

ARQUITECTURA DEL PROCESADOR

ESTRUCTURA GENERAL

Una computadora digital es un sistema de procesadores, memorias y dispositivos de entrada/salida interconectados. Veremos aquí estos componentes y su interconexión. Procesadores, memorias y dispositivos de entrada/salida son conceptos claves por lo que iniciaremos el estudio de la arquitectura del procesador analizando cada uno de ellos.

DESCRIPCION DE LOS COMPONENTES DE UN PROCESADOR

En una visión muy general, un ordenador lo podemos asimilar a la siguiente situación. Supongamos una oficina, donde tenemos un "Señor" sentado frente a un escritorio, en donde tiene una calculadora, una cierta cantidad de papeles en blanco donde escribir datos, y un contador que le indica el número de la casilla que deber leer a continuación. A su espalda tiene una serie de casilleros numerados en donde puede guardar o sacar información y un lugar preciso tiene una determinada cantidad de tarjetas en donde están escritas instrucciones. Además, en la pared hay un reloj donde se le indica el tiempo que tiene para resolver cada instrucción.

Por otro lado, también tiene a cada lado un canasto, el de su derecha es donde se le introducen datos, requerimientos ó instrucciones y el de su izquierda es donde él entrega datos o respuestas a los requerimientos. Esto se puede apreciar en la siguiente figura (Figura 1).

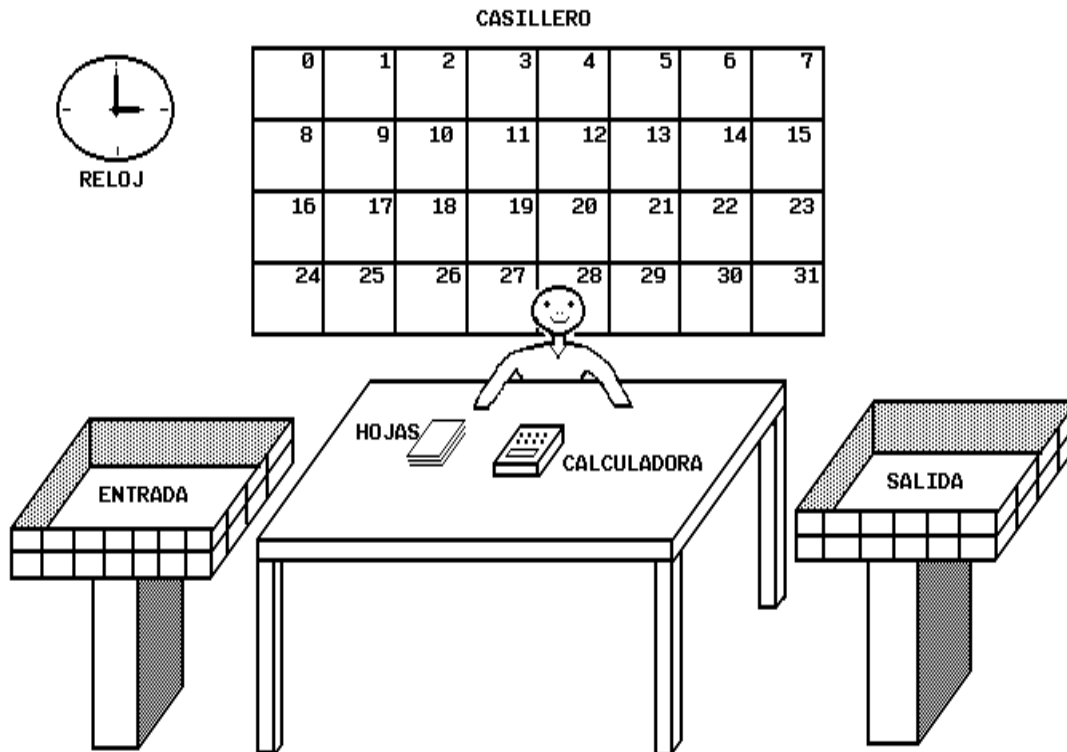


DIAGRAMA ESQUEMATICO ANALOGO DE UNA COMPUTADORA

Figura 1

Universidad Tecnológica Nacional - Facultad Regional Córdoba

ARQUITECTURA DE LOS COMPUTADORES

Es importante destacar, que este "Señor" no sabe hacer nada si no se le indica por medio de las tarjetas con instrucciones.

Veamos como funciona el sistema:

1. El "Señor, que llamaremos Unidad de Control, observa el número que le indica el contador de su escritorio, y se va a la casilla indicada, previo a sumar uno al contador.
2. El "Señor", toma la tarjeta de la casilla indicada por el contador.
3. Copia el contenido de esa casilla en un papel en blanco, e interpreta lo que allí se le dice. Vamos a suponer que en esa instrucción se le pide que sume dos números "A" y "B" y que están en las casillas de las cuales se les da el número, y que el resultado "R" lo guarde en otra casilla de la que también se le indica su número.
4. El "Señor" Unidad de Control, va a la casilla en donde se le dice que está guardado "A" y saca su contenido, que copia en uno de los papeles en blanco que tiene sobre su escritorio.
5. Luego va al número de casilla donde se le dice que tiene "B" y realiza la misma operación, copiando el contenido en otro papel en blanco.
6. Toma la calculadora e ingresa "A" y "B" y los suma.
7. Al resultado "R", lo guarda en el número de casilla que se le indico.
8. Una vez que realizó todo lo anterior, y como ha terminado lo que se le indicaba, se fija en el contador que tiene en la mesa y se examina a que número debe ir a buscar la próxima tarjeta.
9. Va realizando los pasos 1 a 7 hasta que encuentra una tarjeta que le dice que se terminaron las instrucciones. En este caso entrega los resultados en el canasto de su izquierda para que otro pueda interpretarlos, y se queda esperando que se le introduzcan por su derecha nuevas instrucciones.

Ahora que sabemos como funciona el "Señor" y todo su entorno, veamos como relacionamos, por analogía, con las partes constitutivas de una computadora y su funcionamiento.

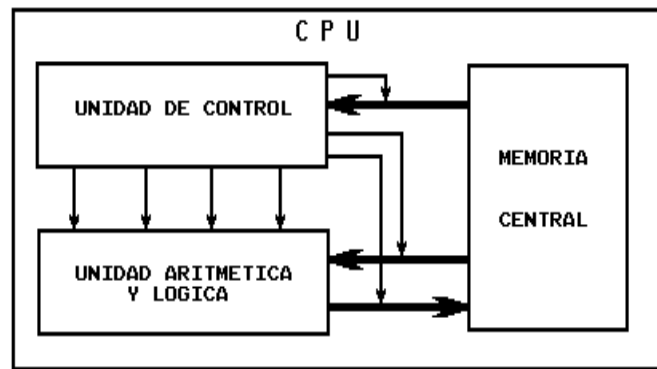
En una visión general, un ordenador se compone de una Unidad Central de Proceso (CPU), en la que tenemos una Unidad de Control (el "Señor"), una Unidad Aritmética y Lógica (la calculadora), los Registros Internos (los papeles en blanco sobre el escritorio) y una Memoria (el casillero a su espalda). Por otro lado tenemos, las Interfaces y Unidades de Entrada/Salida (los canastos a ambos lados). El reloj sobre la pared sirve como elemento de sincronización entre todas las partes del sistema.

En el casillero (memoria) podemos encontrar dos clases de información, por un lado las tarjetas con ordenes acomodadas secuencialmente ("Instrucciones de un programa") y por otro lado los datos, a menudo llamados operandos, con los cuales la máquina efectuará los tratamientos indicados por las instrucciones.

Dentro de la Unidad Central de Proceso (CPU), tenemos dos unidades muy importantes, que tienen correspondencia con las dos clases de información almacenadas. Una es la llamada "Unidad de Control", o también llamada "Unidad de Instrucciones", para las instrucciones del programa, y la segunda, es la "Unidad Aritmética y Lógica", la que realiza operaciones aritméticas sobre los datos, tales como sumar, restar, etc., u operaciones lógicas tales como comparar, o analizar si el contenido de los datos son verdaderos o falsos de acuerdo a la instrucción, etc.

La figura 2 muestra esquemáticamente como pueden resumirse las principales funciones de la CPU.

ARQUITECTURA DE LOS COMPUTADORES



ORGANIZACION GENERAL DE UNA UNIDAD CENTRAL DE PROCESO

Figura 2

La Unidad de Control, que contiene el "Contador de Programa" y el "Registro de Instrucción", extrae de la memoria, en la dirección indicada por el Contador de Programa, la nueva instrucción a ejecutar y la copia en el Registro de Instrucción, el que tiene el siguiente formato:

CODIGO DE OPERACION	DIRECCION 1º OPERANDO	DIRECCION 2º OPERANDO	DIRECCION DEL RESULTADO
---------------------	-----------------------	-----------------------	-------------------------

En donde el "CODIGO DE OPERACION" indica que tipo de operación deberá realizar, y las direcciones de los operandos y del resultado, indican donde ir a buscar los operandos y donde guardar el resultado respectivamente.

Una vez que ha decodificado (interpretado) el tipo de operación a realizar, busca los operandos y los copia en los registros internos de la "Unidad Aritmética y Lógica" (ALU) para que esta realice la operación indicada en el "Código de Operación".

Tal como se ha indicado hasta aquí, la máquina puede ejecutar un programa inicialmente cargado en la memoria central, sobre datos registrados en la memoria central y almacenar los resultados en la memoria central a medida que son obtenidos.

Ahora es necesaria dotarla de medios para comunicarse con el exterior; este es el papel reservado a las Unidades Periféricas.

Existen dos grandes clases de unidades periféricas: "Las Unidades de Comunicación" (lectoras de tarjetas, impresoras, unidades de visualización de rayos catódicos, teclados, etc.), que permiten el diálogo con el exterior; y las "Memorias Auxiliares" ó "Unidades de Almacenamiento Secundario" (discos magnéticos, disquetes, cintas magnéticas, etc.), cuyas capacidades de almacenamiento son muy superiores a las de una memoria principal.

Estas Unidades Periféricas se conectan a la memoria central ya sea directamente o a través de unidades especializadas en la gestión de transferencia de información. Estas "Unidades de Transferencia de Información", se ocupan de gobernar los canales o buses.

En resumen podemos representar un ordenados como un conjunto ensamblado de diferentes unidades, cuyo funcionamiento está regido por un programa registrado en la memoria principal. La Unidad de Control gobierna la

ARQUITECTURA DE LOS COMPUTADORES

ejecución de las operaciones pedidas por el programa. Si la operación es un cálculo, es la Unidad Aritmética y Lógica quien lo realiza, al igual que sí se debiera efectuar una comparación lógica para tomar una decisión. Si es una transferencia de información con el exterior, se cede el control a un canal o bus.

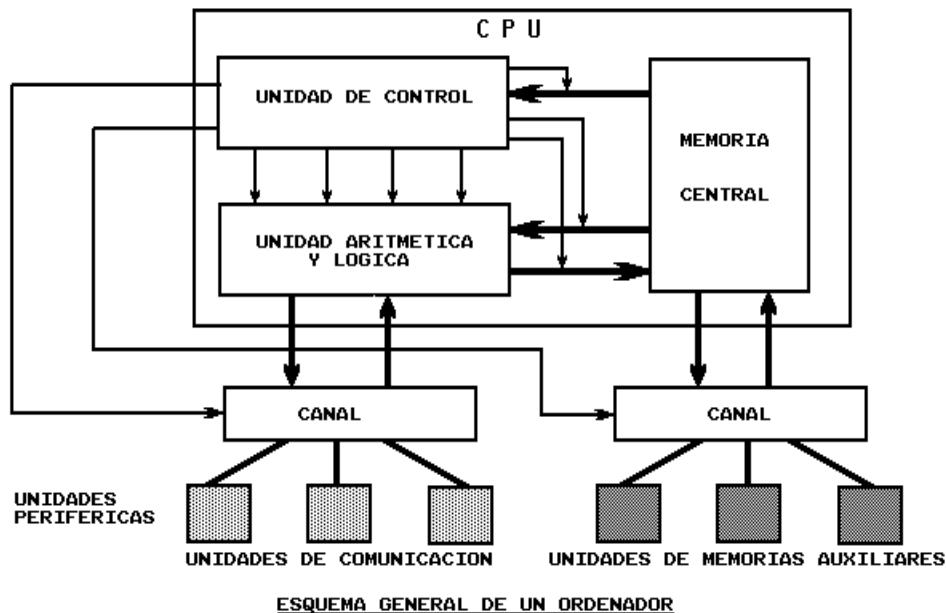


Figura 3

MICROPROCESADORES

En la Figura 4 se puede apreciar la organización de una micro-computadora con un solo bus o canal. La Unidad Central de Procesamiento (CPU) es el "cerebro" de la computadora. Su función es ejecutar programas almacenados en la memoria central tomando las instrucciones examinándolas y luego ejecutándolas una a continuación de otra.

La CPU se compone de varias partes. La Unidad de Control, que se encarga de traer las instrucciones de la memoria principal y determinar su tipo. La Unidad Aritmética y Lógica que realiza operaciones como la suma o la función booleana AND, necesarias para llevar a cabo las instrucciones.

La CPU también contiene una pequeña memoria de alta velocidad utilizada para almacenar los resultados intermedios y cierta información de control. Esta memoria consta de varios registros, cada uno de los cuales tiene cierta función.

El registro más importante es el "Contador de Programa" (PC = Program Counter), que indica la **dirección de la próxima instrucción** que debe ejecutarse. El nombre "Contador de Programa" a veces confunde ya que no cuenta nada, pero su uso es muy difundido.

Otro registro muy importante es el "Registro de Instrucción", que contiene la **instrucción que se está ejecutando**. No obstante, la mayoría de las computadoras tienen otros registros, algunos de los cuales son accesibles a los programadores.

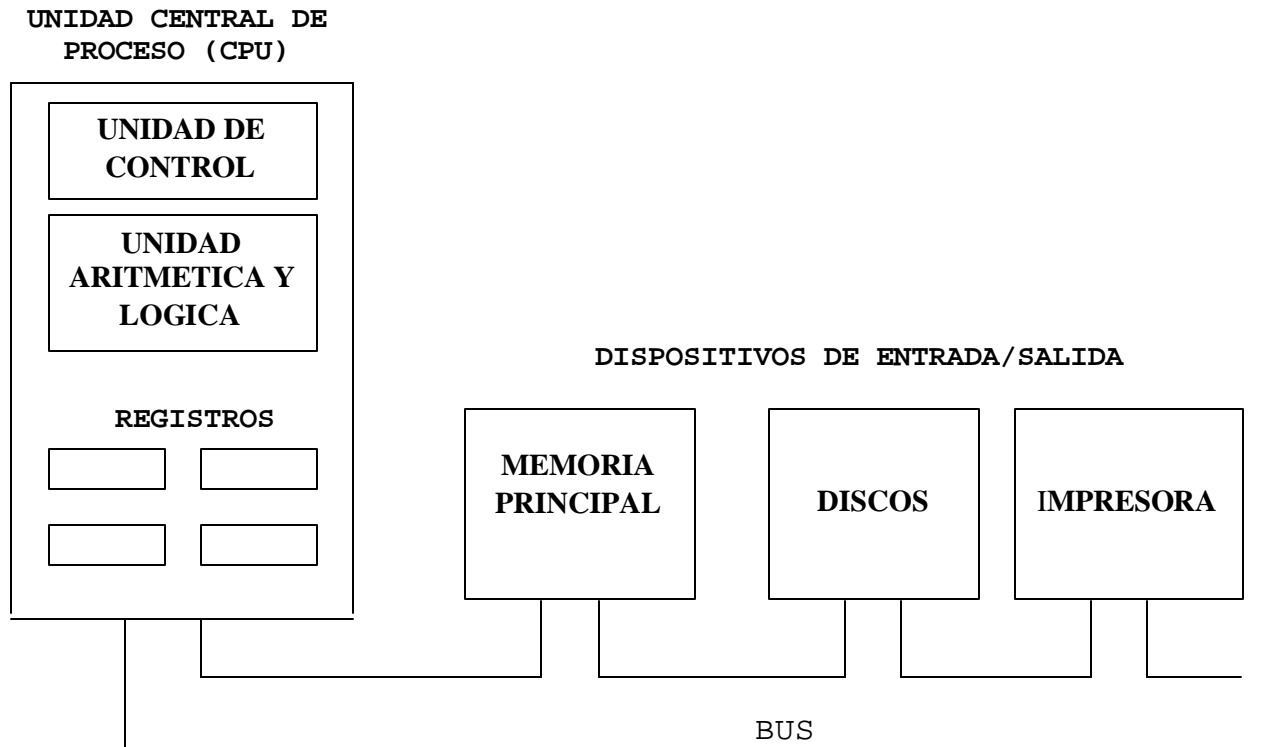


Figura 4 : Organización de una Computadora sencilla

EJECUCION DE LAS INSTRUCCIONES

En la CPU se ejecutan instrucciones en una serie de pequeños pasos:

1. Extrae de la memoria la siguiente instrucción y la lleva al registro de instrucción.
2. Cambia el Contador de Programa de modo que señale la siguiente instrucción.
3. Determina el tipo de instrucción que acaba de extraer.
4. Verifica si la instrucción requiere datos de la memoria y, si es así, determina donde están situados.
5. Extrae los datos, si los hay, y los carga en los registros internos de la CPU.
6. Ejecuta la instrucción.
7. Almacena los resultados en el lugar apropiado.
8. Vuelve al punto 1 para empezar la ejecución de la instrucción siguiente.

A esta secuencia de pasos se la denomina ciclo de máquina y es la forma básica de operación de toda computadora.

Esta descripción de como funciona la CPU se parece mucho a un programa. El hecho de que sea posible escribir un programa que imite el funcionamiento de la CPU demuestra que no es necesario que un programa sea ejecutado por una CPU "física", es decir, un dispositivo compuesto de elementos electrónicos, sino que puede utilizarse otro programa que extraiga, examine y ejecute las instrucciones.

Un programa que realiza esto se llama "intérprete".

Esta equivalencia entre el intérprete y los procesadores físicos es muy importante en la organización de las computadoras. Después de haber especificado un lenguaje de máquina "L" para una determinada computadora, el equipo de diseño puede decidir si quiere construir un procesador fí-

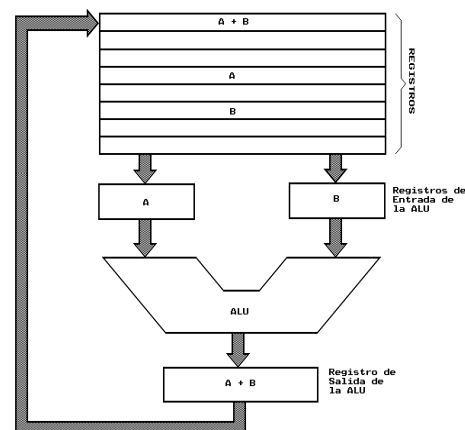
sico para ejecutar programas en "L" o si prefiere escribir un intérprete en su lugar que realice la misma función.

Dado que un intérprete se divide en pasos pequeños las instrucciones que ejecuta, la máquina en la que funciona puede ser mucho más simple y por lo tanto, menos cara de lo que costaría un procesador de hardware para la máquina. Por estas razones, los programas en computadoras modernas, al nivel de máquina convencional, son ejecutados por un intérprete que corre en una máquina mucho más primitiva y al que se llama "nivel de microprogramación".

El conjunto de todas las instrucciones que tiene a su disposición un programador en un nivel se denomina "conjunto de instrucciones". La cantidad de instrucciones varía de una máquina a otra y de nivel en nivel. Por ejemplo, para el nivel de máquina convencional, el conjunto suele tener de 20 a 300 instrucciones. Un conjunto de instrucciones grandes no tiene por que ser mejor que uno pequeño. De hecho suele ocurrir lo contrario. Un conjunto de instrucciones grandes indica, a menudo, que las instrucciones no son muy generales. Los compiladores para alto nivel como Ada, Pascal, etc, se desempeñan mejor en máquinas con conjuntos de instrucciones reducidos y bien definidos. A las máquinas con conjuntos de instrucciones muy reducidos se las denomina "RISC" (Reduced Instruction Set Computer). Este tipo de máquinas no utiliza microprogramación y son muy rápidas. A las máquinas con conjuntos de instrucciones normales o extendidos se las llama "CISC" (Completed Instruction Set Computer).

ORGANIZACION DE LA CPU

La organización interna de una parte clásica de la CPU se debe al diseño de "Von Neumann", y se muestra en la Figura 5. A esta parte se la denomina la "trayectoria de los datos", y consiste en los registros (normalmente de 1 a 16) y en la ALU (Unidad Aritmética-Lógica). Dichos registros alimentan dos entradas a la ALU (A y B en la figura), los cuales retienen información mientras la ALU realiza sus cálculos.



TRAYECTORIA DE LOS DATOS EN UNA MÁQUINA CLÁSICA DE VON NEUMANN

Figura 5

La Unidad Aritmética y Lógica realiza sumas, restas y otras operaciones sencillas sobre el contenido de los registros de entrada, colocando el resultado en el registro de salida, el cual a su vez puede ser almacenado en la memoria.

Las instrucciones pueden dividirse en tres categorías:

- de registro a memoria.
- de registro a registro.
- de memoria a memoria.

Las instrucciones de registro a memoria permiten que una palabra de memoria sea extraída a los registros, donde pueden ser usadas como entradas para la ALU en instrucciones subsecuentes.

Una instrucción típica de registro a registro extrae dos operandos de los registros, los lleva a los registros de entrada de la ALU, lleva a cabo alguna operación con ellos y almacena el resultado en otro registro.

Una instrucción de memoria a memoria extrae sus operandos de la memoria y lleva estos a los registros de entrada de la ALU, donde realiza su operación y escribe el resultado en la memoria.

La operación de flujo de datos es el corazón de la mayoría de las CPU y define en gran parte lo que una máquina es capaz de hacer.

MEMORIA

La memoria es aquella parte de la computadora en donde se almacenan los programas y los datos. No existirían las computadoras digitales sin una memoria en la que los procesadores pudieran leer o escribir información.

Puede considerarse como un conjunto de células, cada una con posibilidad de almacenar una información (palabra), dato o instrucción. Las células están numeradas y la unidad de control conoce cada célula por su número, llamado "Dirección". La Unidad de Control puede pedir leer el contenido, o escribir una nueva información en una célula de una dirección determinada. Para realizar estas operaciones, la Unidad de Control proporciona la dirección de la célula implicada a un registro asociado a la memoria central, llamado "Registro de Dirección", o también "Registro de Selección" de memoria.

El dispositivo de selección de memoria analiza el contenido del registro de dirección y sensibiliza la célula implicada, ya sea para una lectura, o para una escritura. Si se trata de una lectura, la información almacenada en la célula ser transferida a un segundo registro, también asociado a la memoria central, llamado "Registro de Intercambio" ó "Registro de Palabra".

En el caso de una escritura, previamente habrá sido preciso cargar en este registro con la información que se quiere transferir a la célula en cuestión.

La operación de lectura no destruye la información almacenada en la célula, mientras que la operación de escritura destruye la información almacenada, sustituyéndola por una nueva información. La figura 6 muestra esta relación.

La unidad básica de memorias es el "dígito binario" al que se denomina "bit". Este puede contener un "0" ó un "1" y es la unidad más simple posible.

Normalmente se dice que las computadoras utilizan aritmética binaria porque es "más eficiente". En realidad lo que quieren decir es que la información digital puede almacenarse distinguiendo entre valores de alguna cantidad física continua, como un voltaje o la corriente. El sistema de numeración binario necesita únicamente diferencias ente dos valores, en consecuencia es el más fiable para codificar información digital.

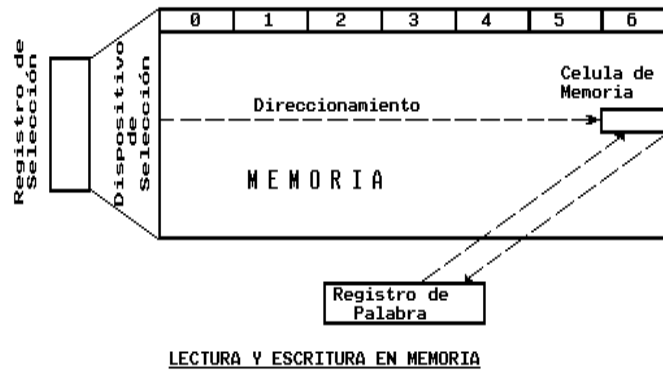


Figura 6

Algunas máquinas, tales como los grandes computadores, anuncian tanto aritmética decimal como binaria. Este truco lo logran utilizando 4 bit para almacenar un dígito decimal. Con cuatro bits se tiene 16 combinaciones que se utilizan para codificar los dígitos 0 al 9, dejando 6 combinaciones sin usar. A continuación se muestra el número 1944 codificado en forma decimal y en binaria pura, usando 16 bits en cada ejemplo:

DECIMAL : 0001 1001 0100 0100

BINARIO : 0000011110011000

En el formato decimal, 16 bits pueden almacenar los números 0 al 9999, teniendo únicamente 10.000 combinaciones, mientras que un número binario puro de 16 bits puede almacenar 65.536 combinaciones diferentes. Por esta razón se dice que el binario es más eficiente.

sin embargo, supongamos que se inventara un dispositivo electrónico altamente fiable que pudiera almacenar directamente los dígitos 0 al 9 en diez niveles de voltajes. Cuatro de estas unidades podrían almacenar cualquier número decimal entre 0 y 9999. Los cuatro elementos proporcionarían 10.000 combinaciones. Con tales dispositivos, obviamente, el sistema decimal es más eficiente.

DIRECCIONES DE MEMORIA

Las memorias constan de cierto número de "celdas" (ó posiciones) cada una de las cuales puede almacenar una porción de información. Cada celda tiene un número asociado, denominado "dirección", por el cual los programas pueden referirse a ella. Si una memoria tiene "n" celdas, tendrá las direcciones de 0 a (n-1). Todas las celdas de una memoria tienen la misma cantidad de bit. Si una celda tiene "k" bits, podrá tener cualquiera de sus "2k" combinaciones. Las celdas adyacentes tienen direcciones consecutivas.

Las computadoras que utilizan sistemas binarios también expresan las direcciones de memoria como números binarios. Si una dirección tiene "m" bits, el número máximo de celdas direccionables es directamente "2m".

El número de bits de la dirección está relacionado con el número de celdas direccionables directamente en la memoria y es independiente del número de bits por celda. Una memoria con "212" celdas de 8 bits y otra de "212" celdas de 60 bits, necesitarán, cada una, direcciones de 12 bits.

A continuación se dan algunas celdas de computadoras comerciales:

Burroughs B1700	1 bit por celda.
IBM PC	8 bits por celda.
DEC PDP-8	12 bits por celda.
IBM 1130	16 bits por celda.
DEC PDP-15	18 bits por celda.
Honeywell 6180	36 bits por celda.

La importancia de la celda reside en que es la unidad direccionable más pequeña. En los últimos años los fabricantes de computadoras han establecido como norma una celda de 8 bits, la que se denomina "byte". Esto bytes se agrupan en "palabras". Una computadora con una palabra de 16 bits tiene 2 bytes por palabra, mientras que otra con una palabra de 32 bits tiene 4 bytes por palabra.

La relevancia de la palabra está en que la mayor parte de las instrucciones operan sobre palabras enteras, por ejemplo, suma de dos palabras. De ahí que una máquina de 16 bits tendrá registros de 16 bits e instrucciones para manipular palabras de 16 bits, en tanto que una máquina de 32 bits tendrá registros de 32 bits e instrucciones (sumar, restar, mover, etc.) para manipular palabras de 32 bits.

ORDENAMIENTO DE BYTES

Los bytes en una palabra pueden numerarse de izquierda a derecha o de derecha a izquierda. En principio esto podría parecer irrelevante, pero como veremos, esto tiene grandes implicaciones.

La figura 7(a) presenta una parte de memoria de una computadora de 32 bits, cuyos bytes están numerados de izquierda a derecha (máquinas de la familia Motorola). La figura 7(b) muestra la presentación análoga de una computadora con numeración de derecha a izquierda (familia Intel).

Dirección	BIG ENDIAN				Dirección	LITTLE ENDIAN			
Byte	0	1	2	3	Byte	3	2	1	0
	4	5	6	7		7	6	5	4
	8	9	10	11		11	10	9	8
	12	13	14	15		15	14	13	12
	Palabra de 32 bits					Palabra de 32 bits			

(a) Memoria Big endian

(b) Memoria Little endian

Figura 7

Para el primer caso, en donde comienzan los números se numeran por el extremo mayor (es decir los bytes de orden superior), estas computadoras se denominan "big endian", en contraposición a las llamadas "little endian" de la figura 7(b). Estos términos (en inglés) se deben a Jonathan Swift, quien en su libro "Los viajes de Gulliver", satirizó a los políticos que iniciaban una guerra debido a la discusión acerca de si los huevos debían romperse por el extremo mayor o menor.

Es importante entender que en ambos sistemas un número entero de 32 bits, con un valor de por ejemplo 6, se representa con los bits 110 en los tres bits más hacia la derecha (de orden inferior) de la palabra y ceros en los restantes 29 bits hacia la izquierda.

ARQUITECTURA DE LOS COMPUTADORES

En el esquema "big endian", estos bits se localizan en el byte 3 (ó 7, ú 11, ó 15, etc), mientras que en el esquema "little endian" corresponden al byte 0 (ó 4, ú 8, ó 12, etc). En ambos casos la palabra en que se ubica este número entero tiene la dirección 0 (cero).

Si las computadoras solo almacenaran enteros, entonces no habría problemas, sin embargo, muchas aplicaciones requieren de una mezcla de enteros, cadenas de caracteres y otro tipo de datos. Considérese un simple registro consistente en una cadena alfabética (por ejemplo nombre de los alumnos) y el de enteros (como puede ser su número de legajo, su edad, etc.). Para completar la palabra, la cadena se termina con uno ó más bytes de ceros.

Las dos representaciones son correctas, pero los problemas comienzan cuando una máquina trata de enviar un registro a la otra a través de una red. Imagínese que la "big endian" envía un registro a la "little endian" byte por byte, empezando por el byte 0. Eso sin considerar que los bits también se invierten dentro de una palabra.

Obviamente no es una solución simple, ya que no es lo mismo tratar datos numéricos que alfabéticos. Debe quedar claro que la falta de una norma en el ordenamiento de bytes, constituye una grave molestia cuando se intercambia información entre máquinas diferentes.

CODIGOS CORRECTORES DE ERRORES

La memoria de una computadora puede en ocasiones cometer errores debido a picos o caídas de voltajes en el suministro de energía eléctrica u otras causas. Para prevenir tales errores, la mayoría usan códigos de detección o de corrección de errores.

Cuando se usan estos códigos se agregan bits extras a cada palabra de memoria en una forma especial. Cuando se lee una palabra de la memoria, se verifican los bit adicionales para ver si ha ocurrido un error.

Para entender como se manejan los errores, es necesario entender que es en realidad un error. Supongamos que una palabra de memoria es de "m" bits de datos a los que se agregan "r" bits redundantes o de verificación, siendo "n" la longitud total (es decir $n = m + r$). Para referirse a una unidad de "n" bits, conteniendo "m" datos y "r" bits de verificación, se utiliza el término "palabra código".

Dadas dos palabras códigos cualquiera, es posible determinar en cuantos bits difieren. Por ejemplo, entre las palabras código 10001001 y 10110001, hay una diferencia de 3 bits.

El número de bits en el que difieren dos palabras códigos se llama "distancia de Hamming". Esto significa que si dos palabras están separadas ente sí por una distancia de Hamming de "d" bits, se necesitarán "d" de verificación para convertir un código en otro.

Con una palabra de memoria de "m" bits, existen "2^m" combinaciones de bits legales, pero por la forma en que se calculan los bits de verificación, únicamente "2^m" de las "2ⁿ" palabra códigos son válidas. Si una lectura de memoria da una palabra inválida, la computadora sabe que ha ocurrido un error de memoria. Dado el algoritmo para calcular los bits de verificación es posible encontrar las dos palabras de código cuya distancia de Hamming sea la mínima.

Las propiedades de detección y corrección de errores, dependen de su distancia de Hamming. Para detectar "d" errores de un bit se necesita de un código con una distancia "d + 1", porque con dicho código no es posible que "d" errores de un solo bit puedan cambiar una palabra válida en otra palabra válida.

De la misma manera, para corregir errores de un solo bit se requiere de un código con una distancia "2d + 1", pues así las palabras códigos legales se encuentran tan separadas entre sí, que aun con "d" cambios, la palabra original está más cerca que cualquier otra y puede ser unívocamente determinada.

Como ejemplo sencillo de códigos detectores de errores, consideremos que agregamos un bit de paridad, el cual se elige de modo que el número de bits con valor 1 sea par (o impar). Dicho código tiene una distancia de 2, ya que cualquier error de un solo bit produce una palabra código con la paridad equivocada. Se puede usar para detectar errores sencillos.

CLASIFICACION Y ORGANIZACION DE LAS MEMORIAS

En la descripción de los componentes de un procesador, dentro de la Unidad Central de Proceso, se encuentra la Unidad de Memoria Central ó Principal, en la que se almacenan las instrucciones, los datos u operandos, los resultados parciales y los resultado finales del proceso.

Cuando en un procesador, el sistema es especializado y el número de bits a memorizar es pequeño, se suelen utilizar registros de entrada/salida en paralelo o en serie disponibles como bloques funcionales de 4 a 8 bits en escalas de integración media (MSI) en la mayoría de las tecnologías.

Cuando la cantidad de información a memorizar es elevada y en especial cuando el sistema digital de proceso es programable, y han de memorizarse las instrucciones del programa, además de los datos, nunca se necesita tener acceso a toda la información simultáneamente. Por ello los elementos que memorizan información se pueden agrupar constituyendo una unidad de memoria en la que sólo se pueden acceder simultáneamente a un cierto número de aquellos.

"Se define por lo tanto una unidad de memoria como un conjunto de dispositivos biestables (que pueden encontrarse en dos estados distintos) agrupados de tal manera que solamente es posible introducir información o leer la que poseen simultáneamente en un grupo reducido de ellos".

Las características que ha de poseer una unidad de memoria dependen:

- De la forma en que van a presentar la información a su entrada.
- De la forma en que se obtienen la información a su salida.
- De la cantidad de información a memorizar.

Nos referiremos exclusivamente a las memorias que forman parte integrante de un sistema digital de proceso y que están físicamente situadas en el mismo recinto físico que la Unidad de control y la Unidad Aritmética y Lógica. Estas memorias constituyen lo que se denomina la Memoria Central del sistema. Estudiaremos los parámetros más importantes de una memoria y la división de las memorias de acuerdo con estos parámetros. Analizaremos los diferentes tipos de memorias desde el punto de vista funcional independiente de la tecnología.

PARAMETROS Y CARACTERISTICAS MAS IMPORTANTES DE UNA MEMORIA

CAPACIDAD

Es la cantidad de información que puede almacenar. Dado que la información se almacena en el sistema binario, la capacidad se mide en la unidad de medida de información en dicho sistema, que es el "bit".

La capacidad se puede considerar, por lo tanto, igual al numero "N" total de bits. Pero en general, la información se almacena en grupos de bits denominados "posiciones" o "palabras", formadas por un cierto número "n" de bits accesibles simultáneamente.

El número de palabras o posiciones de una memoria viene dado por lo tanto por el cociente:

$$m = \frac{N}{n}$$

A la selección de una determinada posición de memoria para introducir información en ella o leer la que contiene, se la denomina "direccionamiento" de dicha posición. Para que se aproveche al máximo la capacidad de selección, se hace que "m" sea una potencia de "2".

El número de variables binarias necesarias para poder seleccionar "m" es "b", tal que:

$$m = 2^b$$

La unidad de capacidad, en lo que se refiere al número de posiciones de memoria se considera igual a 1024 (por ser la potencia de 2 más próxima a 1000) y se la denomina "k".

Para seleccionar 1024 posiciones de memoria se necesitan 10 bits, ya que:

$$m = 2^b = 2^{10} = 1024$$

Como practica, el alumno puede comprobar que si "b = 14", la memoria tiene una capacidad de 16K. El número total de posiciones de una memoria de "n . k" es "n x 1024". Por ejemplo, una memoria de "32 k" posee: 32 x 1024 = 32798 posiciones. La capacidad total de una memoria se suele indicar mediante el número de posiciones y el número de bits de cada posición.

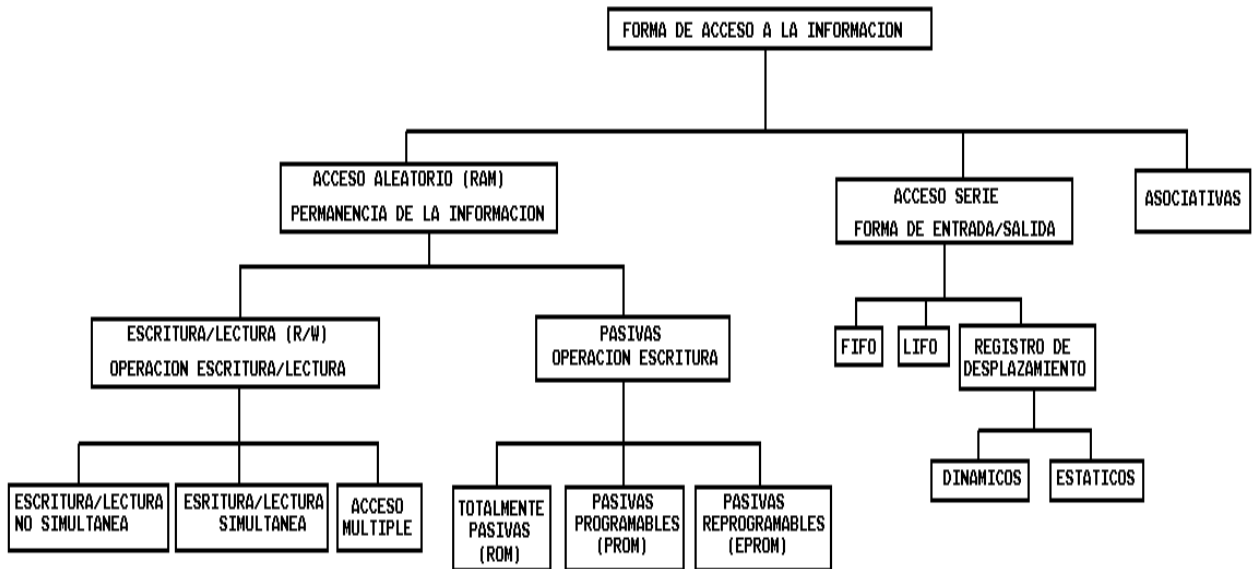
FORMAS DE ACCEDER A LAS POSICIONES DE MEMORIA

La acción de introducir información (escribir ó grabar) en una posición de memoria, o leer la que contiene, se denomina "acceso a la memoria". Tal como se dijo, nunca es necesario leer o escribir simultáneamente todas las posiciones de una memoria. Por ello existen varias formas de acceso a una posición de memoria, lo que da lugar a la división de las memorias indicadas en la figura 8.

En esta figura se representa una visión panorámica de los diversos tipos de memorias y la interrelación que existe entre ellas.

En esta clasificación se incluyen solamente las memorias que no poseen partes en movimiento y que en general se encuentran en el mismo recinto físico que la Unidad de Control y la Unidad Aritmética y Lógica.

El almacenamiento de información en cantidades elevadas (cientos de miles de "k" palabras), tanto de datos como de instrucciones de los sistemas programables, se realiza actualmente en sistemas físicos que poseen partes en movimiento y reciben el nombre genérico de periféricos.



CLASIFICACION DE LAS MEMORIAS

Figura 8

MEMORIAS DE ACCESO ALEATORIO (RAM = Random Access Memory)

Son aquellas en las que el tiempo que tarda en leerse o escribirse en una posición es el mismo para todas las posiciones de la memoria. Aleatoriamente se pue-

ARQUITECTURA DE LOS COMPUTADORES

de seleccionar cualquier posición de la memoria para introducir información o leer la que contiene. Por lo tanto, el tiempo de acceso a cualquier posición de la memoria es el mismo independientemente de su ubicación o situación. En la figura 9 se representa el esquema de bloques general de una memoria de acceso aleatorio.

Se supone que solamente se puede seleccionar simultáneamente una posición de memoria para leer su contenido o introducir información en ella.

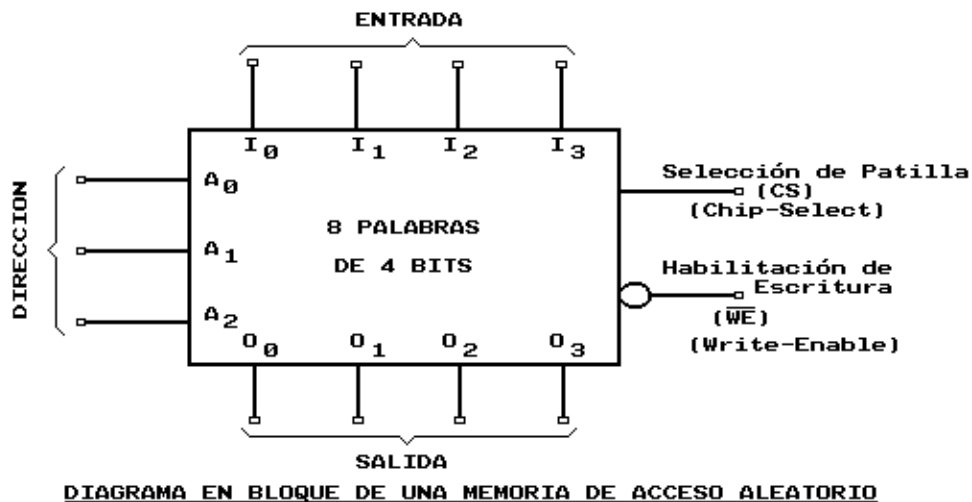


Figura 9

La memoria está organizada en "n" bits, cuya información se desea leer o escribir información simultáneamente.

La selección de la posición cuya información se desea leer o escribir se realiza mediante "b" bits en paralelo, denominados "bits de dirección".

El número total de posiciones que se puede seleccionar es:

$$m = 2^b$$

La asignación de las patitas de una pastilla de circuito integrado (IC = Integrated Circuit) de memoria, como las especifica el fabricante, se indica en la figura 9.

La memoria allí representada tiene la capacidad para almacenar (memorizar) 8 palabras de 4 bits cada una. La pastilla tiene 8 registros individuales "R0, R1, R2 ... R7" y cada registro consta de 4 flip-flop.

Los registros se identifican por los niveles lógicos en los terminales: "A2", "A1" y "A0".

Estos bits se denominan bits de dirección y al conjunto de los bits de dirección se los conoce como "dirección".

Cuando: "A2 A1 A0 = 0 0 0", se accede al registro "R0".

Cuando: "A2 A1 A0 = 0 0 1", se accede al registro "R1".

Y así sucesivamente.

Como hay tres bits de dirección se pueden seleccionar "2b = 8" registros.

Entonces cada uno tiene una dirección y se accede a un registro particular almacenando los bits de dirección del registro en los terminales de dirección.

Los 4 bits que están almacenados, es decir escritos, en un registro direccionado se presentan en la pastilla de memoria en los terminales de entrada "I3 I2 I1 I0". Para escribir una palabra en el registro, este debe direccionarse, la palabra debe presentarse en la entrada, y deben habilitarse o activarse los termina-

Universidad Tecnológica Nacional - Facultad Regional Córdoba

ARQUITECTURA DE LOS COMPUTADORES

les de "Habilitación de Escritura" (WE = Write Enable) y de "Selección de Pastilla" (CS = Chip Select).

En base a la convención adoptada, el diagrama indica que el terminal externo WE se activa en el nivel bajo (activo cuando WE = 0 en un sistema de lógica positiva) y CS se activa en el nivel alto.

En el proceso de lectura, la memoria lleva a los terminales de salida "03 02 01 00" la palabra almacenada en el registro direccionado.

Para leer la memoria presentamos la dirección, habilitamos la entrada CS y ponemos la entrada de WE = 1.

Así pues:

$\overline{\text{WE}} = 0$ escribe en memoria y $\overline{\text{WE}} = 1$ lee de memoria

Por esta razón algunos fabricantes etiquetan los terminales de habilitación de escritura con:

lectura/escritura (R/ $\overline{\text{W}}$)

Tanto para lectura como para escritura la entrada de selección de pastilla "CS) debe estar habilitada, es decir, activada. Si CS no se activa, la pastilla está aislada del mundo exterior. Entonces no es posible ni leer ni escribir en memoria.

El número total de bits en la figura anterior es "8 x 4 = 32".

La forma de disponer los bits en palabras se denomina organización de la memoria. La memoria de la figura anterior se describe como una memoria de 8 palabras de 4 bits/palabra.

Debido a que podemos leer o escribir palabras en la memoria, esta se denomina de "lectura/escritura". Además como podemos escoger cualquier registro de memoria aleatoriamente, tanto para leer como para escribir, también se las conoce como "memorias de acceso aleatorio" ó "RAM = Random Access Memory".

ESTRUCTURA DE UNA RAM SEMICONDUCTORA

En principio la estructura lógica de una RAM semiconductor, del tipo que hemos discutido, se indica en la figura 10. En esta disponemos de cuatro palabras de 2 bits cada una, es decir, la organización de la memoria es "4 x 2". Nuestro interés se centra en la estructura lógica de la memoria y no en los detalles electrónicos. Por lo tanto, hemos incorporado algunas simplificaciones en comparación con una memoria física real.

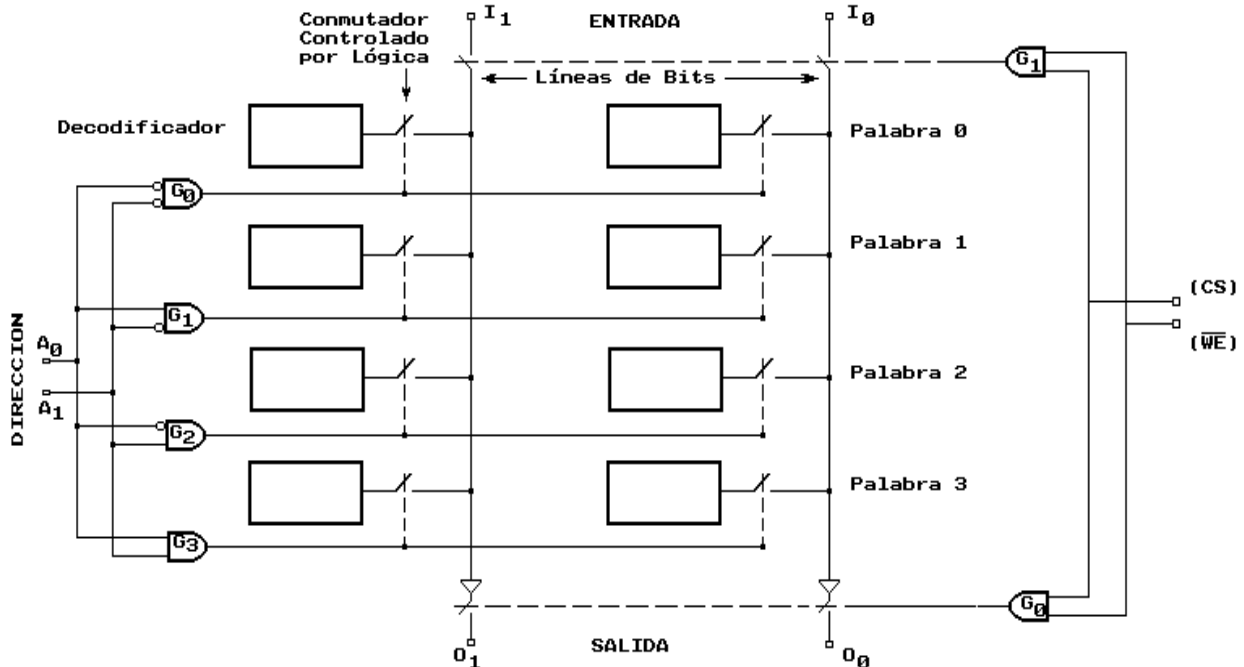
La estructura utiliza el conmutador controlado por lógica, cuando el nivel lógico de la línea de control (línea de trazos que atraviesa el conmutador) es "1", el conmutador se cierra y se establece la conexión; cuando el nivel lógico de la línea de control es "0", el conmutador se abre.

Los bits se almacenan en "Flip-Flop" elementales, es decir formados por dos inversores acoplados. Los bits de dirección "A1" "A0" se aplican al decodificador.

Cuando tenemos, por ejemplo, "A1 = 1" y "A0 = 0", la salida de la puerta G2 está en "1" lógico, mientras que las salidas de las demás compuertas lógicas AND del decodificador están en "0" lógico. Por lo tanto, solo se puede acceder a los flip-flop del Registro de Palabra 2, ya que solamente los conmutadores de estos flip-flops están cerrados y por consiguiente conectados con las líneas de bits. O sea, la entrada de dirección "A1A0 = 10" direcciona la palabra 2 y sólo la palabra 2.

Si la selección de la pastilla está en cero lógico, las salidas de las puertas "G0" y "G1" están en "0" lógico y las líneas de bits no se conectan ni a la entrada de datos ni a los terminales de salida.

Si "CS = 1" y "WE = 0", los conmutadores que conectan "I1" e "I0" a las líneas de bits se cerrarán y los flip-flops asumirán estados correspondientes a los niveles lógicos de las entradas de datos. —



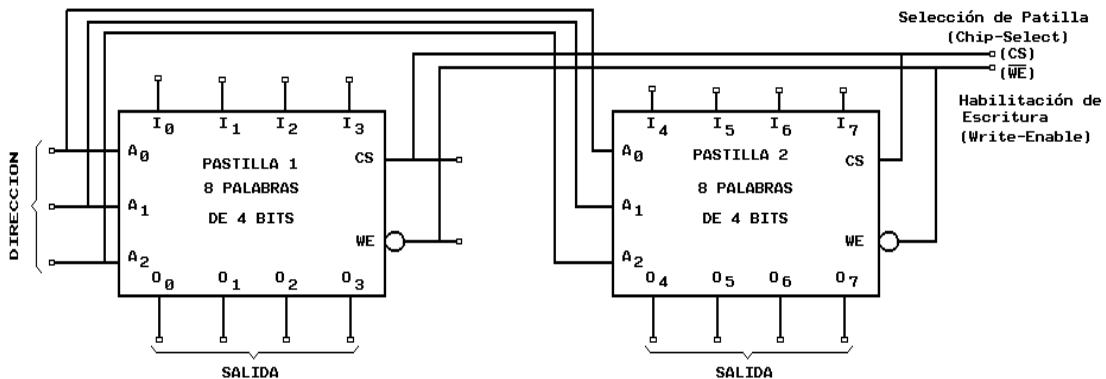
ESTRUCTURA DE UNA RAM DE LECTURA-ESCRITURA DE 4 PALABRAS, 2 BIT/PALABRA

Figura 10

Si "CS = 1" y " $\overline{WE} = 1$ ", los flip-flop se conectarán a los terminales de salida, para que la palabra almacenada pueda ser leída. Comercialmente se disponen de pastillas de Circuito Integrado (IC = Integrated Circuit) que tienen incorporados "65.536 = 216 bits" ó aún mayor en la actualidad, a un precio bastante bajo.

PASTILLAS DE MEMORIA EN PARALELO

Con frecuencia surgen casos en los que no es adecuado el número de palabras de una pastilla, ó el número de bits por palabra, ó las dos cosas simultáneamente. El problema puede remediarse colocando las pastillas en paralelo. En la figura 11 se representa la conexión de pastillas en paralelo para incrementar el número de bits por palabra (pero no el número de palabras).



MEMORIA DE 8 PALABRAS CON 8 BITS POR PALABRA

Figura 11

Hay dos pastillas de 8 palabras en paralelo, 4 bits por palabra, para construir una memoria cuyo número de palabras sigue siendo 8, pero el número de bits se ha

ARQUITECTURA DE LOS COMPUTADORES

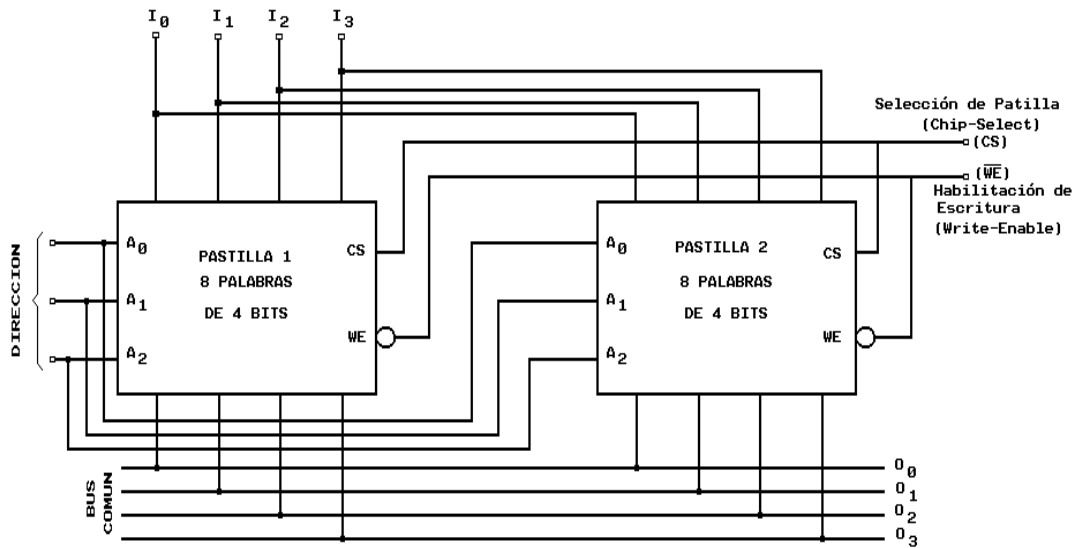
incrementado de 4 a 8. Los 3 bits de dirección se aplican a las pastillas de dirección de ambas memorias. Los terminales "CS" de las pastillas se unen, lo mismo que los terminales WE. Las entradas de selección de pastilla y habilitación de escritura seleccionan y habilitan simultáneamente ambas pastillas.

La pastilla 1 acepta y almacena 4 bits (0,1,2,3), y la pastilla 2, otros 4 bits (4,5,6,7). Por supuesto, se pueden conectar en paralelo más pastillas adicionales. Así, con tres pastillas de 8 palabras, 4bits/palabra, se conseguirá una memoria de 8 palabras, 12 bits/palabra, y así sucesivamente. Cuando se añadan pastillas, la dirección se aplica simultáneamente a los terminales de entrada de dirección de las demás pastillas. Análogamente, todas las patitas CS se conectan entre sí para tener una sola entrada CS y las entradas WE se tratan de la misma forma.

Para dar mayor flexibilidad al problema de los bits por palabra, los fabricantes disponen de pastillas de memoria con palabras de 1 bit. Así pues, encontramos en catálogos de fabricantes memorias cuya organización es "256 x 1", "1024 x 1", "4096 x 1", etc. Una vez seleccionada una pastilla con un número adecuado de palabras, entonces ensamblamos una memoria de "n" bits por palabra, simplemente colocando "n" pastillas en paralelo.

En la figura 12 mostramos la forma de poner en paralelo dos pastillas de 8 palabras, 4 bits/palabra, para obtener una memoria de 16 palabras de 4 bits/palabra. Como antes, los tres bits de dirección se aplican a ambas pastillas, pero en lugar de tener un bit de entrada CS común aplicado a las dos pastillas, cuando se activa la entrada CS de una pastilla, se desactiva la entrada CS de la otra. La entrada CS del sistema de memoria (la de entrada al inversor) ahora es un bit de dirección adicional, que llamamos "A4". Cuando "A4=1", la pastilla direccionada es la 2, y cuando "A4=0", se direcciona la pastilla 1. El bit "A4" de dirección de selección de pastilla selecciona por lo tanto una u otra de las pastillas.

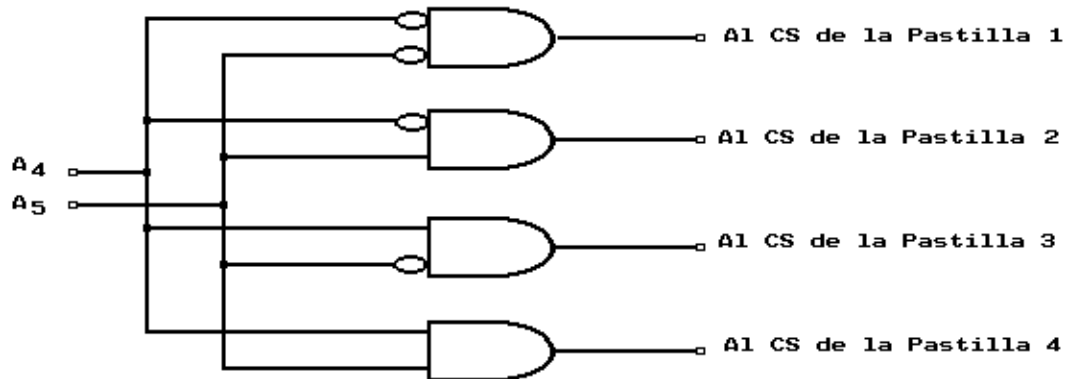
Los bits de selección "A0", "A1", "A2" seleccionan la posición de una palabra particular en la pastilla seleccionada. Los bits de entrada de datos y la entrada WE se aplican en común a ambas pastillas.



PASTILLAS DE MEMORIA EN PARALELO PARA INCREMENTAR EL NUMERO DE PALABRAS, DEJANDO FIJO EL NUMERO DE BITS POR PALABRA

Figura 12

Un problema especial, que no aparecía cuando las pastillas se colocaban en paralelo para incrementar el número de bits por palabra como en la figura 11, surge en este caso que estamos considerando. Ahora una palabra de salida se lee a veces de una pastilla o a veces de otra. Presumiblemente la palabra se transmitirá al mismo destino independientemente de la pastilla que la origine. Por lo tanto, las salidas de las pastillas tendrán que aplicarse a un bus común, ésta es la razón por la que aparece dicho bus en la figura 12.



LOS BITS A4 Y A5 GOBIERNAN UN DECODIFICADOR CUYAS SALIDAS SELECCIONAN UNA DE LAS 4 PASTILLAS DE MEMORIA EN PARALELO

Figura 13

El esquema paralelo de la figura 12, puede, por supuesto, extenderse a más pastillas. Supongamos, por ejemplo, que quisiéramos utilizar este esquema para colocar en paralelo cuatro pastillas con el fin de obtener una memoria de 32 palabras de 4 bits/palabra. Podemos introducir un bit de dirección "A5". Como antes, podríamos aplicar tres bits de dirección "A0", "A1", "A2" a los terminales de entrada de dirección de las pastillas. Los bits de dirección "A4" y "A5" se aplicarían a un decodificador como se indica en la figura 13. Las salidas del decodificador se utilizan entonces para activar las entradas de selección de pastillas.

MEMORIAS PASIVAS

El par metro permanencia de la información, indicado en la figura 8, se mide en forma cualitativa por la diferencia entre el tiempo de lectura y el tiempo de escritura. De acuerdo con este par metro, las memorias de acceso aleatorio se clasifican en memoria de escritura/lectura (R/W) o activas. Se denominan memorias pasivas a aquellas en las que existe gran diferencia entre el tiempo de lectura y el de escritura, el cual es mucho mayor.

A su vez este tipo de memorias se pueden clasificar tal como se indica a continuación:

1 - Memorias totalmente pasivas (ROM = Read Only Memories): son aquellas memorias que nunca pueden ser escritas por el que las utiliza, el cual solamente puede realizar la operación de lectura. La grabación ó escritura la realiza el fabricante. Por lo tanto, para quien la utiliza, el tiempo de escritura se puede considerar infinito y el tiempo de lectura tiene un cierto valor que depende de la tecnología con la cual ha sido fabricada la memoria.

2 - Memorias pasivas programables (PROM = Programmable Read Only Memories): Son aquellas cuyo contenido pueden ser escrito por el usuario una sola vez.

3 - Memorias pasivas reprogramables, ó memorias pasivas programables borrables (EPROM = Erasable Programmable Read Only Memories): Son aquellas cuyo contenido pueden, por un procedimiento especial, ser borradas y programadas por el usuario varias veces. La escritura se realiza por un procedimiento especial, el tiempo necesario es varias veces superior al de lectura.

Estas memorias se denominan apropiadamente de solo lectura, ya que las operaciones de borrado y re-escritura no pueden realizarse mientras la memoria no esté trabajando en un sistema digital. La memoria debe extraerse del sistema y su modificación puede llevar horas.

Como la memoria de lectura/escritura, la de sólo lectura también son de acceso aleatorio. Por consiguiente es bastante inconsistente utilizar la etiqueta "memoria de acceso aleatorio" (RAM) para referirse únicamente a las memorias de lectura/escritura y no a las de sólo lectura. Mucha gente a indicado esa inconsistencia, no obstante, todavía se permite esta terminología.

En una ROM el bit almacenado en cualquier posición, no tiene que cambiarse. Por consiguiente, no es necesario almacenar los bits en flip-flop y por ello una ROM puede ensamblarse completamente a partir de circuitos combinacionales. Realmente una ROM no es más que un conversor de código, y por lo tanto, consta de un decodificador y de un codificador.

MEMORIAS DE ACCESO SERIE

Son aquellas en las que el tiempo que tarda en leerse o grabarse una posición depende de la situación física en el interior de la memoria. Para introducir información en una posición hay que hacerla pasar por todas las que la preceden hasta alcanzar los terminales de entrada.

De forma similar, para leer el contenido de una posición hay que hacer pasar la información por todas las que la siguen hasta alcanzar los terminales de salida. Dentro de este tipo de memorias, y de acuerdo a la forma de entrada-salida, las mismas pueden clasificarse en:

1 - Registros de desplazamiento: Como ya hemos visto, en estas memorias de acceso serie, cada orden externa de escritura o lectura desplaza la información una posición en el interior de la memoria. La orden externa de desplazamiento está constituida por los impulsos de un generador. Los registros de desplazamiento se clasifican en dos clases, de acuerdo con las características de frecuencia de los impulsos de desplazamiento:

a - Registros de desplazamiento estáticos: Son aquellos en los que los impulsos de desplazamiento pueden anularse por tiempo indefinido. la información queda almacenada en las diferentes posiciones de la memoria a las que fue llevada por el último impulso de desplazamiento aplicado.

b - Registros de desplazamiento dinámicos: Son aquellos en las que los impulsos de desplazamiento no pueden anularse por tiempo indefinido porque de esta manera desaparece la información contenida en su interior.

2 - Memorias FIFO (First Input First Output): Son memorias de acceso serie en las que la primera información que entra es la primera en salir. En la figura 14 se representa gráficamente la forma en que se mueve la información en el interior de la memoria FIFO al realizar sucesivas operaciones de escritura lectura.

Se puede deducir que la memoria FIFO se comporta igual que un registro de desplazamiento en la operación de lectura, pero no así en la operación de escritura, en la que la información se desplaza desde la entrada hasta la posición vacía más cercana a la salida.

3 - Memorias LIFO (Last Input First Output): Son memorias de acceso serie en las que la información que entra en la última operación de escritura es la que primero sale en la operación de lectura que se realice. En la figura 15 se indica gráficamente la forma en que se realiza la escritura y la lectura de una memoria LIFO.

El alumno puede deducir que en una operación de escritura, toda la información contenida en la memoria se desplaza una posición hacia la derecha y en una operación de lectura se desplaza una posición hacia la izquierda.

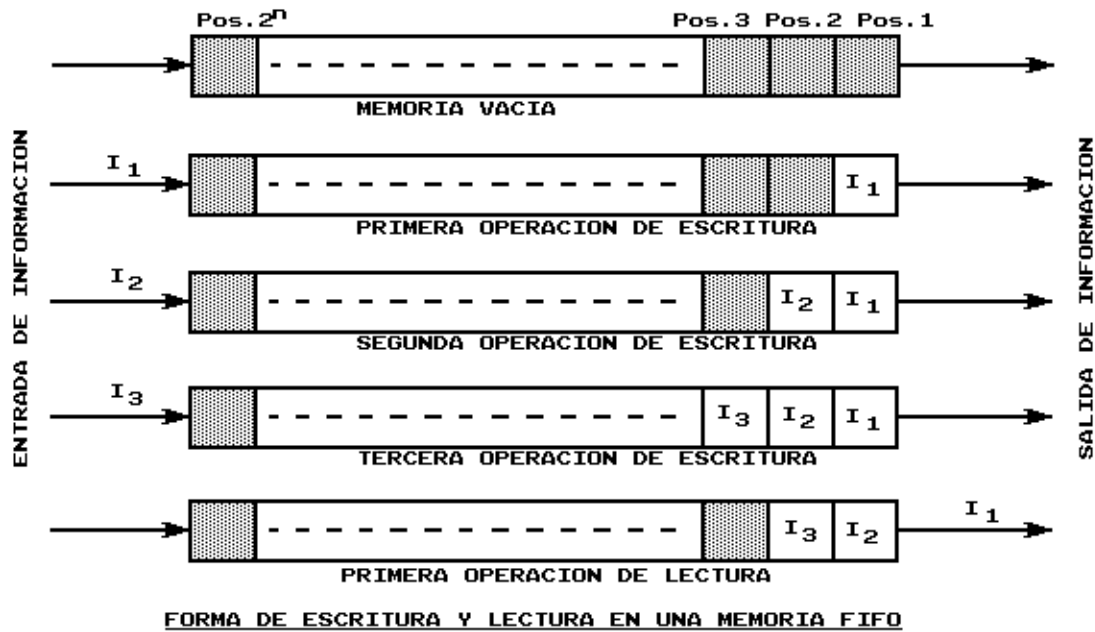


Figura 14

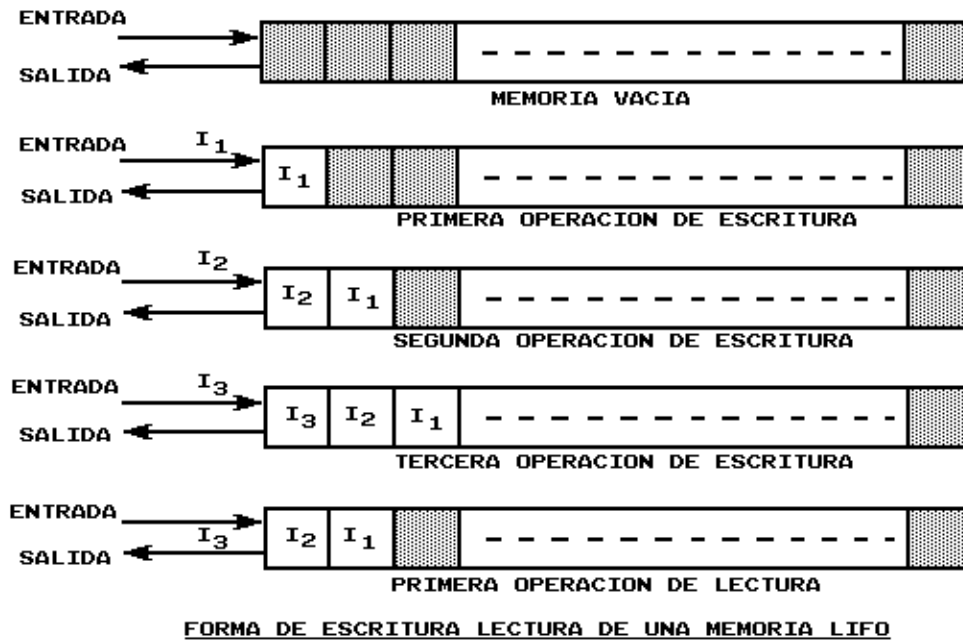


Figura 15

MEMORIAS ASOCIATIVAS

Estas memorias se caracterizan porque la búsqueda de información en la operación de lectura no se realiza indicando una dirección y observando su contenido, sino que se suministra la información a la memoria para observar si ésta la contiene en alguna de sus posiciones.

La información almacenada en cada posición de una memoria asociativa puede estar organizada de dos formas fundamentales:

- a - **Dividida en dos partes ó campos:** Un campo es el de la información que se compara con la que se presenta a las entradas de la memoria en una operación

Universidad Tecnológica Nacional - Facultad Regional Córdoba

ARQUITECTURA DE LOS COMPUTADORES

de lectura. El otro campo es el que da como respuesta la memoria cuando existe coincidencia entre el campo de comparación y la información de entrada.

b - Toda la información contenida en cada posición de la memoria se compara con la información de entrada: La memoria da solamente como respuesta en general un bit cuyo estado indica si existe alguna información en el interior de la memoria que es idéntica a la presentada a la entrada.

La estructura asociativa de una unidad de memorias es adecuada para el almacenamiento de ciertos tipos de organizaciones de datos. Por ejemplo, una tabla de clientes con su dirección como dato asociado, se debe organizar en forma asociativa para que al presentar a la memoria el nombre de un cliente, ésta de a la salida su dirección.

MODOS DE DIRECCIONAMIENTO

Hemos visto que generalmente (aunque no necesariamente) una instrucción consta de una parte de operación y otra de dirección. La parte de dirección puede contener la dirección de un operando utilizado en la ejecución de la instrucción. En otras ocasiones la parte de dirección de la instrucción puede contener, no la dirección del operando, sino la dirección donde se encuentra la dirección del operando. en el primer caso la dirección se describe como directa, mientras que en el segundo caso es una dirección indirecta. Esto es lo que se conoce como "Modo de Direccionamiento".

En computadoras, y microcomputadoras se emplea una amplia gama de modos de direccionamiento, por ello vamos a pasar revista a las diferentes técnicas de direccionamiento. Estas equivalen generalmente a una transformación de la parte de dirección de la instrucción en la dirección que se transferirá finalmente al Registro de Selección de la memoria para obtener la información deseada. Llamaremos a esta última "Dirección Efectiva". El tipo de procesamiento que debe sufrir el contenido de la zona de dirección viene especificado, ya sea por el Código de Operación, cuando este impone un tipo determinado, o por la configuración binaria de una parte de la instrucción, que contiene lo que convenimos en llamar las "Condiciones de Direccionamiento".

No todos los procesadores o microprocesadores tienen todos los tipos de direccionamiento, en algunos pueden existir modos que en otros no existen y viceversa.

Debido a estas distintas formas de direccionar los registros de un computador, la instrucción constar de tres partes:

CODIGO DE OPERACIÓN	CONDICIONES DE DIRECCIONAMIENTO	ZONA DE DIRECCION DE MEMORIA
---------------------	---------------------------------	------------------------------

Veremos a continuación los modos de direccionamiento más comúnmente utilizados en los procesadores actuales:

MODO DE DIRECCIONAMIENTO DIRECTO

En el direccionamiento directo, como ya lo señalamos, la instrucción contiene la dirección de la posición de memoria donde se encuentra el operando.

Este direccionamiento exige un ciclo de memoria para obtener el operando. La figura 16 muestra esta situación.

MODO DE DIRECCIONAMIENTO INDIRECTO

En el direccionamiento indirecto, como también dijimos, la instrucción no contiene la dirección de la posición de memoria donde se encuentra el operando, sino la dirección de la posición de memoria donde se encuentra la dirección del operando.

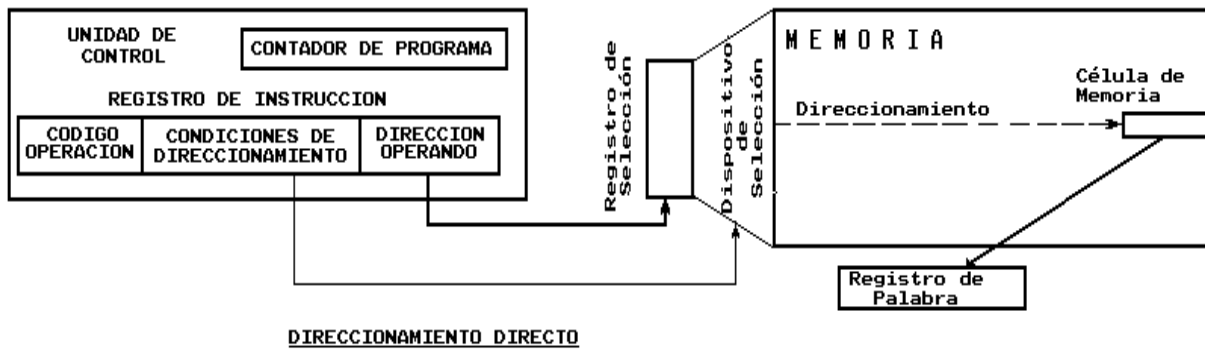


Figura 16

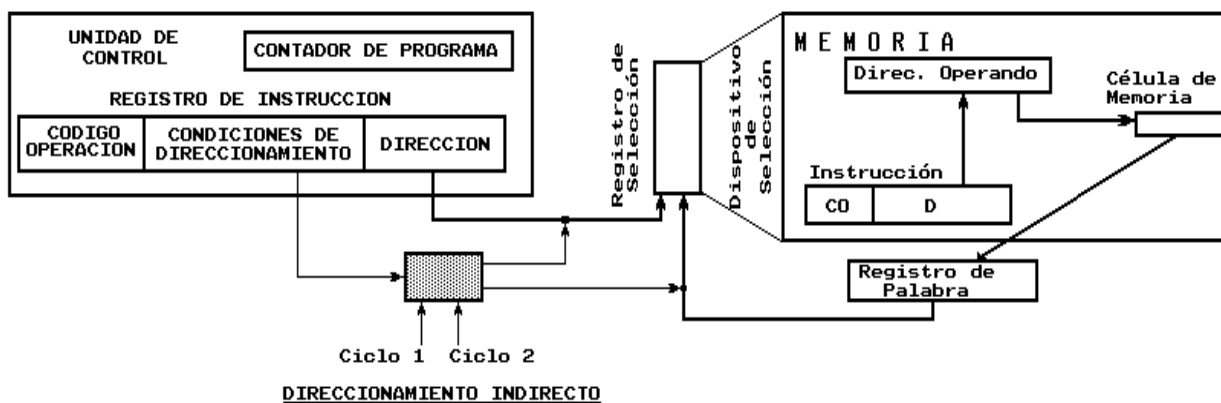


Figura 17

Por lo tanto, la localización de un operando direccionado indirectamente exigirá dos ciclos de memoria: un ciclo para buscar la dirección efectiva, otro ciclo para buscar el operando, de acuerdo a la figura 17.

MODO DE DIRECCIONAMIENTO RELATIVO

La dirección relativa no indica la posición de la información en la memoria en valor absoluto, sino que la sitúa con relación a una dirección de referencia. Esta a su vez, está almacenada en un registro, frecuentemente llamado "Registro de Traslación". La dirección efectiva se obtiene sumando la dirección relativa con la dirección de referencia.

Las técnicas de direccionamiento relativo se emplean especialmente para permitir direccionar a toda la memoria. Es así como un calculador cuya palabra es de 16 bits, si se tomasen 8 bits conjuntamente para el Código de Operación y las Condiciones de Direccionamiento, no quedarían más que 8 bits para la dirección. Esto permitiría direccionar "28 = 256" posiciones de memoria, cuando lo normal es que los ordenadores de este tipo poseen muchas posiciones más. Si no queremos pasar a un formato de instrucción de dos palabras, puede emplearse el direccionamiento relativo para alcanzar directamente determinadas zonas de la memoria. Veamos tres ejemplos:

1 - Direccionamiento por Base y Desplazamiento

En la máquina existe un registro llamado "Registro Base", que contiene la dirección de referencia (primera dirección de un programa o de una zona de datos). A la información que alberga la parte de dirección de la instrucción se le llama "Desplazamiento".

La dirección efectiva es la suma de la "Base" y el "Desplazamiento", como se muestra en la figura 18.

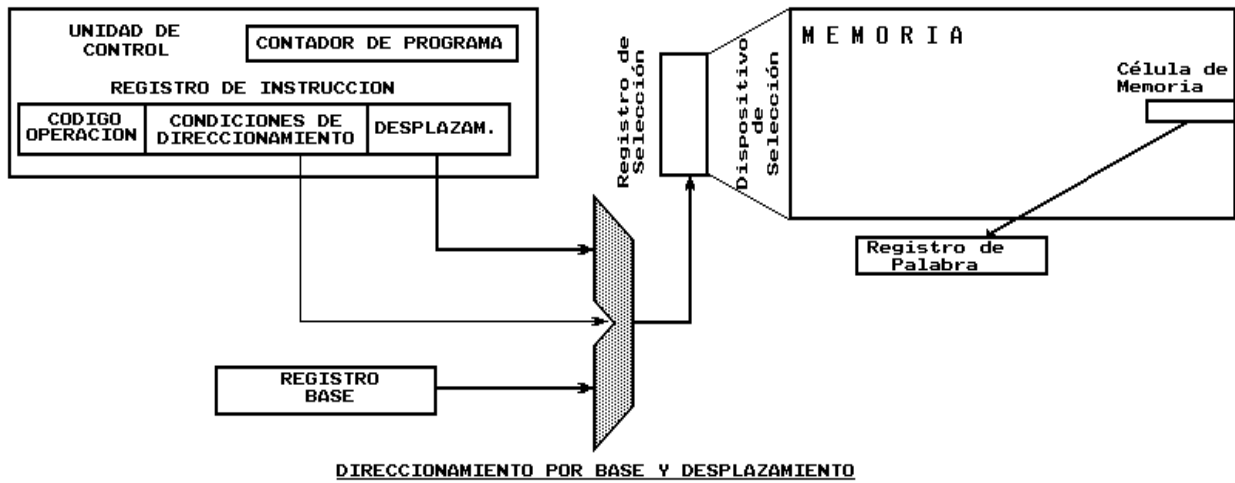


Figura 18

Algunos calculadores admiten varios registros de base. La instrucción debe especificar entonces la dirección del Registro Base escogido

2 - Direccionamiento por Referencia al Programa

El contenido del contador de programa sirve de dirección de referencia. Con este sistema es posible generalmente direccionar dos zonas de memoria a un lado y a otro de la instrucción en curso, según que la parte de dirección de la instrucción se suma o se resta con el contenido del Contador de Programa. La figura 19 muestra este modo de direccionamiento.

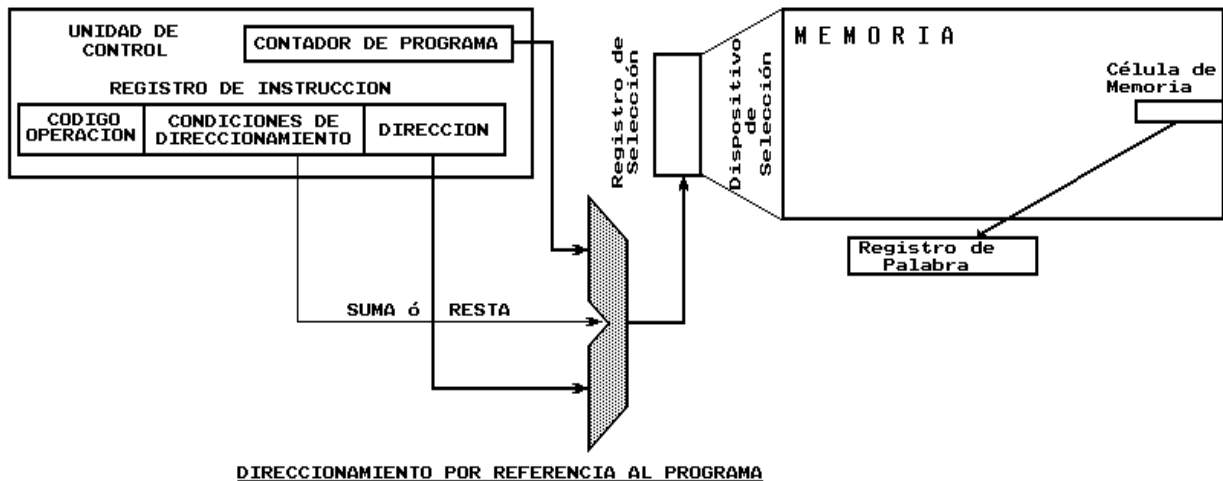


Figura 19

3 - Direccionamiento por Página (ó por Yuxtaposición)

Se considera a la memoria dividida en zonas de "2n" palabras llamadas "Páginas". En general, la parte de dirección de la instrucción contiene "n" bits, por lo que no capacita a la máquina para direccionar más palabras que las que contiene la página (ó dirección de página), y de la parte de dirección de la instrucción, que suministra la dirección dentro de la página.

Este modo se muestra en la figura 20.

ARQUITECTURA DE LOS COMPUTADORES

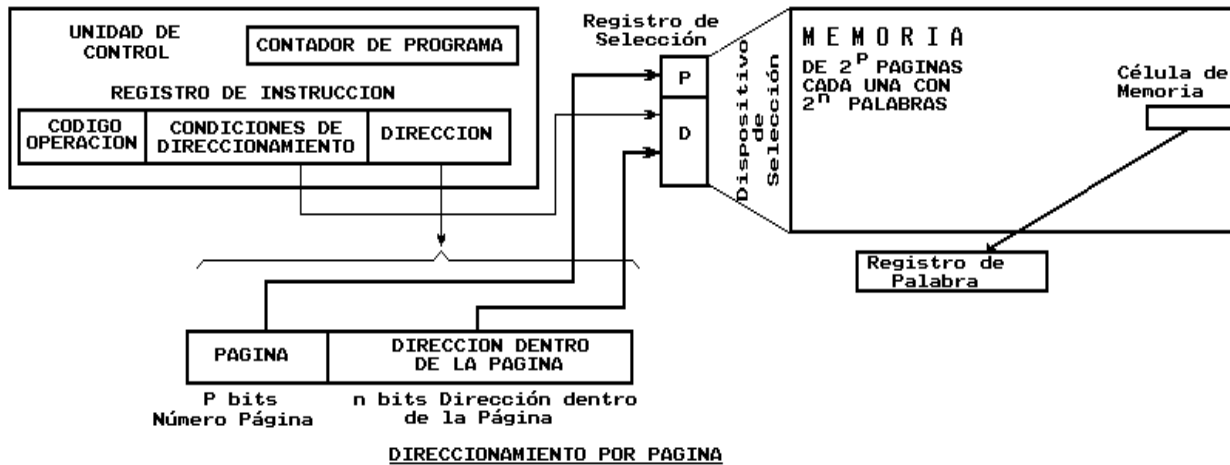


Figura 20

Las condiciones de direccionamiento, en la mayoría de los pequeños ordenadores organizados por páginas, poseen un bit de direccionamiento que, según su valor, implica el direccionamiento absoluto, es decir dentro de la página cero, o el direccionamiento en la página de la instrucción en curso por yuxtaposición de los bits de mayor peso del contador de programa y de la dirección dentro de la página.

Si la compuerta (no dibujada en la figura) está cerrada, hay direccionamiento absoluto en la página cero, si está abierta hay direccionamiento en la página de la instrucción. En este caso el direccionamiento por yuxtaposición puede ser considerado como direccionamiento por referencia al principio de la página en curso.

4 - Complementos acerca del direccionamiento relativo

Al menos uno de estos direccionamiento relativos se encuentra en las máquinas de palabra corta (12, 16 ó 18 bits) donde la zona reservada a la dirección no basta para direccionar a toda la memoria, a no ser que se utilicen instrucciones sobre dos palabras.

Permiten direccionar directamente zonas de memoria (además de las "2n" primeras palabras) sin necesidad de recurrir al direccionamiento indirecto. Las referencias fuera de esta zona hacen necesario un direccionamiento indirecto apoyado en una célula de memoria perteneciente a estas zonas directamente accesibles.

El direccionamiento por base y desplazamiento, lo mismo que un tipo más evolucionado de direccionamiento por página, se utiliza en grandes ordenadores para resolver problemas de asignación de memoria en multiprogramación. En el caso de las PC's se usa un sistema parecido para poder direccionar más memoria.

MODO DE DIRECCIONAMIENTO INDEXADO

En el direccionamiento indexado, como en el relativo, la parte de dirección de la instrucción contiene un número "N" que puede ser positivo o negativo. Sin embargo, para utilizar el direccionamiento indexado, el computador debe estar equipado con un registro especial (distinto del Contador de Programa) empleado para permitir direccionamiento indexado, y denominado "Registro Índice" (I). La posición de memoria donde se localiza el operando se encuentra mediante la suma "N + I".

La dirección efectiva se obtiene sumando a la parte de dirección de la Instrucción el contenido de un registro de la Unidad Central llamado "Registro Índice", llamándose a menudo a su contenido "Índice".

Este registro admite un cierto número de operaciones, como carga, lectura, incremento o disminución en uno, comparación. El programador lo utiliza para tratar, mediante una sola instrucción en un bucle de programa, datos almacenados

ARQUITECTURA DE LOS COMPUTADORES

vectorialmente en células sucesivas (tablas) de la memoria. El direccionamiento correspondiente es indexado, lo que quiere decir que la dirección especificada en la instrucción es la de la primera célula del vector (tabla) y a ella se suma el valor del índice, inicialmente puesto a cero e incrementando en uno cada vez que se ejecute la instrucción de fin de bucle. Esta última compara el índice con el número de elementos del vector y origina un salto al principio del bucle mientras quede algún elemento a procesar.

La figura 21 ilustra el concepto de indexación, con la existencia de varios registros índice.

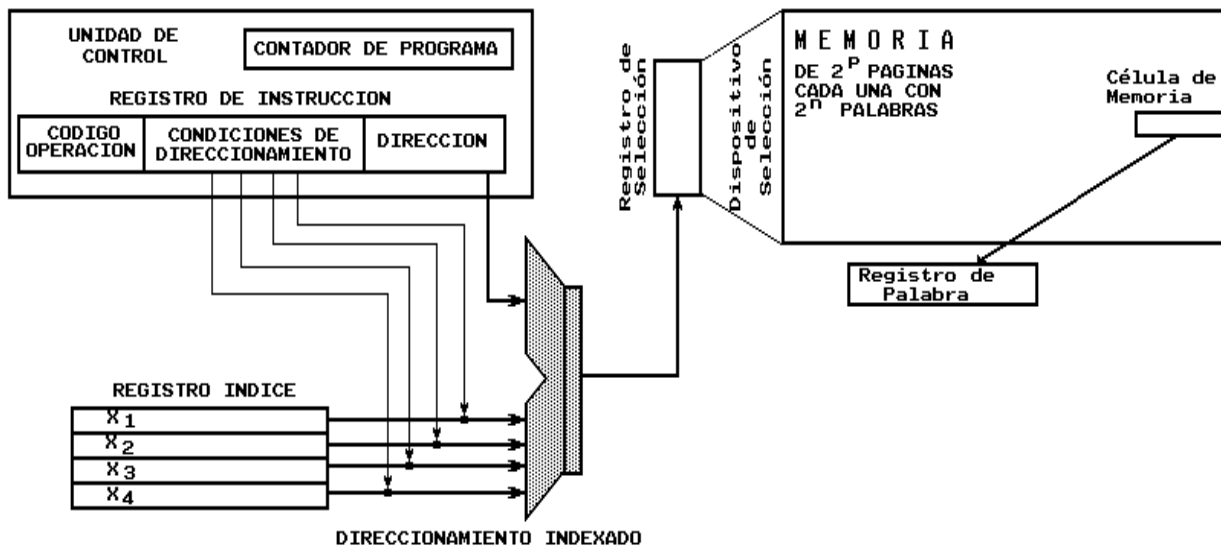


Figura 21

En algunos ordenadores, el Índice se inicializa en $-(n-1)$, donde "n" representa el número de los elementos del vector. La dirección especificada en la instrucción es la última del vector. El índice se incrementa en uno a cada pasada y se sale del bucle cuando el índice es cero.

Otras máquinas poseen dos registros índice, uno contiene el índice y el otro el valor máximo de este índice, ambos resultan comparador al momento de la instrucción a fin de bucle y se origina un salto al principio del mismo mientras no se produzcan coincidencias entre los dos valores.

En algunas máquinas también existen registros de autoíndice incrementables en 1 cada vez que son referenciados. En muchos ordenadores pequeños se implantan los índices en posiciones reservadas de la memoria central, con objeto de economizar el costo de los correspondientes registros.

MODO DE DIRECCIONAMIENTO POR REGISTRO INDIRECTO

Algunas computadoras que incorporan la facultad del direccionamiento de registro indirecto, tienen un registro especial, a menudo llamado "Puntero" (P) (Pointer Register). Este registro "P" contiene la dirección del operando, es decir, apunta a la posición de memoria del operando. Una instrucción que invoque realmente direccionamiento de registro indirecto no tiene bits significativos en su parte de dirección. En su lugar, la instrucción completa se incluye en los bits asignados a la parte de operación de la instrucción. Una instrucción típica que utiliza registro de direccionamiento indirecto debería especificar: "Cargar el acumulador con el operando localizado en la dirección de memoria dada en el registro P".

OTROS MODOS DE DIRECCIONAMIENTO

Otros esquemas comunes para localizar fuentes de operandos o destinos de operandos, denominados "modos de direccionamiento", incluyen los siguientes:

1 - Direccionamiento Inmediato

No es propiamente un direccionamiento, puesto que la parte de dirección de la instrucción no contiene la dirección del operando, sino el operando mismo. Así, la instrucción sería: "Cargar al acumulador directo con 37", significa cargar al acumulador con el número 37. La figura 22 muestra este tipo de direccionamiento.

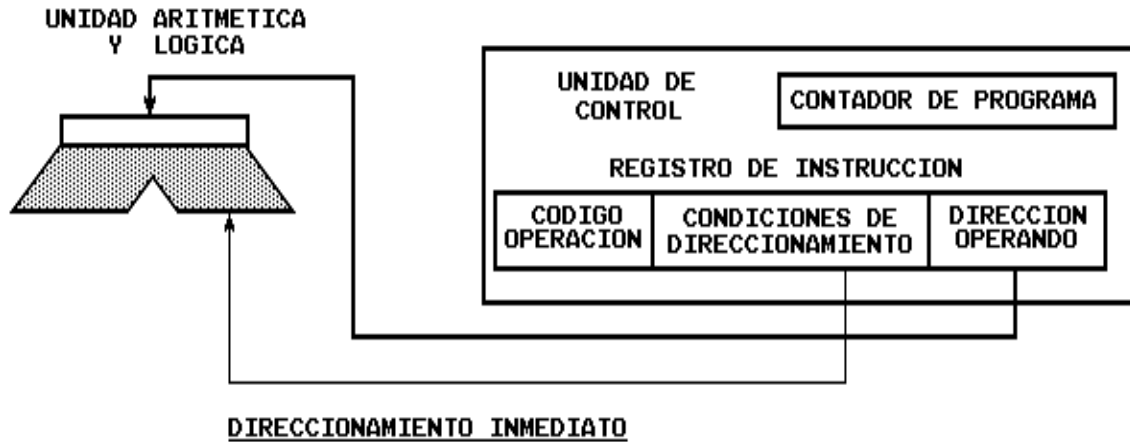


Figura 22

2 - Direccionamiento Inherente

Ordinariamente una dirección que es parte de una instrucción se refiere a una posición de memoria. Cuando una instrucción indica una fuente o un destino de algunos datos y no se direcciona específicamente, ya que no se hace referencia a la posición de memoria, se dice que la instrucción tiene una dirección "inherente". Por ejemplo, en la instrucción "Borrar el acumulador", los "datos" movidos están en una palabra cuyos bits son todos 0 (ceros) y la dirección de destino de estos datos es el registro acumulador. De nuevo, en la instrucción "Mover el contenido del registro R1 al registro R2", R1 es la "dirección" donde se lee una palabra y R2 es la "dirección" donde escribe la palabra.

LA UNIDAD ARITMETICA Y LOGICA

Para pedir al ordenador una operación matemática, por ejemplo una suma, la instrucción debe facilitarle la siguiente información:

- 1 - La clase de operación a realizar, en este caso una suma. Este papel lo cumple el código de operación.
- 2 - La dirección de la célula de memoria que contiene el primer dato, o primer operando.
- 3 - La dirección de la célula de memoria que contiene el segundo dato, o segundo operando.
- 4 - La dirección de la célula de memoria donde debe almacenarse el resultado.

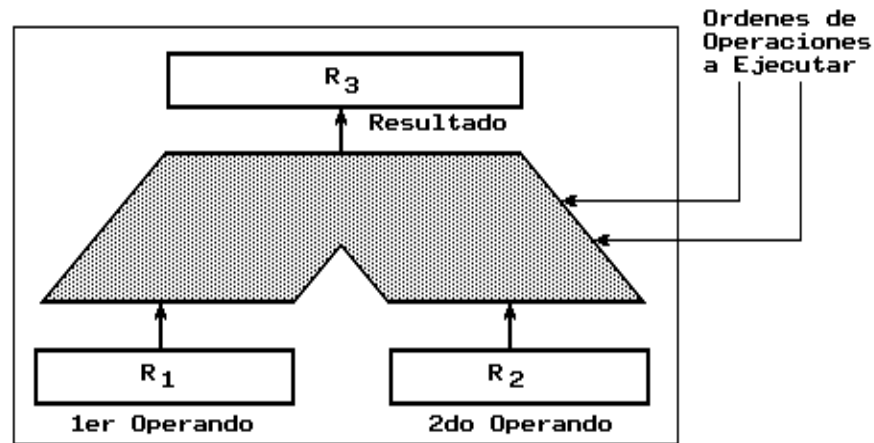
De aquí se deduce la forma de la instrucción, que contiene un código y tres direcciones:

CODIGO DE OPERACIÓN	DIRECCIÓN 1er OPERANDO	DIRECCIÓN 2do OPERANDO	DIRECCIÓN DEL RESULTADO
---------------------	------------------------	------------------------	-------------------------

Universidad Tecnológica Nacional - Facultad Regional Córdoba
ARQUITECTURA DE LOS COMPUTADORES

La figura 23 representa la Unidad Aritmética y Lógica capaz de ejecutar esta operación, la cual está rodeada de tres registros donde se memorizan los dos operandos y el resultado.

La instrucción de suma necesita cuatro accesos a la memoria central, que permiten sucesivamente buscar la instrucción, después el primer operando, luego el segundo operando y, por último, almacenar el resultado. A las máquinas que utilizan este tipo de instrucción se las llama de tres direcciones.



UNIDAD ARITMETICA Y LOGICA (Máquina de 3 Direcciones)

Figura 23

Existen otras máquinas llamadas de una dirección. Su Unidad Aritmética y Lógica posee un registro particular, denominado acumulador, que sirve tanto para albergar el primer operando como el resultado. Esta característica permite instrucciones de una sola dirección: la del segundo operando.

La operación de suma necesita tres instrucciones para:

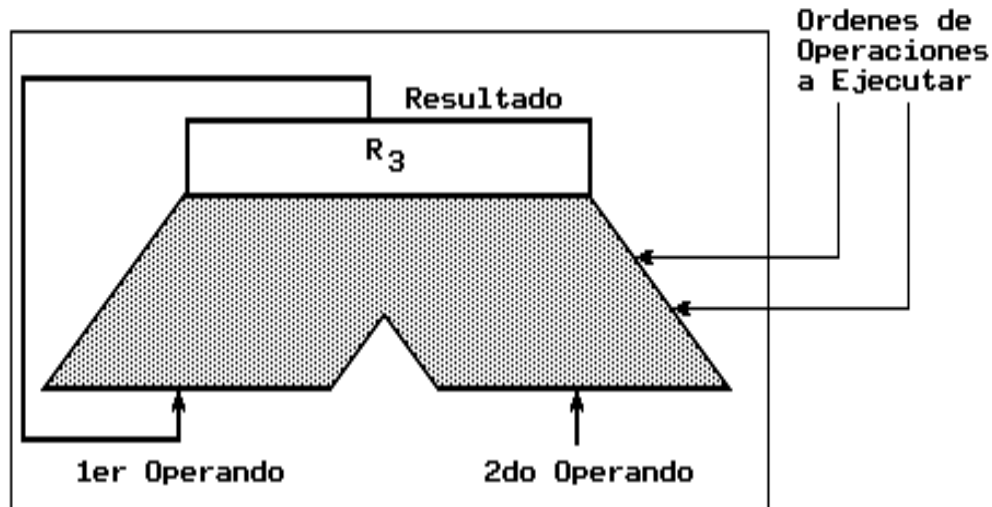
- 1 - Cargar el primer operando en el acumulador.
- 2 - Sumar el segundo operando con el contenido del acumulador.
- 3 - Almacenar en memoria el contenido del acumulador.

Cada una de estas tres instrucciones comportará un código de operación y una dirección.

CODIGO DE OPERACIÓN	DIRECCIÓN
CARGA	DIRECCIÓN 1er OPERANDO
ADICION	DIRECCIÓN 2do OPERANDO
ALMACENAMIENTO	DIRECCIÓN DEL RESULTADO

La Unidad Aritmética y Lógica está esquematizada en la figura 24, donde el acumulador sustituye a los registros R1, R2 de la figura anterior.

El segundo operando puede almacenarse durante la operación en el "Registro de Palabra" asociado a la memoria. Este es el caso de la computadora conocida como "Abacus".



UNIDAD ARITMETICA Y LOGICA (Máquina de 1 Dirección)

Figura 24

LA UNIDAD DE CONTROL

Es la unidad que extrae y analiza las instrucciones de la memoria central. Para ello necesita dos registros:

1 - Uno llamado "Contador de Instrucciones", "Contador Ordinal" ó "Contador de Programa" (PC = Program Counter), que contiene la dirección de la próxima instrucción a ejecutar. Su nombre se debe a que, exceptuado las rupturas de secuencia, este registro va aumentando su contenido en una unidad para pasar a la siguiente instrucción.

2 - Un registro llamado "Registro de Instrucción", que contiene la última instrucción extraída de la memoria. Si bien existen distintos diseños para este registro, vamos a ver el Registro de Instrucción de "Abacus", que es la arquitectura que estamos analizando y que es la que está más difundida. Este Registro de Instrucción se divide en dos partes:

a - Una parte que está conformada por el "Código de Operación", que define el tipo de instrucción a ejecutar (suma, multiplicación, salto, etc.).

b - Otra parte que contiene la/s "Dirección/es del/os operando/s".

La unidad de control contiene además un órgano generador de secuencias llamado "Secuenciador", el que tras analizar el código de operación, distribuye las órdenes al conjunto de unidades del ordenador, tales como la Memoria, Unidad Aritmética y Lógica, etc., a fin de hacerles ejecutar las distintas fases de la instrucción. Esto puede verse en la Figura 25.

El desarrollo de una instrucción de procesamiento en un computador de una dirección, puede descomponerse en tres fases:

- 1 - Búsqueda y análisis de la instrucción.
- 2 - Búsqueda y procesamiento del operando o bien, almacenamiento del operando.
- 3 - Preparación de la siguiente instrucción.

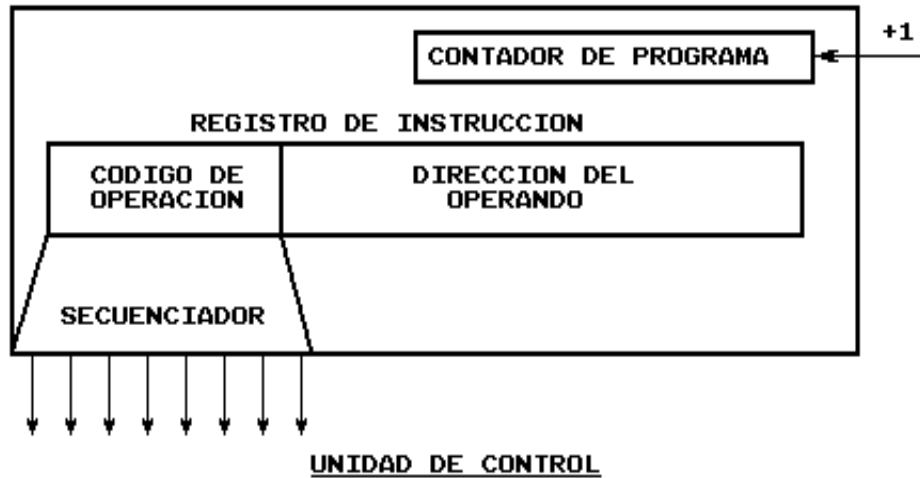


Figura 25

FASE DE BUSQUEDA DE LA INSTRUCCION

La Unidad de Control ordena la transferencia del contenido del Contador de Programa (es decir la dirección de la instrucción por ejecutar) al Registro de Selección, y envía a la memoria una orden de lectura. Una vez terminada esta operación de lectura, la instrucción queda disponible en el Registro de Palabra. Entonces, la Unidad de control ordena la transferencia del contenido de este registro al Registro de Instrucción. Los circuitos especializados de la Unidad de Control pueden entonces analizar el Código de Operación de la instrucción. Esta primera fase es común a todos los tipos de instrucciones. Tal como indica la Figura 26.

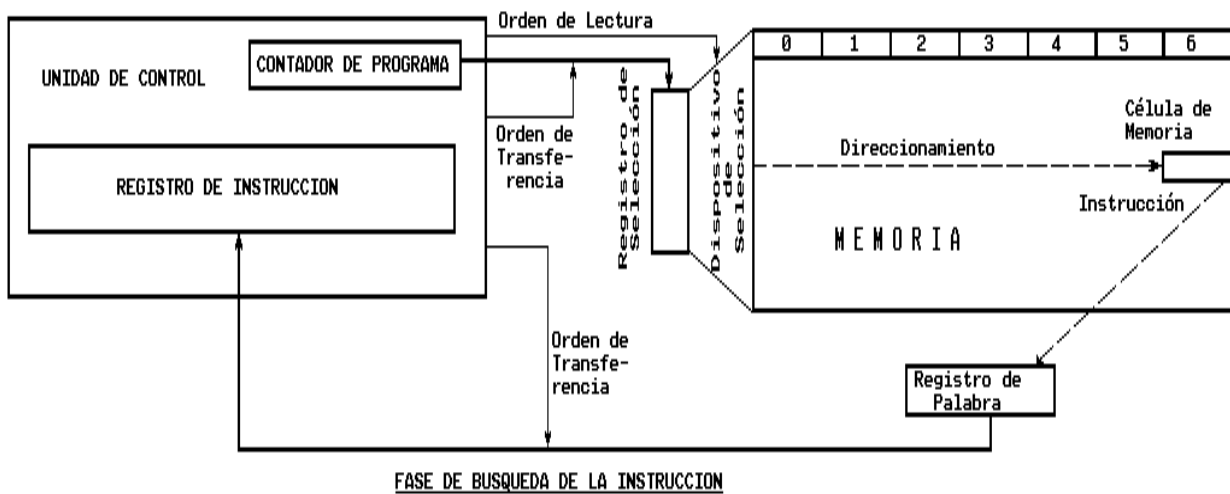


Figura 26

FASE DE BUSQUEDA O ALMACENAMIENTO DEL OPERANDO

Caso de búsqueda del operando, seguido de procesamiento:

Una vez analizado el código de operación de la instrucción, la unidad de control sabe que se trata de una instrucción de procesamiento, con búsqueda previa del operando. La dirección del operando se encuentra en la zona de dirección del Registro de Instrucción. La Unidad de Control ordena su transferencia al Re-

gistro de Selección de memoria y acto seguido ordena una operación de lectura en la memoria.

Al finalizar dicha operación, el operando buscado queda disponible en el Registro de Palabra. La Unidad de Control posiciona los circuitos de la Unidad Aritmética y Lógica para realizar el procesamiento solicitado por el código de operación y ordena la transferencia del operando a la Unidad Aritmética y lógica. El resultado del procesamiento del operando queda almacenado en el Acumulador. Obsérvese que un posible procesamiento puede ser simplemente una transferencia del operando al acumulador. Figura 27.

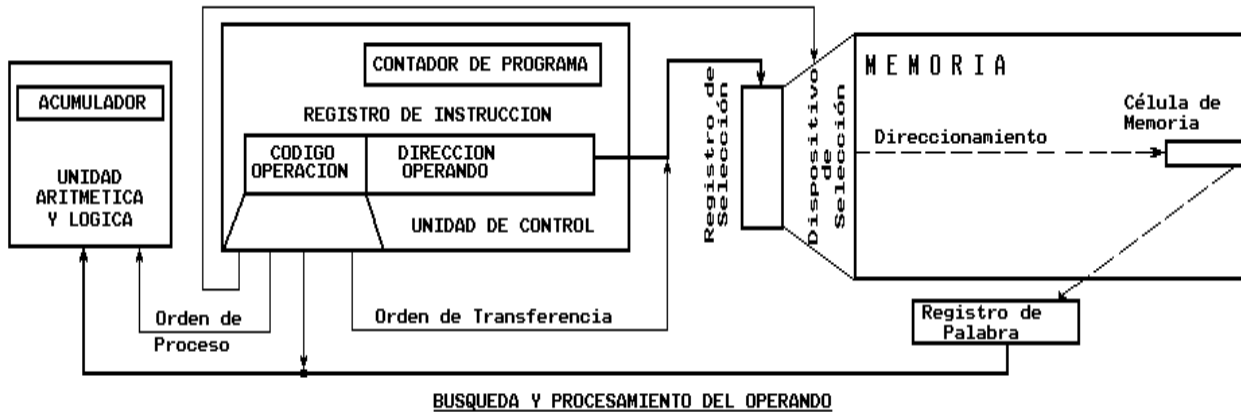


Figura 27

Caso de almacenamiento del operando:

La dirección de almacenamiento del operando se encuentra en la zona de dirección del Registro de Instrucción. La Unidad de Control ordena su transferencia al Registro de Selección de la memoria. El operando a almacenar está en el Acumulador. La Unidad de Control ordena su transferencia al Registro de Palabra. Acto seguido completa la ejecución ordenando a la memoria una operación de escritura. Figura 28.

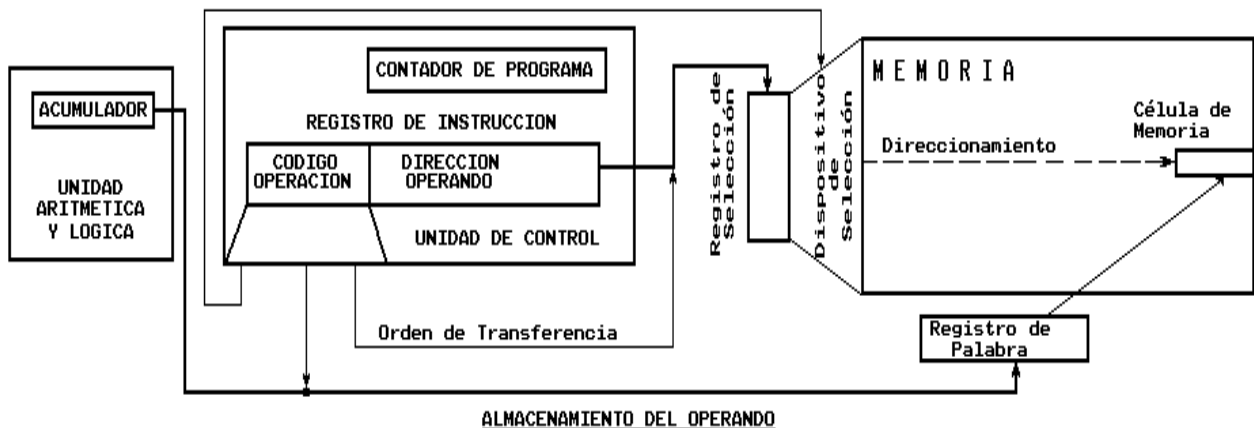


Figura 28

FASE DE PREPARACION DE LA SIGUIENTE INSTRUCCION

Consiste en aumentar una unidad (a menudo se dice incrementar en uno) el contenido del Contador de Programa, de manera que contenga la dirección de la siguiente instrucción.

INSTRUCCION DE RUPTURA DE SECUENCIA

Este tipo de instrucción, también llamado instrucción de bifurcado o de salto, permite modificar el desarrollo secuencial del programa, haciendo que la instrucción de salto sea seguida, no por la instrucción almacenada en la siguiente dirección de memoria, sino por una instrucción cuya dirección viene suministrada por la propia instrucción de salto.

El salto puede ser condicional, o sea que este no tendrá efecto si no se satisface una determinada condición, normalmente relacionada con el contenido del acumulador, en caso de no cumplirse esta condición, el programa continuar en secuencia.

El Código de Operación define la condición y la Dirección indica el emplazamiento de la próxima instrucción por ejecutar en caso de satisfacerse la condición. Esto puede verse en la figura 29.

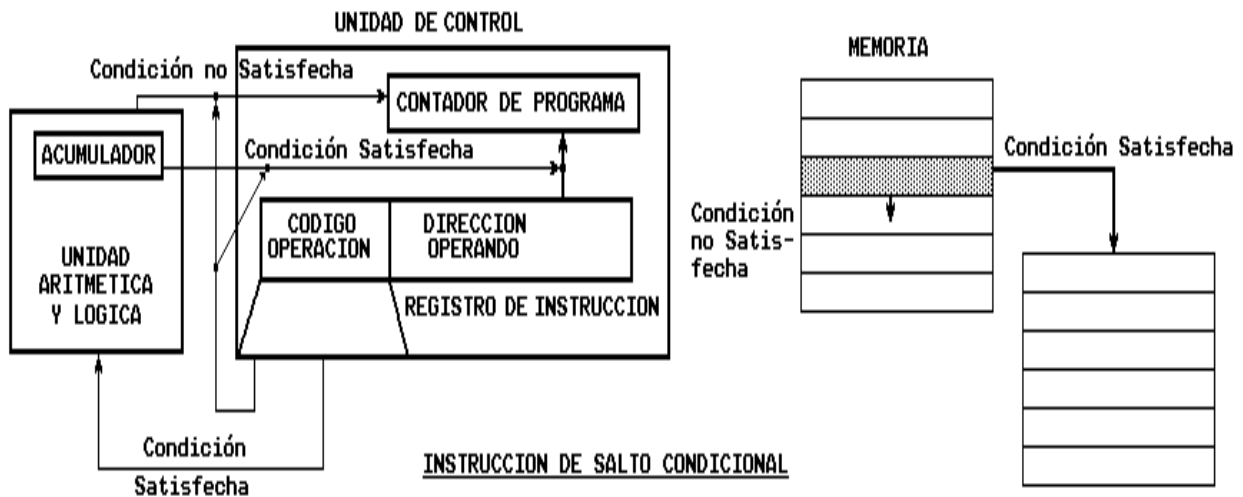


Figura 29

Si la respuesta de la Unidad Aritmética y Lógica es que la condición ha sido satisfecha, la Unidad de Control ordena la transferencia de la dirección al Contador de Programa e inhibe la suma de una unidad a este contador, en caso contrario ordena incrementar en uno al Contador de Programa.

EL CANAL

Es un dispositivo que es capaz, en general, de gestionar la transferencia, tanto en un sentido como en otro, de un conjunto de información entre las Unidades Periféricas y la Memoria Central.

En los ordenadores actuales esta transferencia puede operarse simultáneamente con el desarrollo de un programa de cálculo. Las informaciones transferidas se almacenan en memoria en forma secuencial.

Para inicializar la transferencia, instrucciones especiales de entrada/salida deben suministrar al canal lo siguiente:

- 1 - La dirección de la Unidad Periférica implicada.
- 2 - La dirección para el almacenamiento de la primera información.
- 3 - El número de informaciones a transferir.

En lo sucesivo el canal se ocupará totalmente de la gestión de la transferencia. Por cada información transferida, sumar uno a la dirección de almacenamiento y restar uno al número de informaciones a transferir. En el momento en que toda la información ha sido transferida, advertir a la Unidad de Control a través de un mecanismo llamado "Interrupción". Esto se aprecia en la Figura 30.

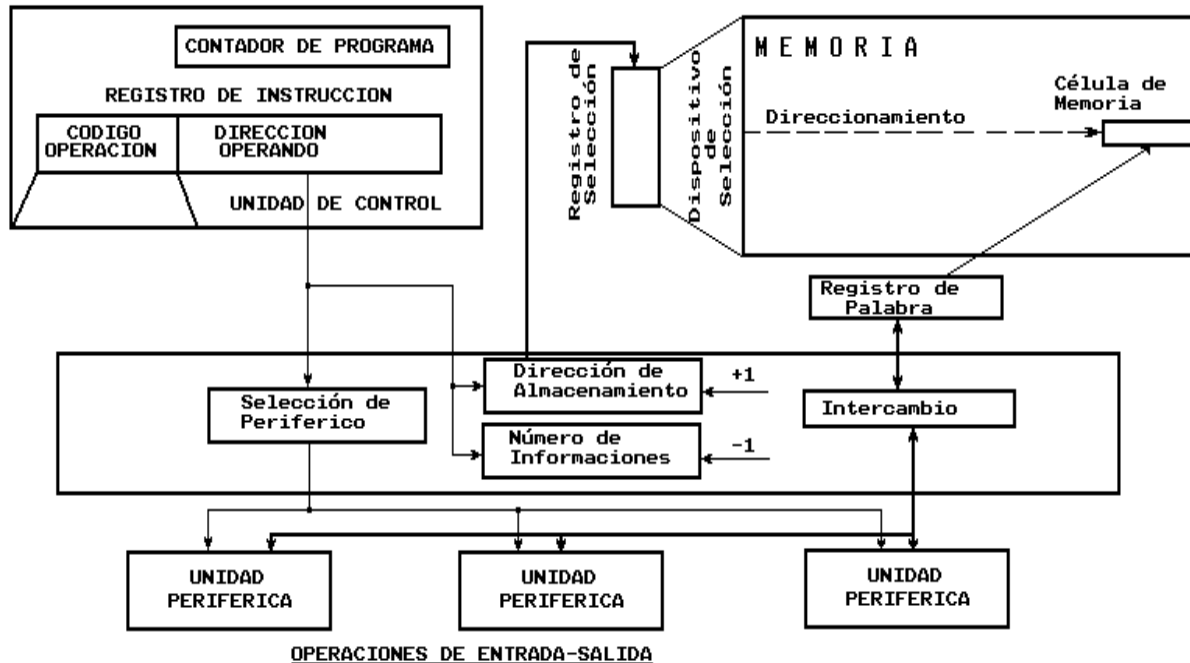


Figura 30

LAS UNIDADES PERIFERICAS

Se distinguen dos grandes clases de Unidades Periféricas:

1 - Las memorias auxiliares: que sirven de soporte de almacenamiento de gran capacidad y de medio de comunicación con el interior del sistema. Estas unidades están compuestas por los discos, disquetes, cintas magnéticas, etc.

2 - Las unidades de comunicación: que podemos a su vez dividir las en:

a - Comunicación entre el sistema y el medio exterior: que permiten el intercambio de información sobre soportes generados o leídos por el hombre, como pueden ser las máquinas lectoras o perforadoras de cintas o tarjetas perforadas, las impresoras, etc.

b - Unidades para el diálogo entre el hombre y el computador: tales como las máquinas de escribir, unidades de visualización, teclados y monitores de terminales, etc.

c - Las unidades para intercambio de información a distancia: como por ejemplo las líneas de comunicación, las redes de computadoras, las terminales, otros computadores, etc.

d - Las unidades para el intercambio de información a través de dispositivos que aceptan o suministran datos analógicos: tales como modems, conversores analógicos/digitales, etc.

La mayoría de las unidades constan de dos partes:

1 - Una parte electrónica, llamada "Controlador" o "Unidad de Control de Periférico".

2 - Una unidad electromecánica que, gobernada por la primera, lee o escribe datos en los dispositivos periféricos.

INTERRUPCIONES

Las "Interrupciones" son ordenes que proceden del exterior y que piden al ordenador ejecutar un programa asociado a la orden. El programa en curso se ve interrumpido para permitir la ejecución del programa solicitado por la interrupción, considerado ahora como prioritario. Terminado este último, se reanuda la ejecución del programa interrumpido.

Es gracias a las interrupciones, como se le comunica a la Unidad de Control que debe realizar o ejecutar una tarea, o bien para avisarle que algo a sucedido ó finalizado, como por ejemplo cuando el canal le avisa que todos los datos solicitados han sido transferidos a la memoria.

Más exactamente, el tratamiento de una interrupción consiste, en general, en memorizar el estado de la máquina desde el final de la instrucción en curso y forzar seguidamente en el Contador de Programa la dirección de la primera instrucción del programa asociado a la interrupción. Este programa, a su vez, termina en una instrucción que restaura el estado que tenía la máquina al momento de la interrupción, lo que permite la reanudación del programa interrumpido.

Existen dos tipos de interrupciones:

1 - Las "Enmascarables": o sea las que pueden ser temporariamente ignoradas, por cuanto no implican un procesamiento o atención inmediata. El ejemplo anterior del canal es una de las interrupciones enmascarables.

2 - Las "No Enmascarables": que no pueden ignorarse por cuanto deben ser atendidas inmediatamente. Entre estas encontramos las interrupciones por errores, tanto de Hardware como de Software, los errores lógicos (división por cero), etc. También tenemos las interrupciones que proceden por ordenes del operador, como por ejemplo la cancelación de un programa, etc.

Podemos clasificar las interrupciones según el campo de aplicación del procedimiento de interrupción como por ejemplo:

1 - Interrupción por errores o averías: falla de la alimentación eléctrica, error de paridad en memoria, etc.

2 - Interrupción por causa del programa (o desvío): instrucción o dirección incorrecta, operaciones imposibles, división por cero, desbordamiento de capacidad, etc.

3 - Interrupción por entrada/salida: generada por el canal para avisar del fin de una operación de entrada/salida ó de una anomalía ocurrida en el transcurso de una operación de entrada/salida.

4 - Interrupción externa: utilizada para avisar a la máquina acerca de cualquier modificación interesante del medio exterior, especialmente en control de procesos industriales.

5 - Interrupción de recuento: para contar impulsos procedentes de un reloj.

Determinados computadores presentan una sola posibilidad de interrupción, en cuyo caso se pasarían todas las causas de interrupción a través de una compuerta lógica "OR", y el programa de interrupción comienza a comprobar un conjunto de indicadores a fin de detectar cuál es la causa de la misma.

Otros poseen varios niveles de interrupciones. A cada nivel de interrupción le corresponde un conductor proveniente de la causa o causas de la interrupción y también de los programas asociados. Dichos niveles pueden estar jerarquizados, lo que significa que están clasificados por orden de las prioridades respectivas, y que un programa de interrupción puede a su vez ser interrumpido por una demanda de interrupción clasificada en un nivel superior de prioridad. El programa interrumpido pasa entonces al estado de espera.

Universidad Tecnológica Nacional - Facultad Regional Córdoba

ARQUITECTURA DE LOS COMPUTADORES

Los pequeños ordenadores, especializados en la guía de procesos, poseen frecuentemente sistemas de interrupción bastante elaborados. En general, presentan varias características del tipo siguiente de organización modular:

Las interrupciones están agrupadas en cierto número de niveles jerárquicos, o sea, cada uno de ellos posee una prioridad diferente. Un programa de un determinado nivel de prioridad puede verse interrumpido por un programa solicitado por una interrupción de nivel superior.

Un nivel agrupa varios subniveles, cada cual con su conductor de interrupción y su prioridad dentro del nivel. Los programas asociados a los subniveles de un mismo nivel no pueden interrumpirse mutuamente, siendo únicamente efectiva su prioridad en caso de elección entre varios de ellos, si se encontrasen simultáneamente en estado de espera.

Un subnivel agrupa, a su vez, varias demandas de interrupciones, cuyas causas son investigadas por prueba de los indicadores.

Una interrupción puede encontrarse en varios estados a saber:

1 - Estado desactivado: el nivel no acepta ninguna demanda de interrupción. un nivel puede ser activado o desactivado por el programa.

2 - Estado de espera: el nivel pasa al estado de espera si recibe una señal de demanda de interrupción. Varias razones pueden diferir la consideración operativa de la interrupción. Primeramente, es preferido que el nivel esté autorizado, puesto que se distinguen dos formas de estado de espera:

Estado de espera inhibido: el nivel ha resultado inhibido por programa, de tal manera que la interrupción ha podido ser memorizada pero no tomada en cuenta por el ordenador. También se suele decir que el nivel ha sido enmascarado.

Estado de espera autorizado: la interrupción puede ser tomada en cuenta por el ordenador, si se han satisfecho las siguientes condiciones:

a - no existe ningún nivel de prioridad superior en el estado autorizado de espera.

b - La unidad central se encuentra en una fase ininterrumpible (generalmente en final de instrucción).

3 - Estado activo: implica que la unidad central tome en cuenta la interrupción y se mantiene durante toda la ejecución del programa de interrupción.

A cada nivel de interrupción se asocian dos señales: una impulsiva, procedente del exterior, para pedir la interrupción, y otra de larga duración que el calculador deja activada hasta tanto no se haya tratado la interrupción, es decir, hasta que el nivel sea capaz de aceptar una nueva demanda de interrupción.

A cada nivel de interrupción se asocia un conjunto de posiciones de memoria, divididas en dos partes: la primera contiene toda la información necesaria para el arranque del programa de interrupción (en especial la dirección de la primera instrucción). La segunda sirve para almacenar la información que caracterizan al estado del programa en el instante de la interrupción, para su posterior reanudación (estado de los diferentes indicadores, Contador de Programa, eventualmente los registros de base, etc.).

El paso de una interrupción al estado activo consiste en memorizar el estado del programa, almacenando la información pertinente en las posiciones asociadas de memoria para instaurar después un nuevo estado del programa.

La última instrucción de un programa de interrupción es una muy especial, cuya finalidad es la de restaurar el estado del programa en el momento de la interrupción, rescatando las posiciones asociadas de memoria. Además desactiva el nivel de interrupción, lo que permite responder a otra posible señal de interrupción.