

Ejercicio 1:

Dados los siguientes bloques de memoria:

- | | |
|-----------------|-----------------|
| 1) 8Kbyte | 5) 16K nibble |
| 2) 256 x 16bits | 6) 32Mbyte |
| 3) 2Kbits | 7) 16K x 32bits |
| 4) 4K x 4bits | 8) 1024Kbyte |

Se pide:

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

$$1) 8K \text{ bytes} = 2^3 \cdot 2^{10} \cdot 2^2 = 2^{15} = 32K \text{ bits}$$

$$2) 256 \times 16 \text{ bits} = 4K \text{ bits}$$

$$3) 2K \text{ bits}$$

$$4) 4K \times 4 \text{ bits} = 2^{12} \cdot 2^2 = 2^{14} = 16K \text{ bits}$$

$$5) 16K \text{ nibble} = 2^9 \cdot 2^{10} \cdot 2^2 = 2^{16} = 64K \text{ bits}$$

$$6) 32M \text{ byte} = 2^5 \cdot 2^{20} \cdot 2^2 = 2^{27} = 128M \text{ bits}$$

$$7) 16K \times 32 \text{ bits} = 2^9 \cdot 2^{10} \cdot 2^5 = 2^{19} = 512M \text{ bits}$$

$$8) 1024K \text{ byte} = 2^{10} \cdot 2^{10} \cdot 2^2 = 2^{22} = 4M \text{ bits}$$

$$a) 2K \text{ bits}$$

$$b) 256 \times 16 \text{ bits}$$

4K bits
16K bits
32K bits
64K bits
4M bits
128M bits
512M bits +

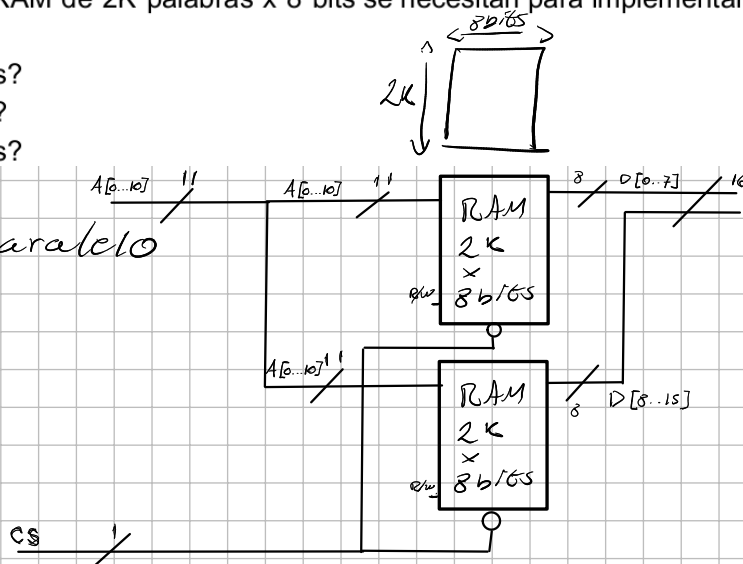
2K bits
4K x 4 bits
8K bytes
16K nibble
16K x 32 bits
1024K byte
32M byte

Ejercicio 2:

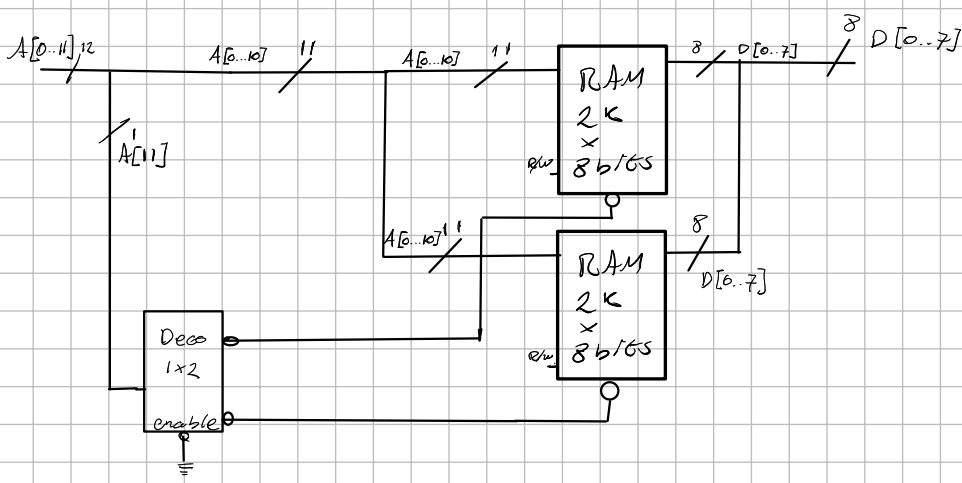
Cuántos "chip" de memoria RAM de 2K palabras x 8 bits se necesitan para implementar un banco de memoria de:

- 2K palabras de 16 bits?
- 4K palabras de 8 bits?
- 4K palabras de 16 bits?

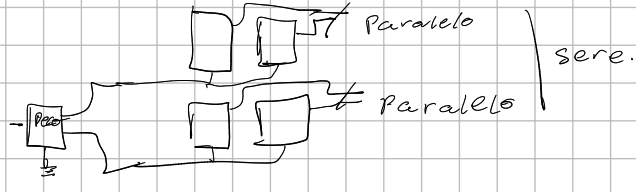
A) 2 bloques en paralelo



B. 4K palabras de 8 bits? 2 en Serie.



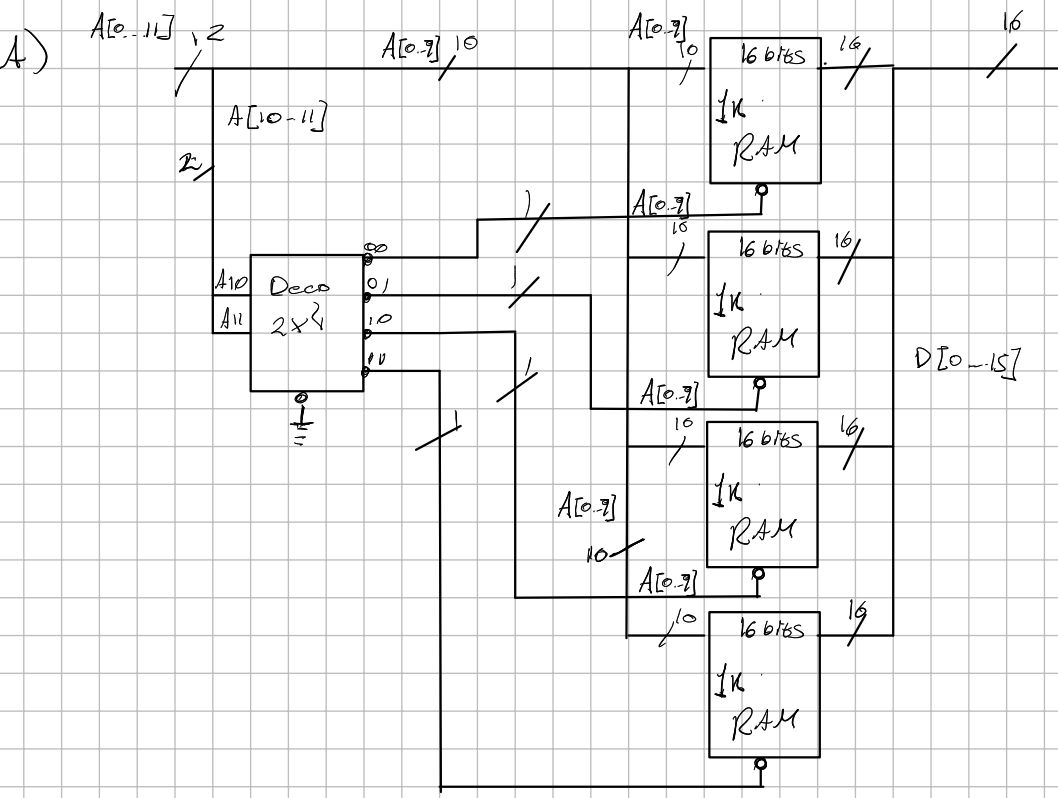
C. 4K palabras de 16 bits? Se necesitan 4 bloques / 2 bloques de 2 Ram en paralelo y luego en serie.



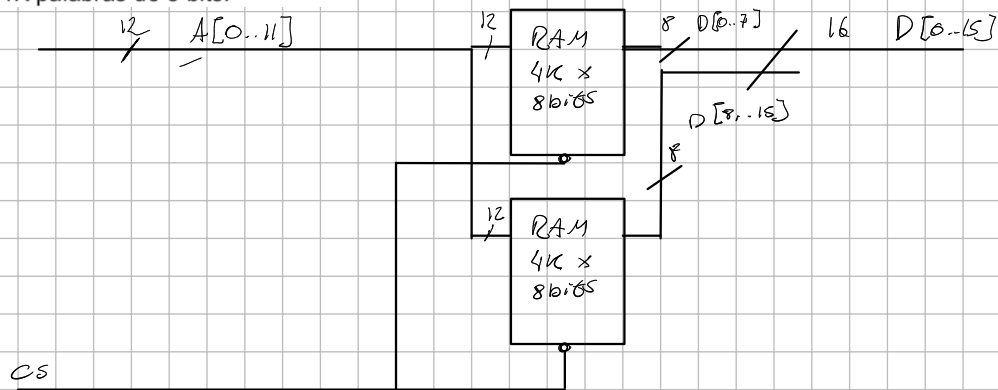
Ejercicio 3:

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

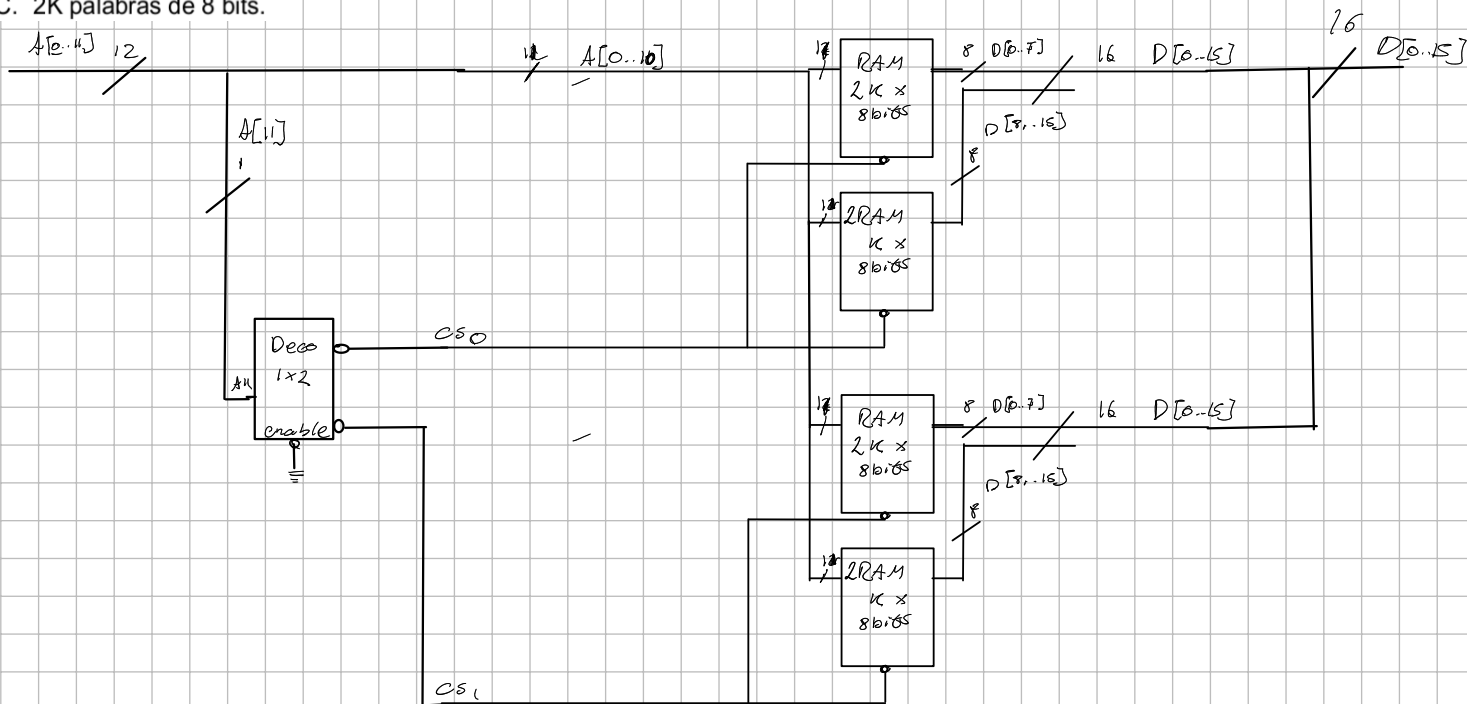
- A. 1K palabras de 16 bits.
- B. 4K palabras de 8 bits.
- C. 2K palabras de 8 bits.



B. 4K palabras de 8 bits.



C. 2K palabras de 8 bits.

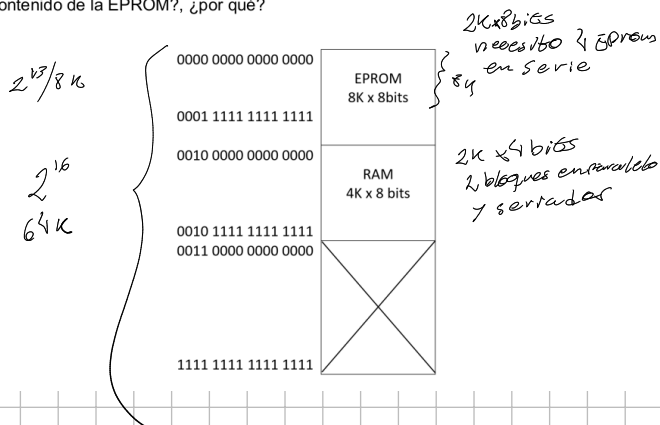


Ejercicio 4:

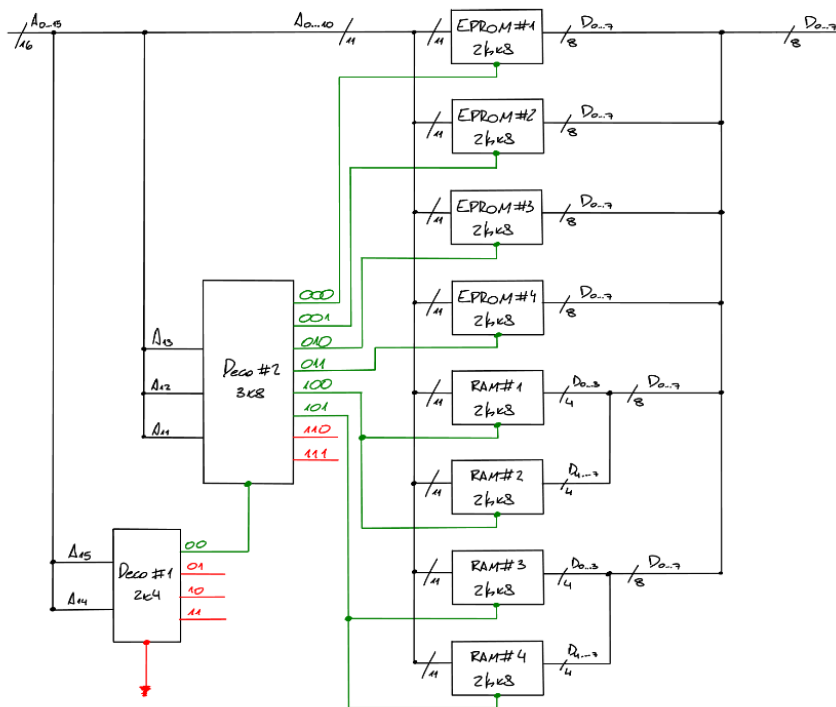
Construir un sistema de memoria como el que se muestra en el mapa de memoria de la figura. Se dispone para su implementación con los siguientes "chip" de memoria: EPROM de 2K x 8 bits y RAM de 2K x 4 bits.

A. Realizar una implementación que NO genere posiciones imagen en el espacio no implementado.

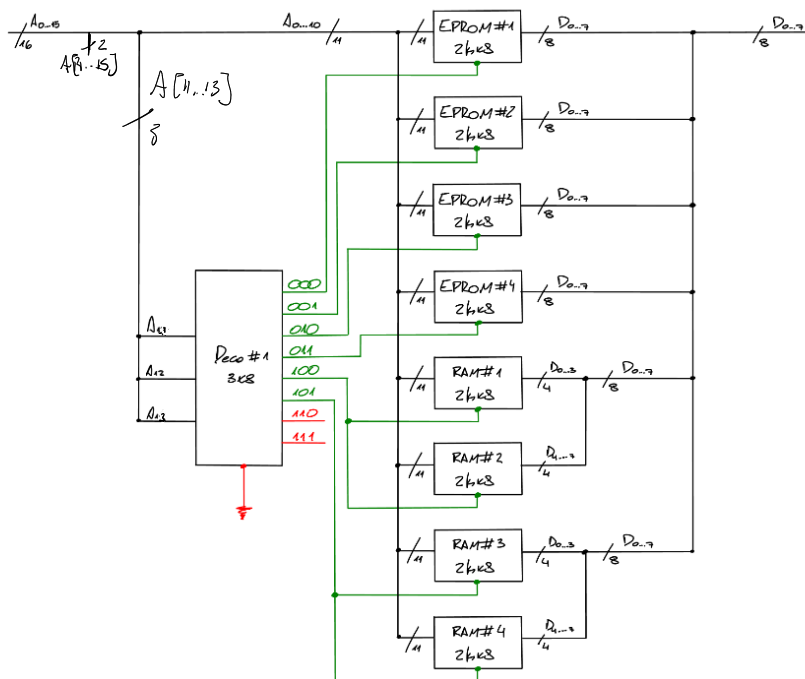
B. Realizar una implementación en la cual se generen posiciones imagen del contenido de la EPROM y la RAM a lo largo de todo el espacio direccionable. Analizar: ¿cuántas veces se replica el contenido de la RAM? y ¿cuántas veces se replica el contenido de la EPROM?, ¿por qué?



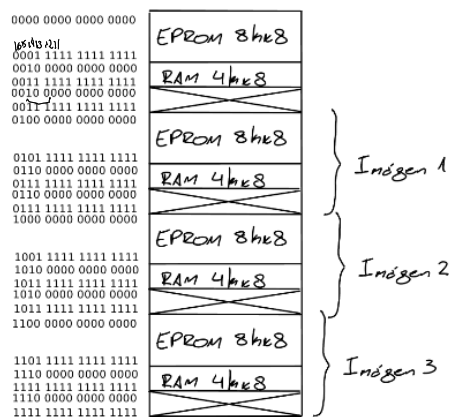
A)



B) Para hacer una implementación que genere posiciones imagen, debo eliminar el decodificador #1 de la implementación anterior:



Como A_{15} y A_{14} no están codificados, en este caso se generan 3 posiciones imagen que se suman a la implementación que hicimos. El mapa de memoria es el siguiente:



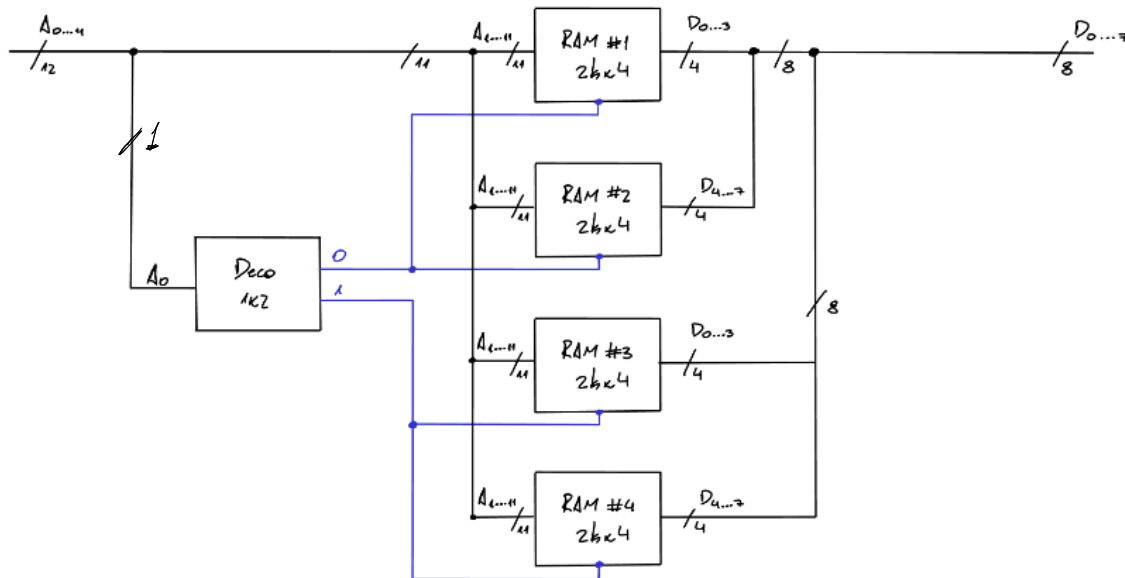
Ejercicio 5:

"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.

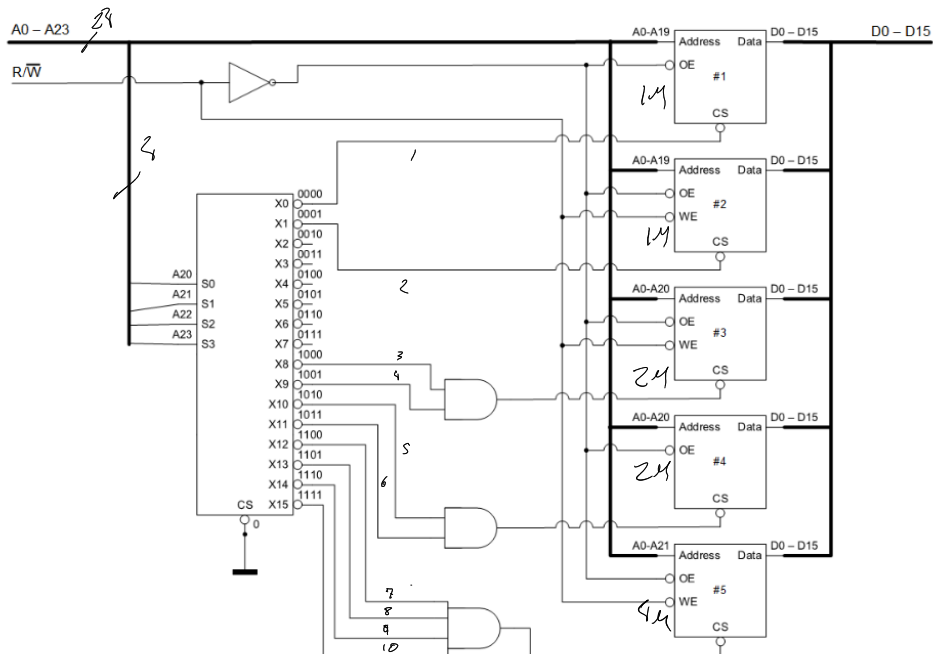
Recordar que en el ejercicio 4 se podía implementar una RAM 4Kx8 a partir de chips 2Kx4.

En este caso, para llevar a cabo la técnica interleaved memory, me alcanza con dar vuelta los cables de selección, es decir que el decodificador elija entre los bits menos significativos:



Ejercicio 6:

Basados en el sistema de memoria mostrado en la figura.

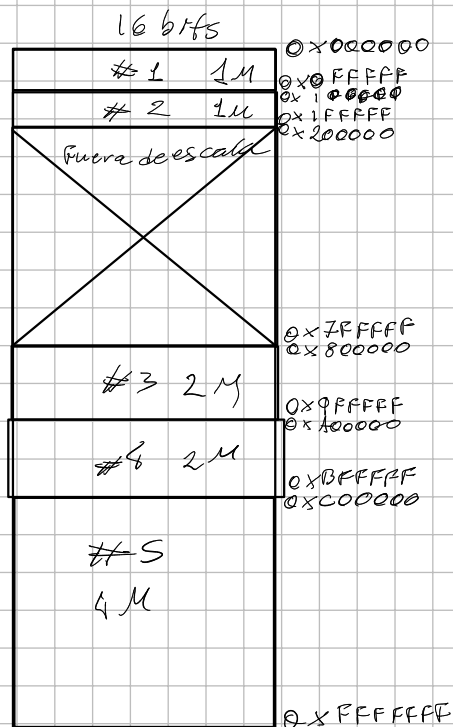


Se pide:

- Calcular el máximo espacio direccionable por el procesador expresado en palabras de 16 bits.
- Desarrollar el mapa de direcciones implementado indicando el inicio y final de cada bloque de memoria.
- Indicar en qué bloque se encuentran las siguientes direcciones:
 - 0x0654321
 - 0x0ABCDEF
 - 0x0FEDCBA
 - 0x0123456
 - 0x2000000

- Diseñar con compuertas lógicas un circuito que proporcione un "1" a la salida cuando la dirección generada por el procesador no esté en el mapa de memoria implementado.
- ¿Esta implementación genera posiciones imagen de algún bloque de memoria?, ¿Por qué?

A) El sistema de memoria tiene 24 cables de direccionamiento, por lo tanto puede direccionar 16 megas de 16 palabras, de los cuales solo 10 se están utilizando.



B) Tanto la salida del decodificador como el *chip select* de las memoria son activos por bajo, esto se indica mediante el círculo que hay en el pin de salida del decodificador y en el pin de entrada del chip select de las memorias. Esto implica que, por ejemplo, cuando los address A23, A22, A21 y A20 sean cero, se activará la salida X0. Es decir, habrá un cero en esta salida y por lo tanto el bloque #1 de memoria se activará.

Dado que al decodificador están conectadas las salidas A23, A22, A21 y A20, cada una habilitará bloques de 1 Mega palabras. En caso en que se quiera habilitar bloques más grandes, se deberán utilizar más de una salida.

El chip #1 tiene conectados los address del A0 al A19 es decir, 20 bits. Y dado que está conectado a la salida cero del decodificador (X0), este chip se va a habilitar cuando A23, A22, A21 y A20 sean ceros. Por lo tanto la dirección de la primer posición de memoria será la 0x000000 y la última 0x0FFFFF.

El siguiente chip (el #2) tiene el mismo tamaño pero está conectada a la salida uno del decodificador (X1), por lo tanto, el chip se habilita cuando las señales [A23, A22, A21, A20] = 0001. La dirección de la primer dirección de memoria será la 0x100000 y la última 0x1FFFFF.

El chip #3 cuenta con 21 señales de dirección, de la A0 a la A20 (21 bits), es decir, la capacidad de direccionamiento es de 2 Mega palabras. El *chip select* de este bloque se conecta a dos salidas del decodificador, las X8 y X9, mediante una compuerta AND. Dado que las salidas del decodificador son activas por bajo, si la salida de alguna de ellas vale cero, la salida de la compuerta AND será cero y por lo tanto se habilitará el chip #3 de memoria. La primer posición de memoria de este bloque será la 0x800000 y la última será la 0x9FFFFF.

La conexión del chip #4 es similar a la anterior, pero la dirección de la primer posición es la 0xA00000 y la última la 0xBFFFFF.

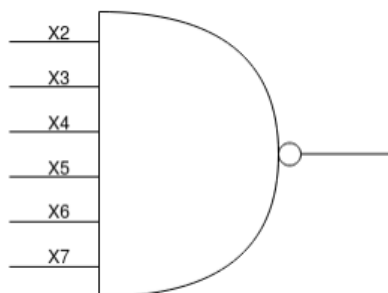
Finalmente, las señales de direccionamiento que están conectadas al último chip de memoria (#5) son las A0 a la A21, es decir, 22 bits. Por lo tanto, este bloque de memoria tiene una capacidad de 4 Mega palabras. Al *chip select*, hay conectadas cuatro salidas del decodificador mediante una compuerta AND. Por lo tanto, la primer posición de memoria es la 0xC00000 y la última la 0xFFFFF.

C. Indicar en qué bloque se encuentran las siguientes direcciones:

- 0x0654321 *no implementado*
- 0x0ABCDEF *#4*
- 0x0FEDCBA *#5*
- 0x0123456 *#2*
- 0x2000000 *Fuera del espacio*

D. Diseñar con compuertas lógicas un circuito que proporcione un "1" a la salida cuando la dirección generada por el procesador no esté en el mapa de memoria implementado.

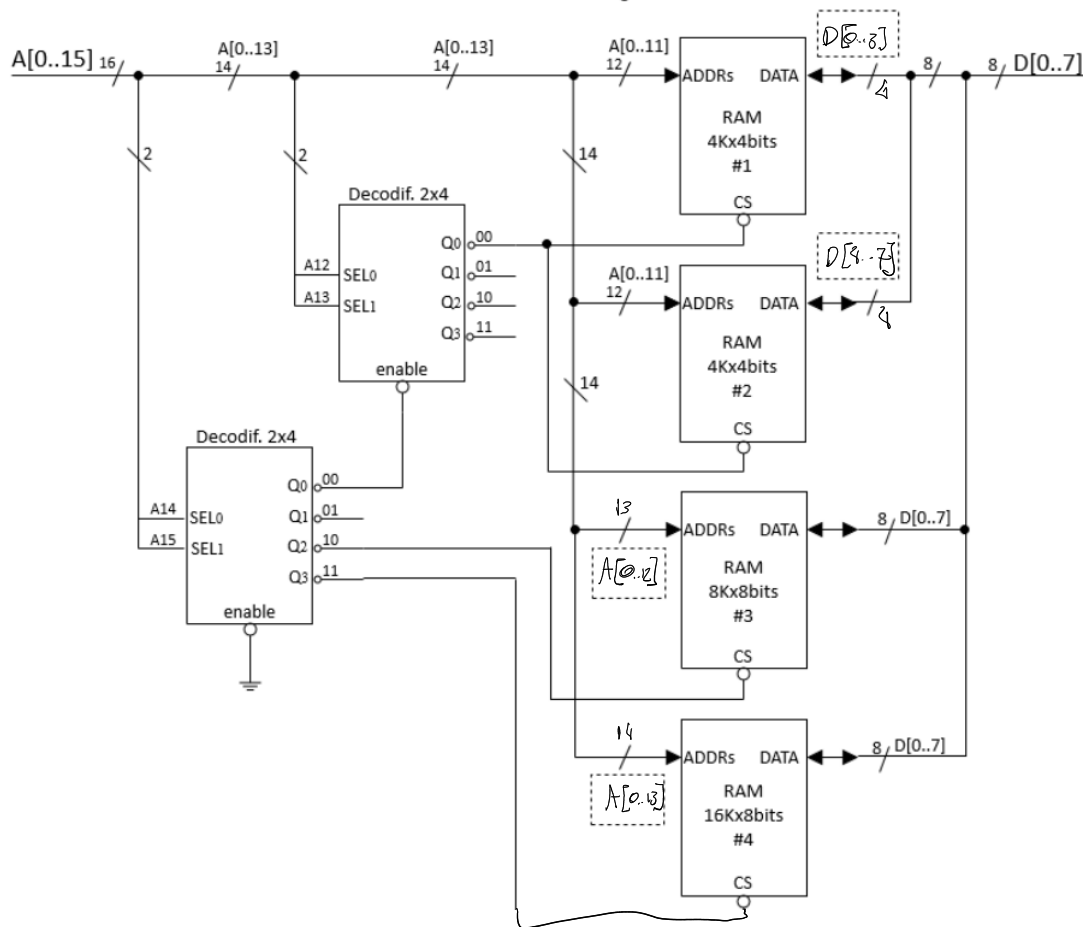
D) Como se dijo en el punto B, la salida del decodificador es activa por bajo, por lo tanto, si alguna de las señales del decodificador que no están utilizadas (X2, X3, X4, X5, X6, X7) se activan, tendremos un cero en dicha salida. Por lo tanto, mediante el siguiente circuito se obtendrá un 1 si se activa alguna de estas señales.



E) No existen posiciones imágenes, ya que, en todos los chips están involucradas todas las señales de direccionamiento. Ya sea, conectadas directamente a las señales de address del chip o mediante el *Chip Select* y el decodificador.

Ejercicio 7

Basados en el sistema de memoria mostrado en la figura:



$$010 = 5$$

$$011 = 7$$

$$1000 = 8$$

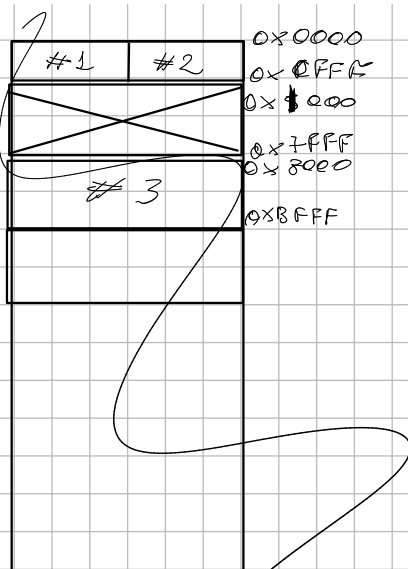
$$1011 = 11 = B$$

$$1100 = C$$

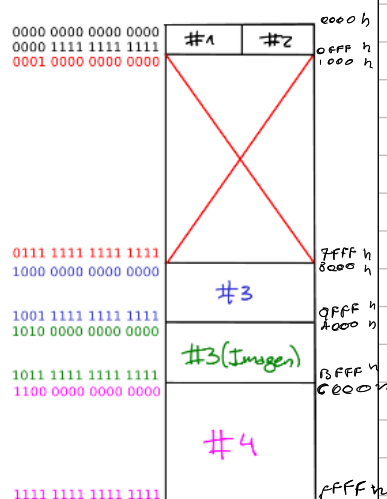
$$1111 = F$$

Se pide:

- Completar los cuadros en línea de puntos con los faltantes de cantidad de señales y su denominación (A[..] o D[..]) para los bloques #1, #2, #3 y #4.
- Desarrollar el mapa de memoria implementado, indicando la dirección de inicio y final de cada bloque. Especificar si se trata de un rango real o de posiciones imagen.
- Responder con **V** (Verdadero) o **F** (Falso) las siguientes afirmaciones:
 - El sistema de memoria implementado NO contiene posiciones imagen.
 - El sistema de memoria contiene segmentos de memoria no implementados.
 - El procesador puede direccionar un total de 64K palabras de 16 bits.
 - Todos los bloques están implementados en posiciones de memoria consecutivas.



Mapa de memoria:



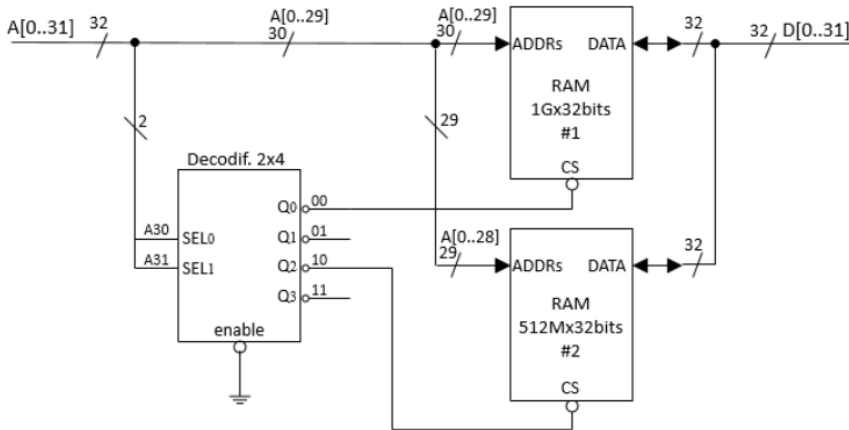
C. Responder con **V** (Verdadero) o **F** (Falso) las siguientes afirmaciones:

1. El sistema de memoria implementado NO contiene posiciones imagen. **F**
2. El sistema de memoria contiene segmentos de memoria no implementados. **V**
3. El procesador puede direccionar un total de 64K palabras de 16 bits. **F** *son de 8 bits*
4. Todos los bloques están implementados en posiciones de memoria consecutivas. **F**

Ejercicio 8:

Basados en el sistema de memoria mostrado en la figura, responder:

- A. Calcular la capacidad total de memoria implementada (expresada en bits).
- B. Dibujar el mapa de memoria, indicando la dirección de inicio y final de cada bloque.
- C. ¿Este esquema genera posiciones imagen? De ser así, especificar a qué bloque corresponde y en qué rango se encuentra en el mapa de memoria del punto B).
- D. Sobre el mismo diagrama de la figura, dibuje la implementación de un bloque de memoria RAM #3 de 1G x 32bits a partir de la dirección 0xC0000000 utilizando la cantidad necesaria de CI's de memoria RAM de 512M x 16 bits y decodificadores.



$$\#1 = 2^{30} \cdot 2^5 = 2^{35} = 32 \text{ Gbytes}$$

$$\#2 = 2^9 \cdot 2^{20} \cdot 2^5 = 2^{34} = 16 \text{ Gbytes}$$

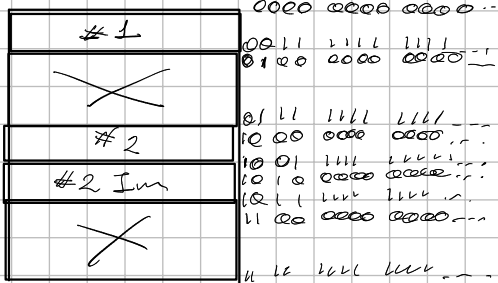
pero tiene una imagen por el bit 29

$$\Rightarrow 32 \text{ Gb} + 16 \text{ Gb}$$

48Gb implementados

$$\text{pero } 2^{32} \cdot 2^5 = 2^{37} = 128 \text{ Gb teóricos}$$

B)



C) Genera posiciones de imagen en el bloque 0x100... - 0xBFF... #2 Im

- D. Sobre el mismo diagrama de la figura, dibuje la implementación de un bloque de memoria RAM #3 de 1G x 32bits a partir de la dirección 0xC0000000 utilizando la cantidad necesaria de CI's de memoria RAM de 512M x 16 bits y decodificadores.

1100 0000...

