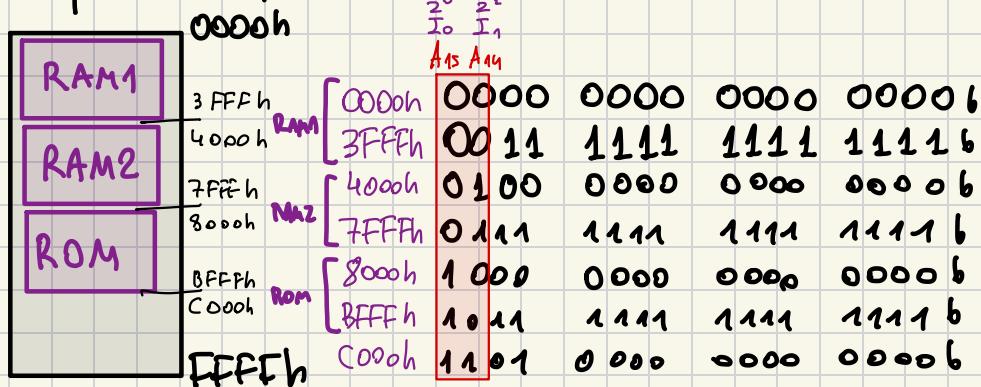
Ejercicio 5

Igual que el ejercicio anterior, pero ahora las memorias son todas de 16K x 16, y se quieren mapear 32K x 16 de RAM y 16K x 16 de ROM.

$$BA \cdot 16 = 2^{16} = 2^{\frac{16}{10}} \cdot 2^{\frac{64}{6}} = 64K \\ \downarrow \\ 16K \cdot 4 \quad \text{Smiley Face}$$

$16K \times 16$ RAM $16K \times 16$ ROM $16K = 2^{\frac{16}{10}} \cdot 2^4 = 2^{14}$

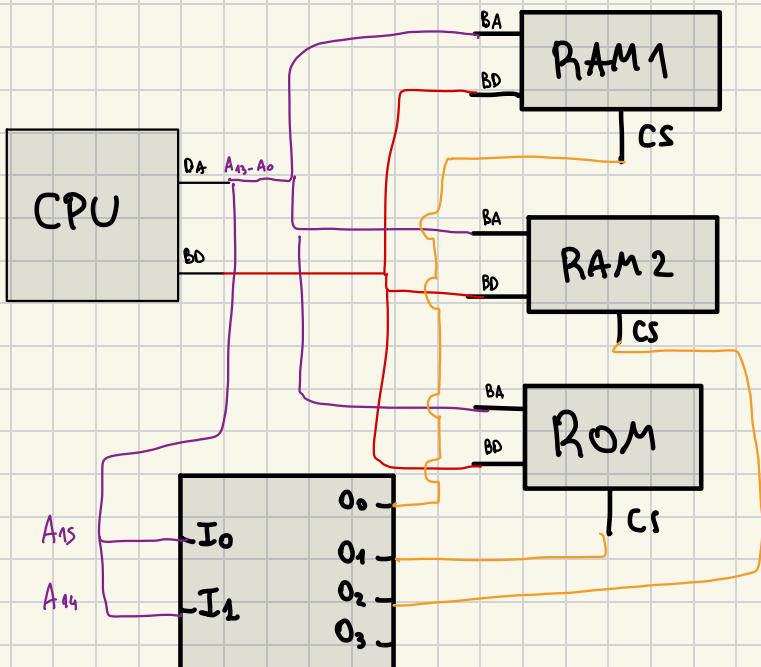
quiero mapear $32K \times 16$ de RAM \rightarrow uso 2 RAM de $16K \times 16$



$$A_{15} = 0 \text{ y } A_{14} = 0 \rightarrow \text{RAM1: } 2^0 \cdot 0 + 2^1 \cdot 0 = 0 \rightarrow O_0$$

$$A_{15} = 0 \text{ y } A_{14} = 1 \rightarrow \text{RAM2: } 2^0 \cdot 0 + 2^1 \cdot 1 = 2 \rightarrow O_2$$

$$A_{15} = 1 \text{ y } A_{14} = 0 \rightarrow \text{ROM: } 2^0 \cdot 1 + 2^1 \cdot 0 = 1 \rightarrow O_1$$



Ejercicio 6

Igual que el ejercicio anterior, pero ahora las memorias son de 16K x 8. ¿Cómo conectaría las memorias para que para el procesador sea transparente?

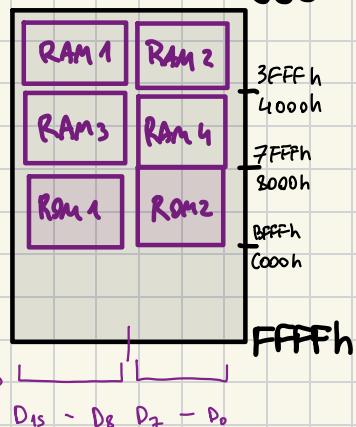
$$BA = 16 = 2^{16} \Rightarrow FFFFh$$

$16k = 2^{10} \cdot 2^4 \Rightarrow \frac{1}{4}$ de las dir. totales
 8BD \rightarrow 1 byte
 pero tengo palabra (cuad) de 2 bytes

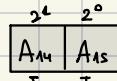
quiero mapear 32k de RAM; Todas memorias 16k x 8

2bytes

0000h



	A ₁₅ A ₁₄	RAM1	RAM2	RAM3	RAM4	ROM1	ROM2
0000h	0000	0000	0000	0000	0000	1000	1000
3FFFh	0011	1111	1111	1111	1111	1000	1011
4000h	0100	0000	0000	0000	0000	1000	1100
7FFFh	0111	1111	1111	1111	1111	1000	1111
8000h	1000	0000	0000	0000	0000	1000	1111
BFFFh	1011	1111	1111	1111	1111	1011	1111
C000h	1100	0000	0000	0000	0000	1100	1111
FFFFh	1111	1111	1111	1111	1111	1111	1111



$$A_{15} = 0 \Rightarrow \text{output} = O_0$$

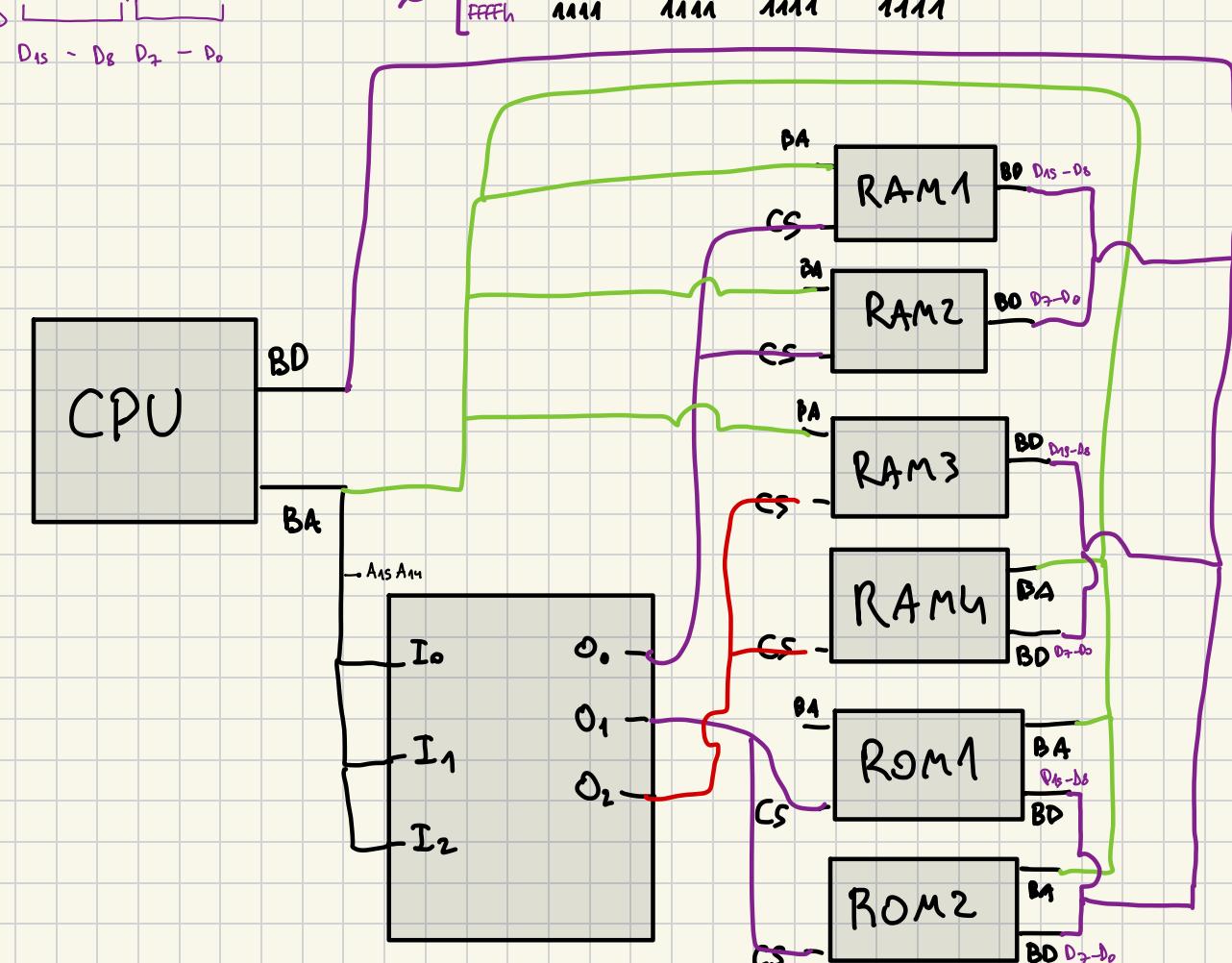
$$A_{14} = 0$$

$$A_{15} = 0 \Rightarrow \text{output} = 2^0 \cdot 0 + 2^4 \cdot 1 = 2 \cdot O_2$$

$$A_{14} = 1$$

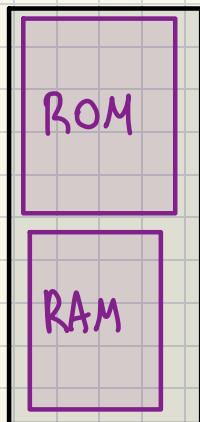
$$A_{15} = 1 \Rightarrow \text{output} = 2^0 \cdot 1 + 2^1 \cdot 0 = 1 \cdot O_1$$

$$A_{14} = 0$$



Ejercicio 7

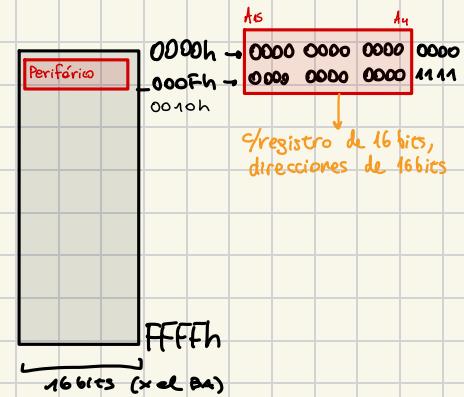
Al circuito del Ejercicio 4 se le desea adicionar un periférico que esté mapeado en la memoria. Posee 16 registros de 16 bits cada uno. ¿Qué es lo único que interpreta el procesador?



	A ₁₅	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000
ROM	0	0111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111
RAM	1	1000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000
		1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111
		FFFF6	80006	FFFF6	FFFF6

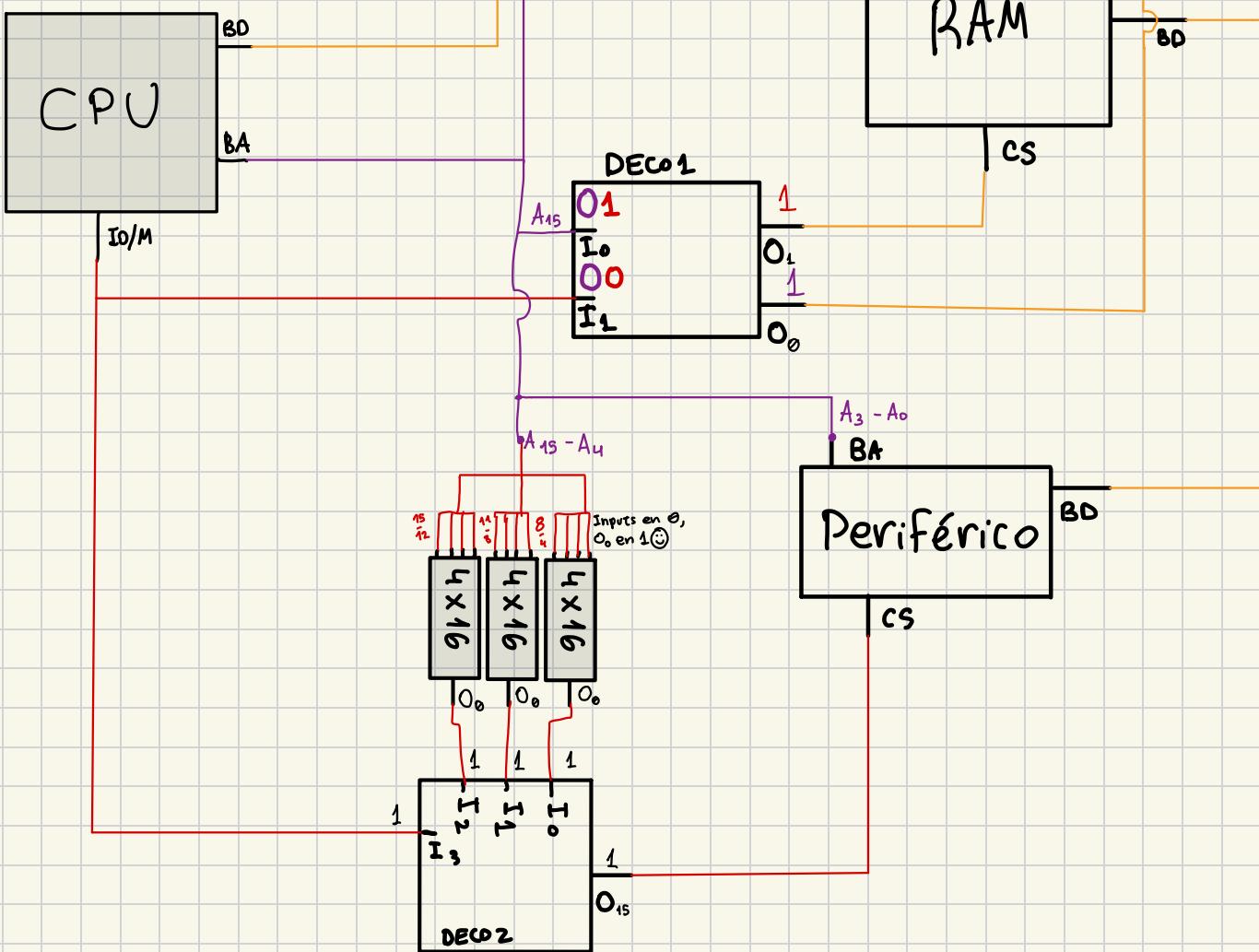
$$A_{15} = 0 \rightarrow \text{ROM} \checkmark \\ A_{15} = 1 \rightarrow \text{RAM} \times$$

FFFFh

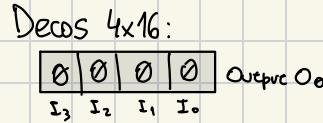


/WR (Cuando vale cero hay una escritura) 0 write
/RD (Cuando vale cero hay una lectura) 0 read

IO/M (Si vale 1: operaciones con ports, si vale 0: operaciones con la memoria)



$$\text{Output: } 2^0 \cdot 1 + 2^1 \cdot 1 + 2^2 \cdot 1 + 2^3 \cdot 1 = 15$$



Deco 1:

I ₀	A ₁₅
I ₁	I ₀

$$A_{15} = 0 \rightarrow \text{ROM Output} = 2^0 \cdot 0 + 2^1 \cdot 0 = O_0 \\ I_{0,M} = 0$$

$$A_{15} = 1 \rightarrow \text{RAM Output} = 2^0 \cdot 1 + 2^1 \cdot 0 = O_1 \\ I_{0,M} = 0$$

Ejercicio 8

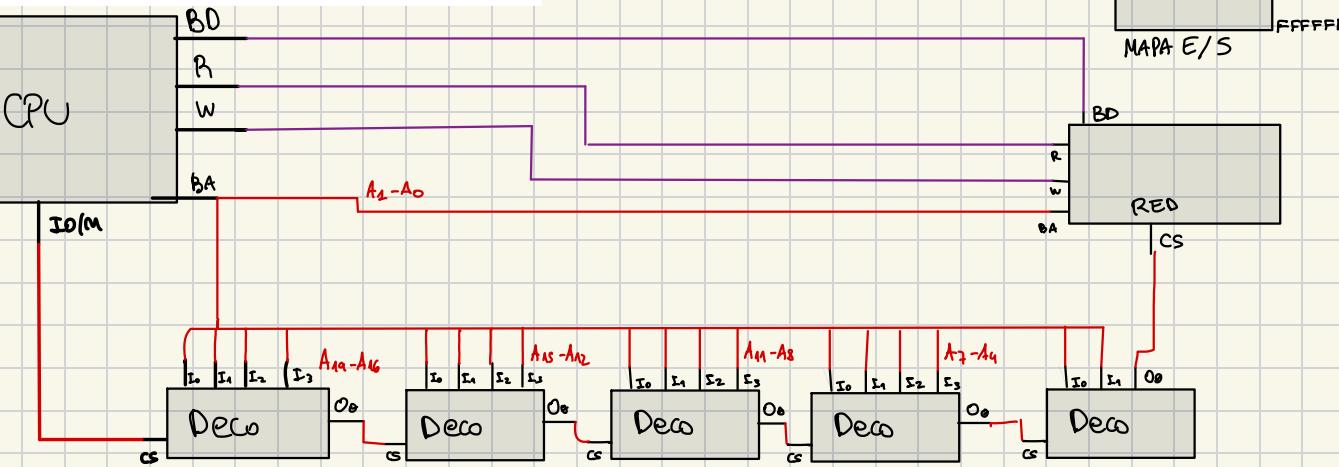
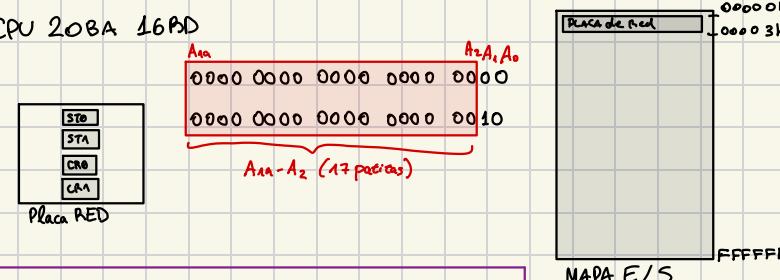
Se dispone de un microprocesador con 20 líneas de bus de address y 16 líneas de bus de datos.

Se desea utilizarlo para enviar y recibir datos a través de una red. Para esto se cuenta con una placa de RED de red con las siguientes características:

- 1 registro ST0, cuyo valor es 1 cuando hay datos esperando ser leídos.
- 1 registro ST1, del cual se pueden obtener los datos recibidos.
- 1 registro CR0, que se setea en 1 para enviar datos.
- 1 registro CR1, en el cual se colocan los datos a enviar.

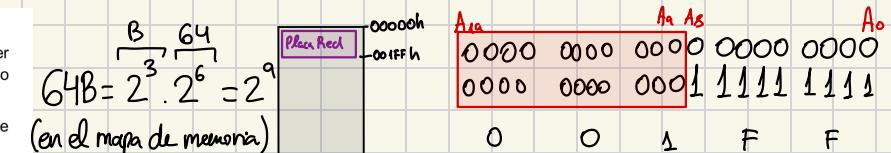
Realice la decodificación completa de este sistema. Mapee la placa de Red en el Mapa de Entrada y Salida.

CPU 20BA 16BD



Ejercicio 9

Explique de qué manera se debe modificar el sistema del ejercicio 8, si en lugar de tener un registro de envío y otro para recepción, la placa tuviese un buffer de 64 bytes mapeado en memoria

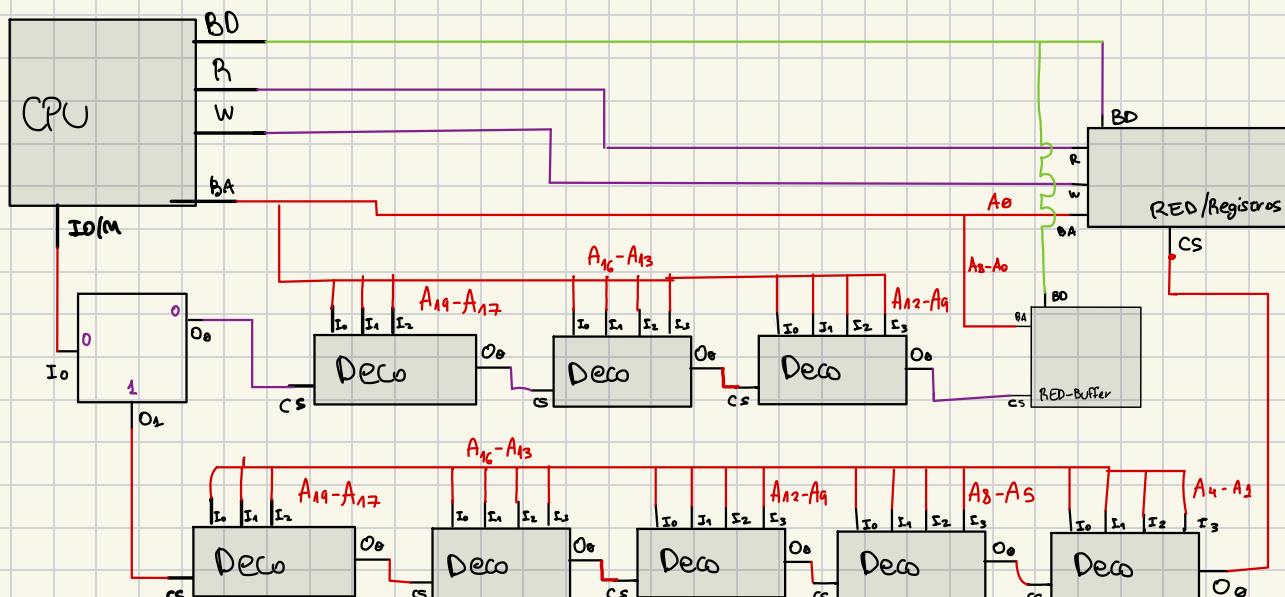
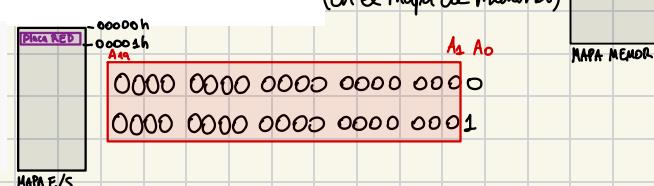


Realice la decodificación completa de este nuevo sistema y muestre el mapa de memoria.

- el término "mapeado en memoria" significa "mapeado en el mapa de memoria estandar" (NO el de e/s)
- Cuando se refiere al mapa de entrada/salida, se indica explícitamente "entrada/salida".

Por ello, tenés que dibujar ahora 2 mapas de memoria separados

- mapa de memoria ----> para el buffer
- mapa de entrada/salida ----> para los 2 registros
- utilizar iomen para direccionar a cada uno de ellos.



Ejercicio 11

32 BA \rightarrow 32 bits \rightarrow 4 bytes

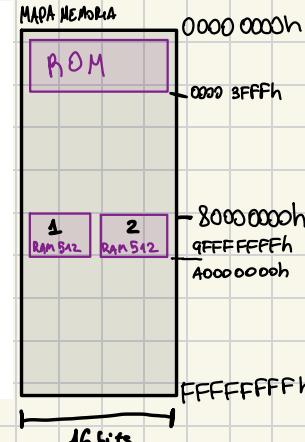
Se tiene un procesador genérico (no es Intel) con 32 líneas de BA y 16 líneas de BD. Un programador necesita memoria suficiente para poder ubicar un programa de 1Gb pero también quiere que la RAM o las RAMs en la que esté ubicado dicho programa tengan la menor cantidad de direcciones posibles para que quepa dicho programa. Es decir que al ubicar el programa deben sobrar la menor cantidad de direcciones posibles en la RAM o las RAMs. Además nos dice que su programa comienza en la mitad del mapa de memoria. Se cuenta con las siguientes RAMs:

- 1Gx16 1G de direcciones y 16 bits \times dir. \Rightarrow 2G de espacio 2bytes
- 512Mx8 512M de dir. y 8bits xdir \Rightarrow 512M de espacio
- 2Gx8 \rightarrow 2G de espacio

Se tiene libertad para elegir entre cualquiera de ellas y cualquier cantidad. Por último, se solicita ubicar una ROM de 1Gx16 al comienzo del mapa.

Indicar la RAM o las RAMs que habrá que elegir y dibujar el mapa de memoria, diagrama y decodificación. Se puede elegir un único decodificador de entre los siguientes:

- 2 a 4
- 3 a 8
- 4 a 16



ROM $1G \times 16 : 1G = 1.1K.K = 2^{30}$ bytes y $1/2$

2 bytes

0000 0000h rango de

3FFF FFFFh direcciones

BA: 30 bytes

A₃₁ A₃₀

0000 0000 0000 0000 0000 0000 0000 0000

0011 1111 1111 1111 1111 1111 1111 1111

16bytes RAM 512Mx8 : $512M = 2^9 \cdot 2^10 \cdot K = 2^{29} \rightarrow 2^{29} \cdot 1/2$

80000000h Start
9FFFFFFFh end
A000 0000h next

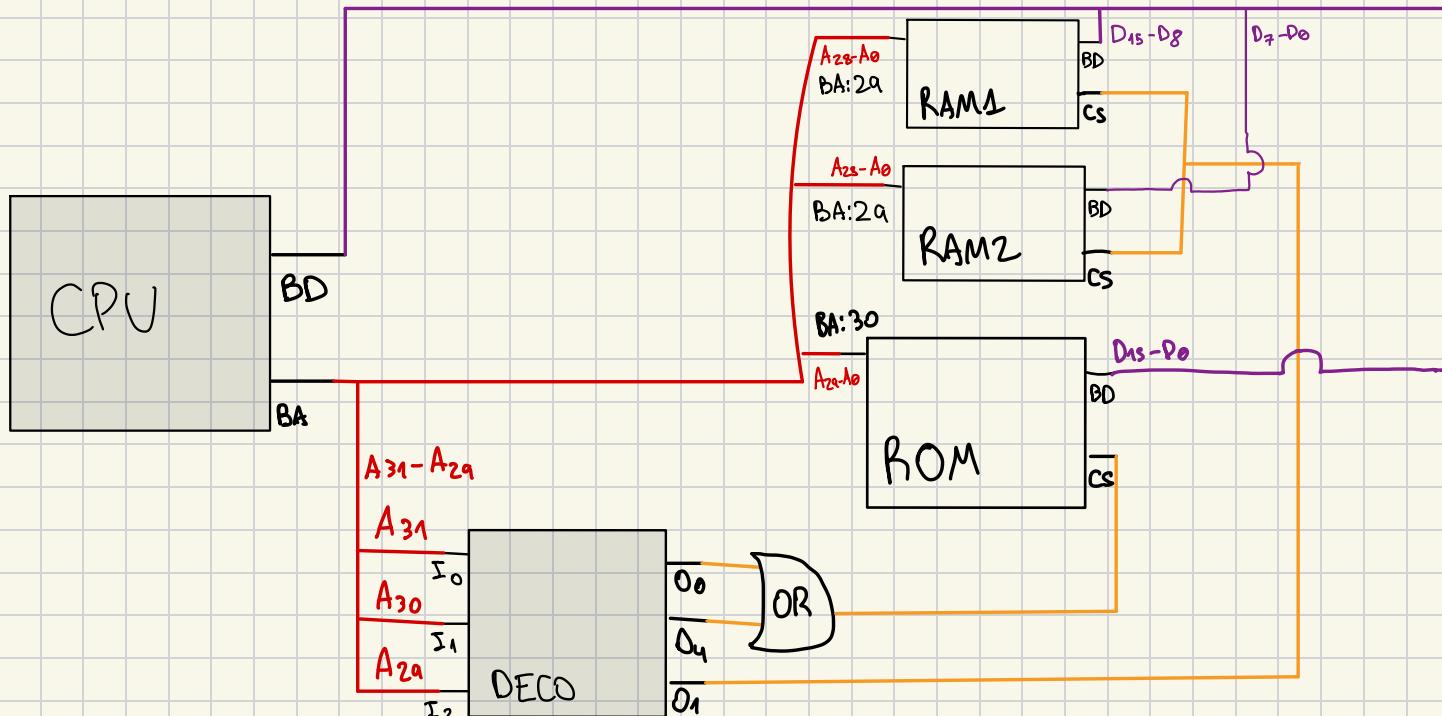
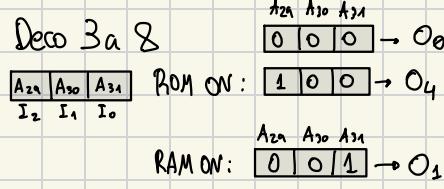
00000000h rango de

1FFFFFFFh direcciones

A₃₁

A₂₉

1000 0 0 0 0 0 0 0
1001 F F F F F F F F

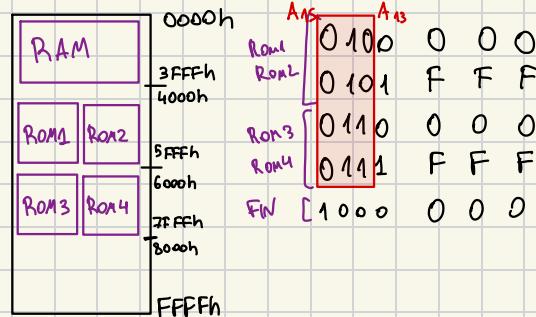


Ejercicio 12

Se está desarrollando un pequeño sistema para encriptar mensajes. El sistema debe soportar 9Kbytes para datos volátiles y 20Kbytes para el programa prefijado de fábrica. Además se desea mapear como dispositivo un integrado que tiene 2 registros de 16 bits en la dirección 0xC0DE. En el primer registro se encuentra la configuración y en la segunda se encuentran los datos pseudoaleatorios para encriptar.

Se cuenta con un microprocesador genérico de 16 bits de bus de datos y 16 de bus de address, RAM de 16K x 16 y ROM 8K x 8

- Dibuja el Mapa de Memoria y de Entrada /Salida
- Haga esquema de todo el circuito con todas las memorias y periféricos
- Haga esquema del circuito Decodificador



Decod1: [A15 A14 A13]

ROM1 & 2 ON: A15 = 0
A14 = 1
A13 = 0

RAM ON:

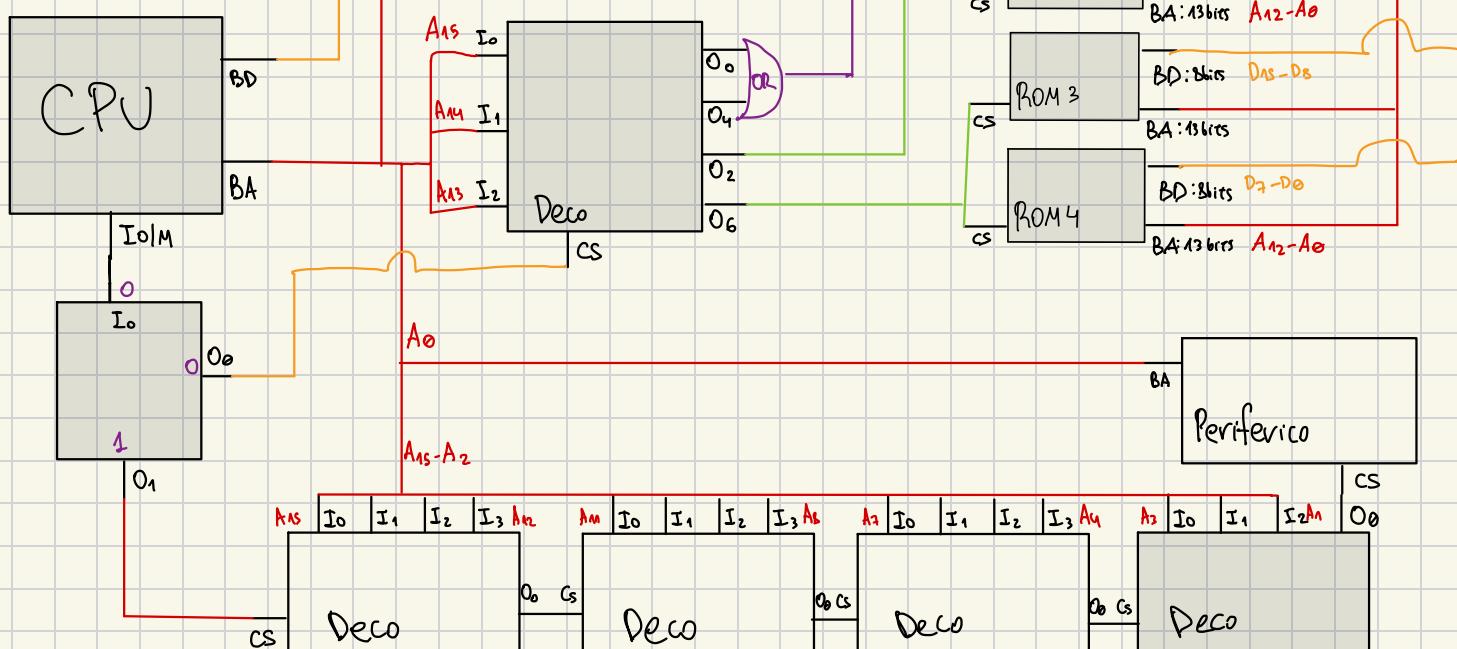
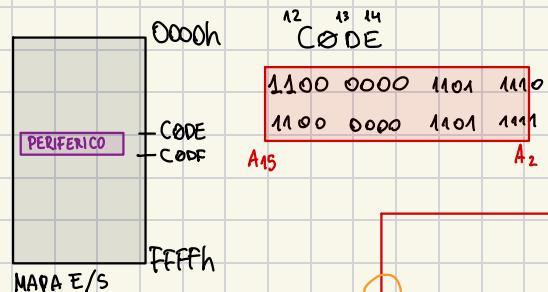
A15 = 0
A14 = 0
A13 = 1

A15 = 0
A14 = 1
A13 = 2

ROM: 0000 0 0 0 ?
0001 F F F ?
L 1FFFh

ROM: 0100 0 0 0 ?
0101 F F F ?
L 5FFFh

ROM: 0100 0 0 0 Start
0101 F F F end → next 0100 0 0 0: 6000h



ROM 8K x 8 : necesita 3 ROMS // $8K = 2^3 \cdot 2^{10} = 2^{13}$ → 13 bits BA

1 byte

RAM 16K, 16 bits: 32k espacio: necesita 1 RAM // $16K = 2^4 \cdot 2^{10} = 2^{14}$ BA 16 bits

RAM: 0000 0 0 0 Start
0001 F F F end → next 0100 0 0 0: 4000h
L 3FFFh

ROM: 0000 0 0 0 ?
0001 F F F ?
L 1FFFh

ROM: 0100 0 0 0 ?
0101 F F F ?
L 5FFFh

ROM: 0100 0 0 0 Start
0101 F F F end → next 0100 0 0 0: 6000h