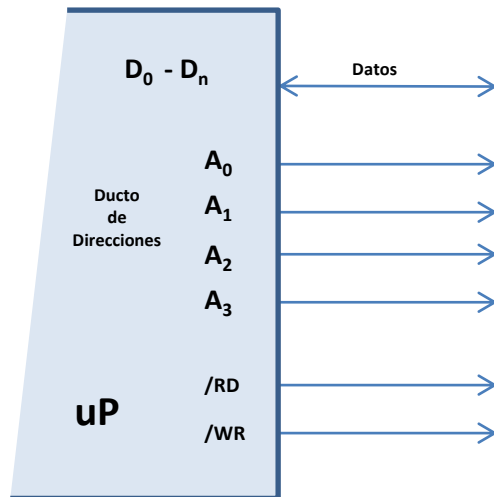


Decodificadores de Memoria

Con fines de explicación considere lo siguiente:

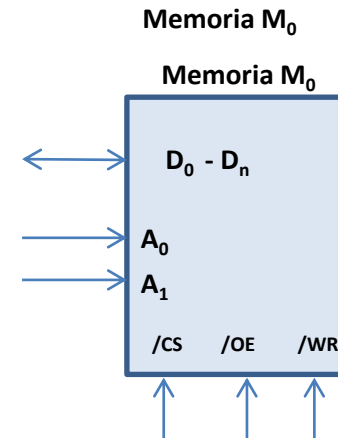
- Un procesador (uP) con un ducto de datos de n bits y un ducto direcciones de 4 bits (A_3-A_0)
- Una memoria (M_0) de un ducto de datos de n bits y un ducto direcciones de 2 bits (A_1, A_0)

Nota: para este caso explicativo no importa el número de bits del ducto de datos.



1

El uP tiene un espacio de direcciones de 2^4 (16 localidades) pues su ducto de direcciones es de 4 bits



2

La memoria tiene un espacio de direcciones de 2^2 (4 localidades) pues su ducto de direcciones es de 2 bits

3

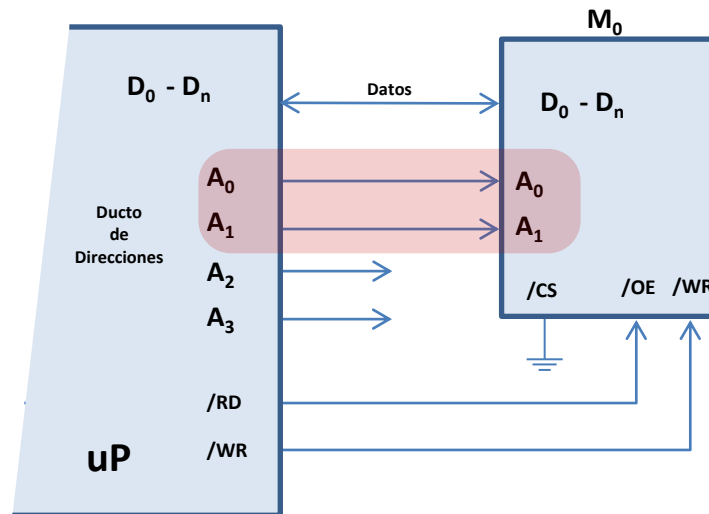
Si se desea conectar la memoria al procesador puede verse que el procesador tiene más líneas de direcciones que la memoria; por tanto tiene más localidades diferentes para acceder (16 en total) que las localidades que tiene la memoria (solamente 4).

Decodificadores de Memoria

4

Para conectar la memoria al procesador lo lógico es conectar los correspondientes ductos (direcciones, datos y control) con se muestra en la siguiente figura.

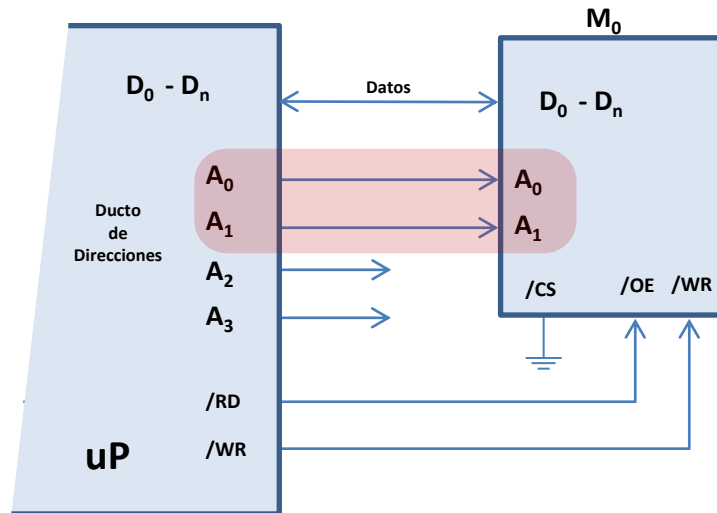
Nota: dado que se tiene una única memoria podemos tener el selector de la memoria (/CS) siempre activo (conectado a tierra) sin causar un conflicto en el ducto.



Decodificadores de Memoria

5

Bajo estas condiciones tenemos entonces la siguiente tabla de direcciones del uP con su correspondiente dirección de la localidad accedida de la memoria.

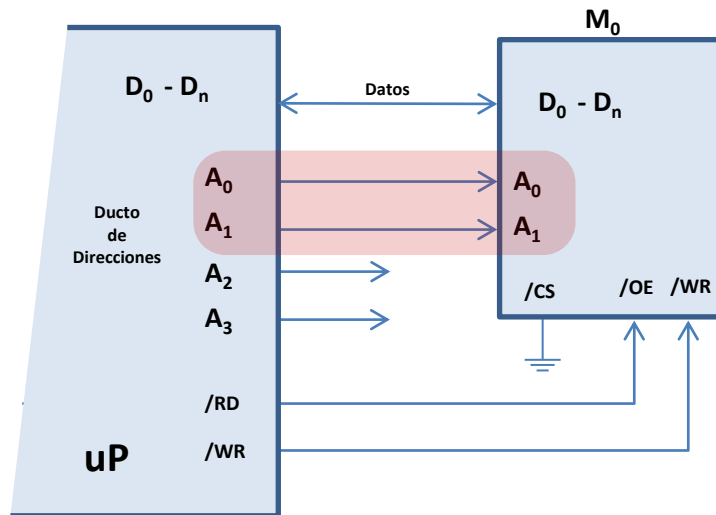


Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	0	4
0101	1	5
0110	2	6
0111	3	7
1000	0	8
1001	1	9
1010	2	A
1011	3	B
1100	0	C
1101	1	D
1110	2	E
1111	3	F

Decodificadores de Memoria

6

Como puede observarse existen diferentes direcciones del uP para las cuales se acceda a la misma localidad de memoria y eso podemos verlo en la tabla donde se muestra que el espacio de memoria (de 4 localidades) se repite a lo largo del espacio de direcciones del uP.

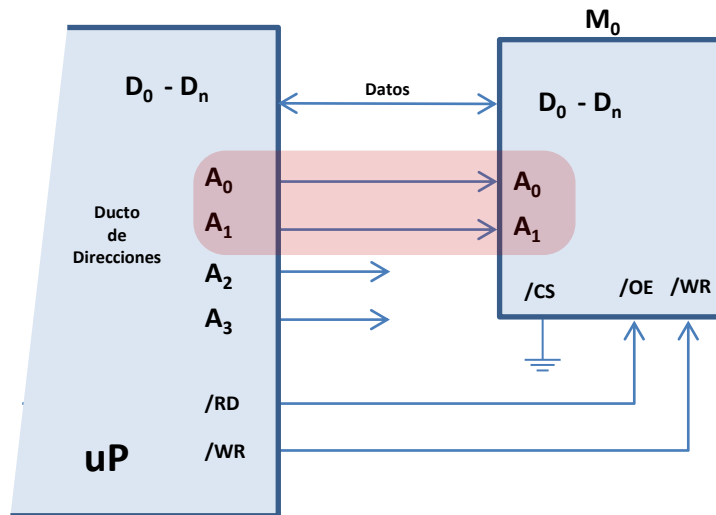


Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	0	4
0101	1	5
0110	2	6
0111	3	7
1000	0	8
1001	1	9
1010	2	A
1011	3	B
1100	0	C
1101	1	D
1110	2	E
1111	3	F

Decodificadores de Memoria

7

Esto puede verse que es debido a que la memoria sólo puede detectar (digamos ver) los dos primeros bits (A_1 y A_0) del ducto de direcciones del uP causando entonces que existan **direcciones espejo**. Es decir una dirección de memoria tiene una correspondiente dirección del uP pero también puede verse reflejada en otra dirección diferente del uP.



Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	0	4
0101	1	5
0110	2	6
0111	3	7
1000	0	8
1001	1	9
1010	2	A
1011	3	B
1100	0	C
1101	1	D
1110	2	E
1111	3	F

8

¿Es problema o no tener direcciones espejo? Depende...

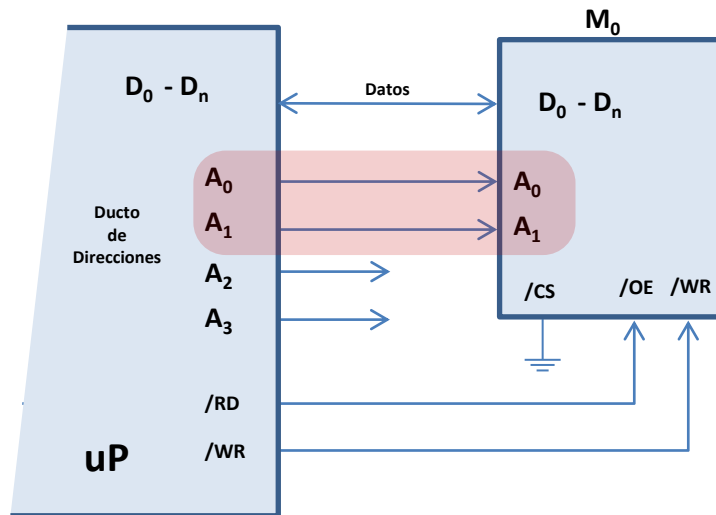
- Si el programador lo sabe entonces no existe problema alguno pues él deberá tener el suficiente cuidado de usar las direcciones correctas.
- Si el programador desconoce la existencia de zona espejos puede pensar que tiene más memoria de la que realmente existe y sobre escribir datos causando así problemas.

Decodificadores de Memoria

9

¿Cómo evitar que existan estas zonas espejo de la memoria?

Se tendría que forzar la asignación de un rango exclusivo de direcciones del uP a la memoria. Como ejemplo podemos decir que la memoria M_0 debería ser accedida exclusivamente en el rango de 0 a 3 de direcciones del uP.



Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	0	4
0101	1	5
0110	2	6
0111	3	7
1000	0	8
1001	1	9
1010	2	A
1011	3	B
1100	0	C
1101	1	D
1110	2	E
1111	3	F

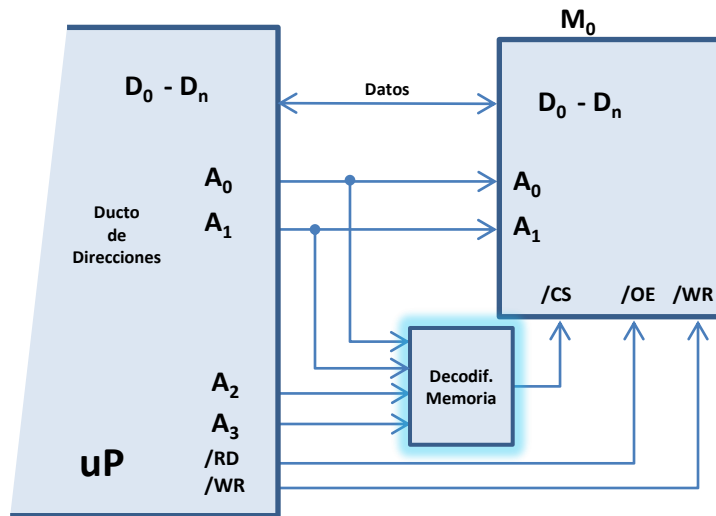
10

Para esto entonces es necesario un decodificador de memoria para que M_0 sólo se active en el rango de direcciones correspondientes.

Decodificadores de Memoria

11

Para esto entonces es necesario un decodificador de memoria para que M_0 sólo se active en el rango de direcciones correspondientes.



Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)	M ₀ /CS
0000	0	0	0
0001	1	1	0
0010	2	2	0
0011	3	3	0
0100	0	4	1
0101	1	5	1
0110	2	6	1
0111	3	7	1
1000	0	8	1
1001	1	9	1
1010	2	A	1
1011	3	B	1
1100	0	C	1
1101	1	D	1
1110	2	E	1
1111	3	F	1

Activar memoria

Desactivar memoria

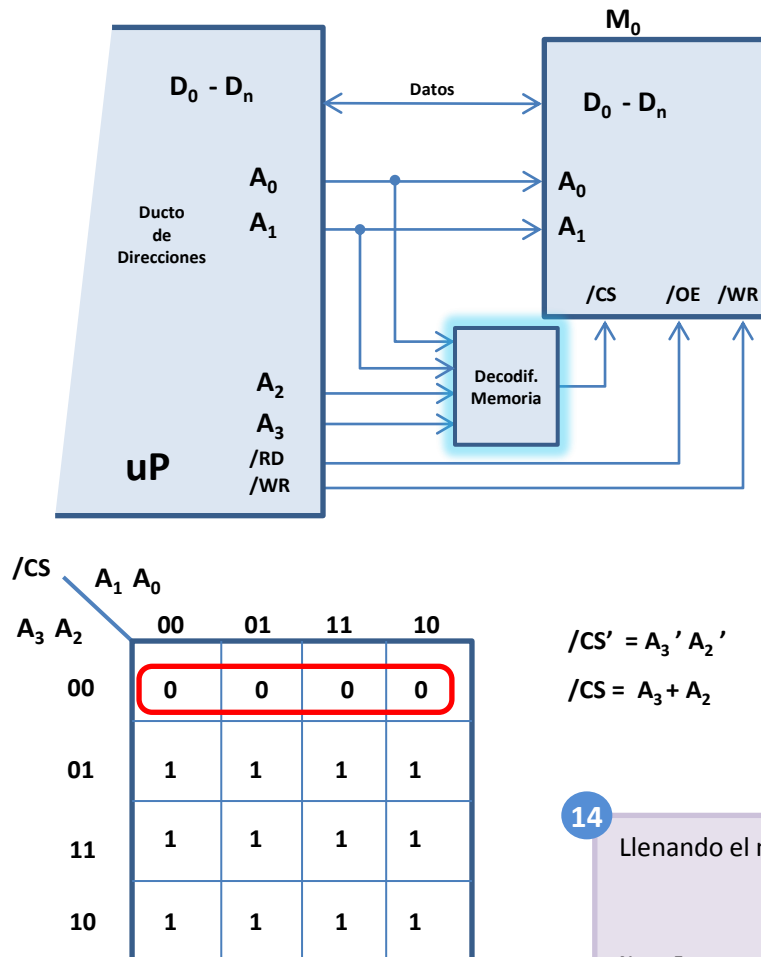
12

Por tanto el decodificador de memoria tendría como entradas las líneas de ducto de direcciones (A₃-A₀) y como salidas el selector de memoria (/CS). Así el decodificador determinará si la dirección presente corresponde al rango y entonces activar (hacer cero) la salida /CS; o desactivar (hacer uno) en cualquier otro caso en la que no corresponde al rango – ver tabla.

Decodificadores de Memoria

13

El decodificador es un circuito combinaciones donde su salida depende de la entrada (para al entrada, tal salida) y para el diseño se hace uso de mapas de Karnaugh .



Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)	M_0 /CS
0000	0	0	0
0001	1	1	0
0010	2	2	0
0011	3	3	0
0100	0	4	1
0101	1	5	1
0110	2	6	1
0111	3	7	1
1000	0	8	1
1001	1	9	1
1010	2	A	1
1011	3	B	1
1100	0	C	1
1101	1	D	1
1110	2	E	1
1111	3	F	1

Activar memoria

Desactivar memoria

14

Llenando el mapa y tomando ceros (son menos) tenemos como resultado:

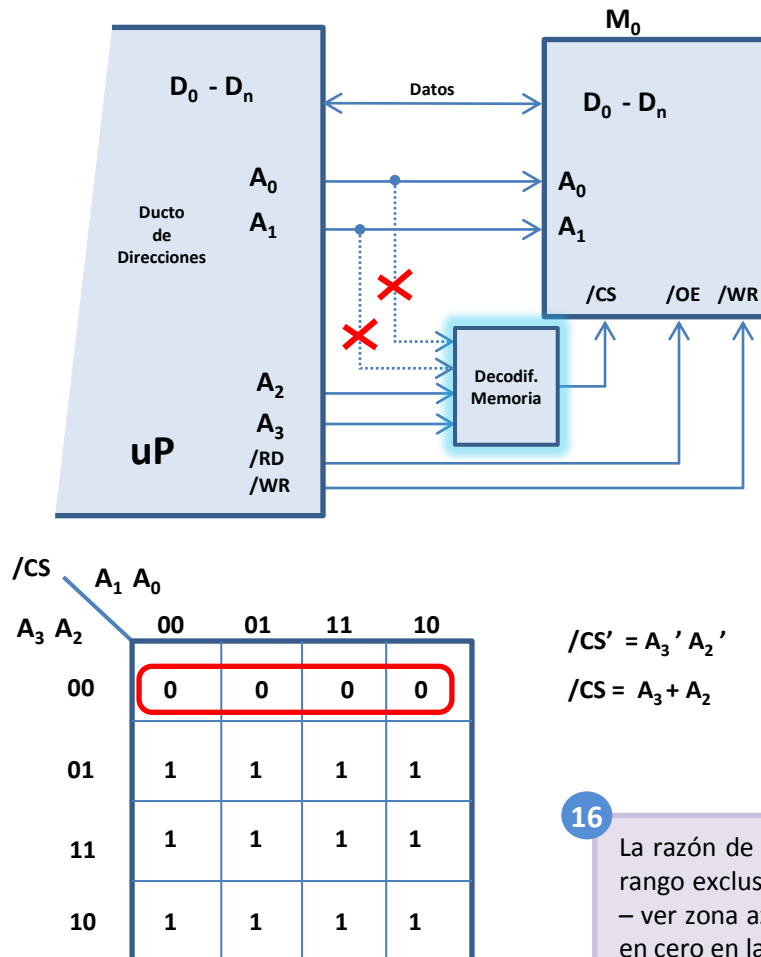
$$/CS = A_3 + A_2$$

Nota: En este caso particular resultaría la misma ecuación si tomamos los unos en lugar de los ceros.

Decodificadores de Memoria

15

Se puede observar que en el proceso desaparecen las líneas A_0 y A_1 – líneas que van directamente a la memoria.



Dirección (uP)	Dir. Mem (Hex)	Localidad (Hex)	M_0 / CS
0 0 0 0	0	0	0
0 0 0 1	1	1	0
0 0 1 0	2	2	0
0 0 1 1	3	3	0
0 1 0 0	-	4	1
0 1 0 1	-	5	1
0 1 1 0	-	6	1
0 1 1 1	-	7	1
1 0 0 0	-	8	1
1 0 0 1	-	9	1
1 0 1 0	-	A	1
1 0 1 1	-	B	1
1 1 0 0	-	C	1
1 1 0 1	-	D	1
1 1 1 0	-	E	1
1 1 1 1	-	F	1

Activar memoria

Desactivar memoria

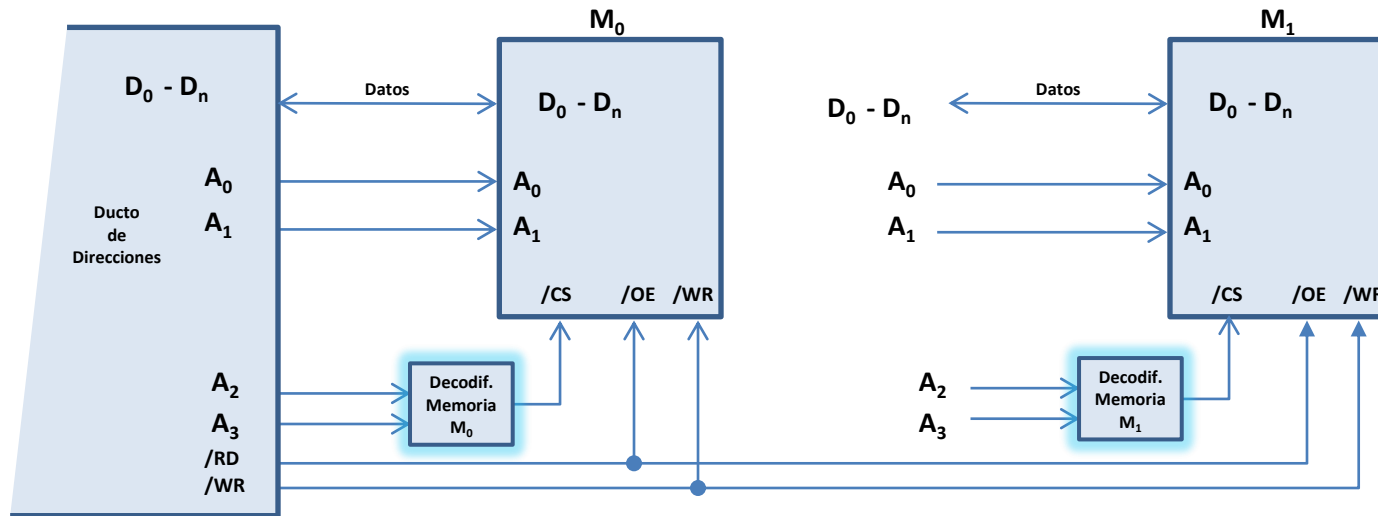
16

La razón de que A_0 y A_1 se eliminan es porque esas líneas no nos ayudan a discriminar el rango exclusivo deseado pues en todo el espacio de direcciones del uP siempre cambiaron – ver zona azul. En cambio A_3 y A_2 si ayudan a discriminar la zona pues éstas permanecen en cero en la zona exclusiva que se desea activar la memoria.

Decodificadores de Memoria

17

¿Qué sucede si ahora tenemos dos memorias?



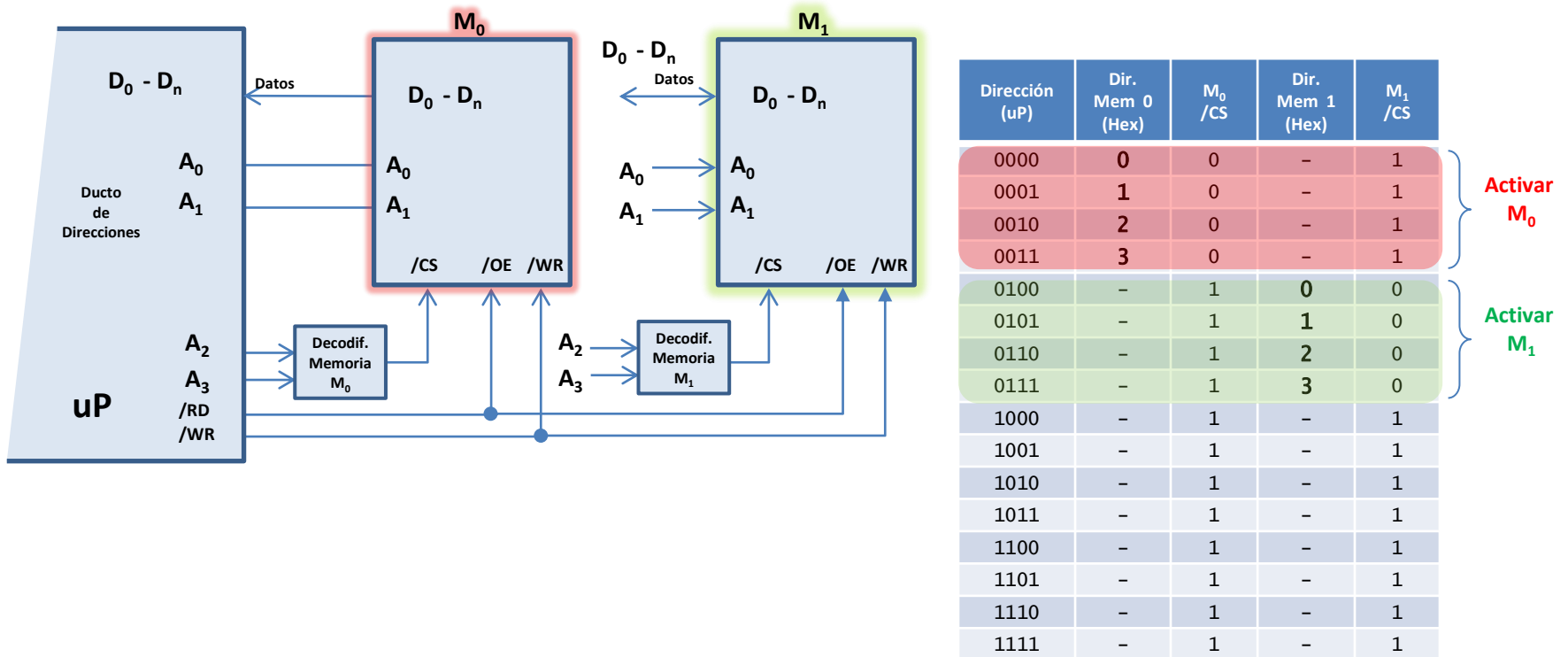
18

Para cada memoria debe existir un decodificador para que cada una de las memorias se active exclusivamente en su rango de direcciones válido.

Decodificadores de Memoria

19

Un ejemplo de esto sería el siguiente esquema de dos memoria de 4 localidades cada una.



20

Cada memoria tiene su decodificador y para su diseño para cada uno se realiza su propio proceso según el rango a activar