

UNIDAD 1. Funcionamiento y Arquitectura del Ordenador.

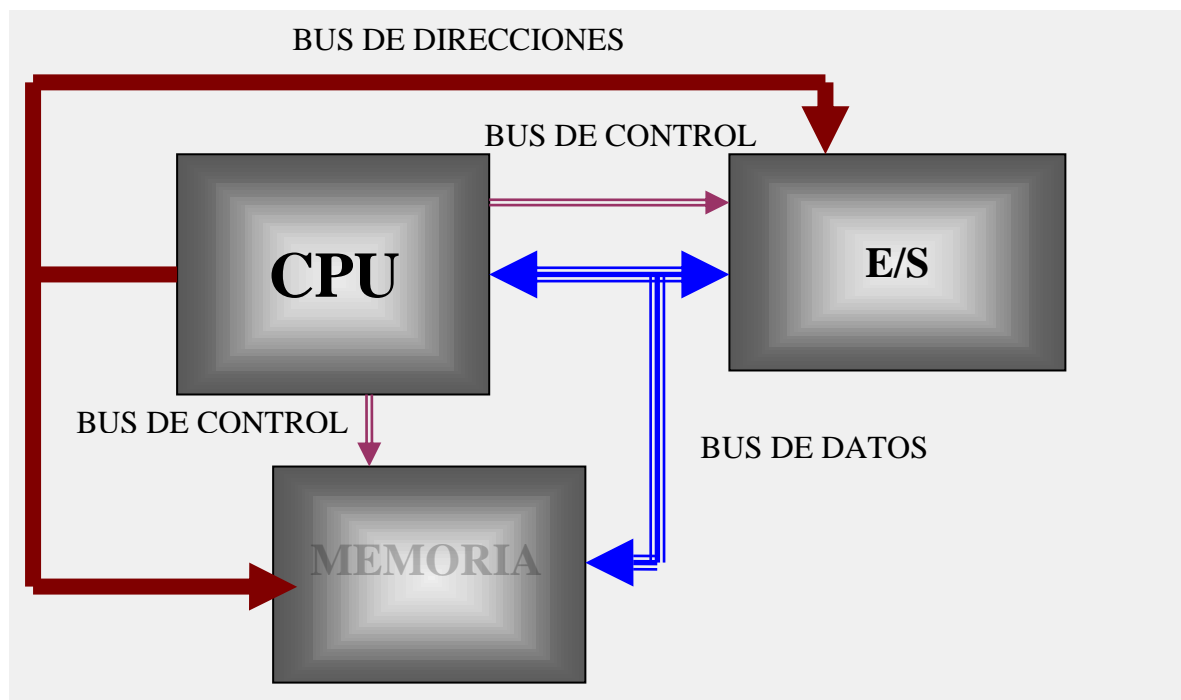
1.	COMPONENTES FÍSICOS.....	2
a.	<i>Unidad Central de Proceso</i>	3
b.	<i>La unidad aritmético-lógica</i>	6
c.	<i>Memoria</i>	7
d.	<i>La unidad de entrada/salida y los periféricos</i>	11
e.	<i>Buses de comunicación</i>	15
2.	SISTEMAS LÓGICOS	17
a.	<i>Formato de las instrucciones</i>	18
I.	Instrucción de tres operandos	19
II.	Instrucción de dos operandos	19
III.	Instrucción de un operando	20
IV.	Instrucción.....	20
b.	<i>Modos de direccionamiento</i>	21
I.	Direccionamiento inmediato	21
II.	Direccionamiento Directo	21
III.	Direccionamiento indirecto	21
IV.	Direccionamiento relativo	22
c.	<i>Fases de una instrucción</i>	23
I.	Fase de carga de una instrucción.....	23
II.	Ejecución de una instrucción.....	25

1. COMPONENTES FÍSICOS.

Los principales componentes físicos de un sistema informático son:

- la unidad central de proceso, CPU.
- la memoria,
- la unidad de entrada/salida
- y unos elementos que sirven de unión a todos los anteriores llamados buses.

Todos ellos se relacionan entre sí según el siguiente esquema conocido como arquitectura Von Newman:



a. Unidad Central de Proceso

Esta unidad suele abreviarse como UCP (unidad central de proceso), pero es muy frecuente el uso de la abreviatura inglesa CPU (central processing unit).

Físicamente está formado por circuitos de naturaleza electrónica que se encuentran integrados en una pastilla o chip denominada microprocesador.

La unidad central de proceso es un elemento muy importante en una computadora y se encarga de organizar todo el trabajo y realizar los cálculos aritméticos y lógicos; así como de supervisar todas las operaciones del sistema informático. A menudo se dice de ella que es como del cerebro que controla y gobierna todo el sistema.

¿Cómo consigue hacer todo esto?

Para ello extrae, una a una, las instrucciones del programa que se tiene alojado en otro de los componentes físicos denominado memoria central, las analiza y emite las órdenes necesarias para su completa realización.

La unidad central está conectada con la memoria y con los periféricos por medio de unos canales, a través de los cuales circula la información. Estos canales reciben el nombre de *buses*.

La información que circula entre la unidad central y la memoria es de dos tipos: las *instrucciones* y los *datos*.

- Las instrucciones son las órdenes que configuran el programa que la computadora debe ejecutar.
- Los datos son conjuntos de símbolos que representan un valor.

La unidad central está formada por dos partes claramente diferenciadas:

- la unidad de control
- y la unidad aritmeticológica.

Unidad de control

La unidad de control se encarga de gobernar el resto de unidades. Para gobernar necesita mandar mensajes y enviar los datos a procesar por unos canales especiales que también gobierna. Todas estas órdenes llamadas *señales de control* y los datos van canalizados por unas vías llamadas *buses*.

Para realizar su tarea la UC necesita conocer, por un lado, la instrucción y, por otro, una serie de informaciones adicionales que deberá tener en cuenta para coordinar, de forma correcta, la ejecución de la instrucción. El resultado de la interpretación de dichas informaciones son una serie de órdenes a los diferentes elementos de la computadora.

Como informaciones adicionales a las instrucciones podemos ver los impulsos de reloj y los indicadores de estado. Los impulsos de reloj van a sincronizar todas las actividades que se lleven a cabo. Los indicadores de estado son una serie de valores que se modifican según resultados de las operaciones anteriores guardando una memoria histórica de los acontecimientos precedentes para que, en función de dichos acontecimientos, pueda la UC tomar decisiones.

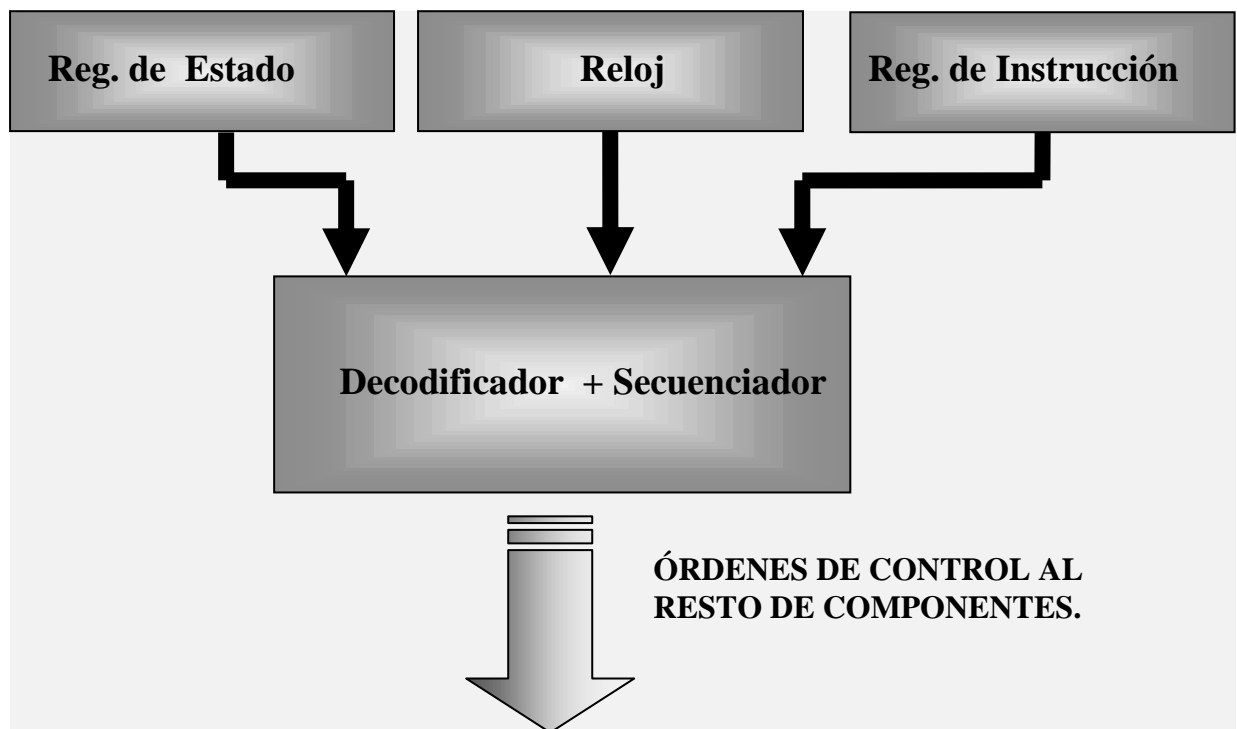
Los componentes principales de la UC son:

- Un elemento formado por 2 componentes que decodifica y secuencia las instrucciones. El primer elemento decodifica la instrucción para ver de que se trata, esto lo realiza un **decodificador**. Y una vez sabe de que se trata, el segundo irá ejecutando la instrucción paso a paso, es decir una instrucción no se ejecuta en un solo paso, sino que es una secuencia de pasos, necesitando cada paso para su realización las señales de control correspondientes. Así pues el **secuenciador** es el encargado de generar en cada paso las señales de control necesarias.
- Varios elementos de memoria denominados registros:
 - El registro que almacena la instrucción mientras el intérprete está traduciendo su significado: **Registro de Instrucción (RI)**. El resto de las instrucciones

permanecen en la memoria, esperando que les toque su turno de ejecución.

- El registro que guarda cuál es la dirección de la próxima instrucción, es decir, guarda la referencia de la siguiente instrucción que se debe realizar, para poder ir a buscarla una vez que finaliza la ejecución de la instrucción en curso; este registro se conoce con el nombre **Contador de Programa (CP)**.
- Los indicadores de estado están agrupados en un registro denominado **Registro de Estado (RE)**.

En la siguiente figura se esquematiza el conjunto de señales que utiliza la UC y las que genera.



b. La unidad aritmético-lógica

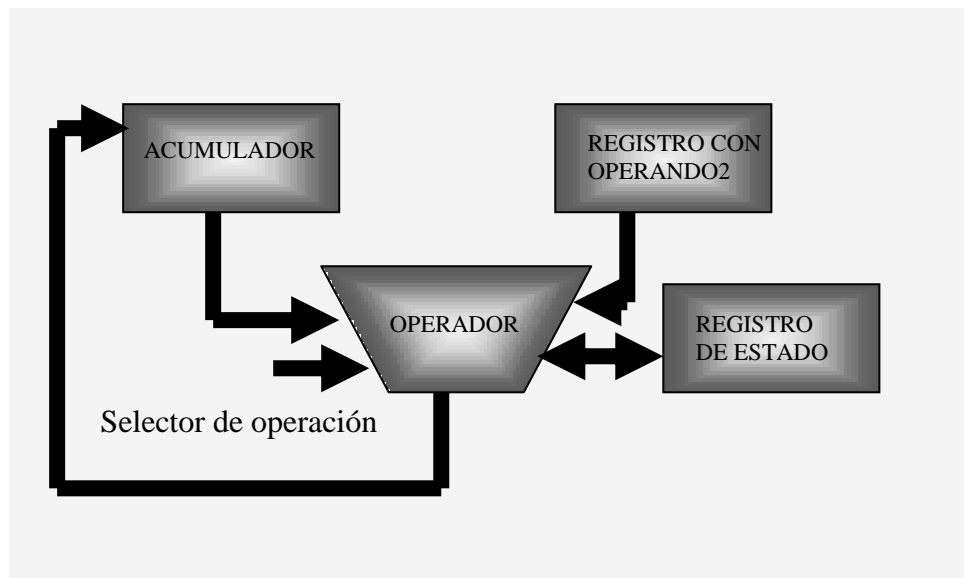
La unidad Aritmético – Lógica (UAL) es la encargada de realizar los cálculos; siempre bajo la supervisión de la unidad de control. Los datos sobre los que se realizan las operaciones se denominan **operandos**. Al elemento encargado de ejecutar las operaciones se le denomina **operador**, y esta formado por una serie de circuitos electrónicos que son capaces de sumar dos números binarios o hacer las operaciones lógicas elementales: disyunción, conjunción y negación.

Para que el operador realice la operación, los operandos se llevan a la UAL y se guardan en unos registros denominados registros de trabajo. El resultado de la operación se guarda también en un registro antes de ser llevado a la memoria o a la Unidad de Entradas y Salidas. Frecuentemente se utiliza un mismo registro para guardar uno de los operandos y, también, el resultado, a este registro se le conoce con el nombre de **Acumulador**.

El operador, además de calcular el valor de la operación, modifica el registro de estado según el resultado de la operación. Así, si el resultado es un valor negativo, se modifica un bit (cantidad mínima de información en binario. El binario es el lenguaje interno de los ordenadores. En binario solo hay dos valores posibles: 0 y 1) de dicho registro, llamado bit negativo o bit N, poniéndose a 1; por el contrario, el bit N permanecerá en estado 0 mientras el contenido del acumulador no sea negativo. De igual forma indicará la UAL a la UC si el resultado ha sido cero, o si ha producido algún acarreo, etc.



Unidad aritmético-lógica sin acumulador



Unidad aritmético-lógica con acumulador

c. Memoria

La memoria es la unidad donde se almacenan los **datos** y las **instrucciones** que constituyen los programas.

Hay dos tipos básicos de memoria: memoria **interna** o central y memoria **externa** o **masiva**. En este apartado vamos a ocuparnos de la memoria central o también conocida como memoria RAM o principal y dejaremos para más adelante la memoria externa (discos, cintas ...).

La memoria no es nada más que un conjunto ordenado de **celdas o posiciones de memoria**, numeradas de forma consecutiva, capaces de retener información, mientras la computadora está conectada. Para entenderlo imaginad un enorme casillero donde en cada casilla pudiéramos almacenar una combinación de 8 unos y ceros.

A cada posición se puede acceder por medio de un número que la identifica. Dicho número se conoce con el nombre de **dirección de memoria**. Mediante esta dirección se puede acceder de forma directa a cualquiera de ellas independientemente de su posición; se dice, por ello, que la memoria central es un soporte de información de acceso directo.

Dirección de Memoria	Contenido de cada celda
0	10101010
1	11010100
2	11010101
....	01010100
1023	00010100

No hay que confundir los términos celda o posición de memoria con el de **palabra**, ya que esta última es la cantidad de información que puede introducirse o extraerse de la memoria central de una sola vez. Información que se puede leer o escribir en un único golpe de reloj. El tamaño habitual de la palabra suele ser 16, 32, 64 bits.

El tamaño de palabra es un conjunto de posiciones de memoria.

El tamaño de la memoria se mide por bytes (8bits) y sus múltiplos:

Bit: unidad mínima de memoria. Se corresponde con un “1” o con un “0”.

Byte: 8 bits

Kilobyte = 2^{10} bytes

Megabyte = 2^{10} K = 2^{20} Bytes

Gigabyte = 2^{10} M = 2^{20} K = 2^{30} Bytes

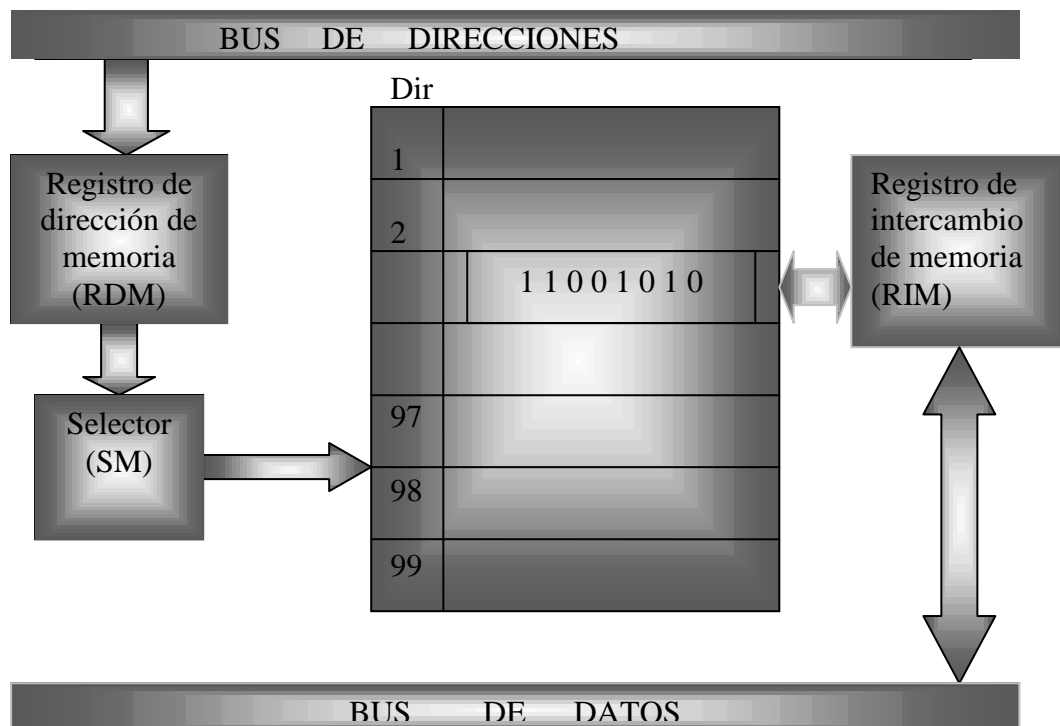
Terabyte = 2^{10} G = 2^{20} M = 2^{30} K = 2^{40} Bytes

Modo de funcionamiento.

La acción de guardar una información en una palabra de la memoria se llama **acceso de escritura**, y la acción de recuperarla, **acceso de lectura**. Los accesos son coordinados siempre por la UC.

La memoria tiene asociados 2 registros para la realización de las operaciones de lectura / escritura y un dispositivo encargado de seleccionar la celda de la memoria implicada en una determinada operación.

- **Registro de Dirección de Memoria (RDM)**, antes de una operación de lectura o escritura se ha de colocar en este registro la dirección de la celda implicada en la operación de lectura o escritura.
- **Registro de Intercambio de Memoria (RIM)** : contiene el dato a L/E
 - si se trata de una operación de lectura de memoria, este registro es el que recibe el dato de la posición de memoria señalada por el RDM.
 - si se trata de una operación de escritura, este registro contiene el dato que vamos a grabar en la posición de memoria referida por el RDM.
- **Selector de Memoria (SM)**, este dispositivo se activa cada vez que se produce una orden de lectura o escritura en memoria, conectando la celda de memoria, cuya dirección figura en el RDM con el RIM, posibilitando el intercambio de información en ambos sentidos.



Descripción de una **operación de escritura**.

- La UC indica a la MP que desea escribir.
- La UC envía la dirección en la que se va a poner el dato. Este se guarda el RDM.
- La UC envía las señales de control necesarias para que el dato a escribir se almacene en el RIM
- La UC da la orden de inicio del acceso. Tras lo cual el SM conecta la celda de memoria indicada por el RDM con el RIM posibilitando el paso de información desde el RIM a la celda de memoria.
- La UC espera el tiempo necesario para que la MP realice la escritura.

Descripción de una **operación de lectura**.

- La UC indica a la MP que desea leer.
- La UC envía la dirección de la que queremos leer. Ésta se guarda el RDM.

- La UC da la orden de inicio del acceso. Tras lo cual el SM conecta la celda de memoria indicada por el RDM con el RIM posibilitando el paso de información desde la celda de memoria al RIM.
- La UC espera el tiempo necesario para que la MP realice la operación de lectura. Tras la cual queda el dato almacenado en el RIM y de ahí se dirigirá a su destino siguiendo de nuevo las órdenes de la UC.

La **velocidad de la memoria** viene determinada por el tiempo de ciclo y el tiempo de acceso.

- Tiempo de acceso de lectura: tiempo transcurrido desde la orden de L sobre la memoria hasta que el dato está disponible en RIM.
- Tiempo de acceso de escritura: Tiempo transcurrido desde la orden de E hasta que el dato se encuentra estable en la posición de memoria.
- Tiempo de acceso: Tiempo medio entre tiempo de acceso en lectura y tiempo de acceso en escritura.
- Tiempo de ciclo: o ciclo base de memoria. Tiempo que necesita la memoria hasta que está dispuesta para efectuar otra operación de E/S. Siempre será mayor que el tiempo de acceso.

Normalmente estos tiempos son fijos para una memoria y están condicionados por el tamaño de la palabra.

d. La unidad de entrada/salida y los periféricos

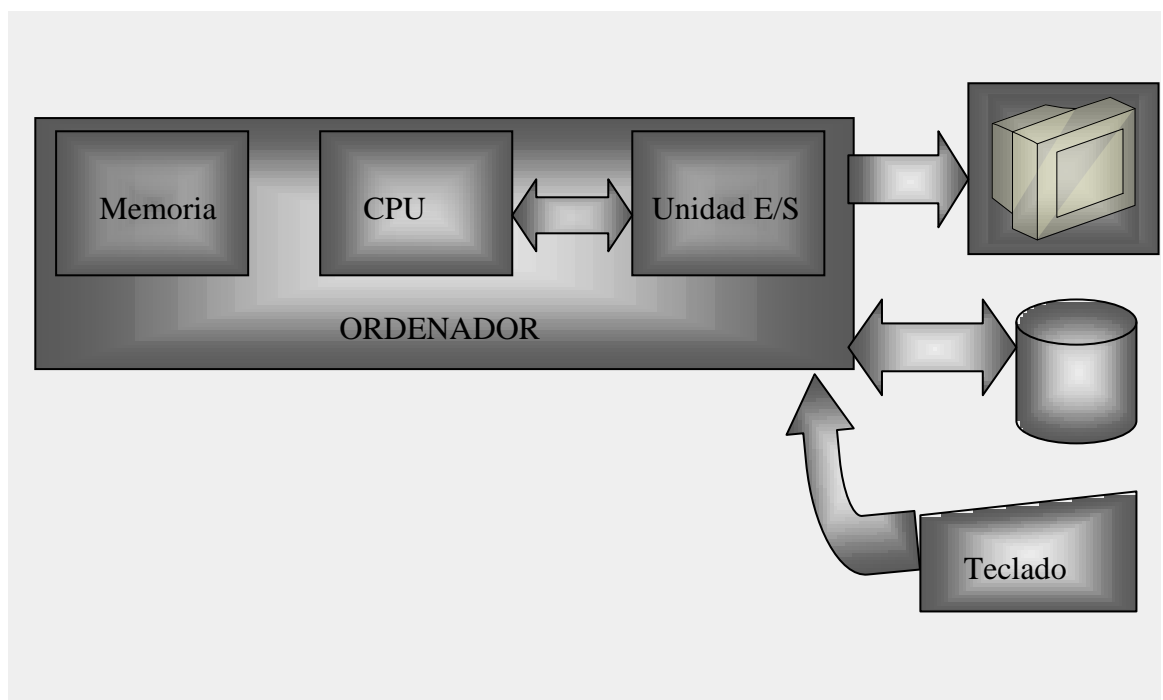
Los periféricos relacionan a la unidad central y su memoria interna con el mundo exterior. Hay una gran variedad de periféricos, algunos tan conocidos y necesarios como el teclado, la pantalla o la impresora, y otros más especializados, como el escáner de imagen óptico. Los periféricos se dividen en periféricos de **entrada**: elementos que permiten la entrada de datos desde el exterior al sistema, por ejemplo: ratón, teclado, escáner.... y periféricos de **salida** que permiten la salida de información desde el sistema al usuario; por ejemplo: pantalla, impresora,

altavoces. También existen dispositivos de **entrada/salida**, que permiten el flujo de información en los dos sentidos.

Entre ellos y la CPU se encuentra la **unidad de entrada/salida** (otros nombres: módulo E/S, controlador).

El porqué de no conectar directamente los dispositivos externos al bus que comunica la CPU y la memoria radica en aspectos de control, velocidades diferentes, tamaño de datos diferentes... todo esto desemboca en una necesidad de adaptar unas capacidades (de periféricos) con otras (CPU). Este módulo E/S tiene las siguientes características:

- Permite la comunicación entre CPU–Periféricos.
- Son genéricos: un módulo de E/S puede gestionar dispositivos externos de varios fabricantes
- A veces pueden ser incluso auténticos procesadores.



La coordinación de la comunicación entre los periféricos y la CPU la realiza la Unidad de E/S. Obsérvese que ésta no es un periférico sino un dispositivo que gestiona a los periféricos siguiendo las ordenes de la CPU; es decir, la Unidad de E/S recibe de la Unidad de

Control información sobre el tipo de transferencia de datos que debe realizar (si es de entrada o de salida) y qué periférico debe utilizar; si es de salida recibirá también el dato que debe enviar y el momento de la operación.

Entonces, la Unidad de E/S seleccionará el periférico y ejecutará la operación teniendo en cuenta las características propias de cada periférico. Una vez ejecutada la orden avisará a la UC de la terminación de la transferencia.

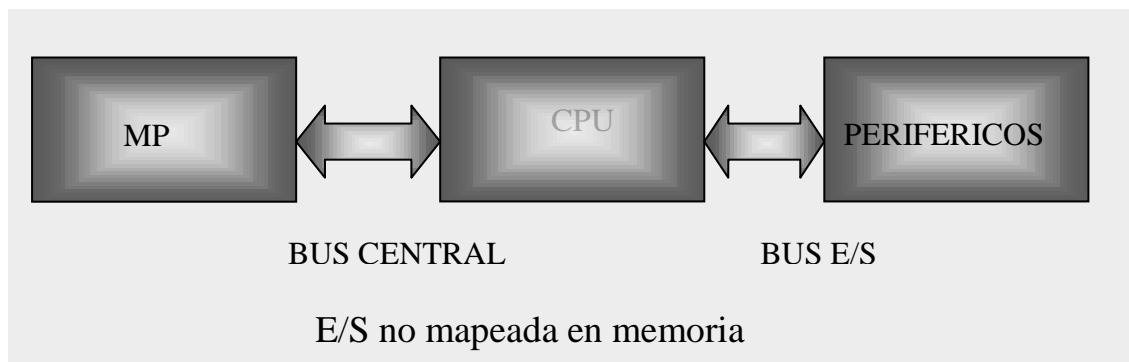
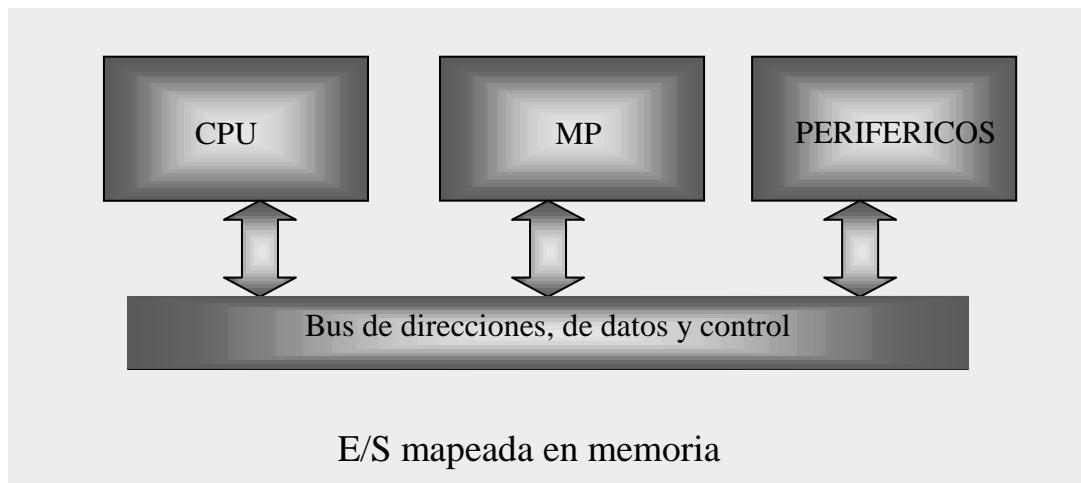
Cada periférico tendrá asignado un número o dirección que servirá para identificarlo. Cuando la UC quiera seleccionarlo enviara dicho número a la Unidad de E/S

Para solucionar el problema de la imposibilidad de saber cuanto tiempo durara una transferencia de información con un periférico se han desarrollado diversas técnicas de comunicación entre la CPU y los periféricos:

- Lo más sencillo es que la CPU, cuando desee hacer una transferencia de información con un periférico, **pregunte** a la Unidad de E/S si dicho periférico se encuentra disponible. Si no lo esta, debe repetir la pregunta una y otra vez hasta obtener una respuesta afirmativa, en cuyo caso se inicia la transferencia de información.
- Si se desea obtener mayor rendimiento del ordenador, se puede emplear otro método que se denomina sincronización mediante **interrupción**. La característica de este método es que la CPU, en lugar de dedicarse a preguntar a la Unidad de E/S por el periférico que desea utilizar, lo que hace es indicar a la Unidad de E/S que desea hacer una transferencia con el periférico, y seguidamente, si no esta el periférico preparado, empieza otra tarea, olvidándose momentáneamente del periférico. Cuando esté preparado, la Unidad de E/S indicara a la CPU que puede realizarse la transferencia; entonces, la CPU interrumpirá la tarea que esté realizando y atenderá al periférico. De esta forma, la CPU no pierde tiempo esperando al periférico.

Existen varias formas de interconectar las unidades funcionales de la computadora entre si:

- Una de ellas emplea un único conjunto de buses, al que se conectan todas las unidades. Esto obliga a que, en un determinado instante, sólo dos unidades puedan estar haciendo uso del mismo; si hay que enviar algo a la tercera unidad, se deberá esperar a que quede libre. La principal ventaja de este sistema de interconexión es su bajo costo, y la principal desventaja, su mayor lentitud. Además, si se emplea el mismo bus de direcciones para indicar la de un periférico y la de una posición de memoria, no podrá coincidir el número de ningún periférico con ninguna dirección de memoria; si así lo hicieran, se producirán errores al seleccionarse a la vez el periférico y la posición de memoria. Por lo tanto, se tendrá cuidado en asignar a los periféricos direcciones que no coincidan con las de la MP. En estos casos se dice que los periféricos están mapeados en memoria, ya que la CPU no distingue entre las transferencias a periféricos y a MP.
- Otra forma de interconectar las unidades de la computadora consiste en emplear conjuntos de buses diferentes para la Unidad de E/S y la MP. De esta forma, aunque se encarece el sistema, se gana en velocidad y se pueden emplear direcciones iguales para nombrar periféricos y a posiciones de memoria. La CPU, si quiere seleccionar un periférico, utiliza el bus de la Unidad de E/S, y si quiere seleccionar una posición de memoria, utiliza el bus de memoria. En este caso se dice que los periféricos no están mapeados en memoria.



e. Buses de comunicación

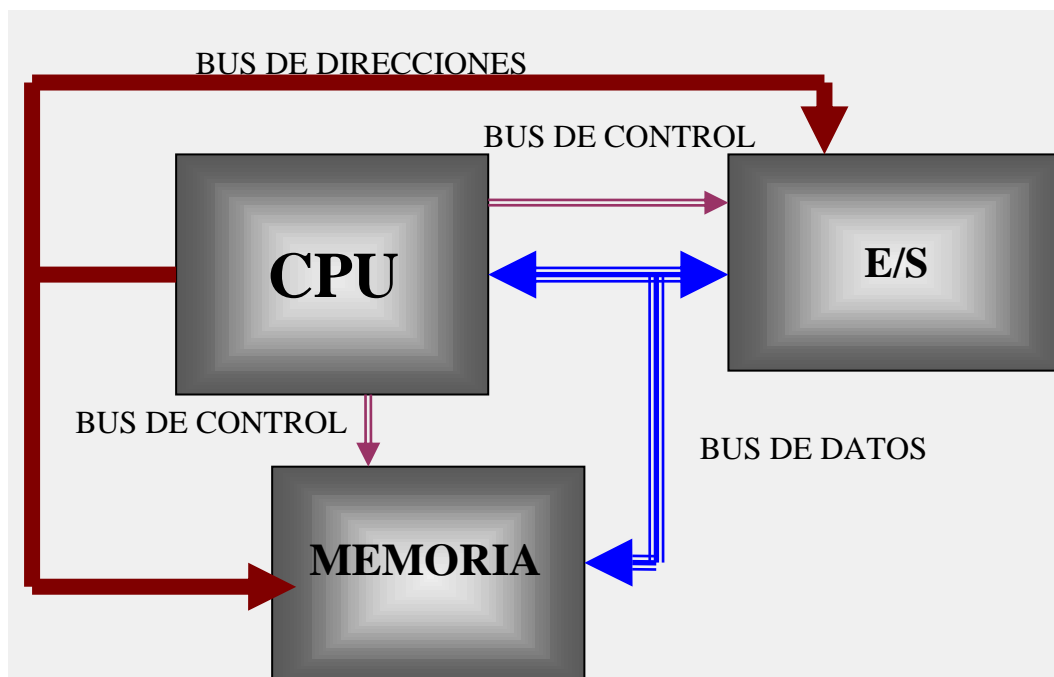
Como ya hemos indicado anteriormente la CPU, la memoria y los periféricos deben estar íntimamente conectadas; aparece en este momento el concepto de bus: el bus es un conjunto de líneas (cables eléctricos) que enlazan los distintos componentes del ordenador, por ellas se realiza la transferencia de datos entre todos sus elementos.

Un bus de 16 bits transfiere simultáneamente esa misma cantidad de bits entre 2 unidades cualesquiera. En función de la naturaleza o tipo de información que estas líneas transmiten nos encontramos con diferentes clases:

- De **control**: forman parte de él las líneas que seleccionan desde dónde y hacia dónde va dirigida la información, también las que marcan la secuencia de los pasos a seguir para dicha transferencia. Transmiten información (señales) de control, procedente de la unidad de control y cuyo

destino puede ser la memoria, para arrancar ciclos de L o E, o cualquier elemento de otra unidad funcional (periféricos).

- De **datos**: por él, de forma bidireccional, fluyen los datos e instrucciones entre las distintas partes del ordenador. En un BUS sólo puede haber una transferencia en un instante. Desde el BUS se pueden cargar simultáneamente varios registros, pero sólo un registro puede volcar información al BUS en un instante. El tamaño del BUS de datos determina el ancho de palabra
- De **direcciones**: como vimos, la memoria está dividida en pequeñas unidades de almacenamiento que contienen las instrucciones del programa y los datos. El bus de direcciones consta de un conjunto de líneas que permite seleccionar de qué posición de la memoria se quiere leer su contenido. También direcciona los puertos de E/S. El tamaño del BUS de direcciones determina el espacio de direccionamiento de memoria.



2. SISTEMAS LÓGICOS

En general se puede considerar al software como la parte lógica que dota al equipo físico de capacidad para realizar cualquier tipo de trabajos, es decir, como un conjunto de ideas y procesos, desarrollados por personas, los cuales marcan las directrices de trabajo de un ordenador. También lo podemos definir como el conjunto de Aplicaciones y Programas que permiten operar con el ordenador, así como controlar y coordinar los distintos elementos del hardware.

El elemento lógico o software, para estar presente en un sistema informático, debe almacenarse en un soporte físico: memoria central, memorias auxiliares.

Existen distintos tipos de software:

- **Software básico.** Es el conjunto de programas que el equipo físico necesita para tener la capacidad de trabajar. Este conjunto de programas constituyen el Sistema Operativo. Una definición más técnica nos presenta al sistema operativo como el soporte lógico que controla el funcionamiento del equipo físico, ocultando los detalles del hardware y haciendo sencillo el uso del ordenador al usuario.
- **Software de aplicación.** Está formado por el conjunto de programas que han sido diseñados para que el ordenador realice un trabajo, tanto genérico (paquetes de software) como específico (gestión de nóminas,..) y los datos. Tipos de Software de aplicación:
 - **APLICACIONES ESPECÍFICAS:** Son aplicaciones desarrolladas a medida de las necesidades del usuario.
 - **PAQUETES STANDARD:** Se compra el paquete y se paga, y ya podemos trabajar con el sin ningún problema.
 - **FREEWARE:** Son paquetes que han sido donados por los programadores y solo hay que pagar el soporte y gastos de envío. Muy barato. También se le llama software de Dominio Público o software libre.

- SHAREWARE: Son paquetes los cuales podemos pedir pagando un precio muy bajo. Una vez probados, si se quiere la versión completa habrá que pagar el resto hasta completar su valor. Aún así suelen ser más baratos que los paquetes estándar.
- Lenguajes de programación (entornos de desarrollo): programas para desarrollar otros programas.

a. Formato de las instrucciones

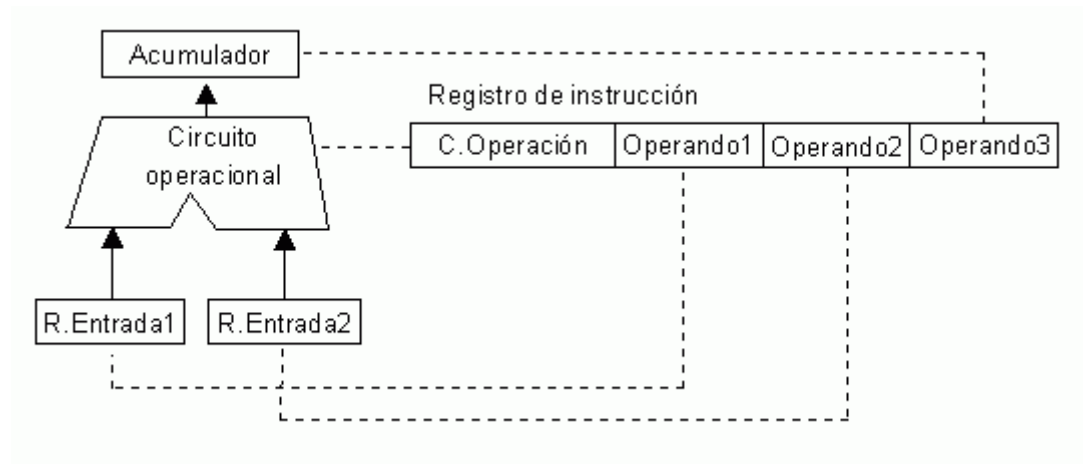
Las instrucciones que es capaz de realizar la UCP se denominan **instrucciones máquina**. El lenguaje que se utiliza para su codificación es el **lenguaje máquina** y, de acuerdo con su función, se clasifican como:

- Instrucciones de cálculo (aritmético y lógico) (SUM, RES, MUL, AND, OR, ORX).
- Instrucciones de transferencia de datos (realizan el movimiento de datos entre la memoria principal y los registros).
- Instrucciones de control de flujo (Instrucciones de salto incondicional o condicional).
- Instrucciones de entrada/salida (instrucciones para transferir datos desde el interior de la computadora a un dispositivo exterior o viceversa, así como enviar órdenes a dichos dispositivos).
- Etcétera.

Una segunda clasificación de las instrucciones máquina hace referencia a su formato y al número de operandos que intervienen en ellas:

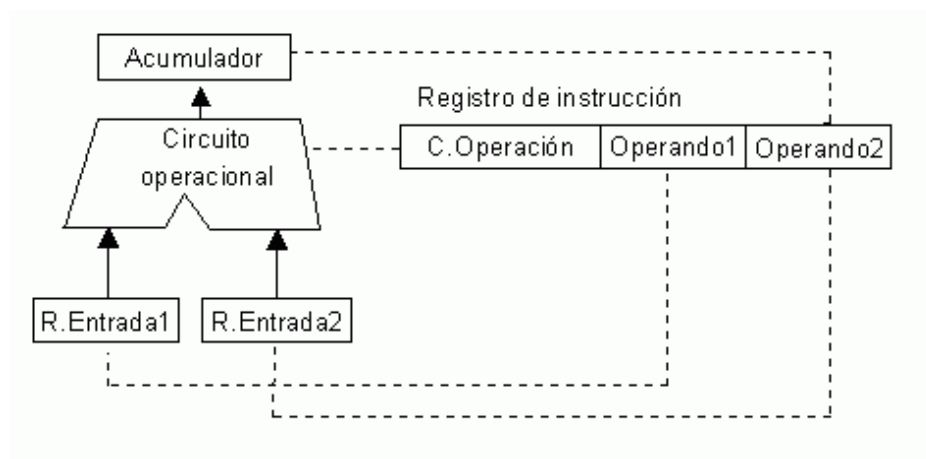
I. Instrucción de tres operandos

Está compuesto por cuatro campos. El primero será el código de la operación y a continuación tres operandos. Los dos primeros son las direcciones de los argumentos que hay que operar y el tercero es la dirección donde se depositará el resultado.



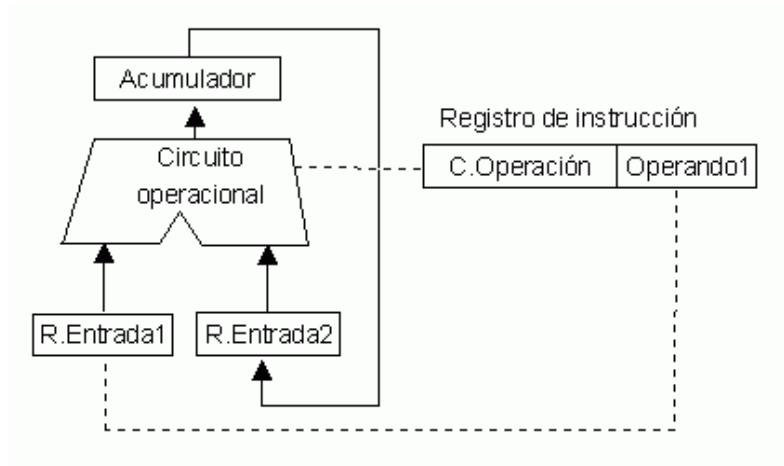
II. Instrucción de dos operandos

Contiene el código de operación y las direcciones de los dos operandos, de los que uno de ellos actúa, además, como receptor del resultado de la operación



III. Instrucción de un operando

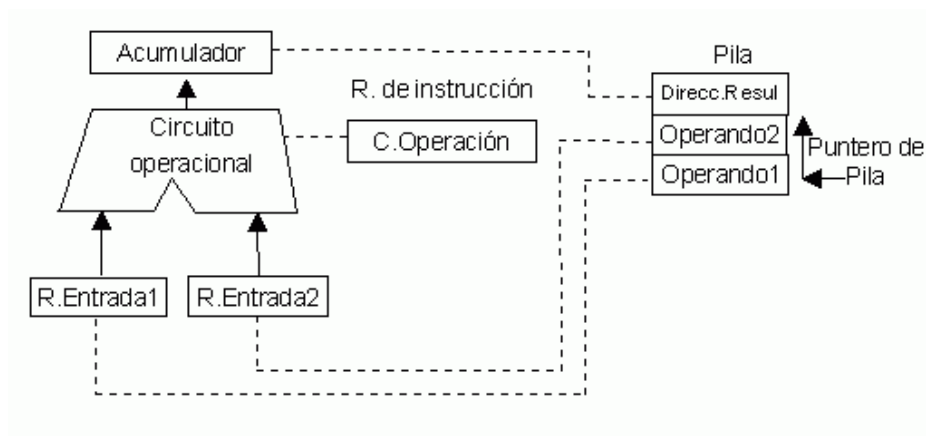
Se utiliza en máquinas cuya arquitectura funciona con filosofía de acumulador. El acumulador de la UAL contiene previamente el primer argumento de la operación, el segundo es el indicado en la dirección de la propia instrucción, y después de ser operados ambos por el circuito operacional, el resultado queda depositado de nuevo en el acumulador.



IV. Instrucción sin operandos

Se utiliza generalmente en computadoras cuya arquitectura tiene filosofía de pila o stack. Una pila está formada por datos almacenados en orden consecutivo en la memoria, existiendo un registro especial denominado puntero de pila, que nos indica la dirección del último dato introducido en ella. Al extraer un elemento de la pila el puntero apuntará al anterior elemento y irá disminuyendo su valor hasta llegar al primer elemento que hayamos introducido.

Estas instrucciones solo tienen el código de la operación y si se trata de una operación de cálculo extraerá los operandos de la pila.



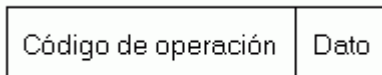
Las computadoras dependiendo de la arquitectura que tengan pueden tener varios tipos de instrucciones.

b. Modos de direccionamiento

Hemos visto diferentes tipos de instrucciones, y en ellas siempre hemos dicho que se encuentra la dirección del dato. Ahora vamos a ver que diferentes modos existen para direccionar.

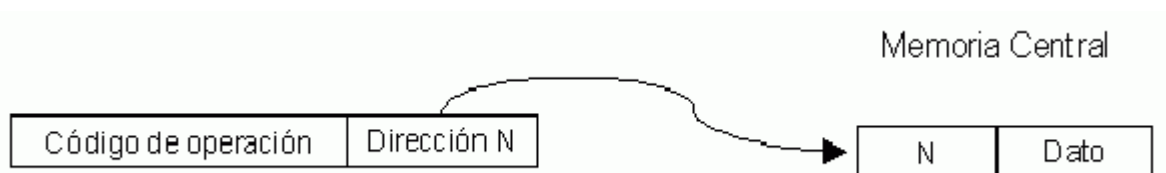
I. Direccionamiento inmediato

En este modo el dato forma parte de la instrucción, es decir, en los operandos de la instrucción no se encuentra la dirección del dato, sino el mismo dato.



II. Direccionamiento Directo

La instrucción contiene la dirección en la memoria central del dato que se va a procesar. Esto obliga a tener que hacer un acceso a memoria para poder extraer el dato y enviarlo a la ALU o hasta la unidad designada por la instrucción.

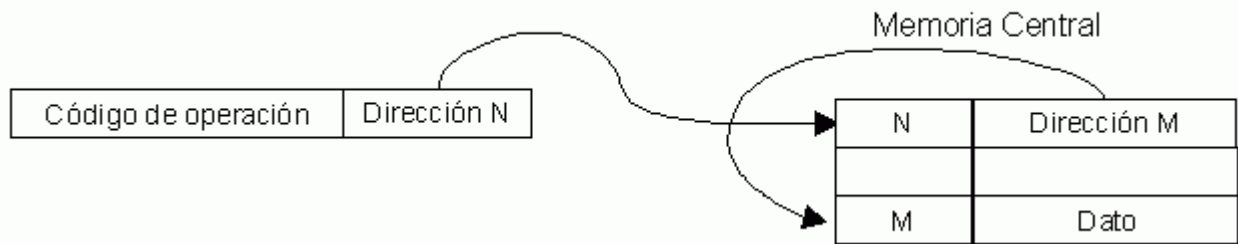


Las características principales de este direccionamiento son las siguientes:

- No son necesarios cálculos previos para determinar la dirección final.
- Necesita un acceso a memoria (ciclo de memoria) para acceder al dato, y trasladarlo hasta la unidad aritmético-lógica o hasta la unidad designada por la instrucción.

III. Direccionamiento indirecto

Este caso es conocido también por indirección. La dirección contenida en la instrucción no es la del dato implicado sino la de una posición de memoria que contiene la dirección de ese dato. Esta posición se denomina dirección intermedia.

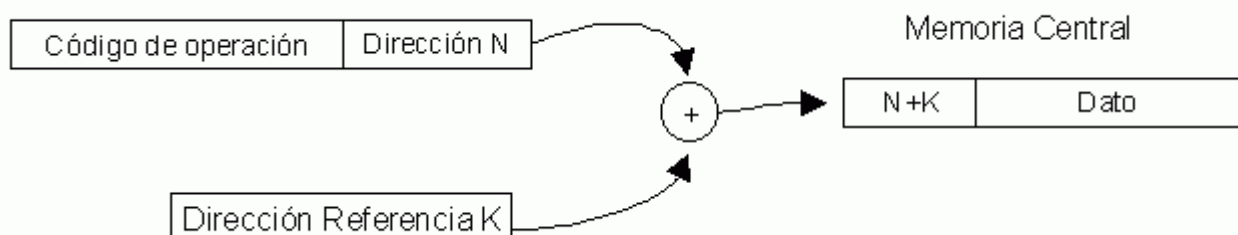


Las características principales de este direccionamiento son las siguientes:

- No son necesarios cálculos previos para determinar la dirección final.
- Necesita dos ciclos de memoria para acceder al operando; durante el primero se accede a la dirección intermedia y en el segundo se accede al operando.
- Con este caso se incrementa notablemente la capacidad de direccionamiento al poder utilizar todos los bits de la palabra de memoria.
- La indirección es un método eficaz para transferir los parámetros de un programa principal a subrutinas.
- El montaje de enlaces en bancos de datos se realiza mediante este direccionamiento de una forma sencilla.
- Este direccionamiento permite con una sola instrucción, de un programa, acceder a distintas zonas de la memoria.
- Necesariamente, todas las computadoras que utilizan memoria paginada necesitan de un direccionamiento indirecto

IV. Direccionamiento relativo

En él, la dirección de memoria donde se encuentra el dato se consigue sumando la dirección contenida en la propia instrucción con una magnitud fija contenida en un registro especial. De esta manera se posibilita el acceso a un conjunto de posiciones determinadas, normalmente consecutivas, a partir de una posición considerada como dirección de referencia.



Existen más modos de direccionamiento distintos como resultado de combinar los mencionados anteriormente.

c. Fases de una instrucción

Para que una computadora pueda ejecutar un programa, éste ha de estar almacenado en la memoria central. La unidad central de proceso tomará una a una sus instrucciones e irá realizando las tareas correspondientes.

Al conjunto de acciones que se llevan a cabo en la realización de una instrucción se denomina **ciclo de instrucción**.

Cada ciclo de instrucción se compone de dos fases:

- **Fase de búsqueda o carga.** En esta fase se transfiere la instrucción correspondiente desde la memoria central a la unidad de control.
- **Fase de ejecución.** En esta fase se realizan todas las acciones que conlleva la propia instrucción.

I. Fase de carga de una instrucción.

Supongamos que se tiene un ejemplo de instrucción aritmética de suma con tres direcciones y direccionamiento directo; es decir, la instrucción contiene el código de operación correspondiente a la suma, los dos primeros operandos están en las direcciones de memoria correspondiente y el resultado ha de quedar en la dirección indicada por el tercer operando.

SUMAR 033 992 993 (Sumar los contenidos de memoria 33 y 992 y almacenar el resultado en la posición 993).

PASO 1.- La UC envía una microorden para que el contenido del registro de contador de programa (CP) que contiene la dirección de la siguiente instrucción (instrucción que corresponde procesar), sea transferido al registro de dirección de memoria (RDM). Esta información viaja por el bus de direcciones.

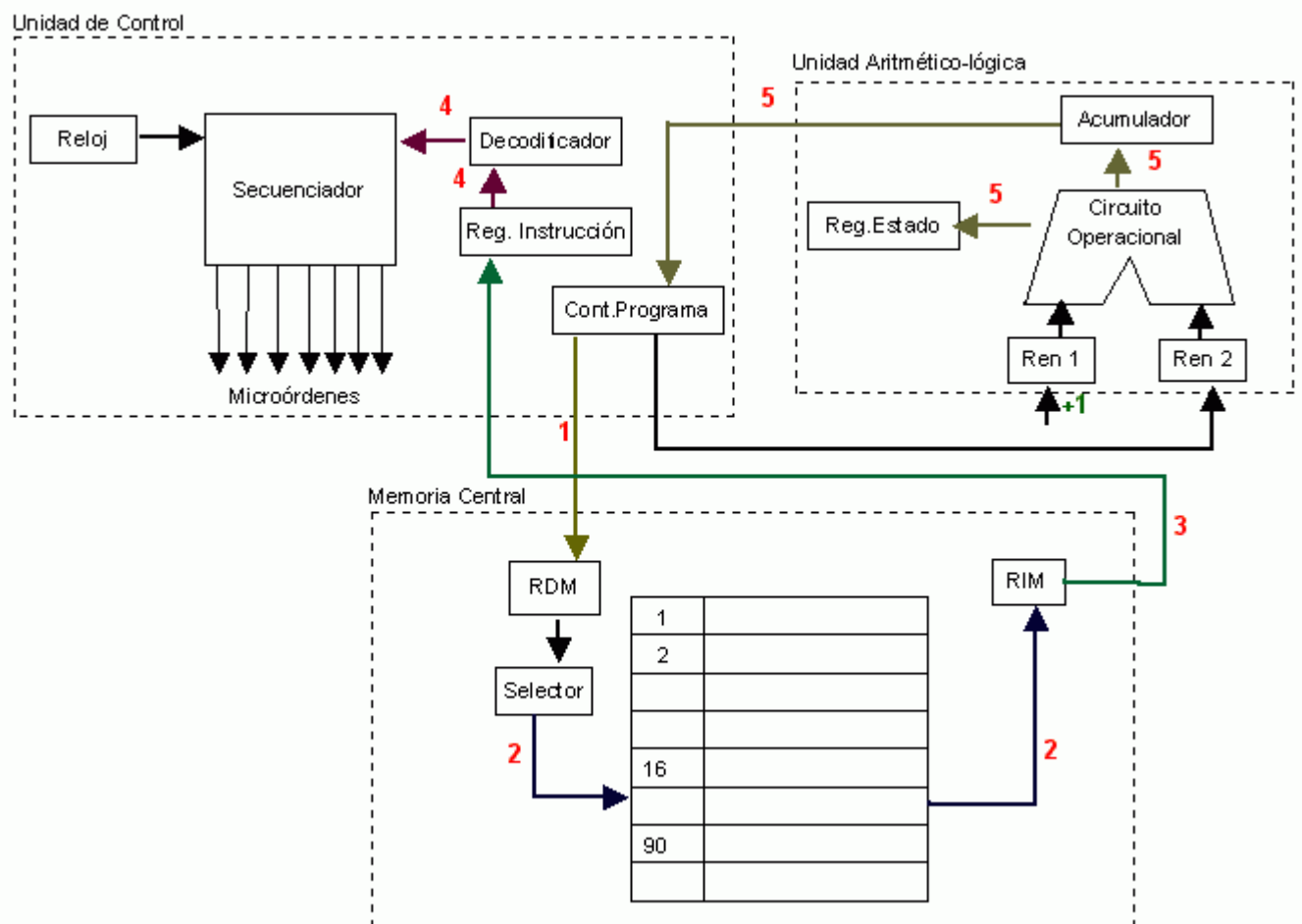
PASO 2.- La posición de memoria que figura en el registro de dirección de memoria (RDM) es utilizada por el Selector para transferir su contenido (instrucción) al registro de intercambio de memoria (RIM).

PASO 3.- Se transfiere la instrucción desde el registro de intercambio de memoria (RIM) al registro de instrucción (RI) por el bus de datos.

PASO 4.- A continuación el decodificador procede a interpretar la instrucción que acaba de llegar al registro de instrucción (RI); en este caso SUMAR, quedando dispuesto para la activación del circuito sumador de la UAL e informando al secuenciador.

PASO 5.- El registro contador de programa (CP) se autoincrementa (utilizando la ALU) con un valor 1 (o n en el caso de que sea ésta la longitud de la palabra de memoria), de tal forma que quede apuntando a la siguiente instrucción situada consecutivamente en memoria. Si la instrucción en ejecución es de ruptura de secuencia, el contador de programa (CP) se cargará con la dirección que corresponda.

Cuando lee una instrucción de salto a una subrutina, lo primero que hace es almacenar el estado de partida (desde la subrutina que llama) en el que se encuentra, en la zona de stack (pila). Cuando encuentre la instrucción de regreso leerá de la zona de stack a que punto debe regresar. Hay que decir, que esta zona de stack que gestiona por el método LIFO (Last In First Out).



II. Ejecución de una instrucción.

La ejecución se realiza en los siguientes pasos, teniendo en cuenta que si la instrucción no hubiese necesitado operandos, no se ejecutarían los pasos 1 a 6 ni el 8.

PASO 1.- Se transfiere la dirección del primer operando desde el registro de instrucción (RI) al registro de dirección de memoria (RDM) por el bus de direcciones.

PASO 2.- El selector extrae de la memoria dicho dato depositándolo en el registro de intercambio de memoria (RIM).

PASO 3.- Se lleva este operando desde el registro de intercambio de memoria (RIM) al registro de entrada 1 (REN 1) de la unidad aritmético-lógica. (UAL) por el bus de datos.

PASO 4.- Se transfiere la dirección del segundo operando desde el registro de instrucción (RI) al registro de dirección de memoria (RDM) por el bus de direcciones.

PASO 5.- El selector extrae de la memoria dicho dato depositándolo en el registro de intercambio de memoria (RIM).

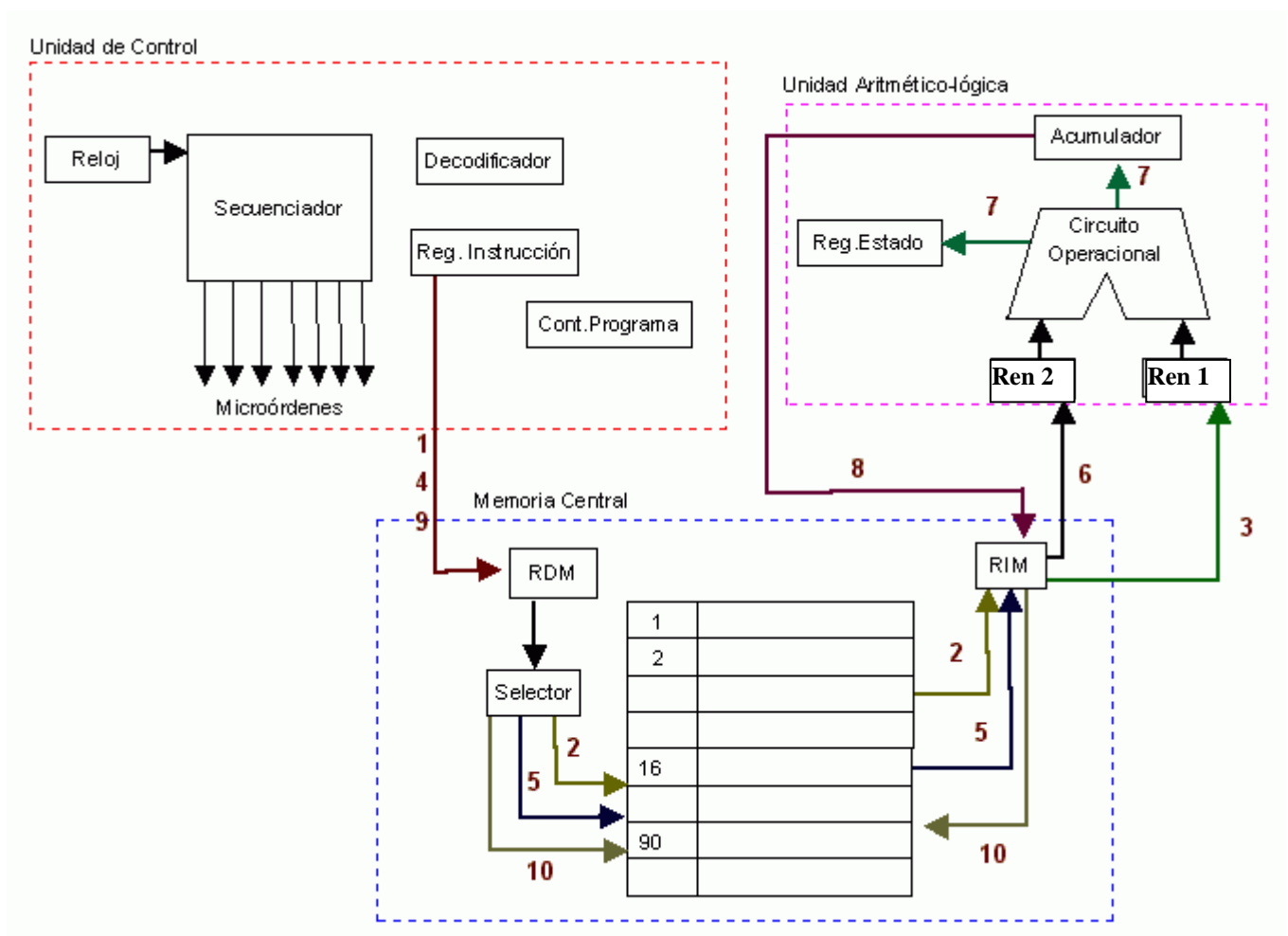
PASO 6.- Se lleva este operando desde el registro de intercambio de memoria (RIM) al registro de entrada 2 (REN 2) de la unidad aritmético-lógica (UAL). Para este traslado se utiliza el bus de datos.

PASO 7.- El secuenciador envía una microorden a la unidad aritmético-lógica (UAL) para que se ejecute la operación de que se trate. El resultado de la operación queda almacenado en el registro acumulador (RA).

PASO 8.- Este resultado es enviado desde el registro acumulador (RA) al registro de intercambio de memoria (RIM) por el bus de datos.

PASO 9.- Se transfiere desde el registro de instrucción (RI) al registro de dirección de memoria (RDM) la dirección donde ha de almacenarse el resultado de la memoria. Se utiliza el bus de direcciones para esta comunicación

PASO 10.- Se transfiere el resultado desde el registro de intercambio de memoria (RIM) a la dirección de memoria indicada en el registro de dirección de memoria (RDM).



Bibliografía . *Material específico para los ciclos formativos de informática. elaborado por Convenio entre la Conselleria de Cultura, Educación y Deporte y la Universidad de Alicante.*