

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Departamento de Electrónica
Cátedra: Técnicas Digitales II - Plan 1995
Versión 2006

CAPITULO III: Memorias

Autor: Ing. Marcelo E. Romeo
meromeo@frba.utn.edu.ar

1. Planteo General.

En este capítulo buscaremos analizar los principios de funcionamiento de las memorias, clasificarlas y estudiarlas conectadas a un microprocesador.

Ampliaremos algunos de los conceptos analizados en 11.2. PROM, EPROM, EEPROM y ROM del Capítulo 11 de Técnicas Digitales I.

2. Principios de funcionamiento.

2.1 Multivibradores biestables con transistores.

Supongamos tener el circuito de la Fig. 1. En dicho circuito admitiremos que todos los elementos conectados a cada transistor, son exactamente iguales a sus equivalentes conectados sobre el otro. Por su parte, ambos transistores tendrán exactamente las mismas características estáticas y dinámicas.

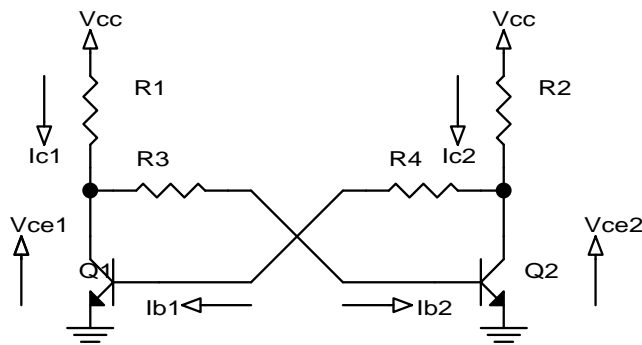


Fig. 1 Diagrama básico de un flip flop con transistores

En tales circunstancias, $I_{C1} = I_{C2}$, $I_{B1} = I_{B2}$ y $V_{CE1} = V_{CE2}$.

Este sistema es absolutamente simétrico, pero, como veremos a continuación *inestable*.

Supongamos que por alguna eventualidad, la corriente de colector del transistor Q₁ aumenta en forma diferencial. Ello producirá un aumento en la tensión sobre el resistor R₁, disminuyendo por aplicación de la segunda regla de Kirchoff V_{CE1} . Esta disminución produce una disminución en la corriente de base del segundo transistor, que produce una disminución en I_{C2} , lo que produce una disminución en la tensión sobre R₂ y aumento en la tensión V_{CE2} . Este aumento genera un incremento en la corriente de base del transistor 1 que a su vez lleva a que I_{C1} crece, lo que fue nuestro punto de partida. Simbólicamente:

$I_{C1} \uparrow$
$V_{CE1} \downarrow$
$V_{BE2} \downarrow$
$I_{B2} \downarrow$
$I_{C2} \downarrow$
$V_{CE2} \uparrow$
$I_{B1} \uparrow$
$I_{C1} \uparrow$

Tabla 1. Evolución del circuito del biestable con transistores al cambiar alguna de sus variables.

El hecho que al aumentar una magnitud eléctrica (I_{C1} en nuestro ejemplo, pero podría haber sido cualquier otra), el sistema refuerza este aumento, se describe diciendo que se trata de un circuito con *realimentación positiva*.

Esta realimentación culminará llevando a Q₁ a la saturación mientras que Q₂ terminará cortado.

Este circuito permanecerá en este estado estable hasta que un pulso negativo¹ en el colector del transistor cortado o un pulso positivo en el transistor saturado invierta el estado de conducción de ambos transistores. En este nuevo estado el circuito permanecerá hasta que por medio de otra señal externa se vuelvan a cambiar los estados.

Esto nos lleva a proclamar que el sistema tiene dos estados estables y se trata de un circuito multivibrador biestable realizado con transistores pero del mismo tipo de los vistos en el Capítulo 7² de Técnicas Digitales I.

2.2 Almacenamiento de información binaria.

En forma introductoria, supongamos que deseamos almacenar información binaria para su posterior empleo. Supongamos que deseamos almacenar 16 bits de información. Según una regla empírica de mejor aprovechamiento del área del circuito integrado, se buscará que la configuración de la memoria sea cuadrada, es decir que tenga la misma cantidad de filas que de columnas .

¹ De los dos éste es el método recomendado, pues una vez que el transistor saturado llegó a la zona activa, su propia ganancia facilitará el cambio de estados.

² 7.2. El Biestable SR

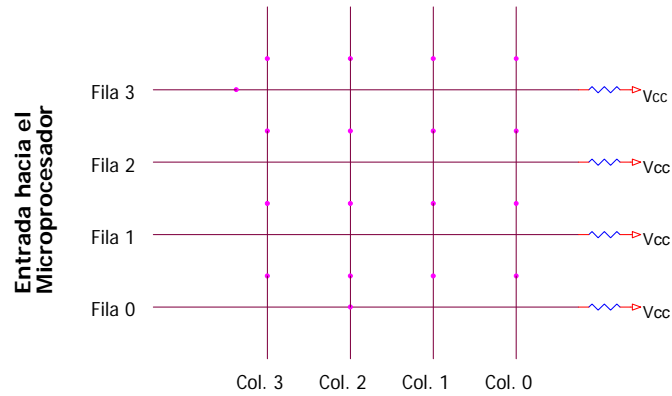


Fig. 2. Esquema inicial de una memoria

Analicemos el esquema de la Fig. 2. Si desde el microprocesador leemos las líneas de entradas allí indicadas, se obtendrá en todo momento "1" en todas las filas. Ello es debido a que al no existir otra conexión, las resistencias de pull up cumplirán su cometido forzando "1" lógico en todas las entradas.

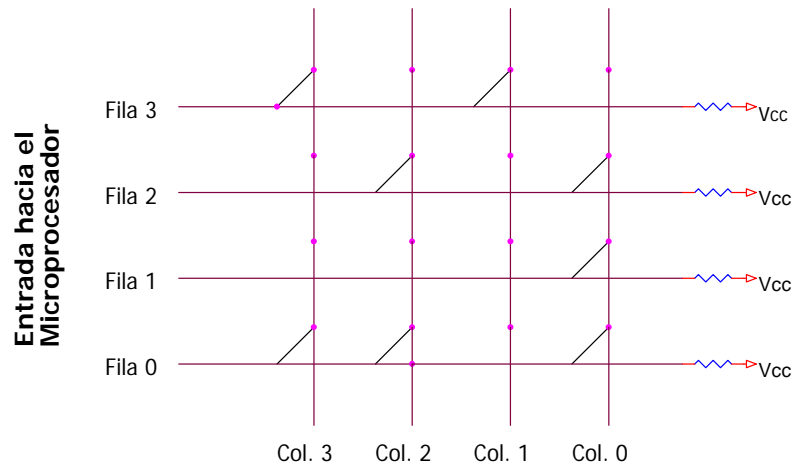


Fig. 3. Esquema de una memoria básica con información.

En la Fig. 3 se observa un esquema similar al de Fig. 2, con el agregado de bits de información por medio de conexiones entre filas y columnas.

Supongamos que colocamos un "0" lógico en una de las columnas y "1" lógico en todas las demás columnas. Como ejemplo, colocamos "0" en la columna 3. Ese "0" aparecerá en las filas 3 y 0, mientras que las filas 2 y 1 quedarán fijas a "1" por los resistores de pull-up.

Bajo esta idea, podremos colocar un decodificador con salidas activas bajas que seleccione una de las columnas, mientras que las demás permanecerán en "1". Podremos "soldar" conductores entre filas y columnas para forzar "0" en esa posición de memoria.

Esta idea es bastante primitiva ya que para modificar el contenido de una posición de memoria deberemos "soldar" y "desoldar" conductores en las intersecciones de fila y columna.

Por otro lado, el bosquejo de la Fig. 3 presenta un inconveniente insalvable que la presencia de conexiones "fantasmas". Supongamos colocar un "0" en la columna 2. Ello producirá que leamos dicho "0" en las filas 0 y 2 como fue nuestro objetivo al conectar dichas filas y columnas.

Ahora bien, como producto de las uniones óhmicas de la columna 2 con la fila 0, dicha fila se hallará a potencial del estado lógico "0". Por la unión entre fila 0 y columna 3, **toda** la columna 3 se hallará a potencial del "0" y por ende, se leerá "0" en la fila 3, lo cual es un error pues sobre la columna 2 no existe unión con la fila 3. Ese valor leído es erróneo y es lo que se denomina conexión fantasma y es producido por el error de implementar los zeros lógicos con uniones puramente óhmicas entre fila y columna.

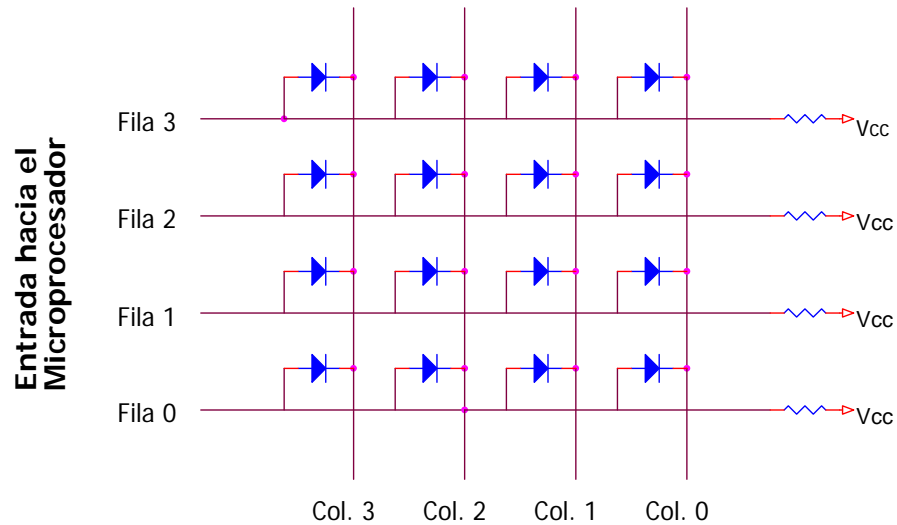


Fig. 4 Esquema de una memoria mejorada con diodos

Una posible solución consiste en colocar diodos en lugar de conductores en las uniones como se observa en la Fig. 4. De esa forma, el "0" lógico forzado por el decodificador en una columna, aparecerá en las filas en las que se hallen los diodos conectados, pero todos los demás diodos quedarán polarizados en inversa por lo que no se producirán conexiones fantasmas.

Seguimos manteniendo la limitación que consiste en que para modificar el contenido de una posición de memoria deberán desoldarse y resoldarse diodos.

La mejor alternativa para dar flexibilidad a la memoria y poder modificar el contenido de la misma, consiste en conectar transistores NPN o MOS en las uniones, con el emisor apuntando hacia lo que antes era el cátodo de los diodos, como se aprecia en la Fig. 5. Esos transistores pueden ser uno de los transistores del flip-flop de la Fig. 1, de forma que pueda cambiarse el contenido de cada posición de memoria cambiando el estado de cada biestable.

En el ejemplo que estamos analizando, se necesitarán 16 biestables ya que se conectará uno en cada intersección de fila y columna. Cada biestable mantendrá la vinculación entre fila y columna hasta que se modifique su estado y mientras se mantenga la alimentación del sistema.

Debe tenerse en cuenta que existirá un circuito adicional (ni dibujado ni analizado aquí para no complicar innecesariamente el esquema) que será el encargado de modificar el estado de los biestables según la información que se desee guardar.

En la Fig. 5, se observa también la aparición de un decodificador con habilitación /CS, que permite seleccionar una columna al aparecer una determinada dirección. También se observa un amplificador separador (buffer tristate) que permitirá volcar el contenido de una posición de memoria (seleccionada por el decodificador) sobre el bus de datos del microprocesador, cuando la señal /OE lo permita.

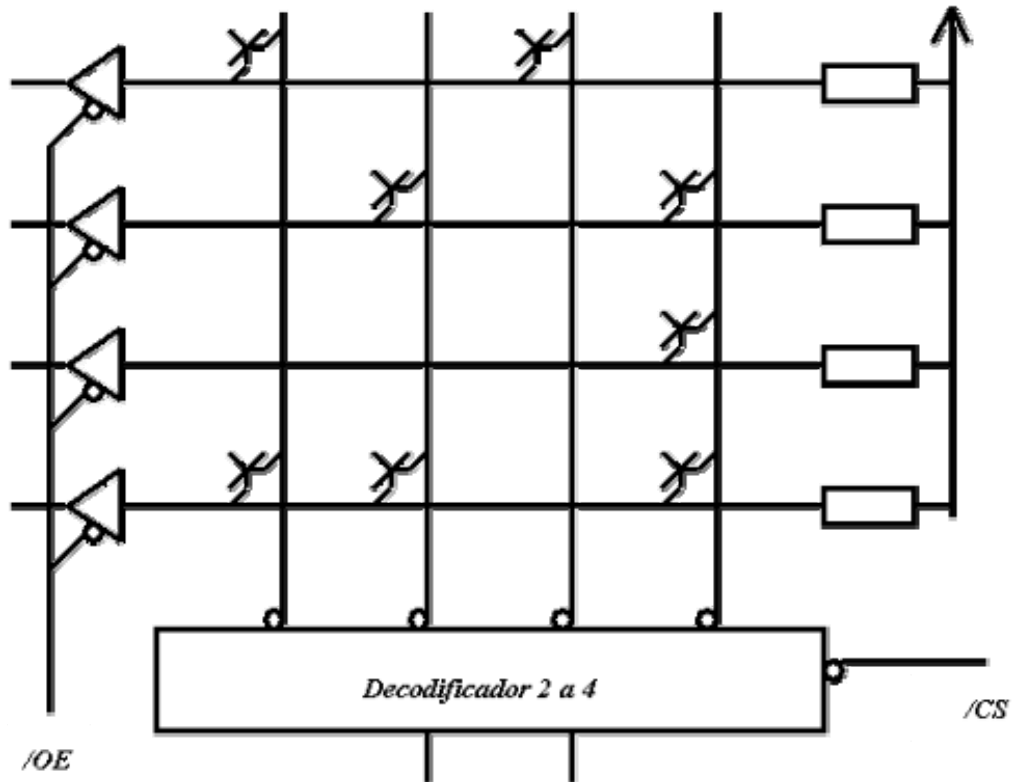


Fig. 5. Bosquejo de una memoria elemental empleando transistores

Normalmente esta señal /OE irá conectada a la línea /RD del microprocesador.

Como hemos visto en el Capítulo II para un ciclo de máquina de lectura de memoria, temporalmente se activará primeramente la habilitación por medio de /CS (producida en el primer ciclo de reloj por la lógica de decodificación) y en el segundo ciclo de reloj la señal /RD habilitará el buffer.

3. Clasificación.

A continuación se realizará una clasificación de las memorias de acuerdo con un conjunto de parámetros. Análogamente a la clasificación de las computadoras en el capítulo I, esta clasificación no es excluyente, ya que un mismo tipo de memoria aparecerá en distintas subclasificaciones.

3.1 Según su funcionamiento.

3.1.1 Memorias Estáticas.

Son aquellas memorias en las que la información permanece en forma permanente en ellas sin necesidad de ningún tipo de acción.

3.1.2 Memorias Dinámicas.

Son aquellas en las que la información se almacena en la carga de un capacitor (parásito o intencional). Por mínimas que sean las corrientes de fuga, el dieléctrico real producirá una pequeña descarga del capacitor. Ello obligará que periódicamente deban reponerse las cargas perdidas. Ese efecto de reposición lleva el nombre de **refresco**. Este tipo de memorias será analizado en detalle en el punto **Memorias Dinámicas** (Memorias Dinámicas)

El tiempo entre refrescos será inversamente proporcional a la temperatura, ya que al aumentar esta última aumentan las pérdidas en el dieléctrico y por ende deberán reponerse las cargas perdidas más rápidamente. Se estima que el refresco deberá proveerse cada 2 a 32 milisegundos³ a 70 °C, con lo tomando dicho tiempo para todo el rango de temperaturas a que funcione el dispositivo será suficiente.

3.2 Según su persistencia

3.2.1 Memorias volátiles

Son aquellas en las que la información se pierde al eliminarse la alimentación. Ejemplo típico son las memorias de una PC. Si se desconecta de la red de alimentación, se pierde la información que se hubiera procesado hasta ese momento.

3.2.2 Memorias Perennes.

Son aquellas en las que la información no se pierde al quitarse la alimentación. Por ejemplo un disquete mantiene su contenido al quitarse la alimentación. Posteriormente incorporaremos en esta categoría a las memorias que definiremos como ROM.

3.3 Según su tecnología.

3.3.1 PMOS.

Las primeras memorias. Lentas (pues los portadores son laguna de escasa movilidad) y requerían una tensión negativa de alimentación. Han caído por completo en desuso.

3.3.2 NMOS.

Emplean electrones como portadores. Son de mayor velocidad que las de la tecnología PMOS y su consumo de energía es importante. Se alimentan con una sola tensión.

3.3.3 CMOS

Debe destacarse que la versión de tecnología MOS complementaria a la que hacemos referencia aquí es la que se empleó desde la década del 60 hasta mediados de la década del 80. Si bien su consumo era bajo (linealmente vinculado con la frecuencia), su densidad de integración pequeña y su velocidad era baja y sólo algunos microprocesadores (familia 1800 de RCA) empleaban memorias de este tipo.

3.3.4 HMOS

Fue una versión de la tecnología NMOS que mejoró la velocidad de su predecesora con disminución de consumo.

3.3.5 HCMOS

Es una versión de alta performance de la vieja tecnología CMOS. Manteniendo el muy bajo consumo, lograron mejorar enormemente su velocidad de operación al punto de ser igual o mejor que la de la tecnología NMOS.

3.4 Según su uso.

3.4.1 De lectura y escritura.

Son memorias alterables constituidas por flip flops. Normalmente serán empleadas para almacenar variables y sobre ella operará la pila del microprocesador. En el circuito de la Fig. 5 se interconectarán uno de los transistores del flip-flop de la Fig. 1, en las intersecciones de filas y columnas. En la jerga técnica se las suele mal llamar **RAM**. Se recomienda analizar la justificación de este nombre mal asignado en Aleatorio (o al azar)..

³ El tiempo máximo entre refrescos depende del tipo de memoria dinámica de que se trate (ver 4.4 Refresco de las memorias dinámicas.

3.4.2 De lectura solamente.

Este tipo de memorias son llamadas habitualmente **ROMs** como abreviatura de Read Only Memories. En el esquema de la Fig. 5, las bases de los transistores se conectarán directa e inalterablemente a V_{cc} ó a $0V$, para almacenar "0" ó "1" lógicos respectivamente. No existirán flip-flops, sino que son simplemente transistores (habitualmente MOS) conectados entre fila y columna.

3.4.2.1 ROMs de máscara⁴.

Las ROMs de máscara son memorias producidas a pedido en fábricas de semiconductores (no en el laboratorio del programador), según los datos provistos por el diseñador del sistema.

Estas memorias son fabricadas en sus múltiples capas en grandes series comunes a todos los usuarios. Solo queda por implementar la última capa de metalización que en definitiva la que conectará las compuertas de los transistores MOS a V_{cc} o tierra.

Con los datos provistos por el diseñador, el fabricante desarrolla esta última capa de metalización que es la que caracteriza cada producto final.

La ecuación económica de la producción de este tipo de ROMs recae fundamentalmente en la producción de las máscaras. Por tal motivo, el costo de estas máscaras (del orden de los miles de dólares) se deberán prorratear entre la totalidad de los circuitos producidos. Por tal motivo, estas ROMs solo se justificarán para las grandes producciones. Obviamente también que una vez producidas las memorias, no pueden realizarse modificaciones en su contenido.

3.4.2.2 PROMs

Para las pequeñas producciones se suelen emplear memorias PROM (*Programmable Read Memory*), de las cuales las primeras que aparecieron fueron en tecnología Schottky TTL. El esquema básico se observa en la Fig. 6. Allí vemos que encontrándose el fusible intacto, el transistor (vinculación entre fila y columna de la Fig. 5) se hallará cortado ($V_{BE} = 0V$), colocando un "1" lógico al ser seleccionada esta celda, mientras que al eliminar el fusible en el proceso de programación, dicho transistor quedará conduciendo, por lo que programaremos un cero lógico.

⁴ Para producir módulos semiconductores monolíticos se deberán depositar varias capas de material con distinto tipo de impurezas y distinto grado de contaminación sobre un cristal de material semiconductor.

Cada una de estas capas involucran un proceso fotolitográfico, donde se deposita una laca fotosensible y se la expone a una radiación UV o de electrones por medio de una máscara. Los lugares expuestos son tratados con sustancias corrosivas y se posteriormente se hace crecer nuevo material con distinto tipo de contaminantes.

Para módulos complejos, este proceso de exposición-corrosión-crecimiento puede llegar a involucrar hasta 20 etapas.

Todo el proceso es idéntico para todas las memorias, salvo la última máscara en la cual las conexiones se hacen siguiendo los requerimientos del cliente.

Los costos de esta última máscara hacen que la solución con memorias ROM sean económicamente factibles para cantidades mayores a las 10.000 piezas.

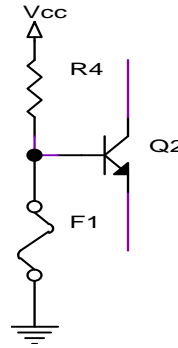


Fig. 6. Diagrama elemental de un bit en una memoria PROM fusible.

3.4.2.3 Memorias EPROM.

Las PROM fusibles son baratas y rápidas (tecnología Schottky), pero tienen como inconvenientes su escaso tamaño (máximo 1 kbyte) por razón de la disipación tolerable y fundamentalmente que al eliminar un fusible por error, el mismo no puede ser repuesto. Ésta última es un inconveniente importante si deseamos trabajar con prototipos en el proceso de desarrollo y depuración, en las que suelen ser frecuentes los agregados, las correcciones y modificaciones.

La solución a este inconveniente la proveyeron las memorias **EPROM** (**Erasable PROM**) que son memorias PROM borrables.

Se basan en el uso de transistores MOS con compuerta aislada y avalancha por inyección (FAMOS en inglés).

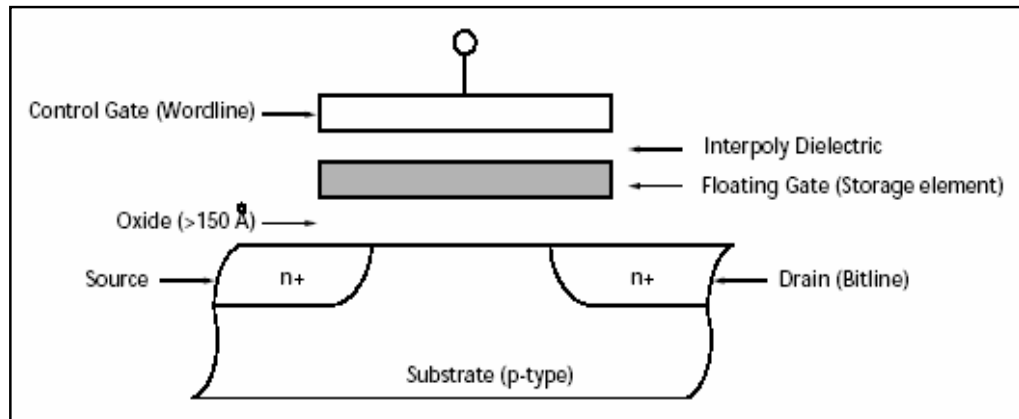


Fig. 7. Celda básica de una memoria UV EPROM.

En la Fig 7 se presenta una celda básica.

El fenómeno que permite la programación se denomina **hot injection** y consiste en aplicar una tensión de programación (propia de cada memoria) V_{pp}^5 , se produce un intenso campo eléctrico en la región del canal de conducción del transistor. Esto produce que algunos electrones salten la barrera del óxido aislante desde la zona de la región del canal hasta la compuerta flotante (floating Gate, como si se tratara de una perforación controlada del dieléctrico).

Durante el proceso de programación, se le aplica una tensión positiva a la compuerta de selección de manera de retener esos electrones en la compuerta flotante.

⁵ Con la cual se logra un altísimo campo eléctrico de 10^7 V/m.

La compuerta flotante se halla rodeada por dióxido de silicio (SiO_2) que es un excelente aislante, de manera que las cargas inyectadas a la compuerta flotante quedan efectivamente atrapadas⁶.

Las celdas que retuvieron cargas quedarán polarizadas en conducción, mientras que las celdas que no retuvieron cargas quedarán polarizados al corte.

Para borrar las celdas programadas con "0", se somete a la memoria a una radiación ultravioleta. Esta radiación de alta energía⁷ excita a los electrones atrapados en la compuerta aislada y les provee la energía necesaria para volver al sustrato y recombinarse.

Las EPROMs están encapsuladas en cerámica con una pequeña ventana circular de cuarzo (transparente a la radiación ultravioleta, mientras que el vidrio común es opaco) para que se pueda borrar dichas memorias. Normalmente con 20 minutos de radiación una EPROM se borrará completamente. Para borrarla con la radiación de un tubo fluorescente se necesitarán tres años o una semana completa de exposición a la radiación solar del 21 de diciembre al mediodía. Por precaución, una vez programadas las memorias, se les colocará una etiqueta opaca sobre el cuarzo para evitar su borrado involuntario.

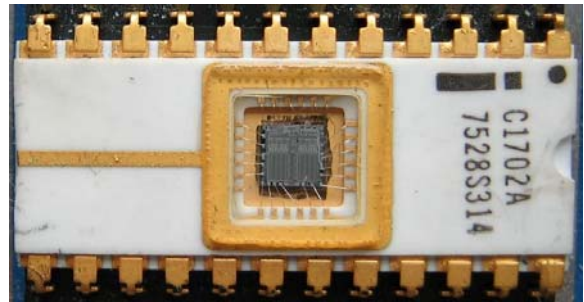


Fig. 8. Encapsulado de una memoria EPROM, en la que se observa la ventana de cuarzo.

Las EPROMs vírgenes no tienen cargas atrapadas, por lo que todos los transistores se hallan cortados y según lo descrito anteriormente todas las celdas se encontrarán en "1".

⁶ Intel garantiza la retención de esas cargas por lo menos por 20 años.

⁷ La energía de una radiación es directamente proporcional a su frecuencia. La frecuencia de la radiación ultravioleta es mayor que la frecuencia de la luz visible, por lo que energía es mayor.

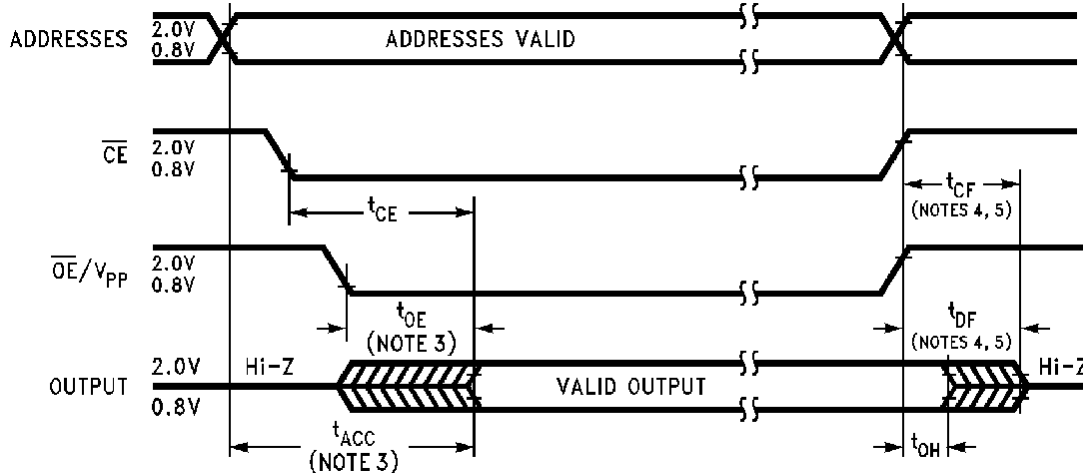


Fig. 9. Diagrama temporal de la lectura de una EPROM.

En la Fig. 9 se puede observar la generación en primer lugar de la señal de selección \overline{CE} , producto de la lógica de decodificación. Cronológicamente después, aparecerá la señal \overline{OE}/V_{PP} . En ella debemos destacar la necesidad que existió de hacer polifuncionales alguna de las patas de conexión de las memorias. En efecto, esta pata servirá para habilitar los buffers de salida (hacia el microprocesador) en la lectura normal o bien para aplicar la tensión de programación en el momento de programación. En la operación normal, esta pata se conectará a la señal de \overline{RD} del microprocesador.

Finalmente, luego de ser seleccionada la memoria y ser habilitados los buffers de salida, el contenido de la celda elegida con las líneas de direcciones, aparecerá sobre el bus de datos.

Un detalle importante es que para el proceso de **hot injection**, la corriente de programación de cada celda es del orden de los 30 mA. Esa importante corriente es imposible de entregar por un generador de tensión interno y debe ser provista externamente.

3.4.2.4 Programación inteligente

La programación de una memoria EPROM puede realizarse en una de dos formas:

- Para los dispositivos de capacidades reducidas (2716 ó 2732) se aplica la tensión de programación (21 V.), simultáneamente se selecciona el dispositivo con \overline{CS} , la dirección de la posición de memoria a programar sobre las líneas A_0 a A_{12} y el contenido de la misma sobre D_0 a D_7 . La programación se realiza aplicando un pulso (activo bajo) de 45 a 55 ms sobre la pata de programación \overline{PGM} . Posteriormente se produce la lectura de esa celda de memoria para verificar que haya sido adecuadamente programada.
- Si cada celda tarda en programarse 50 ms, para programar una 2764 de 8192 posiciones, demoraría como mínimo 6 minutos 50 segundos. Intel introdujo un método de programación abreviada llamada programación inteligente y cuyo diagrama temporal se reproduce a continuación.

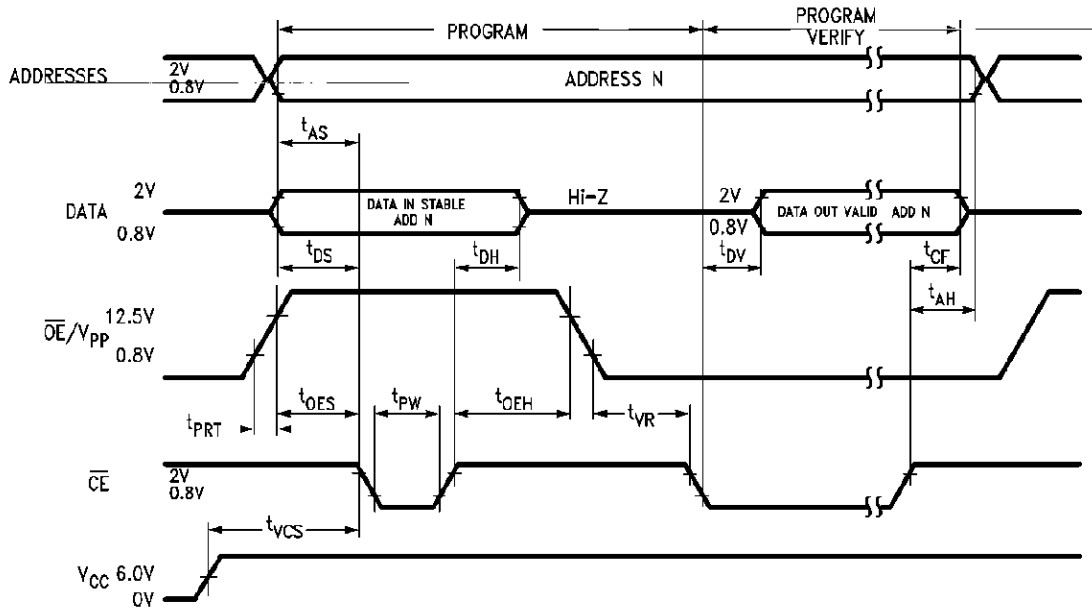


Fig. 10. Diagrama temporal de la programación de una memoria EPROM

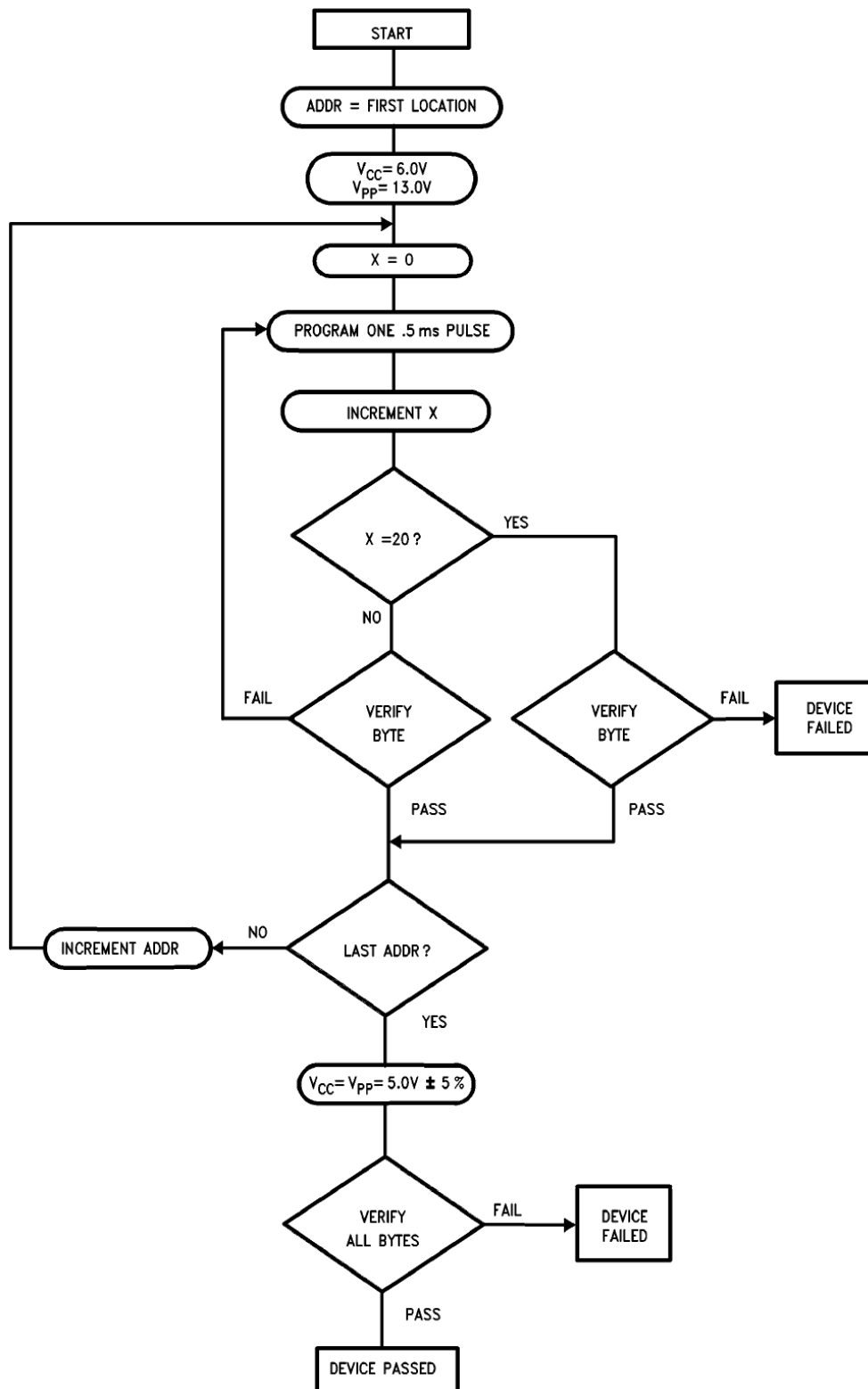


Fig. 11. Diagrama de flujo de la programación inteligente de una Memoria EPROM.

En la misma se aplican pulsos de 0,5 ms con una tensión de alimentación de 6 Voltios y una tensión de programación V_{pp} de 13 Voltios. Se verifica que la posición haya sido adecuadamente programada. En caso contrario se reintentará con esta metodología de programación por 20 veces. Si al cabo de todos esos intentos no se logró verificar la correcta programación se da por fallado al dispositivo. En caso de programación correcta, se pasa a la siguiente posición. Al completar la programación satisfactoria de la totalidad de las celdas, se hace una última verificación de la totalidad de la memoria, comparando lo que se deseaba programar con el real contenido de la misma pero con una tensión de alimentación de 5 Voltios (no de 6 Voltios como en el proceso de programación). En caso de que alguna posición de la memoria difiera entre su contenido real con el teórico, se da por fallada la memoria.

Además, en estas memorias programables en forma inteligente, si se lleva A_9 a 12 V, al forzar A_0 primero bajo y luego a alto se leerán dos bytes que representan respectivamente el fabricante y el código del dispositivo. Esto se hace para que el circuito programador determine de cuál dispositivo se trata para poder adecuar la distribución de patas y la metodología de programación recomendada.

Otros fabricantes (AMD, por ejemplo⁸) han elaborado variantes propias de esta programación inteligente, en las que se mantiene la idea general pero se cambian algunos parámetros por ejemplo que los pulsos de programación son de 100 μ s, la tensión de programación de 12,75 V y la de alimentación durante la programación es de 6,25 V y la de verificación 5,25 V.

Esto pone de manifiesto que un programador de EPROMs genérico deberá imprescindiblemente conocer el fabricante y la denominación de la memoria, ya sea en forma automática o bien ingresándolo por el operador.

3.4.2.5 OTPROM

Para producciones (no para la etapa de desarrollo) de productos ya estabilizados (en los que no se cambiará el programa) y en cantidades pequeñas que no permitan pensar en producir una ROM, se pueden abaratar costos empleando una memoria de similares características a la anteriormente descrita pero sin ventana de cuarzo (costosa de por sí y en su inserción en el encapsulado cerámico) de forma que se puedan programar como una EPROM pero no se las pueda borrar. Estas memorias reciben el nombre de OTPROM (One Time PROM o PROMs de una sola programación).

3.4.3 De lectura mayoritaria.

Las EPROM tienen varias desventajas importantes:

1. La memoria debe ser removida para poder ser borrada.
2. No es posible borrar un solo byte. Todas las celdas deberán ser expuestas a la radiación ultravioleta.
3. La ventana de cuarzo es costosa.

3.4.3.1 E²PROM.

Por todas estas razones es que se han volcado mucho esfuerzos en desarrollar una memoria no volátil borrable en forma eléctrica. El resultado se ha llamado EEPROM (Electrically Erasable EPROM) ó E²PROM.

La diferencia entre la celda de una EPROM y la de una E²PROM, consiste en que la capa de SiO₂ de las EEPROM existente sobre la difusión del drenaje es más delgada. Con una tensión positiva a la compuerta de selección, los electrones son atraídos por la compuerta flotante. Al aplicar una tensión positiva al terminal de drenaje con la compuerta de selección a tierra, descarga (borra) la celda.

⁸ Programming AMD's CMOS EPROMs. Publication #19840 Rev: B January 1997. www.amd.com

Las memorias E²PROM emplean para el fenómeno de recombinación eléctrica (borrado) el principio de la física electrónica denominado hot electron injection (inyección de electrones de alta energía). Ello requiere una importante corriente (varios mili amperes) de la fuente que provee la tensión de programación y obliga a colocar externamente un circuito que provea esa tensión, lo cual encarece la circuitería y aumenta el tamaño del impreso.

Por otro lado, el borrado se debe realizar byte por byte previo a la reprogramación lo cual lleva un tiempo considerable⁹. De hecho las memorias EEPROM se emplean solamente para reemplazos y no para nuevos diseños.

3.4.3.2 Memorias Flash

3.4.3.2.1 Introducción

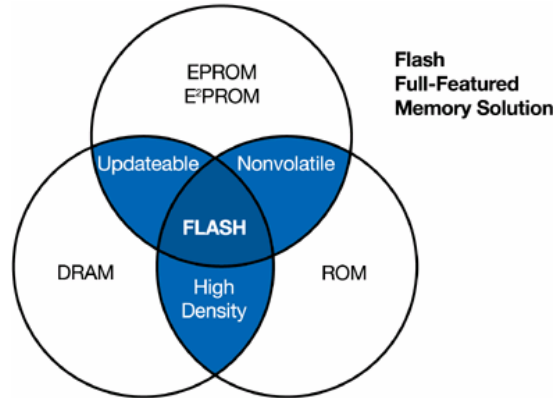


Fig. 12. La memoria Flash como síntesis de características

Las memorias EEPROM fueron un hito en la evolución de los sistemas reconfigurables, pero tanto su densidad como en su tiempo de acceso, dejaron bastante que desear.

Sería deseable mantener las características de no volatilidad y reprogramación de las EEPROM con la alta densidad y bajo consumo de las memorias dinámicas. Estos requerimientos fueron satisfechos con la aparición de las memorias FLASH.

Posteriormente a las memorias EEPROM, se desarrolló la tecnología túnel Fowler – Nordheim bidireccional (tanto para programar como para borrar) que a diferencia de la hot injection, solo requiere unos pocos nano amperes de la fuente de programación, lo cual permite emplear un tipo de generador interno de la tensión de programación denominado bombeador de cargas (charge pump) que consiste básicamente en un oscilador interno y un rectificador – enclavador.

Para el usuario externo, la memoria sólo requiere la tensión de alimentación sin detectar que internamente se está generando la tensión de programación.

⁹ <http://whatis.com/flashmem.htm>

Por otro lado, el borrado no se realiza a nivel de byte (como en las EEPROM) sino a nivel de bloques¹⁰, borrando de una sola acción un conjunto importante de posiciones de memoria, permitiendo a partir de allí la programación individual de cada byte. Justamente el nombre Flash proviene del borrado "instantáneo" de un bloque de la memoria, permitiendo economizar tiempo al no tener que borrar individualmente cada byte.

3.4.3.2.2 Tecnología original.

Intel, uno de los innovadores en tecnología flash, desarrolló una celda como la que se muestra en la **Fig. 13**. En ella se observa que la celda es muy similar a la de una celda EPROM, pero con una separación entre la compuerta y la fuente mucho más fina y más precisamente producida.

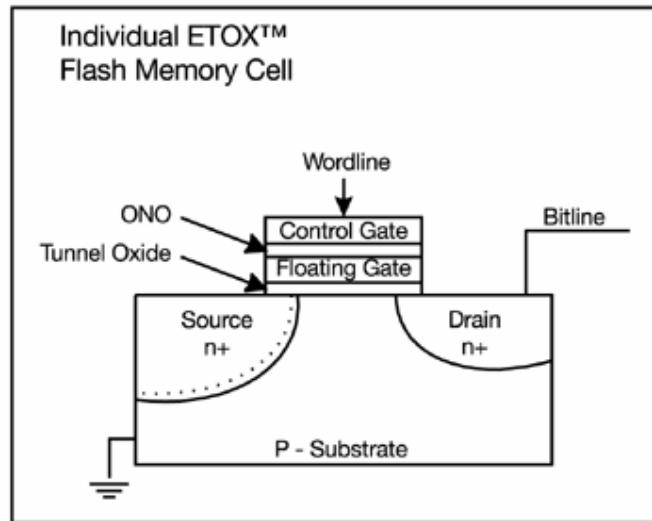


Fig. 13. Celda de una memoria Flash.

La programación se realiza alterando el equilibrio electrónico en la compuerta flotante. Una tensión de 10 a 13 Volts (generada desde el generador de tensión interno) se aplica entre el drenaje y la compuerta de control. Ello provoca que el transistor MOS se comporte como un cañón electrónico (como los de los tubos de rayos catódicos) emitiendo electrones que quedan atrapados en la compuerta flotante.

Las cargas excitadas por el fenómeno anterior, provienen de la línea de bit (datos) y atraviesan el dieléctrico que separa el sustrato P de la compuerta aislada. La compuerta aislada queda cargada negativamente y actuará como barrera entre la compuerta de control y compuerta flotante.

Un dispositivo específico llamado **cell sensor** monitorea el nivel de cargas almacenadas en la compuerta aislada. Si es mayor al 50% de la tensión de programación se computa como "1". Si es menor como "0".

Una flash virgen o borrada tiene todos sus bits en "1".

3.4.3.2.3 Tecnología Strata Flash.

La más reciente evolución tecnológica de Intel, ha desarrollado un tipo de celda llamada Strata Flash. Esta tecnología permite almacenar dos bits en cada transistor.

El secreto de esta tecnología radica en generar cuatro niveles distintos de tensión para marcar cuatro valores lógicos (no sólo “0” y “1” como hasta ahora). Estos cuatro valores lógicos serán codificados como 00, 01, 10, y 11 (borrado), como se puede ver en la Fig. 14.

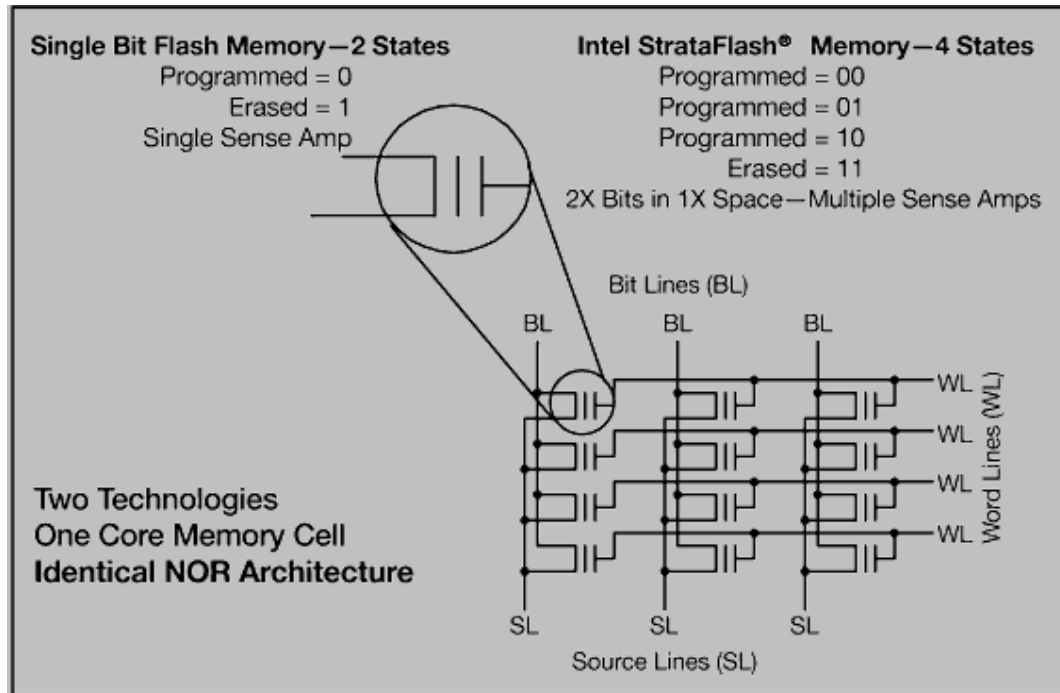


Fig. 14. Celdas de un solo bit y de dos bits.

Para ello se requiere una importante precisión tanto del proceso de grabación como de lectura de la celda y empleando una topología NOR multi - nivel muy confiable ideada para aplicaciones de alta densidad.

3.4.3.2.4 Estructura interna de las Flash

Las memorias flash se caracterizan por estar divididas en bloques. Esa disposición facilita y agiliza el borrado pues en lugar de borrar un byte se borra un bloque completo y también simplifica la reprogramación pues se puede alterar un solo bloque de memoria sin modificar los demás. Como veremos más adelante, esta sectorización también va a facilitar la operación de lectura.

En el caso de la AT29C010A de AMD los 128 Kbytes están organizados como 1024 sectores de 128 bytes o bien la 28F128J3A de Intel (16 Mbytes) compuesta por 128 bloques de 128 Kbytes cada uno de ellos.

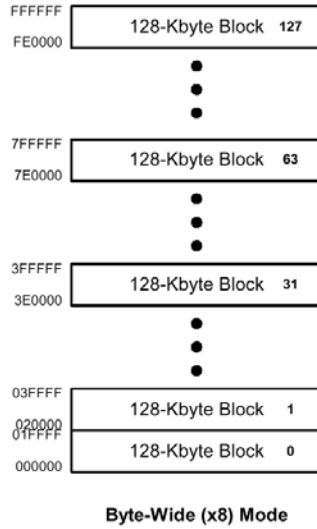


Fig. 15. Esquema de una memoria Flash Intel de 16 Mbytes

3.4.3.2.5 Detalles de la programación de las memorias Flash¹¹.

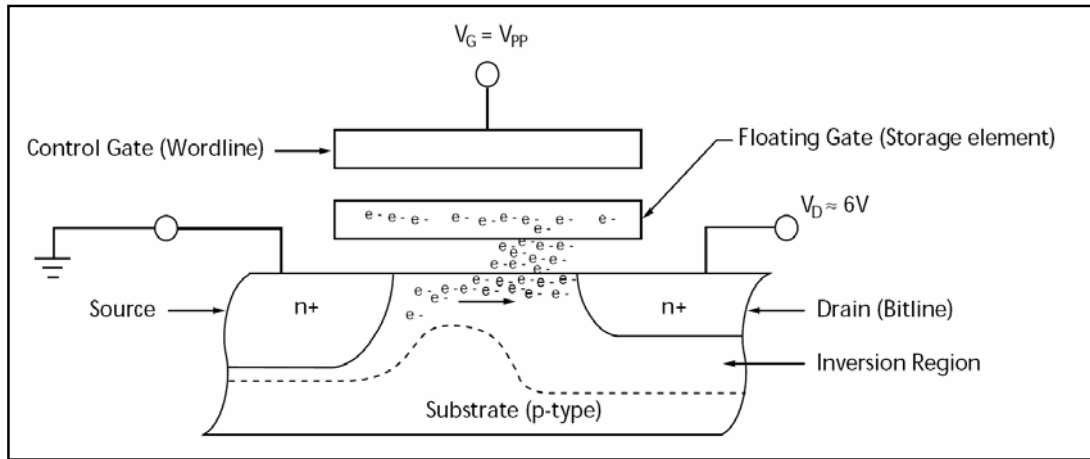


Fig. 16 Programación de un bit de una memoria Flash

Se observa en la Fig. 16 el proceso por el cual al aplicar una tensión de alimentación de 6 Volt y una tensión de compuerta de control de V_{pp} se genera una conducción en el canal y esos electrones saltan a la compuerta aislada donde podrán permanecer años si no son borradas antes.

¹¹ Atmel Flash Programmable Erasable ROM – Application Note 1 (AN-1)

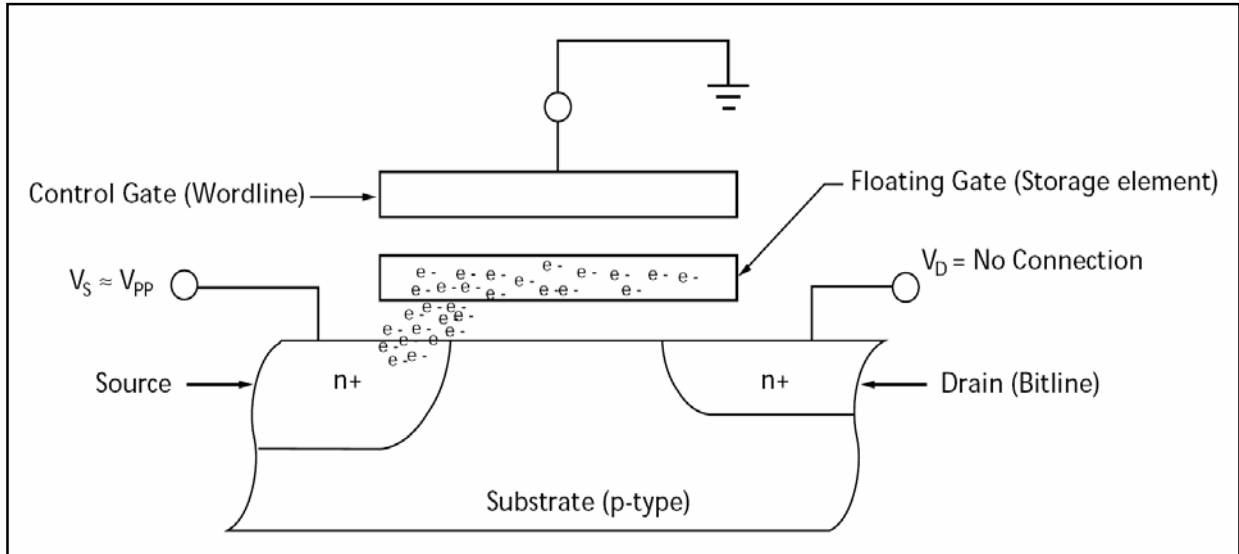


Fig. 17. Borrado de una celda Flash

En la Fig. 17 se ve el proceso de borrado de una memoria flash. Se aplica la tensión de programación a la fuente mientras que la compuerta de control se conecta a 0 V. Ello produce la migración de las cargas almacenadas en la compuerta aislada hacia la fuente donde se recombinan y queda la compuerta aislada totalmente descargada.

Debe destacarse que el proceso de grabado y borrado de las celdas Flash emplean mucha menos energía que para una memoria EEPROM, por lo que se logran densidades importantísimas.¹²

Estas memorias incorporan la novedad de tener registros que permiten saber el fabricante (byte 00000), el modelo de memoria (byte 00001) como podemos observar en la Fig. 18. Esa información es imprescindible para que los programadores universales fijen los tiempos y las características propias de la programación.

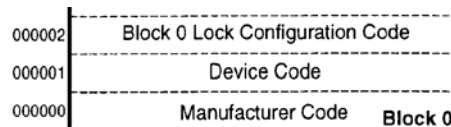


Fig. 18. Ubicación de los códigos de fabricante y dispositivo

¹² Las memorias Flash que se emplean tanto en los pen-drives como en las tarjetas de expansión de cámaras fotográficas, palms, etc. Hoy supera el Megabyte, obteniéndose precios por byte inferiores al 0,01 centavos de dólar.

Si bien los costos actuales por byte son superiores a de los discos rígidos tradicionales, la tendencia es a la disminución de las diferencias y al aumento de la densidad, por lo que se estima que en 5 años estarán en valores comparativos.

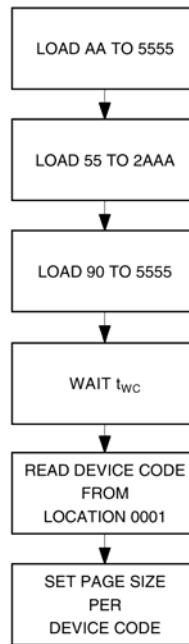


Fig. 19. Lectura del byte identificador de cada memoria Flash.

Para acceder a estos bytes identificatorios, se deberá habilitar una operatoria especial escribiendo tres bytes prefijados en tres posiciones de memoria prefijadas en forma consecutiva (**Fig. 19**), luego esperar el tiempo de ciclo de escritura (t_{wc}) y leer en la posición 00001 un código que permite identificar a de la memoria. Con esta identificación de la memoria se puede fijar el tamaño de los sectores y características generales de la programación. Toda esta operación se realiza con la tensión normal de operación.

Si se desea, también se puede acceder a este byte identificador aplicando 12 Volts a la pata A9 de la misma. Esta metodología se mantiene por compatibilidad con las EPROM.

Para verificar la culminación de la programación, se puede leer repetidas veces una posición de memoria cualquiera. Mientras se este programando el bit 6 cambiará de estado de lectura a lectura. Cuando se termine de programar el sector deseado, dicho bit se mantendrá constante.

Por medio de una secuencia similar a la de la determinación del tipo de memoria a programar, se puede dar el comando de proteger o borrar un determinado sector de memoria o bien el de copiar un buffer interno de 32 bytes (el que es normalmente escrito por el programador) sobre una determinada zona de la memoria.

El estado en que se encuentra la memoria después de haber iniciado alguno de los procesos anteriores, se puede determinar leyendo un registro de estado disponible.

3.4.3.2.6 Acceso paginado de las memorias Flash.

Estas memorias suelen tener un modo de acceso a la información de mayor velocidad, denominado de acceso paginado.

La idea consiste en mantener un conjunto¹³ de los bits mas significativos del bus de direcciones fijos apuntando a una dada página (o sector) y por medio del cambio de los bits menos significativos solo se elige el byte dentro de la página, dejando la señal de habilitación (/CE), de habilitación de los buffers de salida (/OE) permanentemente activas, permitiendo aumentar la velocidad de lectura al tener la misma página permanentemente seleccionada.

¹³ La cantidad de líneas de direcciones a mantener fijas depende del tamaño del sector y por ende del byte identificador que se lea al comenzar la programación.

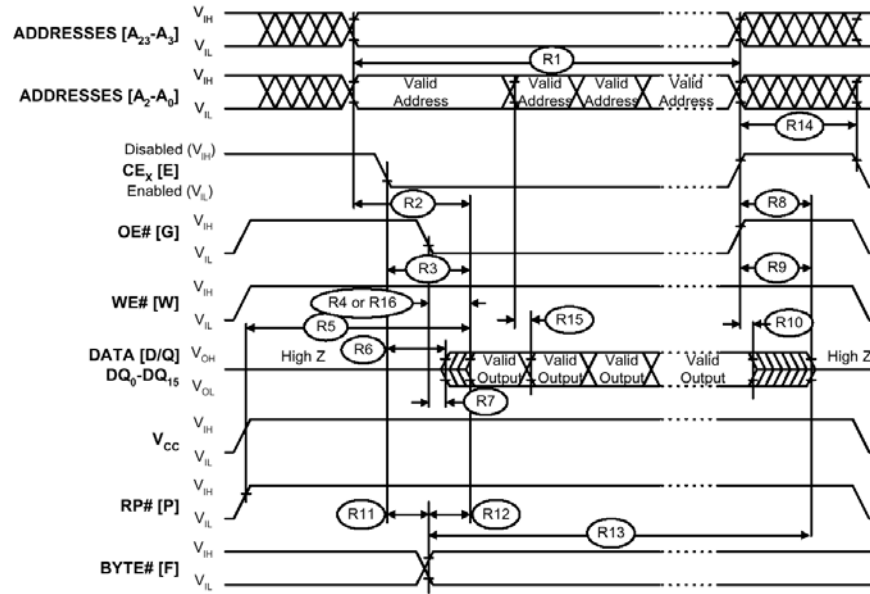


Fig. 20. Operación de lectura paginada.

Las memorias flash, al no requerir circuitería externa pueden ser programadas directamente sobre el impreso final. Es común que los fabricantes de equipos “a medida” o actualizables, preparen el circuito con las flash sin programar y en el momento de la venta de un equipo se programe la memoria con la última versión de programa disponible o con el programa requerido por el usuario.

Existen algunas variantes en la estructura interna de la memoria, como la que también presenta Intel para sus últimos modelos y que vemos en la Fig. 21.

3.4.3.2.7 Bloques simétricos y asimétricos.

Puede darse la situación de tener bloques simétricos o asimétricos (de distinto tamaño). En el caso del asimétrico, el bloque más pequeño se puede emplear para almacenar parámetros fundamentales o la rutina de arranque de una PC, ambos que requieren un tratamiento especial.

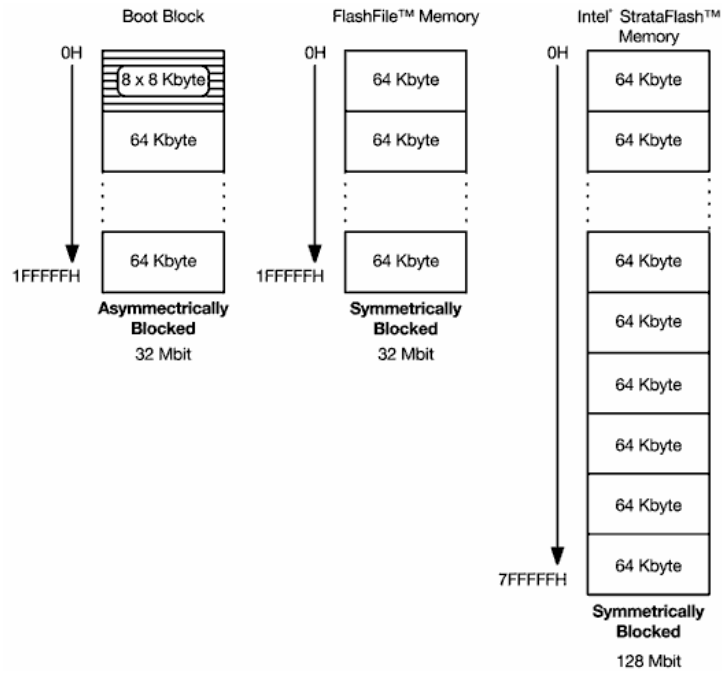


Fig. 21. Arreglo de memoria flash simétrico y asimétrico.

3.4.3.2.8 Modos de trabajo.

Una flash puede ser accedida por varios motivos. Tres de los cuales son:

- a) Para ser leída.
- b) Para ser escrita.
- c) Para ser borrada.

En el caso de lectura, se presentan tres modos de lectura:

a.1) **Lectura asincrónica:** la lectura tradicional.

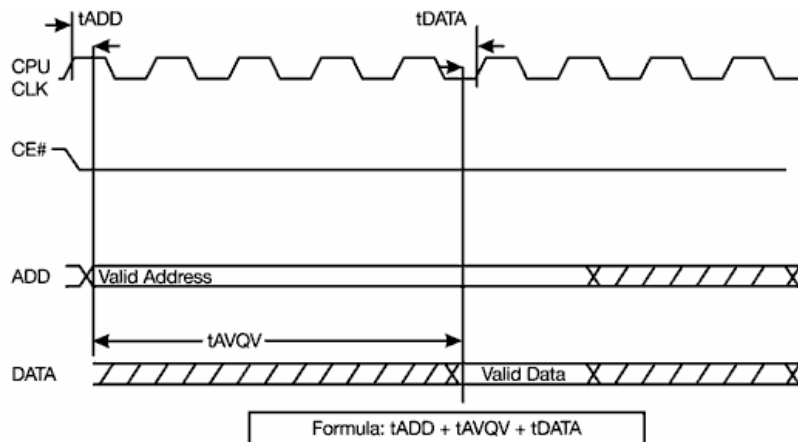


Fig. 22. Lectura asincrónica

El significado del recuadro inferior "Formula" se refiere al tiempo que transcurre desde el primer ciclo de reloj de la lectura hasta que los datos se hallan presente y estables en el bus de datos.

La interpretación cabal de esta expresión la tendremos después de ver "5 - **Tiempos de características**" de una memoria.

Tiempo de acceso.

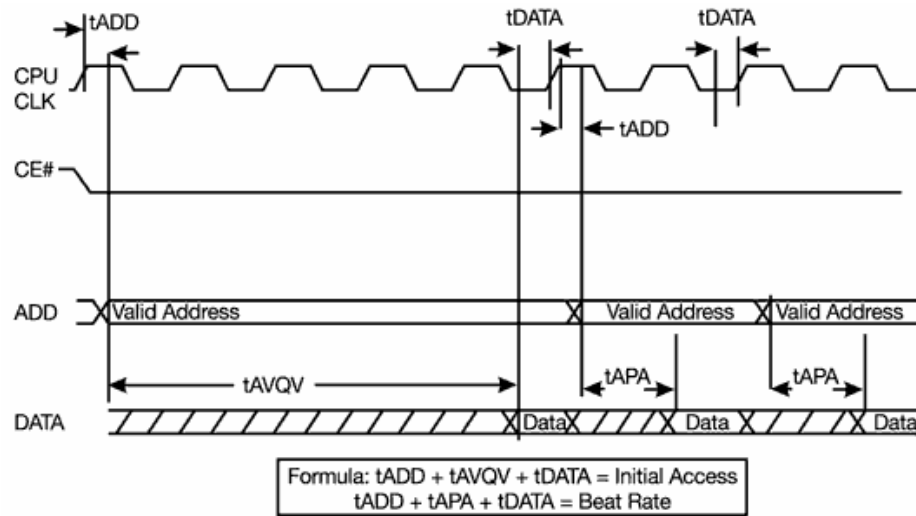


Fig. 23. Acceso paginado asincrónico.

a.2) Modo paginado asincrónico: Mejora el comportamiento del sistema. Consiste en transferir una página de la Flash a un buffer de memoria interna de alta velocidad. El primer ciclo de lectura se realizará a la velocidad estándar pero los subsiguientes a una velocidad de 20 ns por byte.

a.3) Acceso sincrónico por ráfagas: Permite operar hasta a 66 MHz. Requiere de tres señales de interfaz. Una entrada de reloj (CLK), una salida de sincronización con el microprocesador (/WAIT, como se verá en **5. Tiempos de características** de una memoria.

Tiempo de acceso.) y otra entrada ADV (address valid Input o entrada de validación de direcciones) y que se emplea para indicarle a la flash que se ha colocado una dirección válida. En esta operación por ráfagas, la Flash retiene la dirección inicial y a partir de ella, comienza a poner sobre el bus de datos los contenidos de las sucesivas posiciones de memoria en cada ciclo de reloj.

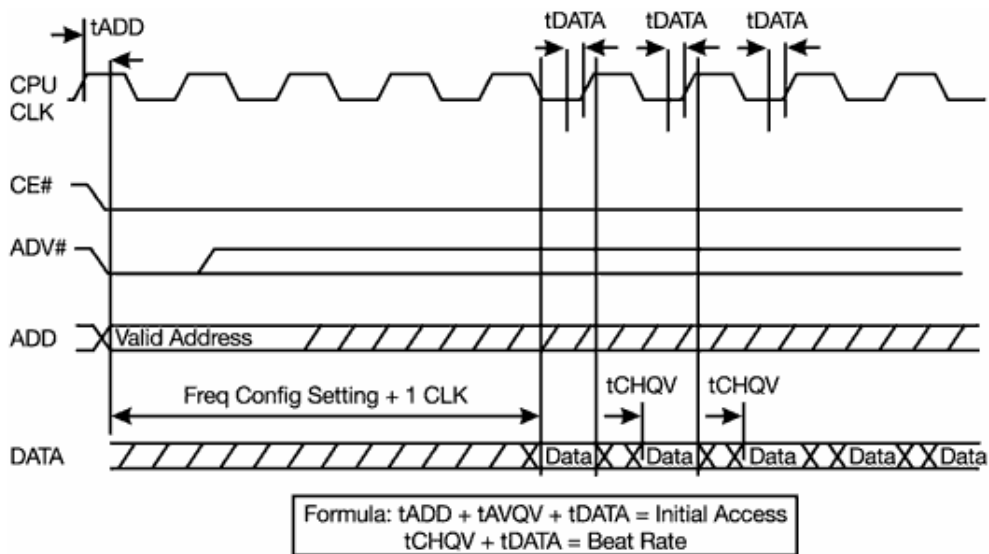


Fig. 24. Lectura sincrónica por ráfagas.

3.4.3.2.9 Aplicaciones de la flash.

En las placas base de las PCs actuales se ha incorporado memorias flash de forma que el usuario de la PC pueda actualizar vía Internet su memoria BIOS sin tener que quitarla de su zócalo.

Justamente esa facilidad fue la que permitió que varios virus actuasen sobre las flash desprogramando la BIOS y dejando inutilizable la computadora. Al respecto los fabricantes de placas base tomaron dos medidas:

1. Protegieron algunos bloques de la flash (los fundamentales para permitir la recuperación del sistema desde Internet en caso de catástrofe).
2. Instalaron la FLASH en un zócalo (las primeras versiones vinieron soldadas a la placa) de forma que en caso de emergencia pudiera ser retirada y cambiada por una nueva.

Las principales características de las memorias flash actuales son¹⁴:

- Tiempo de acceso a una página (se profundizará posteriormente): 25 ns.
- Tiempo de acceso inicial: 65 ns.
- Tensión de alimentación típica (correspondiente a los microprocesadores actuales) 2,7 a 3,6 V ó 5 V.
- Por lo menos un millón de operaciones de borrado – escritura por bloque.
- Buffer de escritura (típicamente de 32 bytes), para que la CPU transfiera a alta velocidad una cantidad de bytes dada por el tamaño del buffer y se pueda dedicar a otra actividad hasta que se terminaron de grabar esos 32 bytes.
- Protección contra escritura de zonas de memoria sensibles.
- Borrado por bloques.
- Identificación en determinados registros internos de la memoria de sus principales características para que el programa de grabado pueda ser flexible y adaptarse a distintos tipos de memorias.

Las memorias flash han encontrado innumerables aplicaciones en PCs actualizables, teléfonos celulares, cámaras digitales, ruteadores de redes, tarjetas PC Card (antes PCMCIA) para computadoras portátiles, reproductores de música en formato MP3, etc.

3.4.3.3 FeRAM.

	FeRAM	SRAM	DRAM	EEPROM	Flash
non-volatile	✓	✗	✗	✓	✓
small cell size	✓	✗	✓	✗	✓
word-wise read&write	✓	✓	✓	✓	✗
low-power capable	✓	✓	✓	✗	✗
cheap	✓	✗	✓	✗	✓
10 ¹³ write cycles	✓	✓	✓	✗	✗
fast data write	✓	✓	✓	✗	✗

Fig. 25. tabla comparativa.

¹⁴ <http://www.amd.com/products/nvd/techdocs/techdocs.html>

Este tipo de memoria se halla en fase experimental y dice reemplazar los flip – flops de las memorias semiconductoras por pequeñísimos dominios magnéticos que permitirán lograr memorias no volátiles de dimensiones muy similares a las actuales, pero con un consumo infinitamente menor, al punto que una computadora portátil experimental, puede estar en el modo suspendido (hibernado) por semanas con la cara de una batería estándar.

En la Fig. 25 vemos una tabla comparativa presentada por el fabricante de las FeRAM¹⁵ que muestran sus potenciales ventajas. La historia tecnológica ha demostrado que no siempre los productos que parecen muy ventajosos en su planteo, terminan siendo viables en su producción masiva.

3.4.3.4 Memorias de lectura y escritura con respaldo (backup) de batería.

Desde la aparición de las memorias CMOS se implementó un sistema sencillo, económico y confiable que permite almacenar la información trascendente en caso de caída de la alimentación. El sistema consiste en disponer de circuito de conmutación que permita que la memoria opere con la tensión de alimentación cuando la misma este presente o bien que conmute la alimentación (sólo de la memoria) a una batería apropiada cuando la alimentación principal desaparezca.

Sobre este circuito de conmutación deberá caer una tensión despreciable, ya que aún algunas memorias pueden presentar el fenómeno de latch-up¹⁶. Este fenómeno estuvo presente en las primeras generaciones de memorias y consiste en que al ser 0,6 V mayor la tensión de alguna de las líneas de direcciones o datos de la memoria que la tensión de alimentación, se dispara un tiristor parásito de dichas memorias. Dicho tiristor sólo se puede desactivar desconectando la memoria y la excesiva circulación de corriente puede dañar en forma irreversible a la memoria.

Un circuito comercial que permite realizar esta conmutación es el MAX691 de MAXIM Semiconductors, que tiene un detector de nivel de tensión de alimentación inferior a los 4,85 V. En tal caso se conmuta automáticamente a la tensión de la batería y fuerza la señal de /WR de la memoria a "1", garantizando que no se escriba la misma en forma indeseada durante el lapso en que el microprocesador deja de actuar normalmente.

Este circuito tiene otras funciones como ser la de resetear al microprocesador y tiene incorporado un monoestable de supervisión (watchdog) de la operación normal del sistema.

3.4.4 Nomenclatura

Históricamente, los fabricantes de memorias siguieron ciertas pautas que fueron respetadas por las nuevas empresas que ingresaron en el mercado. En líneas generales, los fabricantes coincidieron en designar a sus memorias como:

2 N mmm

Donde N indicará el tipo de memoria de que se trata:

¹⁵ www.infineon.com

¹⁶ En las hojas de datos debería decir que la memoria RAM CMOS es "latch-up free", o sea exenta del fenómeno de latch-up.

N	Significado
3	ROM
7	EPROM
8	E ² PROM
9	FLASH

Y *mmm* representa la cantidad de kilo bits de la memoria. Entonces, por ejemplo, una 27256, es una memoria EPROM (N = 7) de 256 kilo bits, o sea de 32 Kbytes.

En las memorias EPROM y FLASH de última generación, al incrementarse la capacidad, se cambió la nomenclatura. Una memoria EPROM CMOS de 1 Megabit debería designarse como 27C1024. Sin embargo de la conoce como 27C010. En este caso *mmm* = 010 representa la cantidad de kilobits de la memoria.

Así entonces la memoria EPROM de AMD **27C040**, será una CMOS de 4 Megabits que esta configurada como 512 Kbytes.

3.4.5 Resumen.

Memoria	Características
SRAM	Static Random-Access Memory. Muy alta velocidad, alto consumo de energía, moderada densidad de integración que aumenta los costos de una determinada cantidad de memoria (por fuente de alimentación, cantidad de integrados y área de impreso)
DRAM	Dynamic Random Access Memory. Alta densidad, bajo costo, alta velocidad, moderado consumo de energía.
ROM	Read-Only Memory. Sistema maduro, alta densidad, confiable, bajo costo por unidad, necesidad de máscara que consume tiempo de generación y alto costo. Recomendable para altos volúmenes de producción con código estable (no sujeto a modificaciones).
EPROM	Electrically Programmable Read-Only Memory. Alta densidad. Debe exponerse a la radiación ultravioleta para su borrado.
EEPROM or E2PROM	Electrically Erasable Programmable Read-Only Memory Borrable a nivel de byte, menor confiabilidad, alto costo, menor densidad de integración, alto consumo de programación.
FLASH	Bajo costo, alta densidad, alta velocidad, arquitectura preparada para alta velocidad, bajo consumo de energía, alta confiabilidad

3.5 Según su acceso

3.5.1 Secuencial.

En estas memorias el tiempo que tarda en obtenerse un dato depende de la ubicación (dirección) del mismo dentro del dispositivo. El ejemplo típico es la información almacenada en una cinta magnética. La información será accedida en un tiempo muy corto si se halla al comienzo de la cinta y demorará más tiempo de acuerdo a su ubicación dentro de la misma.

3.5.2 Aleatorio (o al azar).

En estas memorias el tiempo que tarda en obtenerse la información es independiente de la dirección de la celda seleccionada. Ese tiempo será el mismo ya se trate de la posición 0000H o bien de la FFFFH.

Bajo esta definición se esconde una de los tantos usos inadecuados o bien paradójicos que se dan en la terminología técnica.

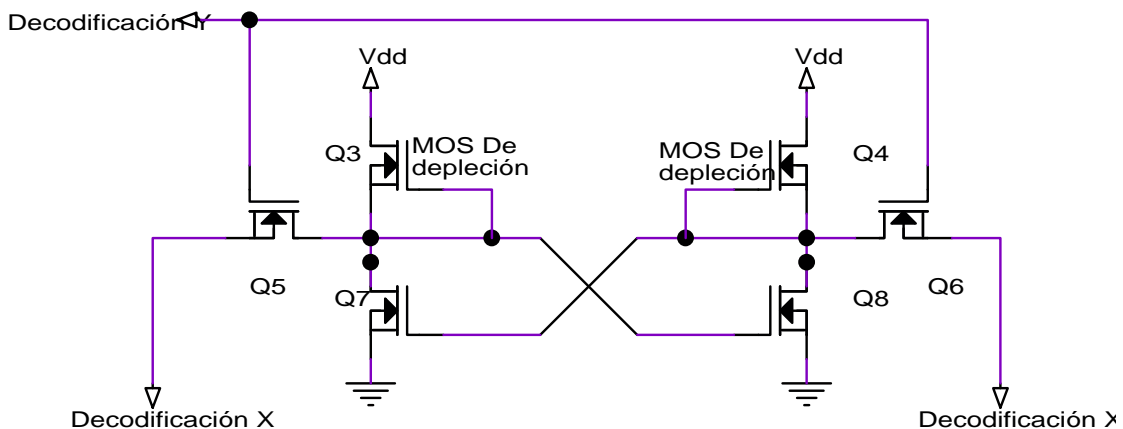
Las memorias ROM descritas en **De lectura solamente**, son de acceso aleatorio y por ende se las debe agrupar como RAM (Random Access Memories o Memorias de Acceso Aleatorio).

Normalmente se presentan como contrapuestas las memorias ROM (o EPROM) para almacenar el programa y las de lectura y escritura, que almacenan variables, mal denominadas RAM, que, como describimos recién caracterizan otro concepto y no el uso de la memoria¹⁷.

4. Memorias Dinámicas

4.1 Introducción.

Las memorias estáticas tradicionales, como la que se observa en la Fig. 26, están formadas por 6 transistores (4 para el flip-flop propiamente dicho y dos de conexión) son aquellas en las que las celdas se hallan formadas por multivibradores biestables. En dichos Flip-Flops¹⁸, uno de los dos transistores se halla cortado, mientras que el otro se encuentra en estado de conducción, esto produce una considerable disipación de energía, limitando consiguientemente, la cantidad de Flip-Flops que se pueden integrar por unidad de superficie.



¹⁷ Probablemente esta designación proviene de las primeras computadoras en las que el programa se hallaba en tambores magnéticos (secuenciales) mientras que la memoria de datos era de acceso aleatorio y estaba constituida por flip-flops con relevadores o válvulas de vacío.

¹⁸ Los transistores MOS de depleción, al unirse la compuerta con la fuente, se comportan como resistores y son mucho más fáciles de integrar que los resistores.

Fig. 26. Flip Flop básico con transistores MOS

Si bien esta estructura resultará la más rápida para leer o escribir, la densidad de integración es moderada y presentaremos a continuación otro tipo de memorias que si bien son algo más complejas para operar y que requieren alguna lógica adicional, con ellas se pueden obtener una extraordinaria cantidad de bits almacenados por unidad de superficie.

Estas memorias en las que el elemento o celda de almacenamiento es un capacitor reciben el nombre de memorias dinámicas. El estado de carga del mismo representa la información contenida (un uno o cero lógico).

Si los capacitores y transistores que se utilizan fueran ideales, se consumiría energía solamente en la escritura (transferencia de cargas al capacitor), lo que permitiría que el área requerida por bit de información sea muy pequeña.

Para la construcción de este tipo de memorias se utilizan transistores MOS debido a las pocas pérdidas asociadas a este tipo de dispositivos, pero como los capacitores y el resto del circuito a ellos conectados no son perfectos, con el tiempo los capacitores tienden a descargarse, perdiéndose por ende la información asociada. Este problema se soluciona reconstruyendo periódicamente la carga de cada una de las celdas. A esta operación se le denomina **refresco** y es fundamental para la conservación de la información.

4.2 Estructuras típicas de la celda.

4.2.1 Celda de un transistor.

Como el pequeño tamaño de las celdas es una de las mayores ventajas de las memorias dinámicas, la celda ideal es precisamente un capacitor y un transistor. Como se ve en la figura, el dato se guarda como carga en el capacitor C1, y se aísla de las líneas de selección por medio del transistor Q1.

Para leer la información de la celda, se selecciona la misma por medio de la línea "ROW SELECT" llevando a la conducción al transistor Q1.

De esta manera el nivel lógico representado por la carga de C1 pasa a la línea de columna y la corriente que se establece por la misma es detectada como "0" o "1" por un amplificador. Esta lectura es destructiva, pues todo ocurre como si colocásemos dos capacitores en paralelo. Uno es el capacitor que se halla en nuestra celda y el otro (bastante mayor) es el de la línea de lectura. Al repartirse las cargas, el capacitor que almacenó la información, queda prácticamente descargado.

Por este motivo es que se coloca un circuito encargado de reponer automáticamente las cargas originales en el capacitor de datos. Este hecho es trascendente para lo que denominamos refresco, pues la lectura de una celda automáticamente repone las cargas perdidas.

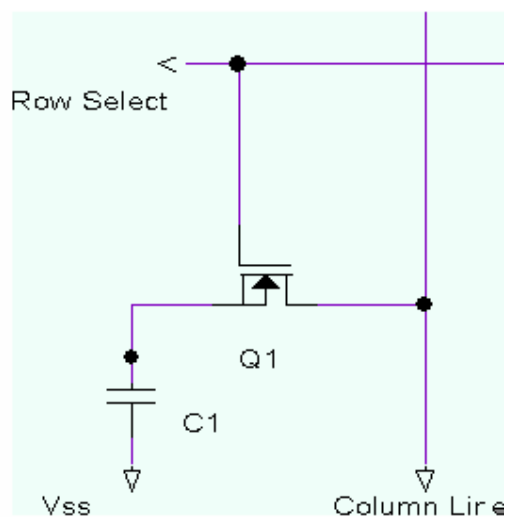


Fig. 27. Celda dinámica de un transistor

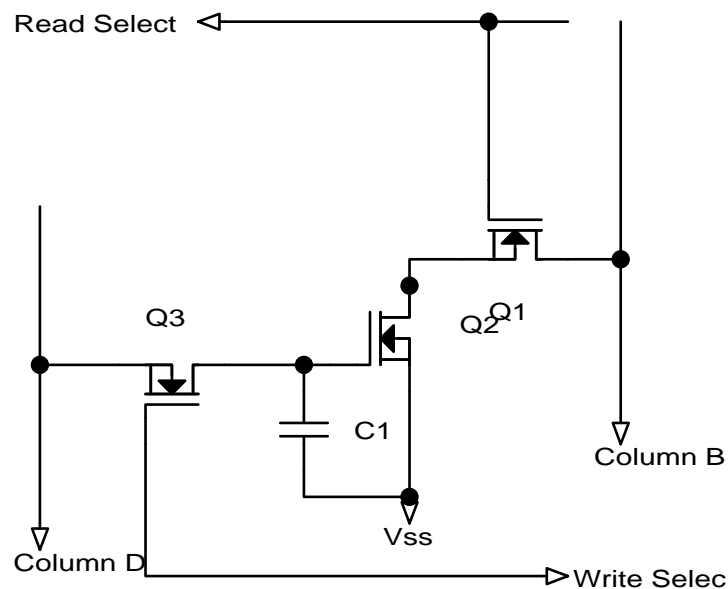
La escritura se efectúa seleccionando la celda por medio de la línea "ROW SELECT", y el dato a ingresar se coloca en la línea de columna. Posteriormente se deselecciona la celda cortando a Q1.

El refresco del dato se realiza efectuando una operación de lectura.

4.2.2 Celda de 3 transistores.

En esta configuración se hace uso de la ganancia de tensión proporcionada por el transistor Q1. A pesar que esta celda posee más componentes y líneas de acceso que la anterior, sus dimensiones son similares a la celda de un transistor, dado que el capacitor será de menor valor ocupando una menor área.

La información es almacenada en forma de carga por el capacitor (en este caso la capacidad asociada con la compuerta de Q1).

**Fig. 28.** Celda dinámica de tres transistores.

Para acceder a la celda las columnas B y D son precargadas a un potencial próximo a Vdd al comienzo del ciclo. La línea de selección de lectura (Read Select) se habilita posteriormente, y el transistor de transmisión Q2 conduce. Si la tensión de la compuerta de Q1 excede de su tensión de umbral, Q1 conduce y la columna B se descarga llegando a un potencial próximo a Vss. Si la tensión en la compuerta de Q1 no es suficiente para hacerlo conducir, la columna B se mantiene precargada a una tensión negativa. El estado de la columna B es luego inspeccionado por el amplificador interno del C.I. (sense amplifier).

Para refrescar este tipo de celda la información de la columna B debe ser invertida y reescrita en la misma. Esto es realizado por un amplificador de refresco que descarga la columna D si la B permanece precargada, y viceversa.

Se habilita posteriormente la línea de selección de escritura (Write Select) y la compuerta de Q1 adquiere una tensión próxima a la de la columna D, refrescando la celda.

Para un ciclo de escritura, sólo debe colocarse la nueva información en la columna D, y habilitarse dicha selección.

4.2.3 Celda de cuatro transistores.

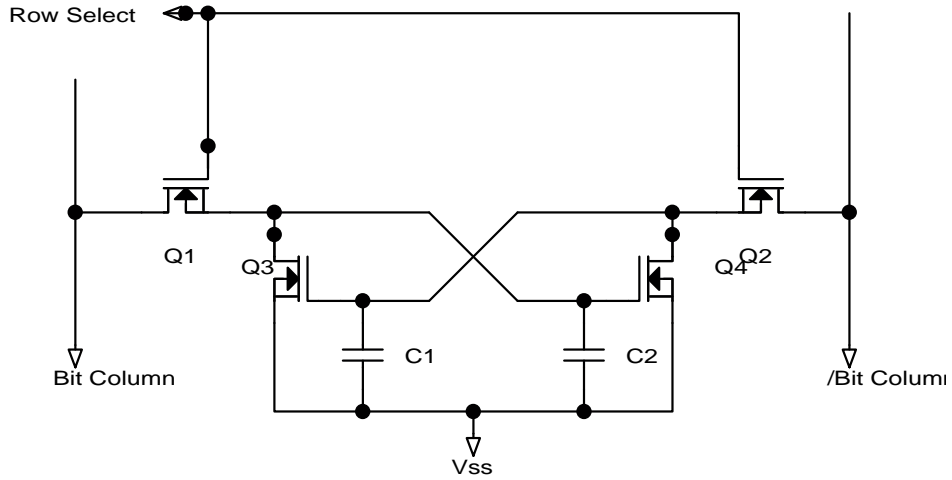


Fig. 29 Celda dinámica de cuatro transistores.

Los dispositivos activos Q1 y Q2 se asemejan en su conexión a un Flip-Flop, pero como no existen resistores de carga, el dispositivo resultante es dinámico, no habiendo circulación permanente de corriente por ninguno de los transistores.

La información es almacenada en las capacidades de compuerta de Q3 y Q4. De los drenajes de los mismos se extraen la información y su complemento, lo que simplifica la actuación de la lógica de refresco.

El mismo consiste en hacer conducir a Q1 y Q2 por medio de la línea ROW SELECT, leyendo el estado de los capacitores en la línea de BIT. Dicho valor es amplificado e invertido por el amplificador exterior y aplicado a la línea /BIT COLUMN, restaurándose de tal forma las cargas perdidas por el capacitor.

El procedimiento de lectura consiste en aplicar en la selección de fila (ROW SELECT) una tensión tal que haga conducir los transistores Q3 y Q4, permitiendo que el estado de los capacitores llegue a las líneas de bit.

Para la escritura se coloca la información sobre las líneas de bit a la vez que se habilita la escritura por medio de la activación de la señal de selección de fila.

4.2.4 Estructura interna de una memoria. Señales /RAS y /CAS

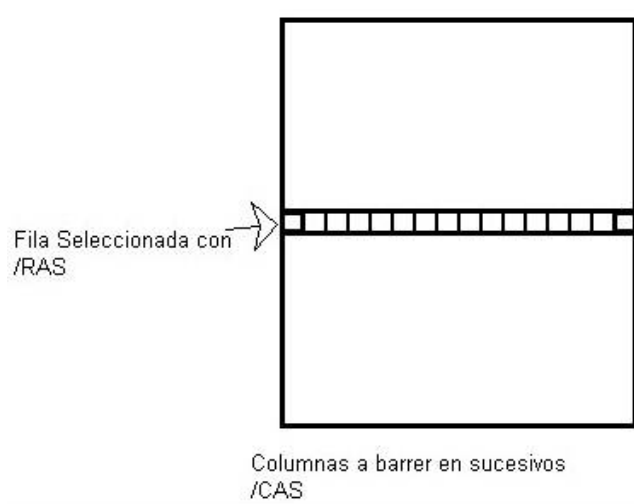


Fig. 30. Bosquejo de una memoria de 16 x 16 bits

Siempre hemos dicho que para optimizar el aprovechamiento del área del integrado se buscaba implementar los circuitos bajo una estructura cuadrada. Ahora le vamos a encontrar una segunda aplicación.

Supongamos tener una memoria de 256 x 1 (256 posiciones por 1 bits de ancho). Al llevarla a estructura cuadrada nos quedaría una estructura de 16 x 16 como vemos en la Fig. 30. Para seleccionar una celda, necesitaremos 4 bits para elegir la fila y otros cuatro bits para elegir la columna.

A fin de minimizar la cantidad de patas empleadas en el integrado, podemos utilizar solo 4 líneas de direcciones y enviar el bus de direcciones multiplexado y agregar una línea para indicar en que momento aparece el direccionamiento de filas (/RAS o Row address strobe) y otra para indicar la presencia de la dirección de columnas (/CAS o Column Address Strobe). Internamente se almacenará en latches una mitad de la dirección hasta que aparezca la otra.

Esta novedad es significativa en memorias de alta capacidad como por ejemplo una de 64 M x 8. Para direccionar 64 Mega palabras necesitaríamos 26 líneas de direcciones, pero con el multiplexado de las mismas solo requeriremos 13 líneas de direcciones más dos para /RAS y /CAS.

Esto además se puede emplear para el refresco de la memoria. En efecto, al leer se leerá una fila completa y por lo tanto se refrescará toda una fila.

Eso es de particular importancia en el manejo de tiempos. Supongamos que tenemos una memoria de 4096 bits en configuración 4096 x 1. El tiempo máximo entre refrescos para esa memoria puede ser de 4 ms (excedido ese tiempo, se pierden los datos almacenados en los capacitores). Ello nos permite calcular el tiempo entre refrescos como:

$$T_{\text{refresh}} = 4 \text{ ms}/4096 = 976 \text{ ns.}$$

Esta memoria tendrá un tiempo mínimo entre lecturas (será definido posteriormente) de, por ejemplo, 120 ns. De todo lo anterior concluimos que casi cada 8 accesos a memoria se deberá dedicar un ciclo al refresco lo cual lentifica al sistema en aproximadamente un 12,5%.

Si aplicamos la anteriormente mencionada estructura cuadrada, las 4096 celdas se repartirán en 64 filas y 64 columnas. Si al leer una fila leemos 64 columnas, al refrescar una fila, refrescaremos 64 columnas.

El cálculo del tiempo entre refrescos será entonces:

$$T_{\text{refresh}} = 4 \text{ ms}/64 = 62,5 \mu\text{s}$$

Ahora, pagando el precio de una ligera complicación del circuito interno de la memoria para elegir el bit de una determinada columna, deberemos refrescar una fila cada 520 accesos a memoria lo cual lentificará al sistema en menos del 0,2 %.

4.2.5 Resumen.

El área requerida para implementar una memoria depende del proceso elegido por el fabricante y por el tipo de celda seleccionada por el mismo. Para una dada tecnología la celda que permite un circuito integrado de menores dimensiones no necesariamente debe ser la que provee la menor área por bit de almacenamiento, pues puede darse el caso de que los circuitos asociados (refresco, selección) ocupen un área considerable.

Las nuevas celdas de un transistor han superado inconveniente previos, logrando desplazar en las memorias actuales a la celda de cuatro transistores.

4.3 La memoria dinámica Hyundai 514100A (Símil a la Motorola 54100A).

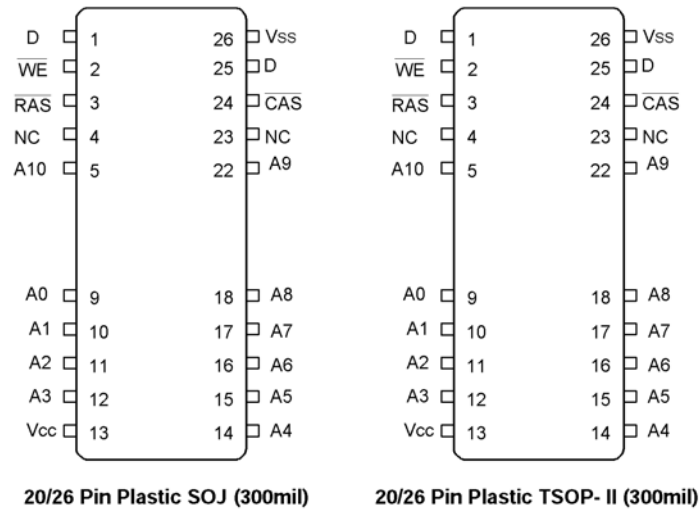


Fig. 31. Diagrama de las patas de conexión de una memoria dinámica.

En la figura anterior vemos la distribución de patas de una memoria dinámica comercial de 4 M x 1. Allí se observa que para minimizar la cantidad de conexiones, se decidió multiplexar las 22 líneas de direcciones en 11 pines denominados A₀ - A₁₁, agregando un par de señales /RAS y /CAS que deberán ser generadas por un circuito externo (controlador de memoria dinámica) juntamente con parte de la dirección que aparecerá sobre el bus multiplexado.

Las señales /RAS (Row Address Strobe) y /CAS (Column Address Strobe) validarán la parte baja y alta respectivamente de la dirección elegida, D será la entrada y salida de datos, /WE indicará operación de escritura o lectura, el resto de las conexiones será para alimentación.

4.4 Refresco de las memorias dinámicas.

Este tema será analizado basándonos en las memorias de 4 Mega bits, probablemente las más estandarizadas a la fecha (existen memorias de hasta 64 Mbits, pero cada fabricante tiene su propia versión de ella).

Los capacitores de cada bit de memoria son de 35 fF (femto faradio ó 10^{-12} F) y debía refrescarse cada 2 ms en las primeras memorias dinámicas. Con la evolución de la tecnología, dicho tiempo comenzó a ser cada vez mayor, como se observa en la siguiente tabla:

Tamaño de la Memoria	Tiempo máximo entre refrescos
16 Kbits	2 ms
256 Kbits	4 ms
1 Mbit	8 ms
4 Mbits	16 ms
16 Mbits	32 ms

Tabla 2. Relación entre el tamaño de la memoria y el tiempo máximo entre refrescos.

El diseño y la operación de la memoria permiten que sólo sea refrescada una fila por vez. El número de refrescos requerido por la memoria dependen de la configuración de la misma.

En el ejemplo que analizaremos en este punto, la memoria Motorola MCM54100A-70 de 4 Mbits de capacidad, esta configurada como 1024 filas de 4096 bits en cada fila¹⁹. (no es cuadrada y se adoptó este formato para refrescar mayor cantidad de bits por vez).

4.4.1 Tipos de refresco.

El refresco puede implementarse de dos maneras:

1. Refresco por ráfagas (burst refresh). En el mismo las 1024 filas de la memoria son refrescadas en forma consecutiva cada 16 ms. El refresco de toda la memoria llevará 133 μ s (130 ns por fila x 1024 filas)
2. Refresco distribuido. Se refrescará una fila cada 15,6 μ s (16 ms / 1024 filas). La duración total del refresco será también de 133 μ s.

Esta memoria de 4 Mbits puede ser refrescada de 3 maneras distintas:

4.4.2 Refresco sólo con RAS (RAS Only refresh - ROR).

Este método requiere que externamente, por medio de un contador, se provea el número de fila a refrescar. El controlador de memoria dinámica colocará el número de fila a refrescar y activará (bajo) la señal /RAS durante un tiempo t_{RAS} . Posteriormente se pasará /RAS al estado inactivo (alto) por un tiempo t_{RP} . /CAS se deberá mantener en inactivo (alto) durante todo el proceso.

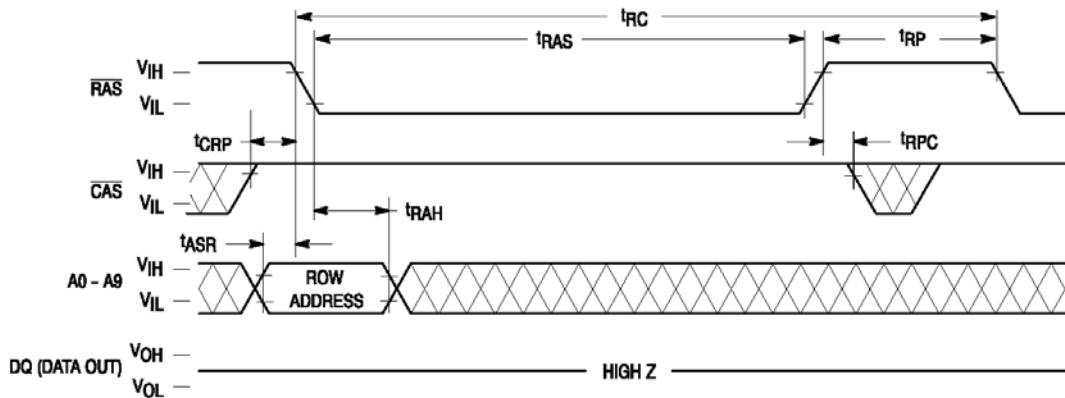


Fig. 32. Refresco sólo con /RAS

4.4.3 Refresco con /CAS antes de /RAS.

En éste método de refresco se empleará un contador interno a la memoria dinámica. Se pasará /CAS de alto a bajo, manteniendo /RAS alto. Transcurrido un tiempo t_{CSR} , se bajará /RAS y se mantendrá baja por un tiempo t_{RAS} .

¹⁹ Para el usuario esta configuración interna es totalmente transparente ya que verá a la memoria como de 4 Mega x 1.

A fin de evitar que la memoria entre en el modo de autotest, durante la transición de /RAS al estado activo bajo, se deberá mantener la señal de habilitación de escritura /WE inactiva.

/CAS deberá ser mantenida baja por un tiempo t_{CHR} después que /RAS pasó a su estado activo (bajo), luego, su esto no interesa.

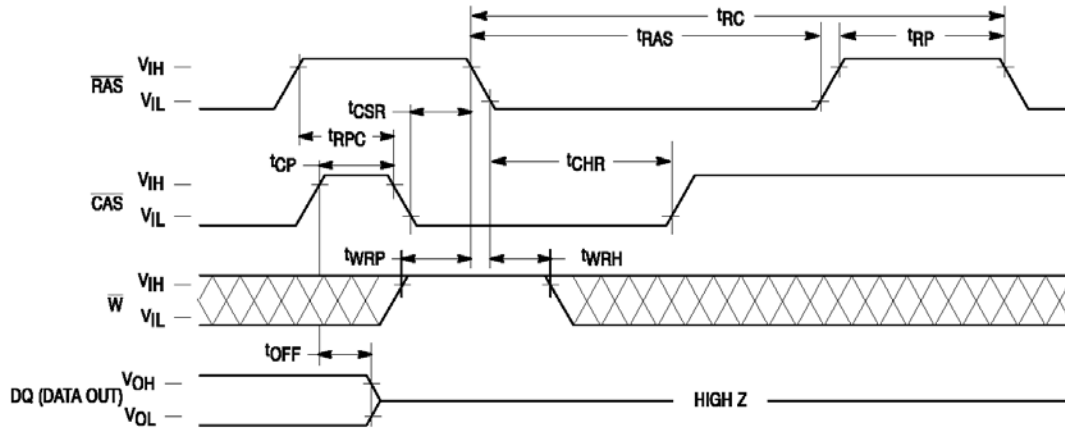


Fig. 33. Ciclo de refresco con /CAS antes que /RAS.

Esta forma de operación es inversa a la acción de lectura normal de una memoria, pues en ella /RAS actúa antes que /CAS. Esta inversión del orden es quien le indica a la memoria que se esta en el proceso de refresco.

4.4.4 Refresco escondido (hidden refresh).

El refresco escondido es un caso particular del refresco /CAS antes del /RAS que mantiene datos válidos en la salida si /CAS se mantiene activo después de transcurrido t_{CHR} , como se observa en la Fig.32.

Con /CAS mantenido bajo, /RAS se conmuta al estado alto luego de transcurrido t_{RAS} y se mantiene en dicho estado por un tiempo t_{RP} , conmutándola nuevamente ahora al estado bajo. Esto comienza otro ciclo de /RAS y se refrescará la siguiente fila cuya dirección es generada por el contador interno de refresco. Mientras que /CAS se mantenga bajo, el dato se mantiene en la salida, resultando en un ciclo de lectura largo.

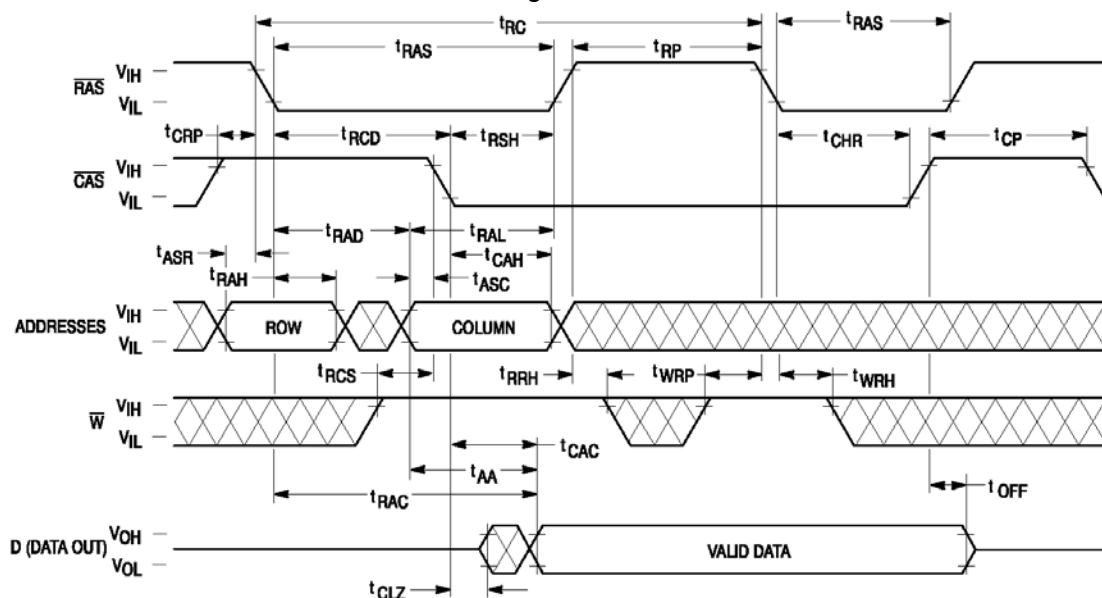


Fig. 34. Ciclo de refresco escondido.

Ya que los datos pueden ser leídos mientras la memoria se esta refrescando, la operación de refresco parece estar escondida por el ciclo de lectura.

El mismo refresco se puede obtener después de iniciar el ciclo de escritura, mientras la señal /WE se mantenga inactiva durante la transición activa de /RAS, a fin de prevenir la entrada en el modo autotest.

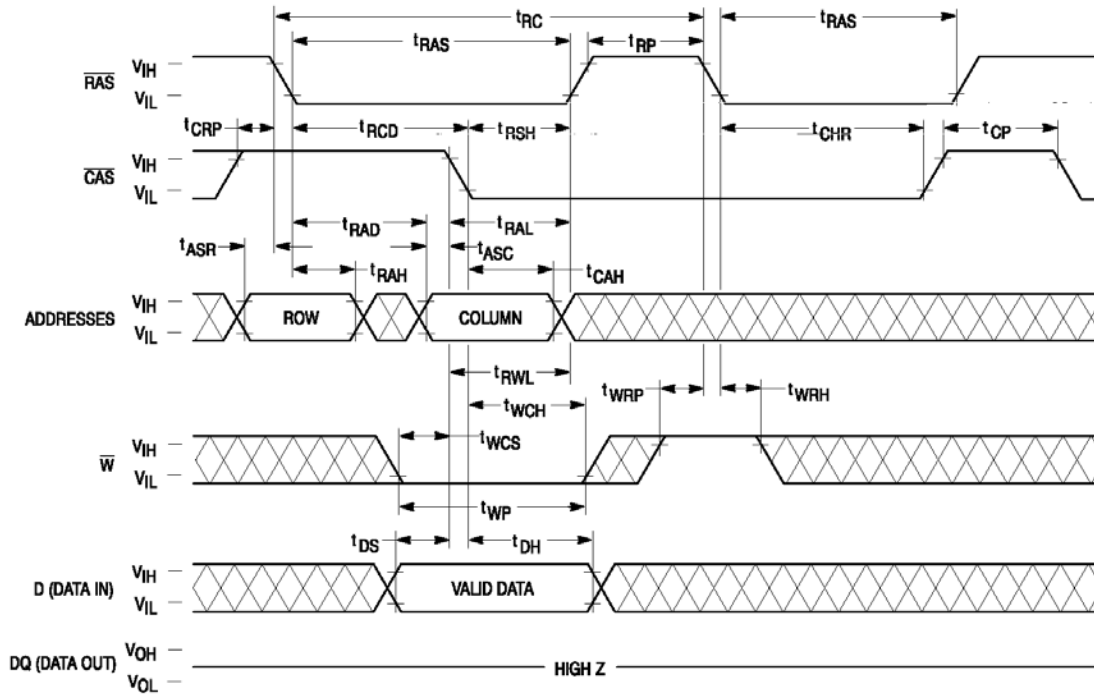
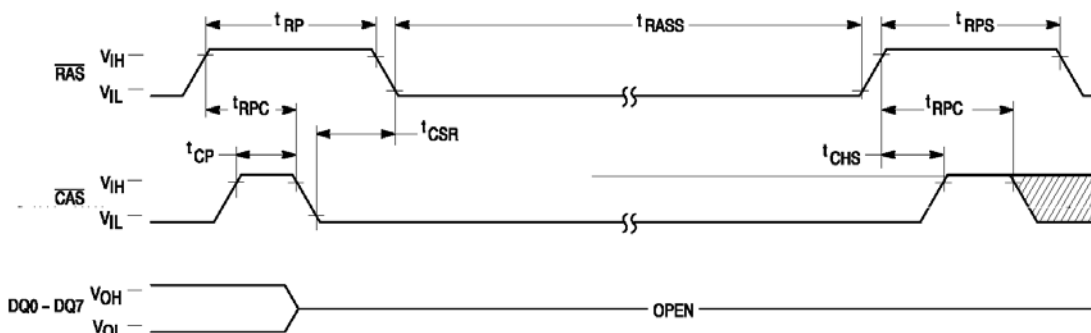


Fig. 35 Ciclo de refresco escondido con escritura temprana.

Como se puede observar en la figura anterior, no podrán realizarse operaciones adicionales de escritura cuando se inició el proceso de refresco escondido con la primera escritura.

4.4.5 Aurrerrefresco.

En las más nuevas memorias dinámicas, se provee una nueva metodología de autorrefresco sin necesidad de contadores y temporizadores externos con respaldo de batería. Ella consiste en una operación inicial similar al ciclo de refresco de /CAS antes de /RAS, pero simultáneamente /RAS y /CAS se mantienen activos (bajos) por un tiempo mayor que t_{RASS} (más de 100 μ s), como se observa para el ciclo de autorrefresco.

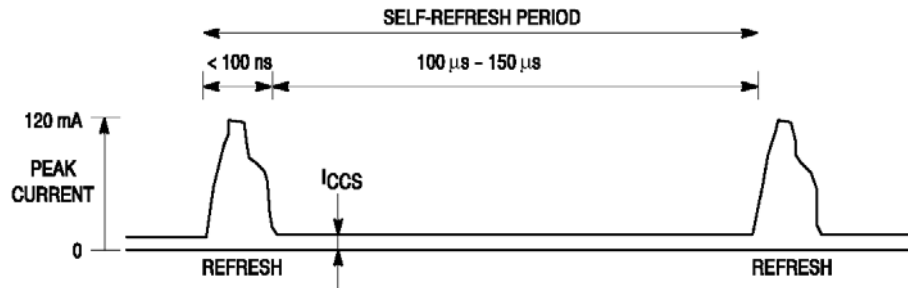


NOTE: Din, \bar{W} , \bar{C} , A0 - A8, A9R = "H" or "L"

"H" or "L"

Fig. 36. Ciclo de Autorrefresco

Al detectar esta situación la memoria dispara un temporizador interno que refresca una fila cada $130 \mu\text{s}$. Cuando se produce este refresco interno, la corriente de alimentación puede crecer hasta 120 mA durante menos de $100 \mu\text{s}$. A pesar de ello, se garantiza que el valor medio de la corriente debida al autorrefresco no podrá superar los $200 \mu\text{A}$.

**Fig. 37. Picos de corriente de autorrefresco.**

Estos picos de corriente deben ser suministrados por pequeños capacitores colocados en las cercanías de la memoria, pues de lo contrario, al circular por el impreso provocarían una fuertísima irradiación electromagnética debido a su enorme contenido en armónicas.

5. Tiempos de característicos de una memoria.

5.1 Tiempo de acceso.

En los diagramas temporales del 8088 se deben tener en cuenta dos parámetros fundamentales para verificar la compatibilidad temporal entre el mismo y las memorias y/o periféricos asociados, esto quiere decir que al esperar datos sobre el bus de datos, la memoria o el periférico lo hayan provisto o bien que al terminar un ciclo de escritura, al retirar el microprocesador los datos, la memoria o el periférico la hayan tomado.

El primero de dichos parámetros es el denominado **tiempo de acceso de direcciones** y que es el tiempo máximo que puede transcurrir desde que las direcciones A₀ – A₁₉ se estabilizan en el bus de direcciones hasta que los datos deben estar presentes en el bus de datos.

El segundo parámetro a tener en cuenta es el **tiempo de acceso de lectura**, que es tiempo máximo que puede transcurrir desde que baja la señal /RD hasta que los datos estén presentes y estables en el bus de datos.. Este parámetro se denomina TRLDV (tiempo desde que baja la señal RD hasta datos válidos) y es de 205 ns para un 8088 de 5 MHz de reloj en modo mínimo (existen algunas diferencias para la operación en modo máximo).

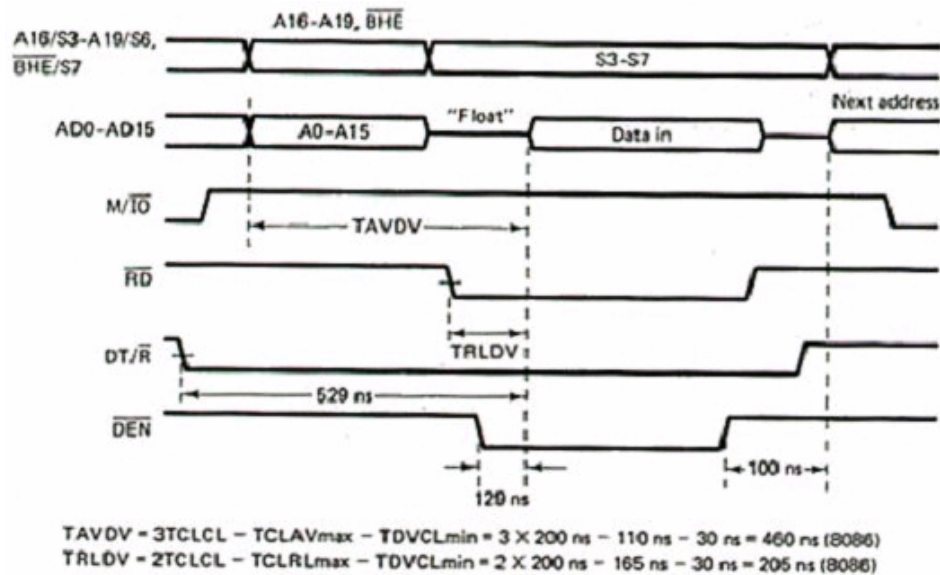


Fig. 38. Diagrama temporal del ciclo de lectura de una memoria.

El tiempo de acceso para las direcciones suele denominarse en forma comercial, simplemente como **Tiempo de acceso** y es un parámetro empleado por los fabricantes de memorias para clasificar la velocidad de sus productos.

Para el caso de que la memoria no sea lo rápido que necesita el microprocesador, ya que no puede aumentarse la velocidad de la misma, se deberá frenar al microprocesador.

Por tal motivo, el microprocesador verifica el estado de la línea de entrada READY en el segundo ciclo de reloj de cada ciclo de máquina. En caso de encontrarla en alto, no realiza operación alguna.

En caso de encontrarse en bajo (**por acción de algún circuito externo**), agrega ciclos de reloj de espera (wait states) hasta encontrar que la línea subió al estado alto²⁰.

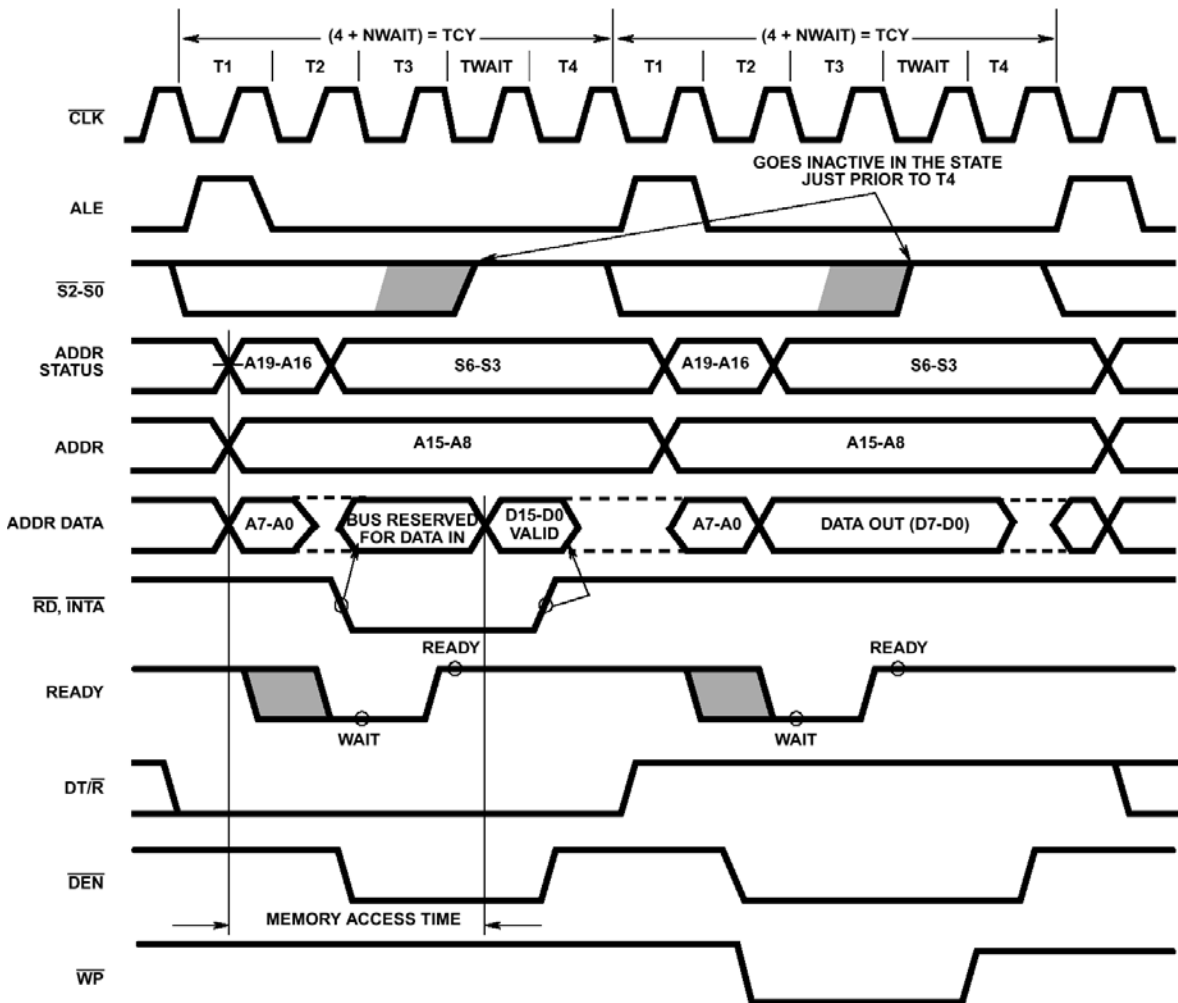


Fig. 39. Diagrama temporal con introducción de wait states.

El circuito de generación de wait states detectará que se ha seleccionado una zona de memoria lenta (no necesariamente toda la memoria tendrá igual tiempo de acceso) y activará un flip-flop que generará una señal que frenará al procesador. Por otro lado se dispondrá de un contador que permitirá reponer la señal de READY una cierta cantidad de ciclos de reloj después (en el circuito planteado, seleccionable con un contacto móvil).

²⁰ Para los microprocesadores que tienen registros dinámicos existirá un tiempo de espera máximo pues en caso contrario se perdería el estado de registros. Será necesario analizar cada caso en particular.

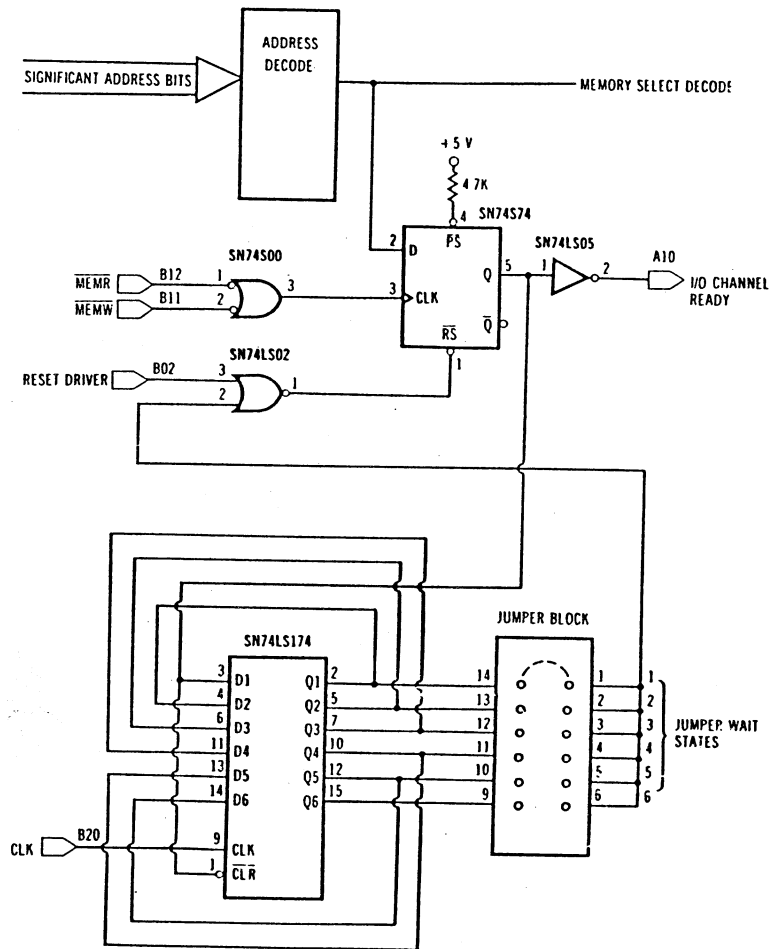


Fig. 40. Circuito de generación de wait states.

5.2 Tiempo de ciclo.

En las memorias dinámicas se suele emplear un parámetro característico adicional que es el tiempo de ciclo (*Cycle time*) y que consiste en el tiempo que debe transcurrir entre dos lecturas consecutivas.

6. La memoria dinámica en las PCs.

6.1 Introducción.

Habitualmente el dueño de una computadora personal le atribuye la velocidad (o lentitud) de la misma exclusivamente al microprocesador con que cuenta.

Queda oculto un tema primordial: Los programas se ejecutan en memoria.

Esto que parece una verdad de perogrullo, encierra una de las razones por las que un sistema puede operar a una velocidad inferior a la posible.

Cada programa o tarea que se abre o inicia ocupa memoria y puede hacer que se ocupe la totalidad de la disponible y obligue al sistema operativo a trasladar alguna zona de memoria al disco.

Al ser el disco un dispositivo electromecánico es mucho más lento que la memoria semiconductora y lentifica toda la actividad al trasladar datos de y hacia la memoria.

Aquí aparece una pregunta trascendente: ¿Debo cambiar el microprocesador o ampliar la memoria?. Cuando una PC dispone de escasa memoria (hoy, con los sistemas operativos y programas usuales, 64 MB) nada se mejorará agregando memoria.

Entonces, al decidir ampliar la memoria, ¿Cómo la debo pedir?.

Hay dos parámetros fundamentales para solicitar una ampliación de la memoria. Las dimensiones físicas de la plaqueta en la que se halla montada la memoria y la tecnología semiconductora de las mismas.

6.2 Dimensiones físicas (factor de forma).

En realidad por medio de una sigla se suelen especificar las dimensiones físicas y la cantidad de contactos de la plaqueta.

6.2.1 SIMMS

Esta sigla corresponde a Single In line Memory Modules. Es decir módulos que tiene conexión de una sola de sus caras de contactos (aunque tengan contactos en ambos lados de la placa).

Estos módulos vienen en dos formatos, de 30 y de 72 contactos.

Las primeras PCs tenían la memoria en la placa principal (recordemos que con el 8088 solo podíamos direccionar hasta 1 MB).

Con la aparición de la PC-AT con el 80286 y 16 MB de memoria posibles, se decidió montar las memorias sobre pequeñas placas de expansión. Fueron las SIMM de 30 contactos.

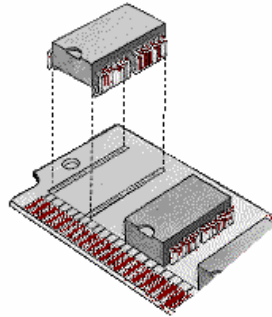


Fig. 41. Montaje de las memorias en módulos



Fig. 42. SIMM de 30 Contactos

El ancho del bus de estas memorias era de sólo 8 bits, por lo que para formar módulos de 16 bits para ese microprocesador se necesitaban dos ó múltiplos de dos módulos.

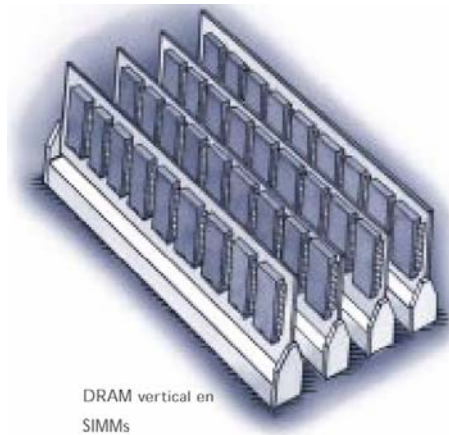


Fig. 43. Inserción de los módulos de memoria perpendiculares a la placa madre.

En los módulos de 30 contactos (como el de la Fig. 42 y Fig. 43), era habitual encontrar 9 circuitos integrados en cada módulo. Uno para cada bits del bus de datos y un noveno para almacenar la paridad. En la placa madre había un generador-verificador de paridad que al escribir cada dato en memoria generaba (por hardware) paridad que se almacenaba en ese noveno circuito integrado y que al leer esa memoria se verificaba. En caso de existir diferencias entre la paridad almacenada y la generada en cada lectura, se genera un mensaje de error catastrófico en la pantalla de la PC.

Con la aparición de los microprocesadores de 32 bits en el bus de datos, en lugar de comenzar a trabajar con 4 módulos, se decidió cambiar el formato y ampliar la cantidad de contactos a 72.

Las diferencias en apariencia entre ambos tipos de SIMMs son: a) Las SIMM de 72 contactos son aproximadamente 2 cm más largas que las de 30 y b) tienen una ranura de polarización entre contactos, como se puede ver en la figura.



Fig. 44. SIMM de 72 contactos con ranura de polarización en el centro

6.2.2 DIMMS

Estos módulos de memoria se asemejan en apariencia a las SIMMs pero tienen conexiones en ambas caras de contactos (Dual In line Memory Module).

Han sido desarrolladas para sistemas que manejen simultáneamente 64 bits de datos (aún cuando el procesador pueda ser de 32) y vienen en un formato de 168 contactos

Uno podría temer confundir una DIMM de 168 contactos con una SIMM de 72, pero no hay posibilidad para que (obviamente sin forzar los zócalos) se inserten erróneamente.



Fig. 45. Comparación entre una DIMM de 168 contactos y una SIMM de 72.

Como vemos en la Figura, las DIMMs son rectangulares, mientras que las SIMMs tienen una muesca lateral y además las DIMMs tienen dos ranuras de polarización (para impedir insertarlas invertidas) entre los contactos.

6.2.3 SO DIMMs

Son módulos DIMM de bajo perfil (Small Outline DIMM) pensados para los notebooks. Los SO DIMM de 72 contactos permiten trabajar sobre buses de 32 bits, mientras que existen módulos de 144 contactos para buses de 64 bits.



Fig. 46. SO DIMM de 72 y 144 contactos.

6.2.4 RIMMS y SO RIMMS.

RIMM es una sigla propietaria de la empresa RAMBUS. Se basa en un bus de 16 bits de alta velocidad (originariamente pensado para los video juegos Nintendo 64) y por ende alta disipación de energía. Por tal motivo, se caracterizan por tener un disipador incorporado al módulo de memoria.

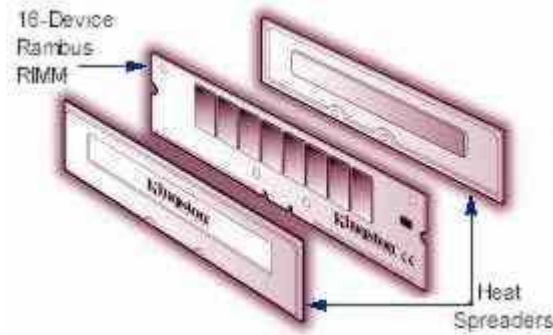


Fig. 47. Rambus con los disipadores separados.

Uno de los parámetros descriptivos del poderío de una PC moderna es la memoria RAM tanto en su tipo, su tiempo de acceso y su cantidad en Mbytes.



Fig. 48. Tres módulos actuales. SO DIMM de 72 contactos, SIMM de 72 contactos y DIMM de 168 contactos.

6.3 Tecnologías de los circuitos integrados.

La tendencia actual es emplear el formato DIMM de 168 contactos, pero ¿Cuál es la tecnología de los circuitos integrados de memoria dentro de dicho formato?.

6.3.1 Memorias FPM (Fast Page Mode)

Cada vez que se accede a una memoria dinámica, se puede aumentar la velocidad en el acceso a la posición siguiente si solamente se aumenta la dirección de la fila (manteniendo la página constante) o aumentando la dirección de la columna (manteniendo la fila constante).

Fast Page Mode:

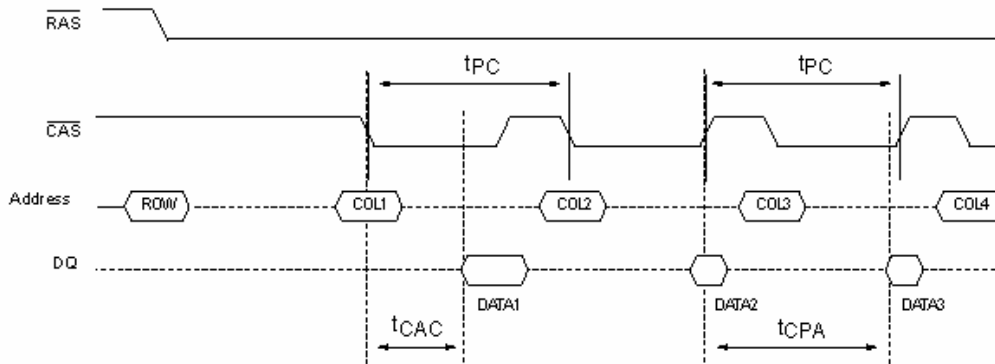


Fig. 49. Diagrama Temporal de una FPM

En la práctica se denomina página a toda la porción de memoria que tiene la misma dirección de /RAS, cambiando sólo el valor de la columna a ser accedida con distintos valores de dirección de /CAS.

En esta nueva generación de memorias, la dirección de /RAS se aumenta en uno automáticamente suponiendo que en el desarrollo lineal del programa se accederán a posiciones de memoria consecutivas.

Este método permite ahorrar 10 ns en un tiempo de acceso de 70 ns.

6.3.2 EDO Memory (Extended Data Output).

Aquí se retiene el dato leído en un registro de salida (de allí el nombre de Salida de Datos Extendida), para comenzar a leer la memoria siguiente, en forma similar a lo que sucedía en las FPM, pudiéndose ganar hasta un 10% de tiempo de acceso frente a aquellas.

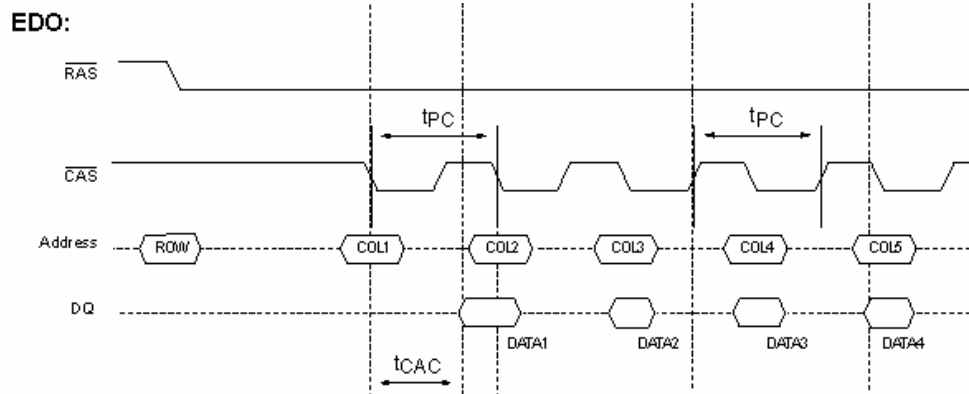


Fig. 50. Diagrama Temporal de una Memoria EDO

Una derivación de estas memorias son las denominadas **BEDO** o Burst EDO (EDO por ráfagas) y se emplean en los buses de microprocesadores de más de 66 MHz²¹. En ellas en lugar de ir leyendo la próxima celda se van preleyendo varias posiciones consecutivas y el microprocesador las leerá una tras otra en una ráfaga.

6.3.3 Memorias Dinámicas Sincrónicas (SDRAM).

²¹ A pesar que el microprocesador pueda trabajar a velocidades mayores (233, 300 ó 400 MHz), el bus interno de la placa base de la PC trabaja a una velocidad de 66 MHz.

La **DRAM sincrónica** es una nueva tecnología de DRAM que utiliza un reloj para sincronizar la entrada y la salida de señales en un chip de memoria. El reloj está coordinado con el reloj de la CPU, para que la temporización de los chips de la memoria y de la CPU estén sincronizados. La DRAM sincrónica ahorra tiempo al ejecutar los comandos y al transmitir los datos, aumentando de esta manera el rendimiento total del ordenador. La SDRAM permite que la CPU acceda a una velocidad un 25% superior a la de la memoria EDO.

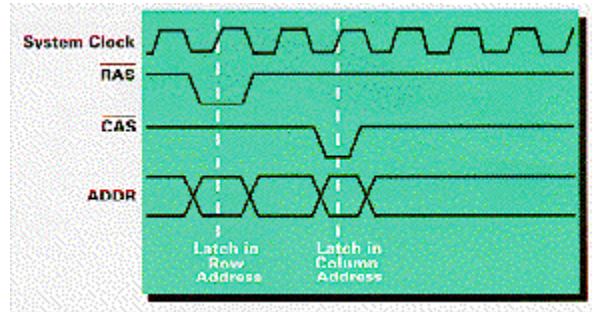


Fig. 51. Diagrama Temporal de una SRAM. Observar la sincronización de RAS y CAS con el reloj



Fig. 52. Presentación de una DIMM de 168 contactos en tecnología SRAM con bus de 100 MHz (PC100).

6.3.4 DDR ó SDRAM II

Double Data Rate SDRAM es la próxima generación de la SDRAM. La DDR se basa en el diseño de la SDRAM, con mejoras que suponen un aumento de la velocidad de transferencia. Como resultado de esta innovación, la DDR permite transacciones de datos en ambos flancos del reloj, con lo que se obtiene el doble de ancho de banda que con la SDRAM estándar. La DDR duplica la velocidad respecto a la tecnología SDRAM sin aumentar la frecuencia del reloj.

6.3.5 RAMBUS.

En la última década, la evolución de la velocidad (MHz) de los microprocesadores fue muy superior a la velocidad de las memorias. Por tal motivo se ha necesitado actualizar tanto la tecnología semiconductora como las técnicas de acceder a la información contenida por ella.

La vinculación entre el microprocesador y las memorias se realiza por medio de lo que se llama el front bus. Actualmente ese bus trabaja a 133 MHz. Actualmente los buses suelen ser de 64 bits, por lo que se leen o escriben hasta 8 bytes por ciclo de bus, es decir:

$$8 \text{ bytes} \times 133 \text{ MHz} = 1064 \text{ Mbytes/s}$$

Este es el límite teórico y que ya está alcanzado por los procesadores de 1 GHz. Por tal motivo RAMBUS e INTEL han desarrollado una interfaz denominada RAMBUS de alta eficiencia y simplificada de 16 bits pero muy alta velocidad (y disipación) con una tecnología exclusiva denominada RSL (Rambus Signaling Logic).

6.3.6 SDRAM

La **SDRAM** es una DRAM fruto de un desarrollo conjunto y, en cuanto a la velocidad, puede representar la competencia más cercana de Rambus. Su desarrollo se coordina a través de un consorcio de 12 compañías desarrolladoras de sistemas y fabricantes de DRAM. La SDRAM es

una extensión más rápida y mejorada de la arquitectura SDRAM que amplía el actual diseño de 4 bancos a 16 bancos. La SDRAM se encuentra actualmente en fase de producción desde 1999.

6.4 Integridad de los datos en la memoria.

Uno de los aspectos importantes que involucra el diseño de un sistema de memoria es el concepto de integridad de la información. Ello se logra por medio de dos técnicas ya conocidas por el alumno de Técnicas Digitales I. Paridad y ECC (Error correction Code o Códigos de Corrección de Datos).

Paridad es el método más común y más sencillo de implementar. Este proceso implica agregar un bit cada 8 bits de datos y que sólo permite **detectar** errores, mientras que ECC es un método más sofisticado que no sólo permite la detección sino también la corrección de errores de hasta 1 bit en cada palabra.

Por una razón de precios y escudándose tras una alta confiabilidad de las memorias, muchos fabricantes han dejado de incorporar a sus módulos algún método de verificación de la integridad de los datos en sus dispositivos.

6.4.1 Paridad

Cuando se emplea paridad en una computadora, se agrega un bit de paridad cada byte de datos. Los dos métodos de paridad (par e impar) operan en forma similar y generarán algún error (una interrupción no enmascarable) que detendrá la operación de la computadora y presentará un mensaje de error.

El método de paridad presenta algunas limitaciones. Como ya se sabe de Técnicas Digitales I, el uso de paridad puede detectar errores pero no corregirlos. Adicionalmente, la existencia de más de un bit de error puede llegar a enmascarar la existencia de los mismos.

6.4.2 Paridad ficticia.

En la operación de los sistemas que empleen paridad normal, cuando se escriben 8 bits sobre la DRAM, se escribe simultáneamente el bit de paridad. Algunos fabricantes, con el fin de abaratar costos, evitan incorporar la memoria del noveno bit y simulan dicho bit, con un circuito sencillo que provea ese bit de paridad en el momento de la lectura.

El problema radica en que siempre se proveerá el bit de paridad adecuado, aún cuando exista un error en la lectura.

	PARIDAD IMPAR	PARIDAD PAR
paso 1	<p>El bit de paridad se fija en 1 (o sea que se "activa") si el byte de datos correspondiente contiene un número par de unos.</p> <p>Si el byte contiene un número impar de unos, el bit de paridad se fija en cero (o sea que se "desactiva")</p>	<p>El bit de paridad se fija en uno si el byte de datos correspondiente contiene un número impar de unos.</p> <p>El bit de paridad se fija en cero si el byte contiene un número par de unos.</p>
paso 2	<p>El bit de paridad y los 8 bits de datos correspondientes se registran en DRAM.</p>	<p>(Igual que con la paridad impar)</p>
paso 3	<p>Antes de que se envíen los datos a la CPU, éstos son interceptados por el circuito de paridad.</p> <p>Si el circuito de paridad identifica un número impar de unos, los datos se consideran válidos. Se elimina el bit de paridad de los datos y los 8 bits de datos se transmiten a la CPU.</p> <p>Si el circuito de paridad detecta un número par de unos, los datos se</p>	<p>(Igual que con la paridad impar)</p> <p>Los datos se consideran válidos si el circuito de paridad detecta un número par de unos.</p> <p>Los datos se consideran inválidos si el circuito de paridad detecta un número impar de unos.</p>

6.4.3 ECC

ECC (Error Correction Code o Códigos que Corrigen Errores) son empleados en sistemas de alto rendimiento y confiabilidad como File Servers de redes o estaciones de trabajo (workstations).

La diferencia esencial entre ECC y paridad, radica en que los primeros pueden detectar y corregir un bit de error en una palabra. Normalmente al producirse un error de un bit en estas computadoras de alta performance, dicho bit es automáticamente corregido por hardware y pasa desapercibido para el sistema operativo y por ende para el usuario.

Según el tipo de controlador que se emplee, se podrán detectar desde 2 hasta 4 bits de error en cada palabra. Tal situación sólo producirá un reporte de error, pero no será factible hacer ninguna corrección.

Existen SIMMs con un ancho de 39 ó 40 bits. Esas memorias han sido diseñadas para operar con ECC. Algunos servidores de alta confiabilidad emplean dos memorias de 36 bits de ancho. 64 de esos bits se emplearán para información y los 8 restantes para ECC.

Estos SIMMs de 36 bits fueron originariamente pensados para operar sobre buses de 32 bits con 4 bits de paridad (2 módulos de 36 bits dan 72 bits → 64 de datos + 8 de paridad).

El empleo de ECC no es una opción que pueda realizar el usuario en la configuración de su computadora. Este sistema sólo estará disponible en aquellos sistemas que **fuero diseñados para ello**. Ello sucede habitualmente con sistemas de alto precio y equipado con procesadores tipo Pentium II.

6.5 Refresco

La velocidad de refresco dependerá del tipo de memorias empleados y variará de 2 ms a 32 ms, según el caso..

La tendencia actual es hacia el autorrefresco, siendo mayoritariamente empleado (por ahora) en notebooks, ya que simplifica el circuito y reduce el consumo de corriente en forma considerable.

6.6 1,8 Volt versus 3.3-volt versus 5-volt

Históricamente las computadoras se alimentaron con 5 V. Con la reducción del tamaño de los componentes, las geometrías actuales no permiten las exigencias de la tensión de alimentación de 5 V. Por todo ello es que la tendencia actual lleva a que los microprocesadores y las memorias operen con una tensión de alimentación de 1,8 ó 3,3 V, con lo que se reduce la potencia disipada y ello permite disminuir aún más el tamaño de los componentes.

6.7 Memoria Caché

Estas memorias serán analizadas en detalle en Técnicas Digitales III. Podemos anticipar que se tratan de memorias estáticas de alta velocidad diseñadas para acelerar la ejecución de las instrucciones por parte de la CPU, en una gran relación a si se empleara la memoria principal de la computadora.

Estas memorias se hallan entre el microprocesador y la memoria principal (la estándar) y por medio de un circuito de alta tecnología que cumple la función de controlar la memoria caché, se copiará una parte del programa en ejecución a la caché.

Como ejemplo, tomemos una placa madre de 100 MHz. A la CPU le tomaría 180 ns para leer información de la memoria principal, mientras que sólo 45 ns cuando la obtiene de la memoria caché.

Las memorias caché pueden ser de dos tipos, la primaria (o de nivel 1), generalmente interna al microprocesador y la secundaria (o de nivel 2) y siempre externa al microprocesador. En caso de no existir la memoria caché interna, la externa cumplirá las funciones de memoria primaria.

La eficiencia a la memoria caché se la da el anteriormente mencionado controlador de memoria caché. Este controlador es el encargado de llenar la caché copiando tramos de la memoria principal y de actualizar a ambas, copiando tramos requeridos a la cache y retirando aquellos tramos de memoria que ya no son útiles o llevan demasiado tiempo sin ser empleados.

La eficiencia de la memoria cache es conocida como hit rate y calcula la probabilidad en que un dato requerido por el microprocesador se halle en la cache y no se necesite traerlo de la memoria principal. Estadísticamente se establece que el hit rate de un sistema de buena performance se halla alrededor del 99%.

7. Bibliografía.

- a) The 8086/8088 family. Design, programming and Interfacing. Uffenbeck. Prentice Hall 1987.
- b) La PC por dentro - 2ª Edición. M.C. Ginzburg.
- c) Nota de aplicación Motorola AN987 - DRAM Refresh Modes (www.design-net.com/cgi-bin/get?/books/apnotes/pdf/an987*.pdf).
- d) www.kingston.com.
- e) www.heacom/hea2/flash/main.htm

8. Cuestionario

1. ¿Qué representan las filas de una memoria? y las columnas? Las columnas son líneas de entrada o de salida de la memoria?
2. ¿Qué ancho de palabra tienen normalmente las memorias que usamos en este curso?
3. ¿Cuándo se producen lecturas o uniones fantasmas? Ejemplifique mediante un dibujo.
4. Indique cual es la primera solución para evitar las lecturas fantasmas. ¿Qué inconvenientes presenta esta primera solución?
5. Explique como se evitan ambos inconvenientes.
6. Dibuje una memoria elemental con todas las líneas de control que usualmente tienen, explicando la función de cada una de ellas y a que líneas del microprocesador van conectadas.
7. ¿Por qué motivo se trata de obtener una estructura cuadrada de circuito integrado?
8. En las memorias dinámicas, ¿También se respeta esa aseveración?. ¿Por qué?
9. ¿Cómo se clasifican las memorias por su funcionamiento? Explique sintéticamente cada tipo.
10. ¿Qué entiende por refrescar las memorias? ¿Qué tipo de memorias tiene que ser refrescadas y por qué? ¿Qué procesos conoce para refrescar las memorias?
11. ¿Cuál es el tiempo de refresco máximo especificado por los fabricantes de memorias dinámicas?. ¿De qué depende?. ¿Qué sucede con la información si no se respeta ese tiempo?
12. ¿Cuáles son las ventajas y las desventajas de las memorias dinámicas?
13. ¿Qué significa EDO, FPM RAM, SRAM, RAMBUS y cache?.
14. Busque en Internet o en manuales hojas de datos de cada una de ellas. Detalle la procedencia y analice las especificaciones de tiempos de cada una de ellas.
15. ¿Cómo se clasifican las memorias de acuerdo con la persistencia de la información? Explique sintéticamente cada tipo. Ejemplifique.
16. ¿Cómo se clasifican las memorias de acuerdo con la tecnología? ¿Cuál de ellas usan la mayoría de los microprocesadores? ¿Por que?
17. Explique por que las memorias comerciales son NMOS y no PMOS comparando las movilidades relativas de los electrones y de las lagunas.

18. Explique por que se coloca un capacitor, normalmente de 0.1 microfaradio, por cada circuito integrado digital (CMOS y TTL) que hay en un circuito impreso. ¿Porqué debe ir lo más cercano posible al circuito integrado?
19. ¿En que momento consumen las compuertas de la familia lógica MOS?
20. Ordene en orden creciente de consumo: CMOS, PMOS y NMOS.
21. ¿Qué ventajas y desventajas tienen los integrados CMOS?
22. ¿Cómo es la densidad de integración de los circuitos CMOS, baja, media o alta?
23. ¿Qué conviene más hacer: un solo integrado por oblea de silicio o hacer 1.000.000 por oblea?. ¿Por que?
24. ¿Cómo se clasifican las memorias de acuerdo con su uso? Explique sintéticamente cada una de ellas.
25. Explique sucintamente el fenómeno de borrado de la información en las memorias EPROMs. ¿Cómo se las programa?
26. ¿Qué características tiene una memoria EEPROM? ¿En que efecto se basa para guardar y para retirar cargas?
27. ¿Cómo clasificaría una memoria de lectura y escritura CMOS con respaldo (backup) de batería?. ¿Cómo se implementa un circuito con estas últimas memorias? ¿Cómo funciona?
28. ¿Cómo se clasifican las memorias de acuerdo con la forma de acceder a la información? Explique sintéticamente cada una de ellas. Analice el tiempo que se tarda en acceder a la información en uno y otro caso. Ejemplifique.
29. ¿Puede una memoria ROM ser RAM? Justifique su respuesta.
30. ¿Qué información le proporcionan los nombres de las siguientes memorias:
31. Describa los nombres de las patas de las memorias:
32. ¿Cuántas líneas de direcciones tienen?
33. ¿En que consiste básicamente un UPS ("Uninterruptible Power System")? ¿Para qué se lo usa?
34. ¿Qué son las memorias NVRAM? ¿Para qué se las utiliza?
35. ¿Cómo puede lograr el mismo efecto que con las memorias NVRAM pero en forma más simple y barata? Dibuje el circuito práctico correspondiente.
36. Explique ¿Cómo puede usar el circuito integrado MAX 691, en lugar del circuito práctico anterior?. ¿Qué ventajas tiene este integrado?
37. Obtenga la hoja de datos del MAX 691 y analice su funcionamiento como mecanismo de seguridad en el caso en que un programa se pierda.
38. ¿Qué señal del microprocesador se usa para frenarlo cuando el mismo se usa junto con un banco de memorias lentas?
39. Defina tiempo de acceso de una memoria. ¿Dónde encuentra este parámetro?
40. A partir de que momento computa el tiempo de acceso de una memoria (tiempo de "Chip Select", tiempo de direcciones válidas, tiempo de "Output Enable", etc.).

41. ¿En qué ciclo de reloj el microprocesador va a leer un dato? ¿Qué sucede con la señal de READ? ¿En qué ciclo de reloj termina un ciclo de lectura de memoria? ¿Cuándo el 8088 lee la información:
 - cuando baja la señal de RD/
 - durante el tiempo en que se encuentra baja RD/
 - o cuando sube RD/?
42. ¿Cómo procede para saber si el tiempo de acceso a una memoria es compatible con el circuito donde se la va a usar? Ejemplifique.
43. Si una memoria determinada no es compatible con un circuito, ¿qué soluciones existen?. ¿Cuál es la más conveniente?
44. ¿Qué tiempo de acceso caracterizaba a las memorias dinámicas viejas? ¿Y a las usadas actualmente?
45. ¿Qué función tiene la línea /OE ("Output Enable") de las memorias? ¿A que línea del microprocesador se la conecta? ¿Cuándo se la habilita?
46. ¿Qué hay que garantizar para que el microprocesador lea correctamente los datos provistos por la memoria?
47. ¿Qué sucede si no se cumple la condición anterior?
48. Si se necesita una memoria con tiempo de acceso de 250 ns, ¿qué memoria elegiría para un diseño si puede optar entre una de tiempo de acceso de 100 ns u otra de 200 ns? ¿Por qué?
49. ¿En que ciclo de reloj sube la señal ALE ("Address Latch Enable")? ¿Y en cual baja la señal RD/ ("Read") cuando se desea leer una posición de memoria?
50. Explique que entiende por CONTENCIÓN DEL BUS ("Bus Contention"). ¿Cuándo se puede producir en una memoria? ¿Cuál es el resultado?
51. ¿Cómo impide que se produzca contención del bus cuando se escribe o lee una memoria? Explique.
52. ¿Qué relación guarda la frecuencia de reloj del 8088 con respecto a la del cristal que usa el 8284? ¿Cuál es la frecuencia máxima que usa el 8088?
53. Defina tiempo de clock: t_{CLCL}
54. ¿Qué valor tiene para la máxima frecuencia del 8088?.
55. ¿Cuál es el ciclo de actividad del clock del 8088?.
56. ¿Qué entiende por t_{CLAV} ?.
57. ¿Qué demoras se producen en el latch 74HC373? ¿Con que criterio calcula el tiempo de propagación del latch teniendo en cuenta las distintas demoras?
58. Cuales son los tres tiempos de propagación que caracterizan al decodificador 74LS138?
59. Explique la ventaja que presenta el decodificador 74HC138 cuando esta habilitado permanentemente.
60. ¿Qué entiende por t_{OE} (tiempo de "Output Enable") de una memoria?
61. Dibuje el circuito practico que permite generar un estado de espera ("Wait State").
62. ¿Cuáles son las diferencia entre una SIMM FPM y una DIMM EDO?.
63. ¿Qué entiende por RAMBUS, video RAM y dual port RAM?.