

Departamento de Arquitectura

Instituto de Computación

Universidad de la República

Montevideo - Uruguay

Notas de Teórico

Memorias ROM

Arquitectura de Computadoras

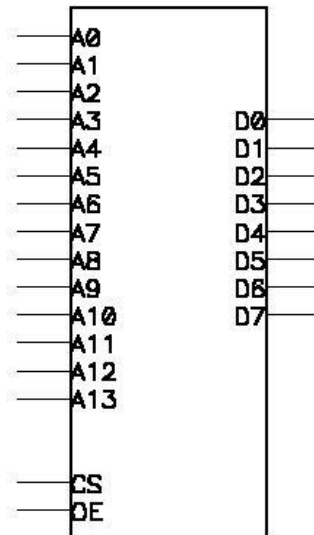
(Versión 4.3a - 2016)

6 MEMORIAS ROM

6.1 Introducción

En este capítulo analizaremos un circuito combinatorio especial, que por su variedad de aplicaciones merece un tratamiento diferenciado del resto.

Este circuito tiene m entradas y n salidas. Para el caso de $m = 14$ y $n = 8$ el símbolo que lo representa es:



Las entradas se identifican por A_0 a A_{13} y las salidas por D_0 a D_7 (en este caso). Para un caso genérico sería: A_0 a A_{m-1} y D_0 a D_{n-1} . Más adelante veremos que son y para qué se usan las entradas CS y OE.

6.2 Características

La tabla de verdad de un circuito ROM está formado por 2^m filas y n columnas:

Entradas								Salidas				
A_{m-1}	A_{m-2}	A_{m-3}	A_2	A_1	A_0	D_{n-1}	D_{n-2}	...	D_1	D_0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	1	0
0	0	0	0	1	0	0	0	0	0
....
....
1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	0	0	1	0	1
1	1	1	1	1	1	1	1	1	1

La colección de 0 y 1 que hemos colocado en las salidas es una cualquiera de las posibles. De hecho podemos representar genéricamente una de las salidas como:

$$D_{ij}$$

siendo

$$0 < i < 2^m - 1$$

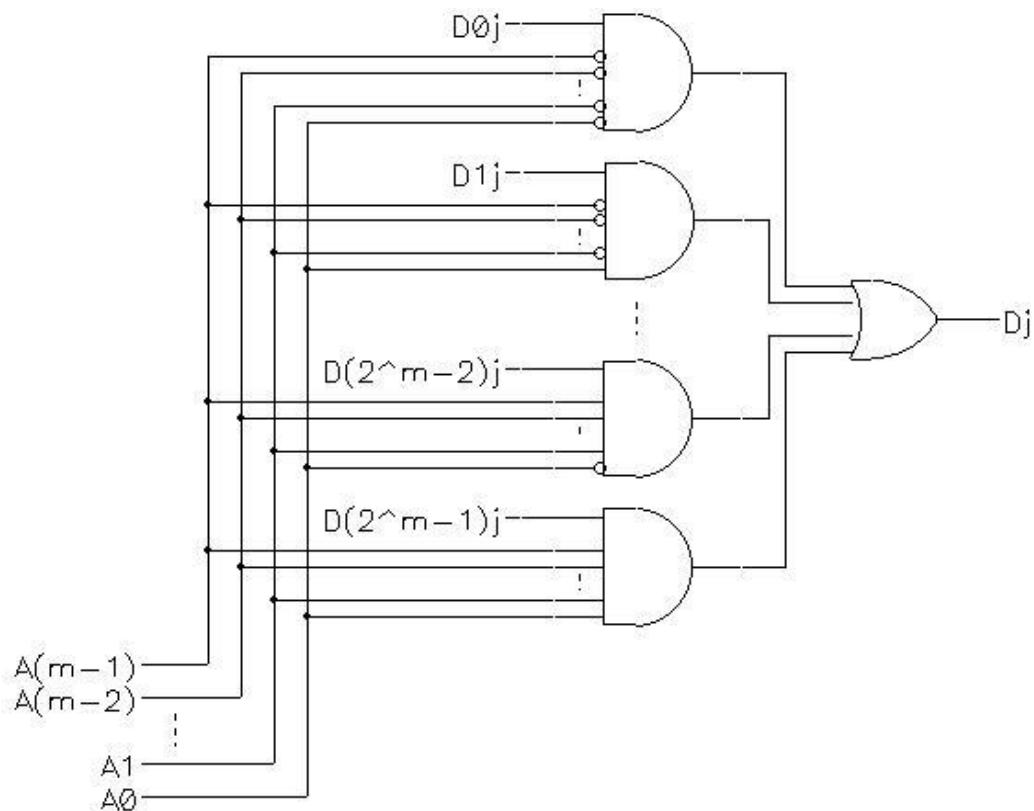
$$0 < j < n - 1$$

Esta tabla de verdad puede verse como la descripción del contenido de un "array" de 2^m elementos que tienen n bits cada uno. Como es un circuito combinatorio, los valores concretos de los bits de los elementos (los D_{ij}) son fijos, predeterminados. Por estas dos características a estos circuitos se los llama Read Only Memory (Memoria de Solo Lectura), o, más brevemente, **ROM**.

Es fácil observar que con una ROM se puede implementar cualquier función lógica de m variables de entrada y n salidas. Basta con especificar el "contenido" de la ROM de manera que los n bits de cada palabra (posición del array) correspondan al valor de la función en el punto (que coincide con el índice del array).

6.3 Circuito Interno

El circuito interno de una ROM tiene, para cada bit de salida, la siguiente forma:



El fabricante de la ROM coloca la entrada de la AND asociada a D_{ij} en 0 ó en 1, en función de la tabla de verdad que le especifiquemos. Para ello conecta la entrada a "tierra" o a "Vcc".

6.4 Variantes Tecnológicas

Las ROMs así construidas tienen el inconveniente que una vez que se fabrican no es posible cambiar su contenido.

Esto no sería un problema significativo cuando usamos las ROMs como circuito combinatorio. Sin embargo el uso habitual de las ROMs es el de almacenar programas fijos (ej: las rutinas de inicio de un computador, el programa almacenado de un controlador de un semáforo, un lavarropas, etc). Los programas tienen correcciones y

mejoras constantes, por lo que es poco práctico (y poco rentable) tener que producir nuevas ROMs cada vez que hay un cambio.

Por ello se fueron desarrollando con el tiempo nuevos circuitos que dieran respuesta a esta situación: por un lado fueran memorias permanentes (no perdieran su contenido al quedar sin energía eléctrica) y por otro pudiera ser modificado su contenido de alguna forma.

6.4.1 PROM

Las PROM son **P**rogrammable **R**OM. Una PROM es una ROM cuyo contenido puede ser definido a posteriori de construida, mediante una actividad de programación que se realiza utilizando un circuito electrónico especial (un Programador de PROMs).

En esencia son ROMs que tienen en su entrada D_{ij} a las ANDs de selección una conexión tanto a ground (0) como a V_{cc} (1). Esta conexión está realizada mediante un fusible, el cual se quema al momento de "programar" el contenido de la PROM. Si quiero grabar un 0 quemo el fusible de la conexión a V_{cc} y si quiero grabar un 1 quemo el fusible de la conexión a tierra.

Estos fusibles no pueden reconstruirse. Cuando se graba una PROM con un cierto contenido no hay marcha atrás.

6.4.2 EPROM

Si bien las PROMs significaron un avance, el hecho de no tener "vuelta atrás" aún significaba una restricción para el uso intensivo de PROMs en el almacenamiento de programas. De esa necesidad no del todo satisfecha surgió la tecnología de las EPROM (**E**rasable **P**ROM).

Una EPROM es una ROM que puede ser borrada. El mecanismo de borrado es totalmente distinto al de programación e implica un proceso de exposición del circuito a luz ultravioleta por varios minutos. La gran ventaja es que puede reutilizar las EPROMs muchas veces borrando su contenido y grabando uno nuevo.

Para ello las EPROM disponen de una ventana transparente en el encapsulado cerámico ó plástico del circuito integrado.



Tomado de Wikipedia © Bill Bertram 2006

Esa ventana expone el propio chip de silicio, de forma de poder irradiar adecuadamente el material con luz ultravioleta de forma de revertir las modificaciones físico-químicas producidas por el proceso de grabación por impulsos eléctricos, mediante un dispositivo específico: Programador de EPROMs.

Esta ventana está normalmente tapada de forma de evitar exponer el silicio a la luz normal (que contiene componentes ultravioletas) para que el contenido de la EPROM no se altere.



De todos modos como el fenómeno también se produce ante la presencia de otro tipo de radiaciones (como los rayos cósmicos) que no pueden detenerse con una etiqueta, el contenido de las EPROMs termina alterándose con el tiempo (aunque, por suerte, este tiempo es sumamente largo, de varias decenas de años).

Como se dijo su principal uso es el almacenamiento de los programas permanentes de un sistema. Su capacidad de desde algunos kilobits hasta del orden de 8 Megabits. Muchas veces están organizadas en palabras de 8 bits (byte).

6.4.3 EEPROM

Las EPROM si bien solucionan el problema de la re-usabilidad de este tipo de memorias, todavía tienen el inconveniente que este proceso es sumamente lento, complejo y requiere retirar la EPROM del sistema para realizar el borrado.

Es así que surgieron las EEPROM (**E**lectrical **EP**ROM), o sea una EPROM cuyo proceso de borrado se hace eléctricamente y puede efectuarse sin retirar el circuito integrado del sistema. Posee otra diferencia importante con la EPROM: una EEPROM normalmente tiene la capacidad de borrar cada bit en forma individual (también hay implementaciones que borran una palabra completa en cada operación de borrado).

Típicamente se utilizan para almacenar los datos de configuración de un sistema. Tienen una capacidad de hasta del orden de 128 kbits. Es frecuente que estén organizadas en palabras de un solo bit.

6.4.4 Flash EEPROM / Flash EPROM / Flash Memory

Este tipo de memoria es una variante de las EEPROM que se desarrolló con el objetivo de mejorar el tiempo de borrado, de forma de habilitar su uso para aplicaciones de almacenamiento masivo.

Si bien el nombre está asociado al concepto de velocidad (lo que se corresponde con lo antedicho), el nombre se origina en la similitud que uno de sus creadores veía entre el proceso de borrado y el destello del flash de una cámara de fotos.

Su aplicación más difundida es la de almacenamiento masivo (reemplazo de discos duros o disquetes), ya que su tiempo de acceso es varios órdenes de magnitud menor que la de dichos dispositivos. Las capacidades de los chips llegan en la actualidad a del orden de 6 Tbits, y están organizados en palabras de 8 ó, más habitualmente, 16 bits.



En la foto siguiente se puede ver la parte interna de una Memoria USB, que actualmente se usa para almacenar información en forma transportable (lo que antes se hacía con disquetes).



6.5 Capacidad y Organización

Las memorias ROM están caracterizadas por una cierta capacidad, que se mide en bits (ó Kilobits, Megabits ó Gigabits) y una determinada organización, que se expresa en cantidad de palabras de tantos bits.

Ejemplos:

- ROM de 8 Kbits, con organización de 1Kx8 (1 Kilo palabras de 8 bits)
- ROM de 8 Kbits, con organización de 4Kx2 (4 Kilo palabras de 2 bits)
- ROM de 1 Kbits, con organización de 1Kx1 (1 Kilo palabras de 1 bit)

6.6 Arreglos de Memorias

Las memorias se utilizan en combinaciones (arreglos) normalmente denominados "bancos de memoria". Estos bancos permiten la construcción de memorias del tamaño requerido por el sistema en base a circuitos integrados disponibles (es habitual que se requiera más memoria de la que puede suministrar un único circuito integrado, o que el tamaño de palabra necesario sea otro).

Es de destacar que estas ideas que estamos desarrollando en el contexto de memorias ROM también se aplican a otro tipo de memorias (RAM) que veremos en otro capítulo y, en particular, a los módulos de memoria (tipo SIMM, DIMM o similar).



En los próximos diagramas circuitales aparecerán las entradas CS y OE. Como ya hemos mencionado el significado y utilización de dichas entradas se verá más adelante en este mismo capítulo. Por ahora consideremos que son entradas que existen en todas las ROMs y deben, por tanto, existir en las ROMs "compuestas" que vamos a diseñar.

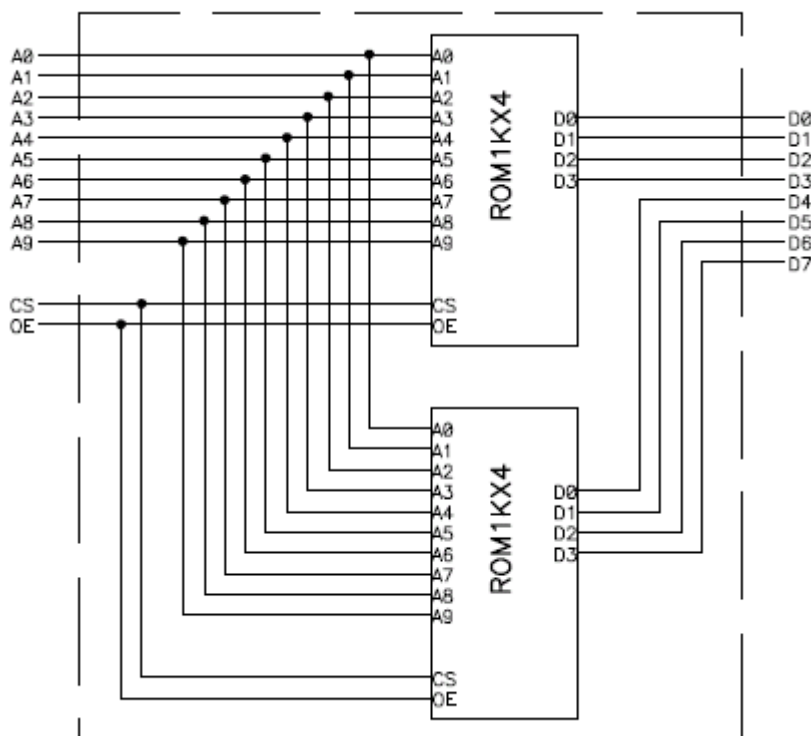
6.6.1 Memoria de mayor tamaño de Palabra

En este caso el objetivo es lograr una memoria ROM de n bits de salida a partir de memorias de menor cantidad de bits (su tamaño de palabra es menor que el requerido).

Veamos como se construye un banco así utilizando un ejemplo: supongamos que queremos construir una ROM de 1Kx8 y disponemos solamente de ROMs de 1Kx4.

Lo primero es determinar la cantidad de memorias requeridas. En este caso vemos que la memoria solicitada tiene una capacidad de 8 Kbits y las disponibles de 4 Kbits. Esto da que se necesitarían 2. Por otro lado el análisis de la organización solicitada y la disponible, confirma esta cantidad.

No es difícil concluir que el circuito de la ROM de 1Kx8 equivalente es el representado por el diagrama esquemático a continuación:



6.6.2 Memoria de mayor cantidad de Palabras

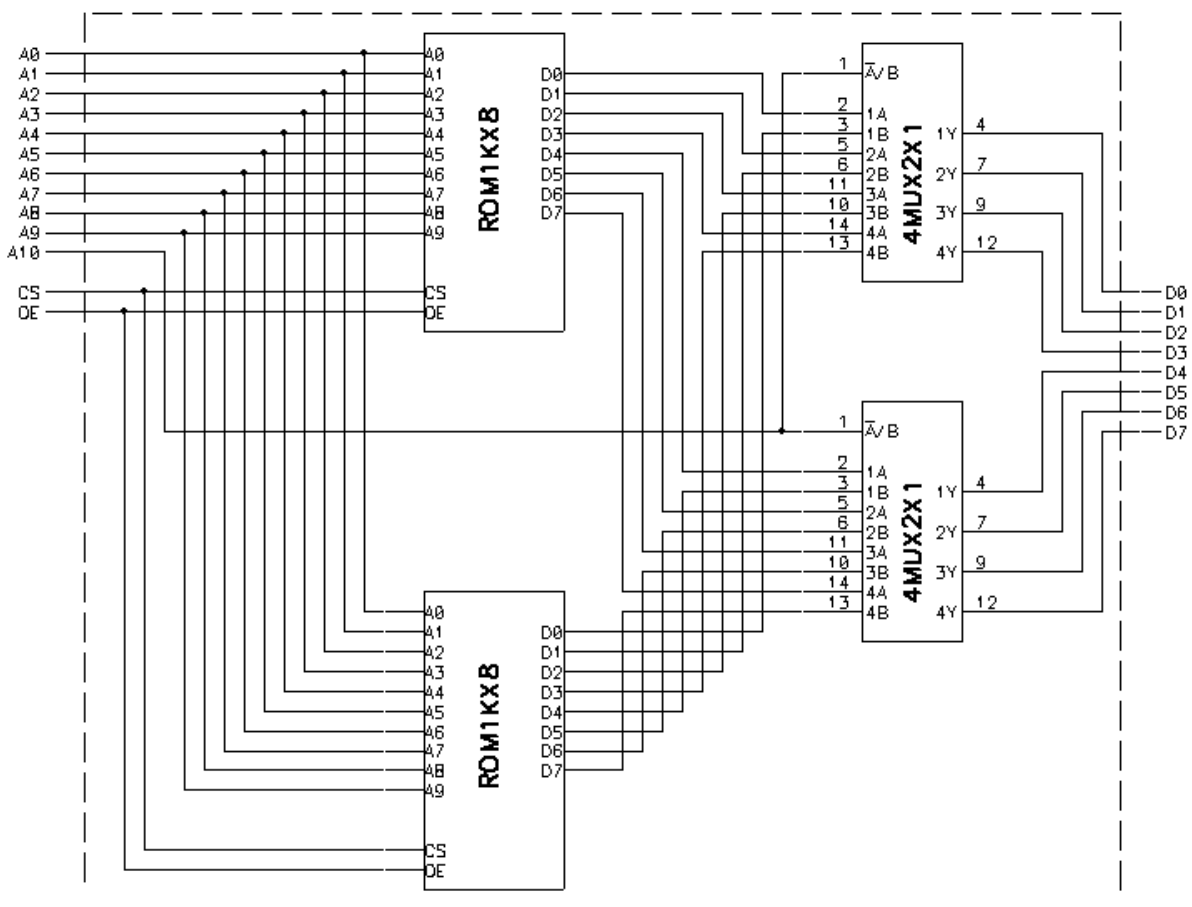
En este caso el objetivo es lograr una memoria ROM de m bits de entrada a partir de memorias de menor cantidad de bits de dirección.

Veamos como se construye un banco así utilizando un ejemplo: supongamos que queremos construir una ROM de 2Kx8 y disponemos solamente de ROMs de 1Kx8.

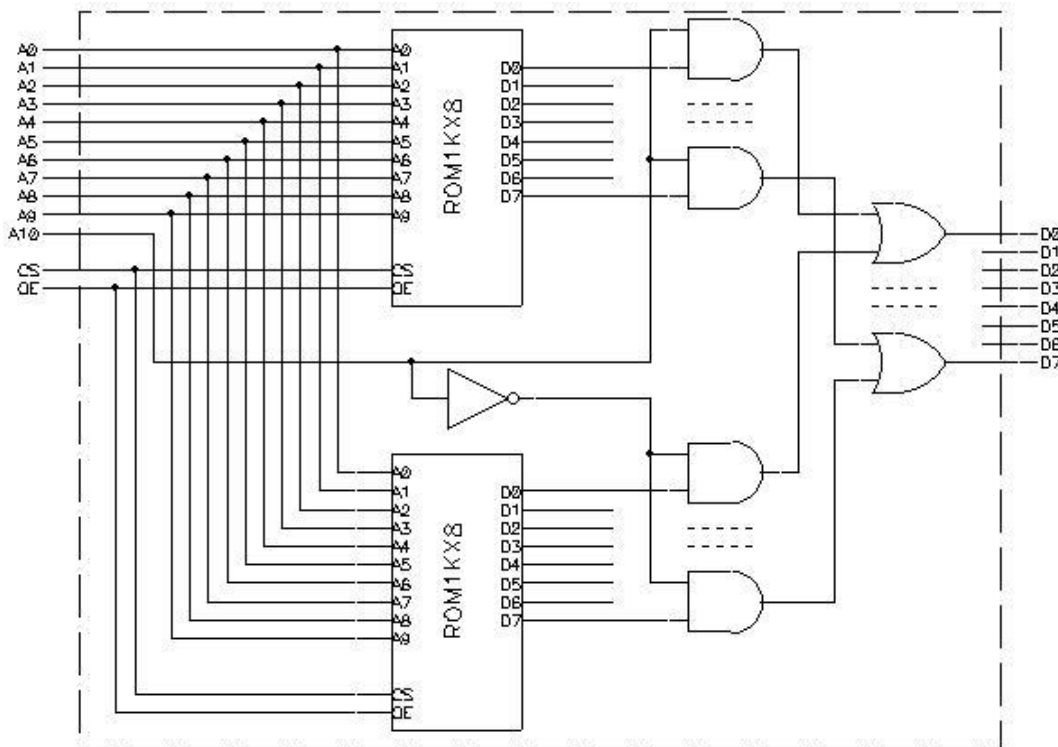
Al igual que en caso anterior lo primero es determinar la cantidad de memorias requeridas. En este caso vemos que la memoria solicitada tiene una capacidad de 16 Kbits y las disponibles de 8 Kbits. Esto da que se necesitarían 2 al igual que en el ejemplo anterior. Por otro lado el análisis de la organización solicitada y la disponible, confirma esta cantidad.

En el caso anterior cada ROM contribuía con una parte de los bits de salida. En este caso cada ROM contribuirá con una parte del rango de direcciones. La ROM a construir tiene 11 bits de entrada, con un rango de direcciones de 0 a 2047. Cada ROM disponible tiene 10 bits de entrada, con un rango de direcciones de 0 a 1023. Por tanto cada ROM contribuirá con la mitad de las posiciones: una de ellas aportará las posiciones de 0 a 1023 y la otra las posiciones de 1024 a 2047. Para seleccionar qué ROM se conecta a la salida utilizamos multiplexores de 2x1, controlados por el bit más significativo de las entradas de dirección.

El circuito que implementa este funcionamiento es:



En el siguiente diagrama los multiplexores se han explicitado en sus equivalentes AND y OR (se ha omitido la repetición de la estructura de compuertas de forma de no sobrecargar el dibujo):



6.7 Chip Select

La entrada CS (Chip Select) permite ahorrar en la implementación la estructura de ANDs que estamos colocando a la salida de las ROMs para elegir cual de ellas conecta a la salida en función del bit más significativo de la dirección.

En otras palabras las compuertas ANDs están incluidas dentro del chip de la ROM y el selector es la entrada CS.

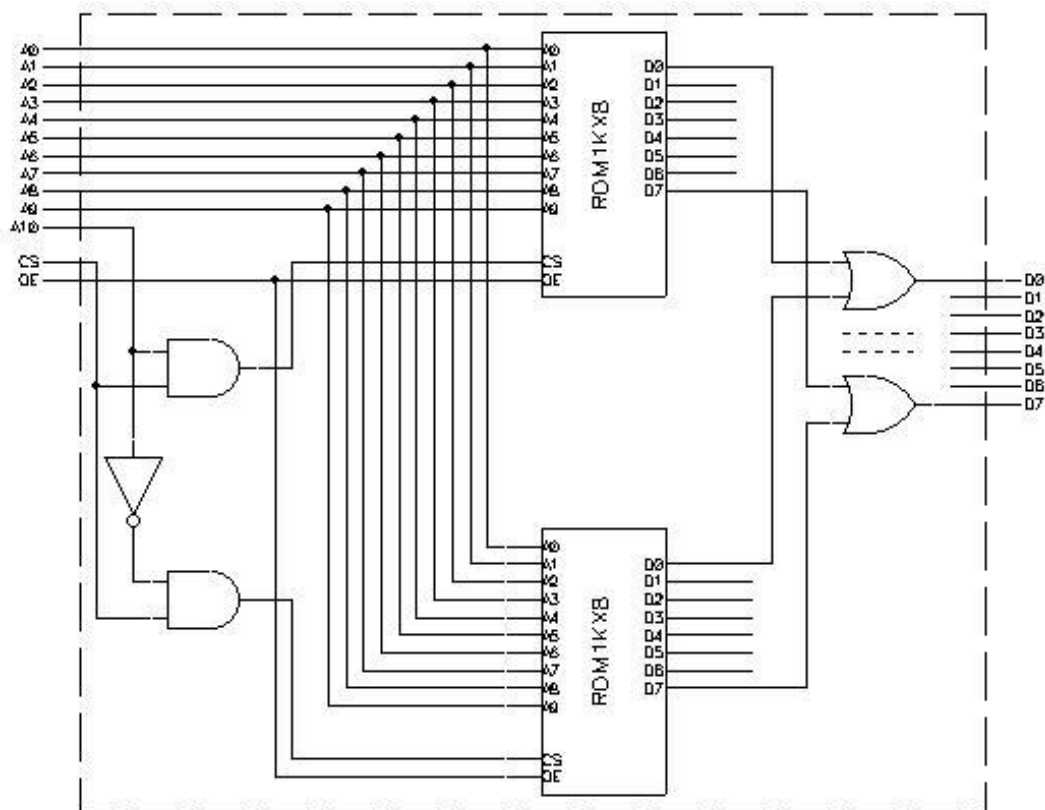
Si $CS = 0$ todas las salidas de la ROM están en 0, con independencia de las entradas de dirección y del "valor" almacenado en la "posición" de la ROM indicada por dicha dirección y si $CS = 1$ las salidas presentan el contenido de la ROM en la posición señalada por la dirección.

En las ROMs más modernas la entrada CS en 0 tiene un efecto un poco distinto por el cual el circuito interno de la ROM pasa a un estado de bajo consumo de energía y el estado de sus salidas tiene un comportamiento similar al "tri-state" que veremos en el próximo punto.

A los efectos del curso consideraremos que el comportamiento es el primero de los descritos (las salidas pasan a estado lógico 0).

6.7.1 Aplicación del Chip Select (CS) a Arreglos de Memorias

El circuito de la ROM de 2Kx8 queda simplificado por el uso del CS de esta manera:



6.8 Lógica de Tercer Estado (Tri-state)

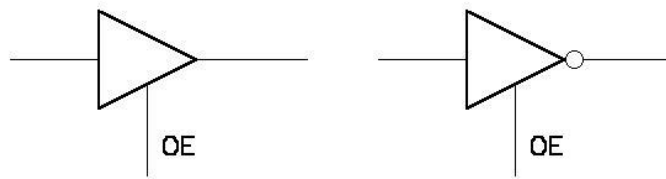
En el ejemplo anterior se ve que las ORs que se colocan a la salida en realidad siempre tienen una de sus entradas en 0. Esto es por la forma de funcionamiento del circuito en donde se utilizan como salida lógica de un selector.

Dada esa propiedad alguien podría verse tentado de quitar las OR y unir las salidas. Eso desde el punto de vista lógico no es posible porque sería equivalente a igualar dos variables de valor en principio distinto (una de ellas con valor 0 y la otra con 0 ó 1). Desde el punto de vista del circuito provocaría un cortocircuito que posiblemente dañe las salidas que se unieran de esa forma.

Para poder implementar ORs "cableadas" (sin necesidad de utilizar compuertas) se diseñaron los circuitos de manera que tuvieran salidas con tres estados posibles (en lugar de los 2 estados lógicos habituales): 0, 1 y Z. El estado Z se denomina también "tercer estado", "estado de alta impedancia", "estado indiferente" ó "tri-state".

Los circuitos que tienen este tipo de salida disponen también de una entrada denominada OE (Output Enable) o similar. Cuando dicha entrada de control está en 0, la salida pasa al "estado de alta impedancia" y cuando la entrada de control está en 1, la salida está en estado lógico 0 ó 1.

Un caso típico de uso de esta tecnología es en los "buffers", ya sean directos o inversores:



6.8.1 Aplicación de Lógica Tri-state con Output Enable (OE) a Arreglos de Memorias

Veamos a continuación como queda del circuito de la ROM de 2Kx8 simplificado por el uso de la lógica de tres estados a través de la entrada de control OE:

