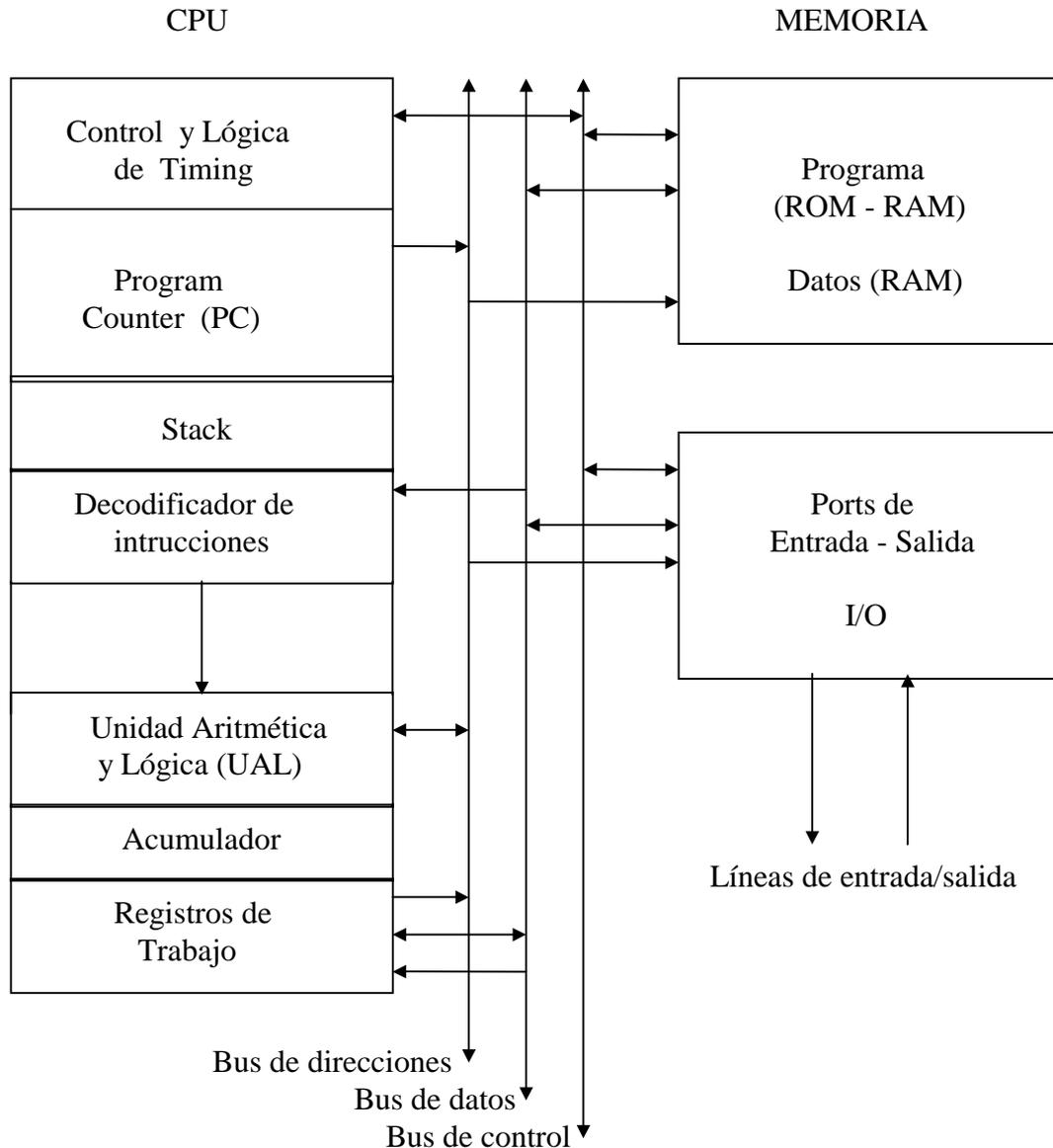


ARQUITECTURA

La organización general o arquitectura de una computadora digital, en un mainframe, una minicomputadora o un microcomputador, responde al siguiente esquema.



La funcionalidad de la C.P.U. se puede dividir en dos subsecciones, una de las cuales está orientada al control de timing y la secuencia de operación del sistema, mientras que la otra ejecuta las funciones aritméticas y lógicas y administra los datos a ser procesados. Un sistema de memoria se conecta a la C.P.U. y es usado para almacenar la lista de instrucciones a ser ejecutadas, conocida como programa, y los datos a ser procesados. La comunicación con el exterior es administrada por dos ports: el de salida OUTPUT y el de entrada INPUT que permiten la transferencia de datos de y hacia los dispositivos externos; tales como keyboards, monitor e impresora. Todos los componentes se comunican entre sí por medio de un sistema de líneas, conocidas como BUS.

SISTEMA DE BUS

Los datos se comunican entre las diversas unidades sobre un conjunto de cables en paralelo conocidos como BUSES. Existen tres tipos de buses: el de datos, de direcciones y el de las señales control.

El bus de dato es del tipo bidireccional y es controlado por la C.P.U. Una línea de lectura/escritura que forma parte del bus de control determina la dirección del flujo de datos a lo largo del bus de dato. Esta línea es generada por la CPU, si está en el modo escritura WRITE permite que los datos salgan de la CPU a otro lugar del sistema; y si el modo es lectura READ la operación es a la inversa. Un sólo dispositivo por vez puede enviar señales por el bus mientras que muchos pueden recibirlas en un instante determinado.

El bus de direcciones se usa para brindar una señal de dirección a la memoria, para seleccionar una ubicación en particular, dentro de ésta. Normalmente el bus de direcciones es una salida desde la CPU hacia la memoria.

El bus de control provee de señales de control hacia y desde la CPU, para gobernar el timing y controlar la transferencia de datos de los otros buses. Entre éstas señales, pueden existir algunas que detienen el funcionamiento de la CPU o desconectan a ésta del sistema de buses. Los buses de direcciones y de datos son normalmente de tres estados. Aparte de los estados normales de 0 o 1 de la CPU y la memoria pueden ser puesto en un estado de alta impedancia que efectivamente desconecta a éstos, de los buses. Esta facilidad puede ser usada cuando se desea tener otros dispositivos conectados al mismo sistema de buses. Esto se usa en sistemas multiprocesadores donde el bus se comparte entre dos ó más CPUs, donde sólo una de éstas puede acceder al bus por vez.

CPU control

Aparte del sistema de timing y lógica de control, la CPU presenta un registro interno llamado program counter, un registro de direcciones y decodificador de instrucciones, un registro de stack o stack pointer y algunas interrupciones lógicas.

Las instrucciones que constituyen el programa a ejecutarse se almacenan en memoria y son normalmente llamadas en una determinada secuencia para ser ejecutadas. La función del program counter es dar la dirección de memoria de la siguiente instrucción a ejecutarse. Al comienzo de un programa, el program counter se cargará con la dirección de la primera instrucción del programa. Cuando el sistema comienza a correr esta dirección se transfiere al bus de direcciones y el código de la instrucción se leerá desde la memoria. Una parte de la instrucción se denomina código de operación u "opcode" y determina el tipo de operación a realizarse. Las operaciones típicas son ADD, SUBTRACT o AND; cada una de las cuales tiene diferentes patrones de opcode. El dato leído de opcode pasa al decodificador de instrucciones, el cual setea el estado interno de la CPU para poder realizar la operación deseada. En éste momento, el contenido del program counter se actualiza para dar la siguiente dirección del próximo opcode. Algunas operaciones requieren datos extras, tanto como, el código de operación; y para éstos casos necesitará 1 ó 2 palabras de datos que se encuentran a continuación del opcode en memoria.

Estos datos son grabados cuando se ejecutan las instrucciones, y en éste caso el contador de instrucciones se incrementa en 2 ó 3 para dar la dirección correcta de la próxima instrucción "opcode". Esta actualización dependerá del decodificador de instrucciones según la instrucción detectada.

La ejecución típica de instrucción está formada por 2 ciclos. -El primero toma el código de memoria => "instruction fetch cycle", y si la instrucción requiere datos extras => existe un ciclo extra para leerlos desde memoria. En algunos casos éstos datos son

direcciones de memoria (donde realmente existe el dato), entonces éstos son transferidos al address bus durante la ejecución de la instrucción.

El timing depende del tipo de CPU que esté usando, dicha velocidad viene dada por las características estructurales (físicas). El clock de timing se genera por medio de un oscilador integrado que sólo requiere de un cristal de cuarzo. Además, éste es necesario para el control de timing de memoria y los canales de entrada - salida.

CPU Ejecución

Dentro de la CPU existe una unidad aritmética y lógica ALU que es un arreglo lógico para obtener una función aritmética o lógica, y asociado con ésta se encuentran los acumuladores. En éstos registros es donde se realizan las operaciones. El uso de los registros internos de la CPU puede aumentar la performance del programa; lo que evita es el acceso a memoria cuando la instrucción se ejecuta.

Algunas CPUs usan registros de memoria como registros generales, permitiendo hacer operaciones directamente en memoria. Existe también registros índices o punteros que se entienden directamente con el banco de memoria, permitiendo el manejo de direcciones en lugar de datos con el beneficio de poder manipular tablas directamente.

Funciones típicas => de la ALU y acumuladores: ADD, SUBTRACT, AND, OR, EXCLUSIVE OR, Complemento. También existe la posibilidad de shift a derecha y a izquierda de datos, rotar a L y R de bits. Otra operación es la de incrementar registros. En algunos procesadores sofisticados también se implementan operaciones de multiplicar y dividir => operaciones lentas si se las realiza en forma de programa interno. Cuando está implementada en forma de hardware extra dentro del chip permite tener operaciones más rápidas de ejecutar.

Una cualidad importante de la CPU es la capacidad de testeo de los resultados ante una operación => zero, negativo, carry y overflow. Carry = existe un acarreo fuera del acumulador. Overflow = el resultado es un número más grande que lo que puede soportar el acumulador. Como resultado del test la CPU puede hacer que realice un Skip, Branch o un Jump en lugar de ejecutar la próxima instrucción.

La operación Skip, hace que la CPU salte hacia la próxima instrucción de la secuencia y la ejecute. En una operación de Branch o de Jump, un nuevo valor se coloca en el program counter y la ejecución se transfiere a otro lugar en el programa. En un Jump la nueva dirección se coloca a continuación del Jump como un dato en memoria. En un Branch el dato a continuación de la instrucción representa un dato relativo que se suma al contenido del program counter para dar la nueva dirección de la próxima instrucción. Tanto en Branch como en Jump, se pueden agregar condiciones lógicas de salto. Cuando el Branch es incondicional el salto se efectúa siempre.

Subrutinas y Stacks

Frecuentemente dentro de un programa existen conjuntos de instrucciones que se repiten. Una solución es reescribirla cada vez que se necesiten, o declararlas como subrutinas. En una subrutina el bloque es tomado como un pequeño programa individual y es almacenado en un lugar de memoria una sola vez. Cuando se las usa se accede a través de un Call. Esto hace que el contenido del program counter sea almacenado temporalmente en otro lugar y en éste se coloca la dirección de comienzo de la subrutina. Al final de la subrutina se coloca un Return que reestablece el contenido antiguo del program counter de manera tal, que se realiza la ejecución de la instrucción que existe luego del Call.

Los procesadores simples tienen un registro para guardar éste program counter, mientras que otros tienen un conjunto de registros llamado "Stack". En tales sistemas el

último registro que se almacena es el primero en ser leído. En la mayoría de los sistemas, el stack es implementado como un lugar en la memoria principal. Para administrarlo se usa un registro interno de la CPU conocido como stack pointer que apunta siempre al tope de ésta pila. En cada llamada, en una subrutina el contenido del program counter se escribe al tope del stack en memoria y el stack pointer es actualizada para apuntar a la siguiente posición libre dentro del stack. Cuando recibe un Return desde una subrutina, el stack pointer se actualiza, pero en forma decreciente automáticamente. Procesadores de tipo generales permiten el uso de llamadas del tipo PUSH y PULL (o POP) para el manejo de datos dentro del stack. Algunos de los procesadores más adelantados tienen más de un stack pointer. Uno dedicado para almacenar la dirección de la subrutina, llamado "system" o "supervisor stack", mientras que el otro que puede ser usado para almacenar datos se conoce como "user stack".

Cuando el stack es implementado en memoria las subrutinas se manejan en forma anidadas. El tamaño del stack determina cual es el nivel de anidamiento. Muchas veces el "return" de una subrutina puede provenir de un lugar cualquiera dependiendo de una condición => no necesariamente está al final.

A veces puede desear transferir datos del programa principal a la subrutina; y puede ser hecho por medio del "pushing" de datos sobre el stack antes de llamar a la subrutina. Una vez dentro de la subrutina, el stack pointer puede manejarse para acceder a éstos datos. Hay que tener cuidado de reestablecer el program counter al valor correcto antes del return, para poder ejecutar la próxima instrucción que se halla luego del Call.

Interrupciones

Cuando se efectúa comunicación entre el procesador y el exterior, puede suceder que el dispositivo externo no se encuentre listo para recibir datos, o que si éstos provienen desde el exterior el procesador se encuentre realizando otra tarea y no pueda recibirlos. Estos tipos de situaciones, podrían ser solucionadas colocando al procesador en un loop de testeo donde se queda esperando una señal de "listo" desde el dispositivo externo. Esto es ineficiente dado que el procesador se encuentra en estado de espera, y la técnica usada en éstos casos es la de usar un sistema de interrupciones. En un sistema de interrupciones, el dispositivo externo entrega una señal en la entrada de interrupción de la CPU. Cuando una interrupción tiene lugar la CPU termina el trabajo que está realizando y realiza un salto a la rutina de servicio de interrupciones, que se entenderán con el dispositivo externo. Cuando se recibe una interrupción el contenido de estado de la CPU es almacenado en el stack. Una vez que se le da servicio a la interrupción, el contenido de los registros de la CPU deben ser reestablecidos desde el stack. Las interrupciones pueden ser anidadas como las subrutinas.

Otro tipo de interrupción, es usada en las tareas de debug de programas, pero también puede ser usada para casos en que se encuentren fallas dentro del programa.

Los procesadores de uso general usan un sistema de vector de interrupciones; donde cada tipo de interrupción contiene la dirección de salto al comienzo de la rutina de administración de dicha interrupción. En algunos sistemas de interrupciones vectorizadas, el tipo de interrupción de que se trata puede ser suministrado vía el bus de datos.

El encendido de la máquina y la secuencia de inicialización es una forma especial de interrupción, normalmente en éstos caso el program counter se carga desde la parte alta de memoria (donde se ubica la BIOS), y direcciona hacia la rutina de comienzo.

Memoria

La memoria de un microcomputador se usa para almacenar tanto programas, como datos. Normalmente los programas se construirán en alguna memoria de sólo lectura (ROM: Read Only Memory) para un sistema dedicado; mientras que los datos se almacenarán en una memoria de acceso directo (RAM: Random Access Memory); y algunos datos como constantes pueden ser almacenadas en la ROM junto con los programas. En computadores de propósitos generales tales como un sistema de desarrollo o una computadora personal (PC: Personal Computer), el programa del usuario es frecuentemente almacenado en RAM, en las máquinas el Sistema Operativo y los programas asociados se almacenaban en ROM. En la actualidad los sistemas presentan sólo los programas de inicialización y varias rutinas útiles almacenadas en ROM para formar lo que generalmente se llama programa MONITOR.

En los sistemas de computación, los datos de gran volumen son almacenados en memorias magnéticas de discos rígidos o cinta (el tipo de grabación en éste tipo de unidades es similar al de audio en una cinta magnética), para una cantidad pequeña de datos se utilizan las RAM y para datos de mediano tamaño se usan las ROM. En la actualidad se usan para manejar volúmenes de datos del orden de GigaBytes (1 GigaByte = 1000 MegaByte; 1 MegaByte = 1000K Byte; 1KByte = 1000Byte) discos compactos del tipo de audio (CDROM: Compact Disk Read Only Memory), de menor tamaño, unos 120mm de diámetro. En éstos sistemas la velocidad de recuperación de datos es superior a cualquier sistema de discos, dado que utiliza un sistema óptico de espejos y un lector láser de arseniuro de galio. También existen discos de éste tipo de tecnología que permiten ser grabados pero que presentan una menor capacidad de almacenamiento, unos 500MegaBytes.

Cuando se busca un gran nivel de producción en cuanto a respuesta de los procesos que se ejecutan lo que se graban son máscaras en CHIP de ROM. Los programas vienen en forma nativa dentro del computador.

Entrada - Salida

Para que el microcomputador sea útil debe ser capaz de comunicarse con el mundo exterior; y con frecuencia ésto es posible vía keyboard o un keypad en el caso de entrada de información, y para la salida, el monitor y/o la impresora. Los datos digitales de entrada o salida son transferidos vía ports de datos, los que básicamente son registros latched conectados al bus de datos y seleccionados por la CPU. En algunos casos la señal es convertida en una señal analógica antes de salir al exterior.

Aparte de las líneas de datos, el port de salida tendrá líneas de control, a menudo llamadas líneas de Handshake. Una de éstas, es de salida del computador para indicarle al dispositivo externo que está listo para transferir datos vía el port. La segunda línea es de entrada al computador, y es usado por el dispositivo externo tanto para indicar que el periférico está listo para aceptar datos o que tiene datos disponibles en las líneas del port; para que el computador lea. Estas señales pueden ser usadas como pedido de una acción en un sentido o una señal de comprendido (acknowledgement) en el otro sentido, o puede ser simplemente para indicar que el dispositivo está ocupado. Las señales de Handshake son importantes, cuando el microprocesador y el dispositivo externo operan a diferentes velocidades.

Longitud de una palabra.

Los microprocesadores y los microcomputadores trabajan con datos del tipo binario, los cuales son administrados como grupos de dígitos binarios (bits); y éstos grupos

son llamados "palabras de datos". Típicamente estas palabras pueden contener 4, 8, 16, ó 32 bits de datos de acuerdo al tipo de sistema que se considere.

En los pequeños y primeros micros, las palabras eran de 4 bits. A ésta agrupación se la llamó nibble, y con éste tamaño el rango de palabras diferentes que podía manejar era de 0 a 15. Las palabras mayores debían ser procesadas en forma fraccionada mediante la ayuda de hardware adicional.

En los microprocesadores de usos generales el tamaño de la palabra es de 8 bits, que se la conoce como byte. Un byte puede tomar representaciones que van desde 0 a 255 => 256 combinaciones y de ésta forma permiten el manejo de caracteres tipo texto, con su modalidad de mayúsculas y minúsculas, y signos de puntuación.

En sistemas más grandes el tamaño aumenta a unos 16 bits permitiendo datos desde 0 a 65535. Algunos de éstos procesadores pueden llegar a incluir registros de 32 bits internos para manejar datos.

En la mayoría de los casos el bus de datos es el que define el tamaño de la palabra de trabajo. Un ejemplo de esto, es el Intel 8088 que pudiendo manejar 16 bits de datos su arquitectura es de 8 bits de bus de datos, y cada transferencia de una palabra necesita de 2 transacciones a través del bus.

El sistema de direccionamiento en procesadores de 8 bits de palabra de dato es de 16 bits de ancho permitiendo manejar una memoria de 65536 posiciones diferentes. Para aquellos de 16 bits de dato presentan 20 ó 24 bits para direccionamiento, permitiendo manejar varios megabytes de direcciones de memoria.

Frecuentemente con el propósito de reducir el tamaño de encapsulado del microprocesador, el bus de datos y de direcciones es usado en forma multiplexada tomando la información de los mismos pines del procesador pero a distintos tiempos.

MEMORIA PRINCIPAL

Unos de los dispositivos de apoyo más importantes para un microprocesador es la memoria, utilizada para almacenar las instrucciones de los programas y los datos a ser procesados. En una computadora la memoria normalmente estará incluida en la misma pastilla que la unidad central de procesamiento. Para la mayoría de aplicaciones este tipo de dispositivo no necesitará memoria externa. Sin embargo para un sistema de microprocesador normal la memoria será externa al chip del microprocesador.

Dos tipos de memoria pueden ser utilizados con un sistema de procesador. La primera es una memoria de lectura y escritura, generalmente llamada RAM (memoria de acceso al azar). Esta es utilizada normalmente como almacenamiento de datos y como almacenamiento para cálculos o resultados intermedios. En un sistema de computación, tal como una computadora de negocios o personal, la RAM puede también ser utilizada para almacenar las instrucciones de los programas, ya que estas variarán de una tarea a otra.

Un problema con la RAM es que si el suministro de energía se retira la información almacenada se perderá. Este tipo de memoria se denomina memoria volátil. Donde las aplicaciones hacen uso del mismo programa en todo momento es aconsejable que la memoria de programa debiera retener sus contenidos, aún cuando la computadora este desconectada, de modo que el programa no necesite ser cargado en la máquina cada vez que se encienda el equipo. Para obtener este comportamiento existe un segundo tipo de dispositivo de memoria, conocido como memoria de sólo lectura o ROM. El almacenamiento en una ROM es de tipo permanente, aunque algunos tipos tienen la facilidad para ser borradas y reprogramadas. Este tipo de memoria son llamadas memorias-permanentes (no volátil).

Las memorias de acceso al azar (RAM).

Una memoria RAM de lectura-escritura contiene un arreglo grande de celdas-de-memoria, cada una de las cuales se usa para almacenar un bit de dato. Un sistema adicional de direccionamiento se incluye en la pastilla, para permitir la selección de celdas o grupos en forma individual que tienen datos a escribirse o a leerse. Normalmente las celdas están ordenadas en filas y columnas, el sistema direccionador se utiliza para seleccionar una fila y una columna con el fin de identificar una celda en particular. La celda cuando se selecciona estará conectada al bus de datos que transfiere información de o desde la celda.

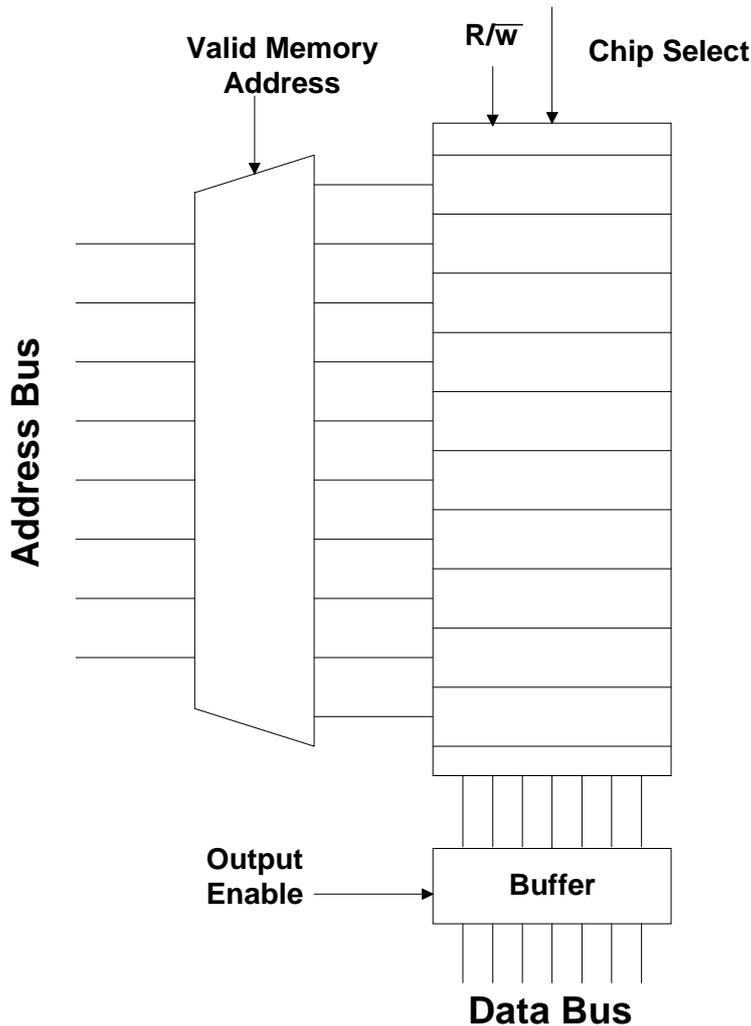
Dos modos básicos de operación interna están disponibles en memorias de tipo RAM, estos son estáticos y dinámicos cada uno tendrá sus ventajas y desventajas.

Las memorias estáticas.

En una RAM estática cada celda contiene un circuito de latching similar a una biestable lógico, que una vez que se establece en un estado lógico retendrá éste hasta que se reprogramme. Así una vez que el dato ha sido escrito en la memoria esta lo retendrá indefinidamente, siempre que la memoria permanezca con energía. Para leer desde este tipo de memoria la dirección deseada se aplica y después de una corta demora la información almacenada aparecerá en las líneas de salida. La demora entre la entrega de la dirección y salida de información se conoce como **Tiempo de Acceso**.

Las primeras memorias de este tipo fueron arreglos que direccionaban de una celda por vez, llegando a dar la posibilidad 1024 x 1 bit de memoria. En un ordenador generalmente tenemos un bus de datos de 8, o 16 bits, para lograr este ancho de palabra habría que tener 8 o 16 memorias de 1024 x 1 bit, direccionándolas en paralelo y cada salida asignarla a una línea del bus de datos. Los tipos posteriores de memoria lograron que

cada vez que se las direccionaba se pudiera ingresar o tomar de a 8 o 16 bits en paralelo de la salida.



La memorias dinámica.

Un problema con el tipo estático de memoria es que la celda utiliza un circuito relativamente complicado y en los inicios de la producción de memoria fue difícil obtener una memoria estática de gran tamaño en un chip de silicio.

Un tipo alternativo de celda es usar para almacenar una carga en un capacitor como memoria. Este tipo de celda de memoria es más simple que un tipo estático, pero debido a las pérdidas de energía los datos se deterioran. Para resolver este problema la información en la celda es refrescada de vez en cuando simplemente leyendo o escribiendo de nuevo la celda, con el fin de restaurar la carga en el capacitor. Este tipo de memoria es conocido como memoria dinámica debido a la necesidad de refrescar en forma continua para retener la información. Durante un ciclo refresco todas las celdas de memoria necesitan ser leídas y escritas. En la mayoría de las memorias dinámicas modernas se organizan tal que la totalidad de una fila o columnas sea refrescada simultáneamente de modo que el número de operaciones de refresco en un ciclo se reduzca.

Debido al tamaño de celda más pequeño, las memorias dinámicas tendrán en general mucho mayor capacidad que las estáticas.

La mayor capacidad implica tener mayor cantidad de líneas de direccionamiento. Por ejemplo una memoria de 64K necesitará 16 líneas de dirección para seleccionar en forma individual las celdas. Con el fin de mantener pequeño el tamaño del encapsulado es

normal para el direccionamiento de la memoria el uso del multiplexado de modo que las direcciones de fila y columna se apliquen en forma sucesiva al mismo conjunto de pines del encapsulado.

Con la dirección de fila se aplica un pulso de RAS (Row Address Strobe) al chip y esta dirección se latchea en un registro interno de la memoria. Luego la dirección de la columna se aplica al mismo conjunto de pines y es acompañada por un pulso de CAS (Column Address Strobe) la que es latcheada en un registro permitiendo seleccionar una determinada celda para la lectura o escritura. La información normalmente aparecerá en la salida mientras la línea del CAS se encuentre en baja después de que una dirección ha sido aplicada. La operación de refresco necesitará sólo de aplicar la dirección para la fila y el pulso de RAS, al refrescar en forma automática la fila.

Las Memorias de Sólo Lectura.

Existen dos tipos de memorias básicas de ROM. Un tipo, es conocida como ROM programada por máscara, el patrón de información se construye cuando se fabrica la memoria y no se puede alterar. Un tipo alternativo a esta es conocida como PROM (Programable Read Only Memory), esta cuando se fabrica tendrá todas sus celdas en el mismo estado que puede ser 0 o 1 de acuerdo con el tipo de dispositivo. Los datos son programados en la memoria seleccionando las celdas y aplicando un pulso grande de energía eléctrica para establecer el nuevo estado. Una vez que la memoria está programada retendrá la información indefinidamente aún cuando el suministro de energía sea retirado.

El tipo normal de PROM tiene vínculos de alambre fusibles o diodos en cada una de sus celdas. Durante la programación estos vínculos serán selectivamente fundidos para alterar el estado de las celdas y así ingresar el patrón de datos deseados en la memoria. Una vez que se ha programado la PROM es similar a una ROM programada por máscara y no puede alterarse, aunque sus celdas no programadas si lo pueden hacer. En la actualidad existen varios tipos de PROM disponibles que son borrables. Uno tipo, generalmente llamado EPROM, puede ser borrado exponiendo la pastilla a luz ultra violeta, y generalmente tienen una ventana de cuarzo en la tapa del encapsulado para permitir que luz penetre en la pastilla. Después de una exposición quizás de 10 o 20 minutos todas las celdas en la pastilla se restauran a su estado original. En este tipo de memoria la información es almacenada como una carga eléctrica dentro de la celda, pero a diferencia de un tipo dinámico RAM el tiempo de almacenamiento para un EPROM es de muchos años. Un segundo tipo de PROM utiliza una señal eléctrica para borrar la información, y esta es llamada EEPROM (Electrically Erasable PROM).

Los PROMs y EPROMs son convenientes para el uso en desarrollo de prototipo y producciones pequeñas, para una producción grande el tipo de máscara ROM generalmente será más económico.

Un tipo de memoria no volátil posterior es la ROM eléctricamente alterable o EAROM, que al igual que una PROM o EPROM usan una programación eléctrica. Este dispositivo, sin embargo es posible usarlo como memoria RAM excepto que es más lento la operación de escritura. La diferencia respecto de una RAM es que la información estará retenida, en gran medida de la misma manera como para un PROM, cuando el suministro de energía sea retirado. Hay aplicaciones, donde un EAROM es útil, particularmente en sistemas de tipo adaptables donde los parámetros del programa pueden necesitar ser variado pero se requiera una memoria-permanente.