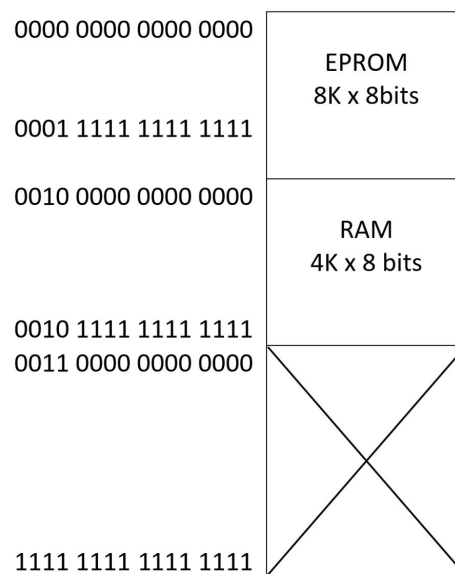


PRÁCTICO 4 - Direccionamiento y Lógica de Decodificación de Memorias

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.

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



Respuesta A:

Para implementar la EPROM de 8K x 8bits se necesitan 4 chip de 2K x 8bits, conectados en serie. Para implementar la RAM de 4K x 8bits se requieren 2 chip de 2K x 4bits conectados en paralelo, para formar 2K x 8bits, y luego 2 de estos bloques conectados en serie. Como resultado, el mapa de memoria implementado tiene la siguiente forma:

Organización del Computador 2020

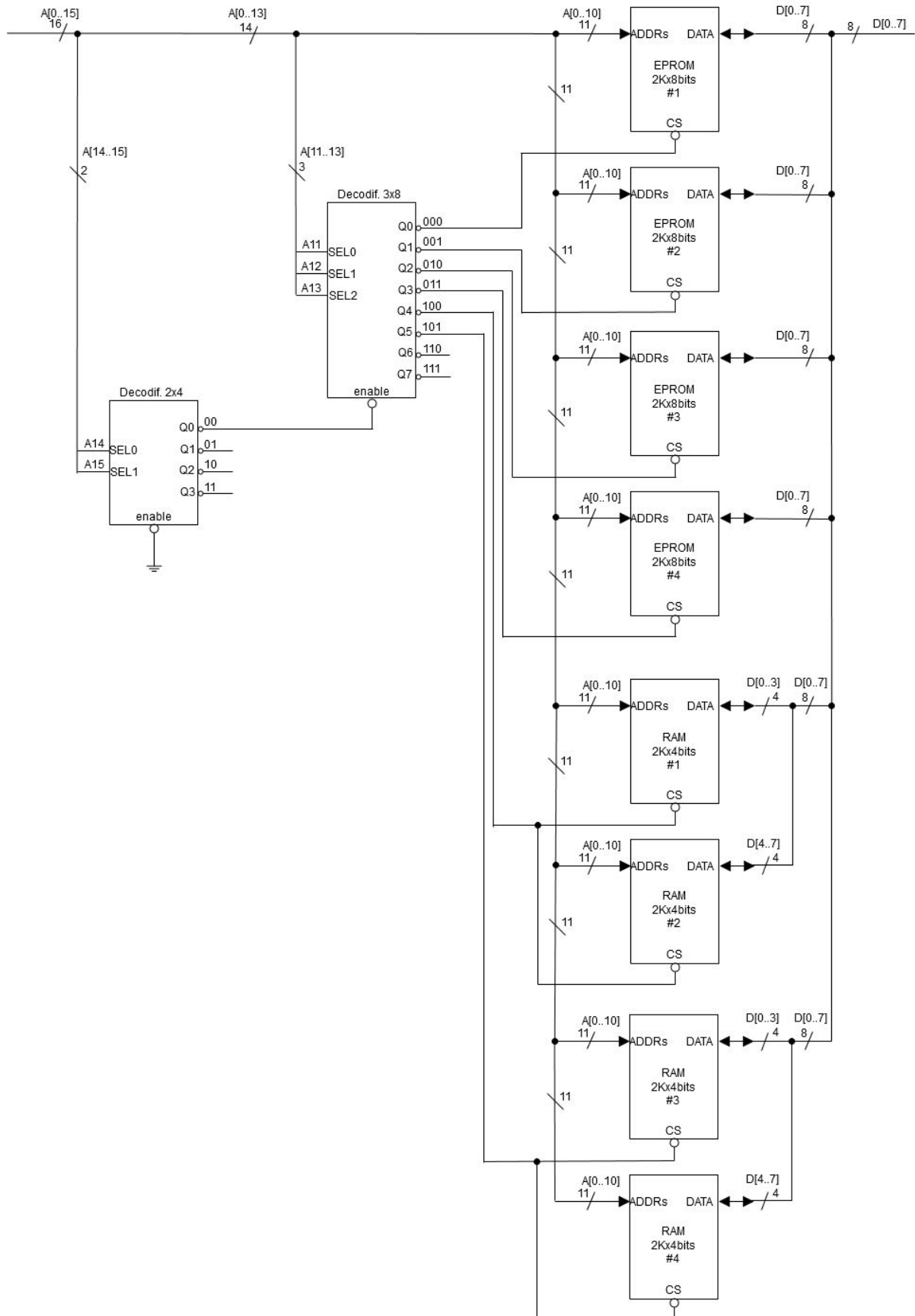
0000 0000 0000 0000	EPROM #1 2Kx8bits
0000 0111 1111 1111	
0000 1000 0000 0000	EPROM #2 2Kx8bits
0000 1111 1111 1111	
0001 0000 0000 0000	EPROM #3 2Kx8bits
0001 0111 1111 1111	
0001 1000 0000 0000	EPROM #4 2Kx8bits
0001 1111 1111 1111	
0010 0000 0000 0000	RAM #1 y #2 2Kx4bits
0010 0111 1111 1111	
0010 1000 0000 0000	RAM #3 y #4 2Kx4bits
0010 1111 1111 1111	
0011 0000 0000 0000	
No Implementado (fuera de escala)	
1111 1111 1111 1111	

Analizando el mapa vemos:

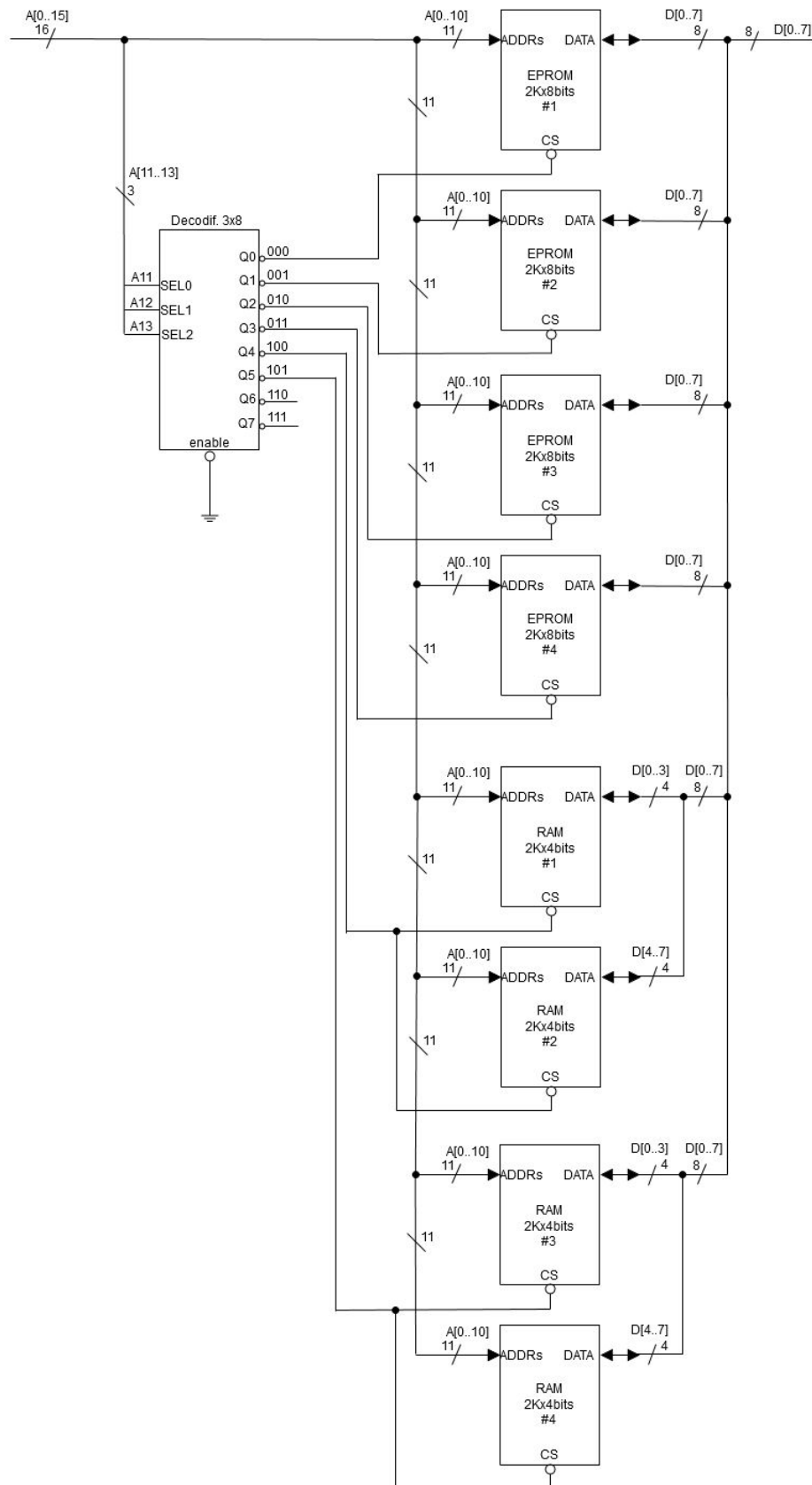
- Para direccionar las 2K palabras dentro de cada bloque de memoria se necesitan 11 bits de address (A10..A0).
- Para direccionar las 8K palabras de la memoria EPROM se requieren 13 bits de address, por lo que se utilizan los bits A12 y A11 para seleccionar cada uno de los 4 chip.
- Para direccionar las 4K palabras de la memoria RAM se requieren 12 bits de address, por lo que se utiliza el bit A11 para seleccionar cada uno de los 2 bloques de 2 chip conectados en paralelo. Notar que por la posición en la que está implementada la RAM, en el comienzo los bits A13 y A12 valen '1' y '0' respectivamente.
- En toda la memoria implementada los bits de address 15 y 14 (resaltados en azul) valen '0'.

Para que la implementación que NO genere posiciones imagen en el espacio no implementado es necesario que para habilitar cada chip de memoria se consideren todos los bits de la dirección, por lo que la implementación tendrá la siguiente forma:

Organización del Computador 2020



Respuesta B:



Esta implementación es igual a la anterior, pero se ha eliminado el decodificador de 2x4 (el que decodificaba A15 y A14), dejando el decodificador de 3x8 siempre habilitado (conectado eléctricamente a '0'). En este caso, los chip de memoria estarán habilitados según corresponda por la combinación de los bits de address A13 a A0, sin importar los valores que tomen A15 y A14.

Con 'N' bits de address, tenemos 2^N combinaciones de dirección. En este caso, A15 y A14 pueden tomar valores '00', '01', '10' y '11'. Como se observa en el mapa de memoria dado, sólo la combinación '00' corresponde al espacio de memoria implementada, mientras que los otros tres casos generarán posiciones imagen.

Analizando el mapa de memoria resultante (abajo) se observa que el contenido de la EPROM y la RAM se replica 3 veces.

0000 0000 0000 0000	EPROM 8Kx8bits	Imagen 1
0001 1111 1111 1111		
0010 0000 0000 0000	RAM 4Kx8bits	
0010 1111 1111 1111		
0011 0000 0000 0000	No Implementado	
0011 1111 1111 1111		
0100 0000 0000 0000	EPROM 8Kx8bits	Imagen 2
0101 1111 1111 1111		
0110 0000 0000 0000	RAM 4Kx8bits	
0110 1111 1111 1111		
0111 0000 0000 0000	No Implementado	
0111 1111 1111 1111		
1000 0000 0000 0000	EPROM 8Kx8bits	Imagen 3
1001 1111 1111 1111		
1010 0000 0000 0000	RAM 4Kx8bits	
1010 1111 1111 1111		
1011 0000 0000 0000	No Implementado	
1011 1111 1111 1111		
1100 0000 0000 0000	EPROM 8Kx8bits	
1101 1111 1111 1111		
1110 0000 0000 0000	RAM 4Kx8bits	
1110 1111 1111 1111		
1111 0000 0000 0000	No Implementado	
1111 1111 1111 1111		