



Universidad Nacional de la Matanza

Departamento de Ingeniería  
e Investigaciones Tecnológicas

# Fundamentos de TIC's

## UNIDAD 4. INTRODUCCIÓN AL HARDWARE DE LOS SISTEMAS DE COMPUTACIÓN

JEFE DE CÁTEDRA

**Mg. Artemisa Trigueros**

COLABORACIÓN

**Docentes de la Cátedra**

**ALDERETE, Claudia**

**AMATO, Alfredo**

**BELLOMUSTO, Hugo**

**CILENTI, Mabel**

**GOMEZ, Sabrina**

**GONZALEZ SANZ, Carlos**

**PÉREZ ARÁUZ, Alejandro**

**RODRÍGUEZ, Rocío**

**TORO, Larisa**

**TRIGUEROS, Artemisa**

**2020**

## Índice

1. Introducción.....	3
1.1. Concepto de computación .....	3
1.2. La Prehistoria de las Computadoras (Generación Cero).....	3
1.3. Generaciones de computadoras .....	4
1.4. Diseño de una computadora.....	5
2. Los módulos y sus interconexiones en el Modelo de Von Neumann .....	9
2.1. Buses.....	9
2.2. Formas de transmisión .....	10
2.3. Buses internos de una computadora .....	10
3. El concepto de Memoria en Informática.....	13
3.1. Memoria ROM.....	13
3.2. Memoria de Lectura Escritura (RWM).....	16
3.3. Modos de acceso a la información .....	16
3.5 Jerarquías de Memorias .....	19
3.6. Organización de Direccionamiento de Memoria y Entrada / Salida .....	19
4. La Unidad Central de Proceso (C.P.U.) .....	20
4.1. Registros físicos .....	20
4.2. Unidad de Control (UC).....	23
4.3. Unidad Aritmética y Lógica (ALU).....	26
5. Fundamentos de la Unidad de Entradas y Salidas .....	30
5.1. Generalidades .....	30
5.2. Funcionamiento de la Unidad de E/S .....	31
5.3 Organización de E/S.....	32
5.4. Atención a Dispositivos de Entradas y Salidas. Interrupciones y DMA. ....	33
6. Fundamentos del Procesamiento de Datos.....	34
6.1 Instrucciones .....	34
6.2 Ciclo de instrucción .....	35
6.3 Ejemplo de procesamiento .....	36
7. Modos de direccionamiento.....	40
7.1 Direccionamiento Absoluto .....	40
7.2 Direccionamiento Relativo.....	40
8. Periféricos y dispositivos de entrada / salida .....	41
8.1 Clasificación De Periféricos Y Dispositivos De Entrada / Salida .....	41
Disco magnéticos .....	42

# INTRODUCCIÓN AL HARDWARE DE LOS SISTEMAS DE COMPUTACIÓN

## 1. Introducción

Cualquier idea, código, secuencia de pasos o procedimientos, requiere de elementos tangibles para poder implementarse. Un arquitecto diseña una vivienda y la construye por medio de máquinas, herramientas, etc. Un cocinero inventa una nueva receta y la prepara en una cocina, con utensilios, balanza, batidora, etc. Un compositor prepara un nuevo álbum y lo graba utilizando piano, micrófonos, parlantes, equipos digitalizadores, etc.

El mismo concepto puede extenderse a una computadora. Este capítulo se introduce dentro del mismo para analizar y comprender el funcionamiento y la constitución de sus partes físicas, tangibles, denominadas genéricamente: **HARDWARE** (ver figura 1).



**Figura 1:** El hardware en los sistemas de computación.

### 1.1. Concepto de computación

El hombre, desde la antigüedad, ha utilizado diferentes objetos para contar: marcas en ramas, dedos, piedras, etc. De hecho, la palabra “cálculo” proviene del latín “*calculus*” que significa *piedra*. Agregando o quitando esos elementos aumentaba o reducía la cantidad representada y de esta forma comenzó a realizar operaciones aritméticas. Posteriormente, con el correr del tiempo, inventó dispositivos que permiten realizar las operaciones en forma automática. Es decir, máquinas que realizan la acción de “calcular”. Se destacan, dentro de este conjunto; el ábaco; que ya se utilizaba varios siglos antes de Cristo en Asia, Grecia y Roma, así como el ábaco mexicano (entre 900 y 1.000 DC) y los quipus incas (ver figura 2).



**Figura 2:** Ábaco japonés y quipu inca, utilizados para calcular

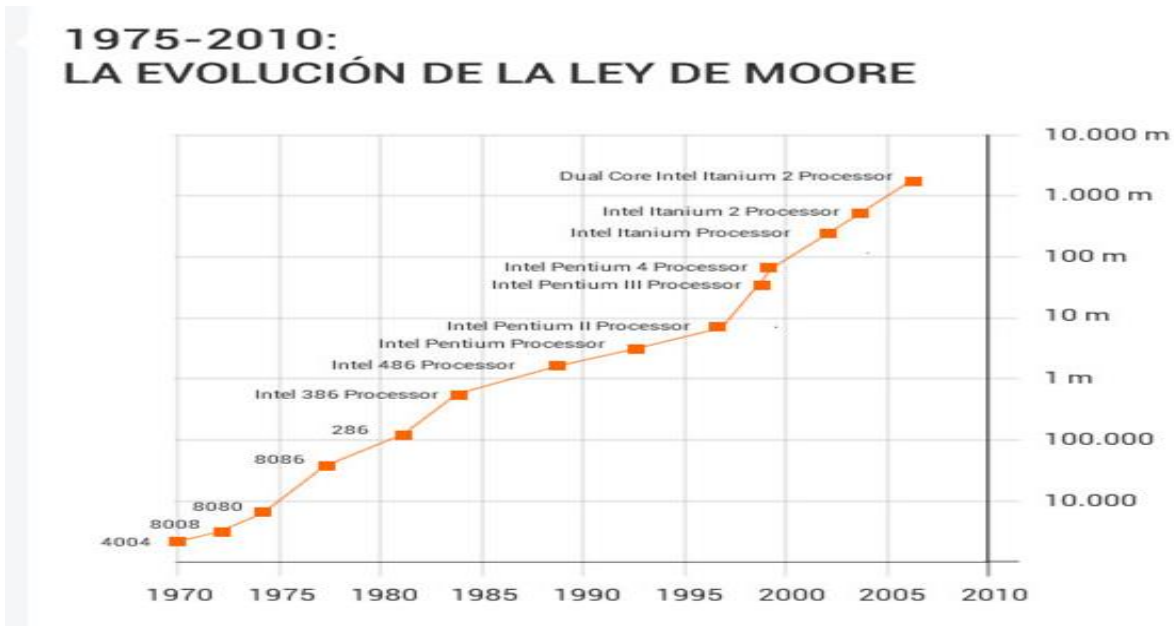
### 1.2. La Prehistoria de las Computadoras (Generación Cero)

Posteriormente se destacan, entre otras muchas invenciones tendientes a facilitar la tarea de calcular, las correspondientes a:

- Blas Pascal (1642), denominada *Pascalina* en su honor.

- La *Máquina Analítica* de Charles Babbage (1832), considerado el “Padre de la Informática” y su colaboradora Ada Lovelace (primera Programadora de la Historia). Dicha máquina era una computadora mecánica automática y programable, aunque debido a la falta de tecnología de la época no pudo ser construida, conservándose un prototipo.
- La *Máquina Tabuladora* de Herman Hollerith, utilizada para el Censo de Población de EEUU del año 1890, que utilizaba tarjetas perforadas para ingreso de datos.

En los últimos años<sup>1</sup>, las computadoras han cambiado más que los automóviles desde su invención. Tanto los índices de performance como de costos se han mejorado exponencialmente. Gordon E. Moore, cofundador de Intel, observó que la cantidad de transistores<sup>2</sup> por circuito integrado<sup>3</sup> se duplica aproximadamente cada dos años (Ley de Moore). Como consecuencia, a lo largo del tiempo los costos bajan y las prestaciones suben (ver figura 3).



**Figura 3:** Evolución de la Ley de Moore (ver referencias al pie).

### 1.3. Generaciones de computadoras<sup>4</sup>

- 1) Primera generación (1945-1953): Las primeras computadoras usaban **tubos de vacío** para los circuitos y tambores magnéticos para la memoria, ocupando habitaciones enormes. El gran consumo de energía y disipación del calor era una causa frecuente de desperfectos. Para programar las computadoras se utilizaba el lenguaje máquina (0 y 1), se utilizaban las tarjetas perforadas para ingresar los datos e impresiones para las salidas.
- 2) Segunda generación (1954-1965): Los transistores reemplazaron los tubos de vacío (componente electrónico) y permitieron construir computadoras más pequeñas, más

<sup>1</sup> En “Historia de las Computadoras”, [http://www.webopedia.com/DidYouKnow/Hardware\\_Software/FiveGenerations.asp](http://www.webopedia.com/DidYouKnow/Hardware_Software/FiveGenerations.asp)  
[https://en.wikipedia.org/wiki/Timeline\\_of\\_computing](https://en.wikipedia.org/wiki/Timeline_of_computing)  
[http://tcm.computerhistory.org/Timeline/DCM\\_Computer\\_Generations\\_1975.pdf](http://tcm.computerhistory.org/Timeline/DCM_Computer_Generations_1975.pdf)  
<http://tcm.computerhistory.org/marketing/guidecomputergenerationsbooklet1975.pdf>

<sup>2</sup> pequeño dispositivo semiconductor que cierra o abre un circuito o amplifica una señal

<sup>3</sup> Chip o pastilla semiconductor de silicio en la cual se insertan componentes electrónicos (incluyendo transistores, diodos, resistencias, condensadores y alambres de conexión)

<sup>4</sup> <https://books.google.com.ar/books?id=QGPHAI9GE->

IC&pg=PA24&lpg=PA24&dq=computer+generation+VLSI&source=bl&ots=FVEttGrW9-&sig=U9RLUjOq\$KomISDZ0A86f4mVGak&hl=en&sa=X&ved=0CD0Q6AEwBmoVChMIvrLj9cqiyAIVQhuQCh0M9Qeg#v=onepage&q=computer%20generation%20VLSI&f=false

rápidas, con menor consumo de energía y más confiables que sus predecesoras. Se comenzó a utilizar un lenguaje simbólico (Assembly) para no utilizar directamente el código binario. Se comenzaron a desarrollar lenguajes de alto nivel como COBOL y FORTRAN. Fueron las primeras computadoras que grababan las instrucciones de los programas en memoria. Además los tambores magnéticos fueron reemplazados por memorias de núcleos magnéticos.

- 3) Tercera generación (1965-1980): Los transistores fueron miniaturizados y ubicados dentro de circuitos integrados, incrementando drásticamente la velocidad y la eficiencia de las computadoras. En vez de tarjetas perforadas e impresiones, los usuarios interactuaban a través de teclados y monitores controlados por un sistema operativo, el cual permitía ejecutar aplicaciones diferentes en una misma computadora. La reducción de tamaño y costos permitió que las computadoras llegaran a una audiencia masiva.
- 4) Cuarta generación (1980-...): Una micro-miniaturización permitió que más de cientos de miles de componentes fueran incorporados dentro de un solo circuito integrado. Las computadoras comenzaron a hacerse verdaderamente masivas por las bajas en los precios.



Figura 4: Las generaciones de las computadoras

## 1.4. Diseño de una computadora

En los inicios, programar era sinónimo de interconectar cables, cambiar switches o directamente, rediseñar físicamente una máquina. Una computadora era construida *con solo un objetivo*, como las calculadoras actuales, que sólo permiten realizar operaciones aritméticas, pero no permiten instalarles software de edición gráfica, procesamiento de textos o un video juego.

Recordemos algunas definiciones importantes:

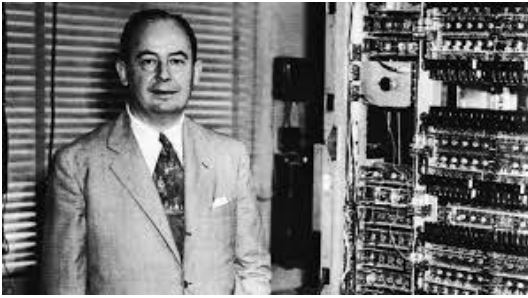
- **Programa:** Un programa es un conjunto de instrucciones que se ejecutan en forma secuencial en base a un algoritmo.
- **Instrucción:** es una orden que será ejecutada por el procesador. Una computadora de **programa almacenado en memoria** es aquella que almacena las instrucciones del programa a ejecutar en memoria principal. Esto permite *cargar* un programa a memoria y ejecutarlo, sin necesidad de cambiar físicamente la computadora.

Una arquitectura define los componentes y la vinculación entre ellos, hay dos arquitecturas de computadoras que se han hecho populares y son las que se describen a continuación:

### 1.4.1. Arquitectura de Von Neumann<sup>5</sup>

**John von Neumann** (Budapest, 1903 - Washington, 1957)

<sup>5</sup> First draft of a report on the EDVAC. Von Neumann, John. Moore School of Electrical Engineering, University of Pennsylvania, 1945. <http://library.si.edu/digitallibrary/book/firstdraftofrepo00vonn>. Biblioteca del Instituto Smithsonian, Washington, D.C.



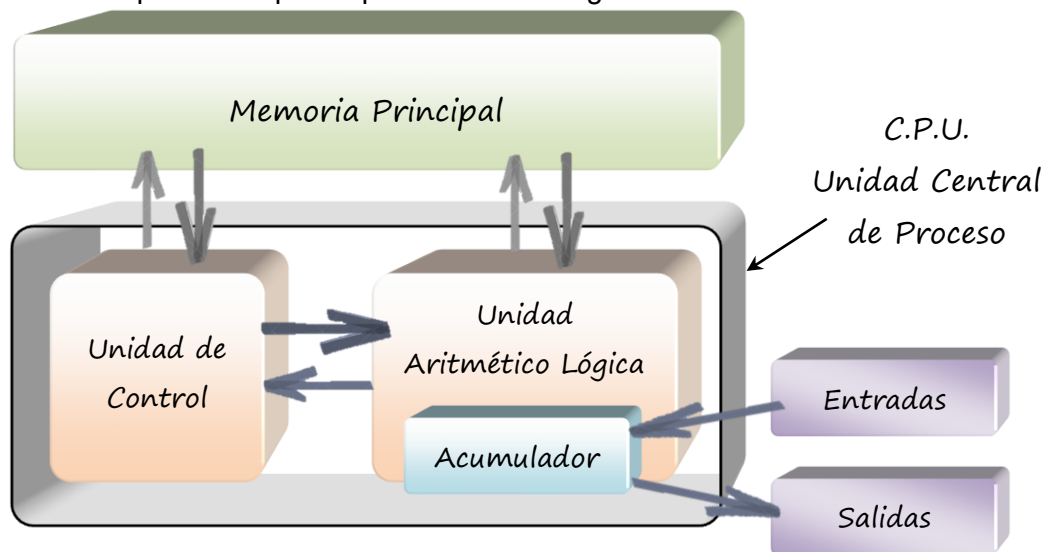
Matemático húngaro nacido en Budapest en 1903, nacionalizado estadounidense. Desde niño mostró unas extraordinarias dotes para las matemáticas. En 1925 se recibió de Ingeniero Químico en Zurich y en 1932 se doctoró en matemáticas en la Universidad de Budapest. En la Universidad de Berlín asistió a los cursos de Albert Einstein. En 1933 se radicó en Estados Unidos, en la Universidad de Princeton.

Realizó contribuciones fundamentales en física cuántica, análisis funcional, teoría de conjuntos, teoría de juegos, ciencias de la computación, economía, análisis numérico, cibernética, hidrodinámica, estadística y muchos otros campos. Es considerado como uno de los más importantes matemáticos de la historia moderna.

En el año 1946 desarrolló su modelo de computadora de programa almacenado.

Falleció en Estados Unidos en 1957.

La arquitectura de Von Neumann describe una computadora de programa almacenado en memoria con los componentes que se presentan en la figura 5.



**Figura 5:** Arquitectura de Von Neumann

Los componentes de la arquitectura de Von Neumann son:

a) Memoria Principal (MP): provee almacenamiento para datos e instrucciones de programa.

b) Unidad Central de Proceso (CPU), compuesta por:

- Unidad de Control (UC): es la encargada de dirigir el funcionamiento de toda la computadora, es el “director de la orquesta”. Ejecuta las instrucciones del programa, le indica a la ALU cuál operación realizar, envía la señal de leer o escribir en memoria, etc.
- Unidad Aritmético Lógica (ALU): como su nombre lo indica, realiza operaciones aritméticas (suma, resta a través del complemento, producto a través de sumas sucesivas) y lógicas (complemento, comparación, entre otras).
- Registros de la CPU: son pequeñas áreas de almacenamiento dentro de la CPU. El Acumulador es un registro donde la ALU vuelca los resultados intermedios.

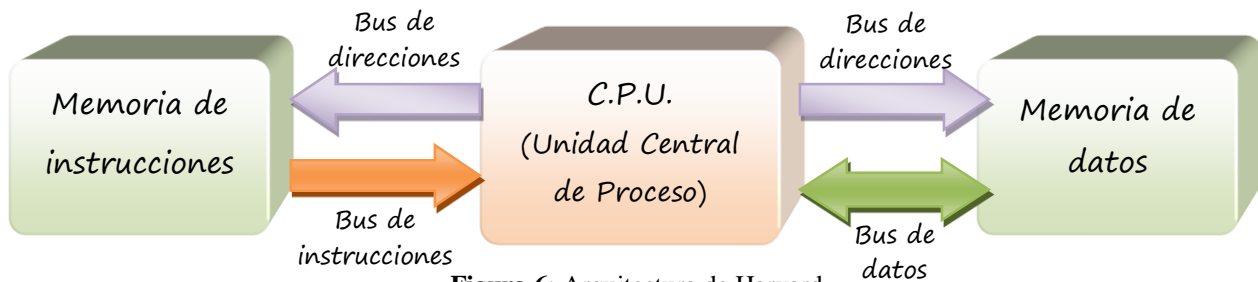
c) Sistema de Entradas y Salidas (E/S o Input/ Output, I/O): Los dispositivos de entrada y salida permiten la interacción de la computadora con el mundo exterior.

#### 1.4.2. Arquitectura Harvard

La arquitectura Harvard fue la utilizada en la primera computadora: Harvard Mark I (1944). La característica más sobresaliente en comparación con la arquitectura de Von Neumann es la



utilización de memorias separadas para las instrucciones y para los datos. Además tiene buses<sup>6</sup> separados para datos e instrucciones, lo que permite acceder a datos e instrucciones en forma simultánea. Comúnmente, las instrucciones del programa están en una memoria de sólo lectura y las de datos en una memoria de lectura y escritura (ver figura 6).



**Figura 6:** Arquitectura de Harvard

En lo que resta del apunte se examinarán las distintas partes de la arquitectura de Von Neumann.

### 1.4.3. PLC. Programmable Logic Controller

Un controlador lógico programable, más conocido por sus siglas en inglés PLC (Programmable Logic Controller) o por autómatas programables, es una computadora utilizada por las distintas ingenierías para automatizar procesos electromecánicos, tales como, control de maquinaria de fabricación, en líneas de montaje, depósitos, atracciones mecánicas, etc.

Los PLC a diferencia de las computadoras de propósito general, cumplen con una función específica, está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico, resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en memorias no volátiles y copias de seguridad.

Un PLC es un ejemplo de un sistema de tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.

Los primeros PLC fueron diseñados para reemplazar los sistemas de relés lógicos, fueron programados en lenguaje llamado “Listado de instrucciones”, con el cual las órdenes de control se le indicaban al procesador como un listado secuencial de códigos en lenguaje de máquinas. Luego para facilitar el mantenimiento de los sistemas a controlar se introdujo un lenguaje gráfico llamado lenguaje “Ladder” también conocido como diagrama de escalera, que se parece mucho a un diagrama esquemático de la lógica de relés.

Los PLCs modernos pueden ser programados de diversas maneras, desde diagramas de contactos, a los lenguajes de programación tales como: dialectos especialmente adaptados de BASIC y C. Otro método es la lógica de estado, un lenguaje de programación de alto nivel diseñado para programar basados en diagramas de estado.

La función básica y primordial del PLC ha evolucionado con los años para incluir el control del relé secuencial, control de movimiento, control de procesos, sistemas de control distribuido y comunicación por red. Las capacidades de manipulación, almacenamiento, potencia de procesamiento y de comunicación de algunos PLCs modernos son aproximadamente equivalentes a las computadoras de escritorio.

<sup>6</sup> Un bus (se definirá más adelante) es un conjunto de conductores eléctricos que pueden estar impresos en pistas metálicas sobre la placa principal de la computadora. A través de este conjunto de conductores circulan las señales que representan los datos.

### Estructura interna

Sus partes fundamentales son la unidad central de proceso o CPU, y las interfaces de entrada y salida. La CPU es el cerebro del PLC y está formado por el procesador y la memoria. El procesador se encarga de ejecutar el programa escrito por el usuario, que se encuentra almacenado en la memoria. Además el procesador se comunica con el exterior mediante sus puertos de comunicación y realiza funciones de autodiagnóstico. La interfaz de entrada se ocupa de adaptar las señales provenientes de los elementos captadores y sensores, tales como botoneras, llaves, límites de carrera etc, a nivel que el CPU pueda interpretar como información. Por otra parte, cuando la CPU resuelve, a través de un programa interno, activar algún elemento de campo, la interfaz de salida es la encargada de administrar la potencia necesaria para comandar el actuador.

### Relé lógico programable (PLR)

En los últimos años, unos pequeños productos llamados relés lógicos programables (PLR), y también por otros nombres similares, se han vuelto más comunes y aceptados. Estos son muy similares a los PLC, y se utilizan en la industria ligera, donde sólo unos pocos puntos de entrada/salida (es decir, unas pocas señales que llegan desde el mundo real y algunas que salen) están involucrados, y el bajo costo es deseado. Estos pequeños dispositivos se hacen típicamente en un tamaño físico y forma común por varios fabricantes, y con la marca de los fabricantes más grandes de PLCs para completar su gama baja de producto final. La mayoría de ellos tienen entre 8 y 12 entradas digitales, 4 y 8 salidas discretas, y hasta 2 entradas analógicas. El tamaño es por lo general alrededor de 10 cm de ancho y 7,5 cm de alto y 7,5 cm de profundidad. La mayoría de estos dispositivos incluyen una pantalla LCD de tamaño pequeño para la visualización simplificada lógica de escalera (sólo una porción muy pequeña del programa está visible en un momento dado) y el estado de los puntos de E/S. Normalmente estas pantallas están acompañadas por una botonera basculante de cuatro posiciones más cuatro pulsadores más separados, y se usan para navegar y editar la lógica. La mayoría tienen un pequeño conector para la conexión a través de RS-232 o RS-485 a un ordenador personal para que los programadores pueden utilizar simples aplicaciones de Windows para la programación en lugar de verse obligados a utilizar la pantalla LCD y el conjunto de pequeños pulsadores para este fin. A diferencia de los PLCs regulares que son generalmente modulares y ampliables en gran medida, los PLRs son por lo general no modulares o expansibles, pero su precio puede ser dos órdenes de magnitud menos de un PLC y todavía ofrecen un diseño robusto y de ejecución determinista de la lógica. En los últimos años se está incluso incorporando en estos pequeños dispositivos, una conexión de red Ethernet con RJ45 que permite configurar y monitorizar el equipo de forma remota.

### Ventajas y Desventajas

Dentro de las ventajas que estos equipos poseen se encuentra que, gracias a ellos, es posible ahorrar tiempo en la elaboración de proyectos, pudiendo realizar modificaciones sin costos adicionales. Por otra parte, son de tamaño reducido y mantenimiento de bajo costo, además permiten ahorrar dinero en mano de obra y la posibilidad de controlar más de una máquina con el mismo equipo. Así como soportar las vibraciones mecánicas generadas por la maquinaria ya que otros dispositivos serían altamente frágiles o propensos a fallas o rupturas.

Sin embargo, y como sucede en todos los casos, los controladores lógicos programables, o PLCs, presentan ciertas desventajas como es la necesidad de contar con técnicos calificados específicamente para ocuparse de su buen funcionamiento y mantenimiento.

### Otros usos

Hoy en día, los PLC no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control.





Figura 6: PLC

#### 1.4.4. Sistemas Embebidos

Son sistemas electrónicos diseñados para realizar pocas funciones en tiempo real, diseñados para cubrir necesidades específicas.

La mayoría de sus componentes se encuentran incluidos en una placa base.

Ejemplos: sistema de control de acceso, electrónica de una máquina expendedora, control de un taxímetro, control de una fotocopiadora o lavarropas y otros electrodomésticos. Son muy utilizados en control de máquinas como tornos. Se programan con lenguajes como CNC y otros.

#### 1.5. Diseño Básico de un Sistema de Procesamiento de Datos

En la Unidad 1 se abordó el tema de sistemas como un conjunto de partes interrelacionadas para cumplir con un objetivo. ¿Cuál es ese objetivo, de modo general? Recibir la ENTRADA de datos y transformarlos en SALIDA de información, la cual ha de ser veraz, oportuna y relevante.



Figura 7: Esquema del procesamiento de datos

El sistema de procesamiento de datos está compuesto por 3 (tres) subsistemas y 3 (tres) interfaces:

- *Subsistemas*: CPU + Memoria Principal + Unidad de Entradas y Salidas
- *Interfaces*: comprende la interconexión de los subsistemas, que se realiza por medio de buses.

## 2. Los módulos y sus interconexiones en el Modelo de Von Neumann

### 2.1. Buses

Un bus es una agrupación de cables que actúan en forma conjunta para interconectar las distintas partes de la computadora. Es una estructura de conexión para comunicar unidades o módulos dentro del ordenador. Son conductores como cables o líneas que interconectan eléctricamente las patas de los chips de las unidades o módulos. Es un sistema digital que transfiere datos entre los componentes de una computadora o entre computadoras, formado por cables o bien, por pistas en un circuito impreso, que puede incluir resistencias y condensadores, además de

circuitos integrados. La finalidad de los buses es reducir el número de rutas necesarias para la comunicación entre los distintos componentes, utilizando un solo canal de datos.

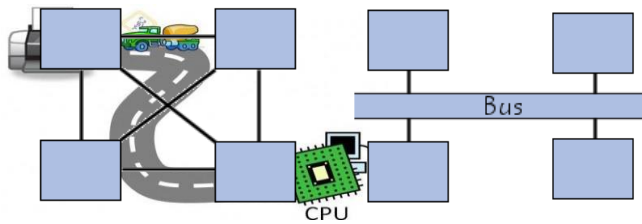


Figura 8: Ejemplo de BUSES

## 2.2. Formas de transmisión

**a) Transmisión en serie:** Los bits circulan por el conector o bus en “fila india”, es decir uno detrás de otro. Esta forma de transmisión funciona correctamente tanto para distancias cortas o largas. Ésta es la forma de transmisión usada en Internet. También puede utilizarse en la comunicación de periféricos con la computadora. Los bits generalmente entran o salen a través del conocido puerto USB (Universal SERIAL Bus).

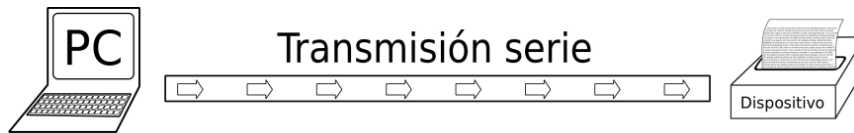


Figura 9: Transmisión en serie (aunque en la figura sólo se ejemplifica un sentido, puede realizarse en ambos).

**b) Transmisión en paralelo:** Transmite varios bits (tantos como una palabra) “uno al lado del otro”. Surge aquí el concepto de “palabra de memoria” como cantidad de bytes que se leen o graban en una sola operación de lectura o escritura. La transmisión en paralelo es más rápida que la transmisión serie, aunque sólo se puede utilizar confiablemente en distancias cortas. En caso contrario, los bits que forman la palabra se “desemparejan”. Dentro de la computadora se transmite “en paralelo” debido a la cercanía entre origen y destino. Esta cercanía permite mantener el paralelismo entre los bits que forman el byte o la palabra.

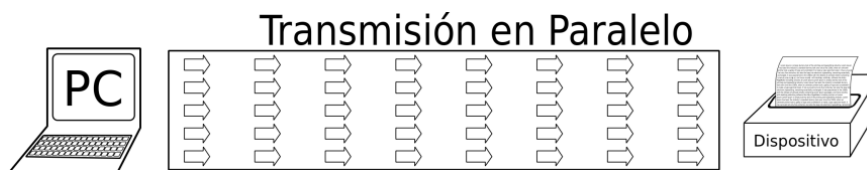


Figura 10: Transmisión en paralelo (aunque en la figura sólo se ejemplifica un sentido, puede realizarse en ambos).

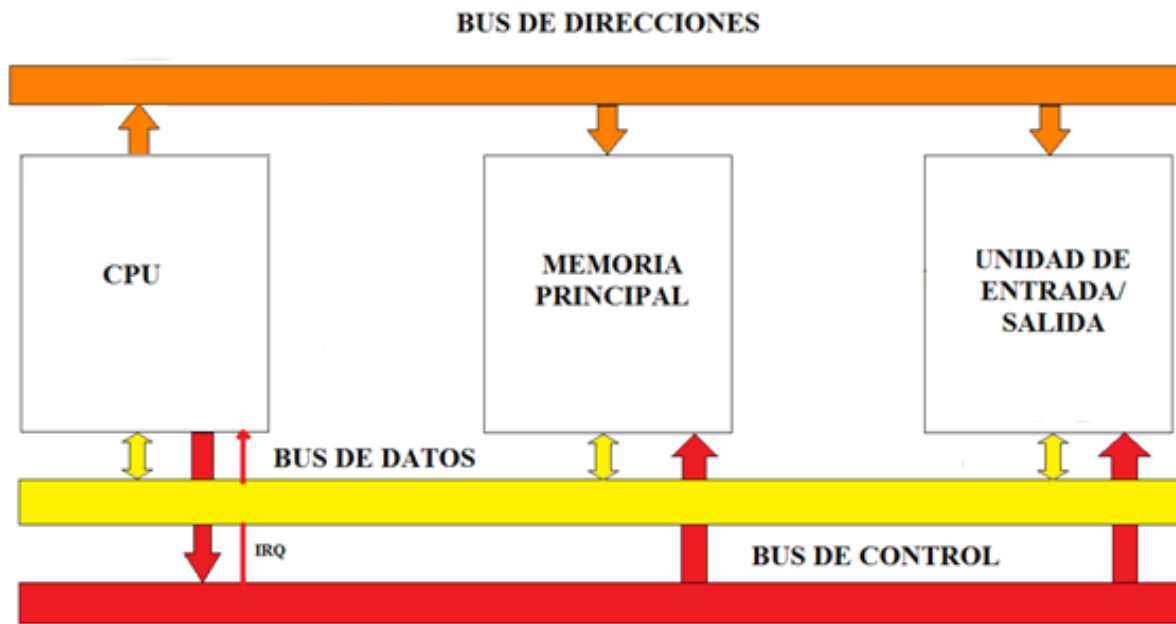
Dentro de la computadora, tal como se vio anteriormente, se usan buses para la comunicación y dicha comunicación es en paralelo. A largas distancias aumenta la probabilidad de errores y se usa la transmisión en serie.

## 2.3. Buses internos de una computadora

Existen tres tipos de buses: 1) Bus de Direcciones; 2) Bus de Datos; 3) Bus de Control.

En la figura 11 (que se presenta a continuación) puede notarse que el bus de direcciones es unidireccional, el bus de datos (e instrucciones) es bidireccional y el bus de control es

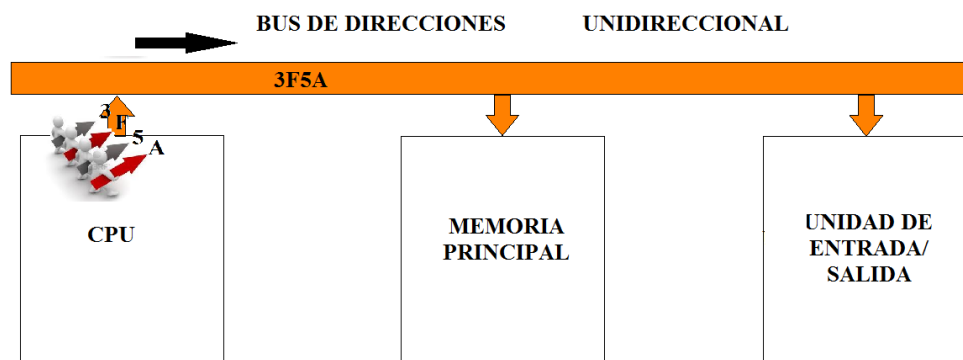
unidireccional excepto para el caso de las señales de interrupción (IRQ) ó acceso directo a memoria (DMA).



**Figura 11:** Los bloques del modelo de Von Neumann y sus interconexiones a través de los tres buses principales.

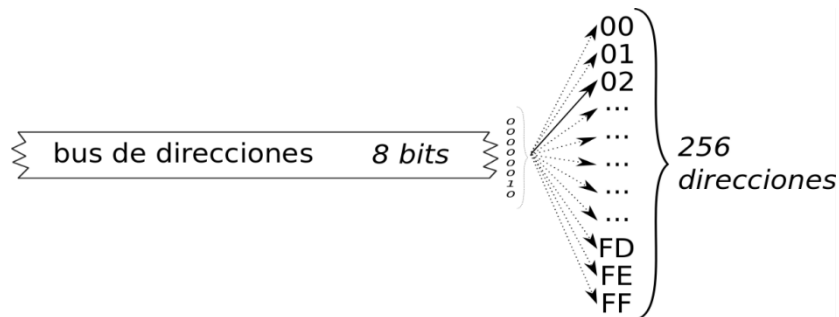
Existen dos formas de transmitir según el recorrido de los bits: UNIDIRECCIONAL, cuando éstos siempre circulan en el mismo sentido o BIDIRECCIONAL cuando lo realizan en ambos sentidos.

**a) Bus de Direcciones:** Es utilizado por la CPU para indicar a la Memoria Principal o a la Unidad de E/S las direcciones donde escribir o leer. El bus de direcciones, como su nombre lo indica, conduce “direcciones” desde la CPU hacia la MP o E/S. Siempre el flujo de información comienza en la CPU y llega a MP o E/S, por eso se dice que el bus de direcciones es un bus UNIDIRECCIONAL.



**Figura 12:** Detalle del bus de direcciones. Las direcciones (por ejemplo, “3F5A” expresada en hexadecimal) “parten” siempre de la CPU y se dirigen hacia la MP o hacia E/S.

La cantidad de líneas del bus de direcciones determina la cantidad de direcciones de memoria que una CPU puede direccionar. Por ejemplo, si el bus de direcciones tiene 8 bits, se puede direccionar una memoria de  $2^8 = 256$  direcciones diferentes (cantidad de combinaciones de 8 bits =  $2^8$ ), por simplicidad se expresan esas direcciones con 2 dígitos hexadecimal en la figura 13.



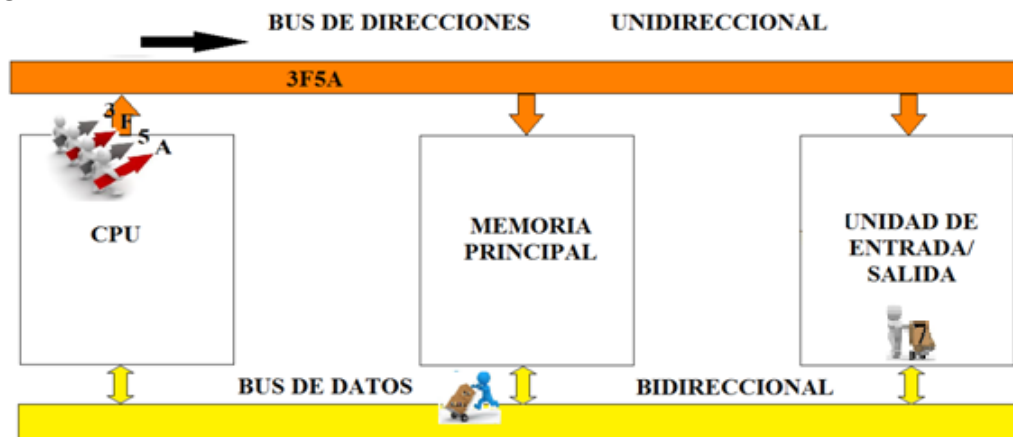
**Figura 13:** Ejemplo de direccionamiento con un bus de direcciones de 8 bits (8 cables).

Si, por ejemplo, el bus de direcciones es de 32 bits (32 cables), entonces, la cantidad de direcciones distintas a las que permite acceder es  $2^{32}$ .

$2^{32} = 2^2 \cdot 2^{30}$  y  $2^{30}$  equivale a 1 Giga, entonces:  $2^{32} = 4$  Giga direcciones de memoria.

Por lo tanto, con un bus de direcciones de 32 bits se puede acceder a 4 Giga direcciones distintas. Luego si en cada dirección se cuenta con un espacio de almacenamiento de por ejemplo 8 bits. La memoria tendrá una capacidad de 4GB (4 GigaBytes).

**b) Bus de Datos:** Permite transferir datos e instrucciones entre la CPU y el resto de los componentes (MP y E/S). En los datos transferidos por este bus también incluimos a las *instrucciones* de los programas. Es un bus **bidireccional**, dado que los datos pueden ir desde y hacia la CPU.



**Figura 14:** El bus de datos puede dibujarse en la parte inferior del diagrama de bloques y es bidireccional. A través del bus de datos viajan datos e instrucciones, en ambos sentidos (bidireccional).

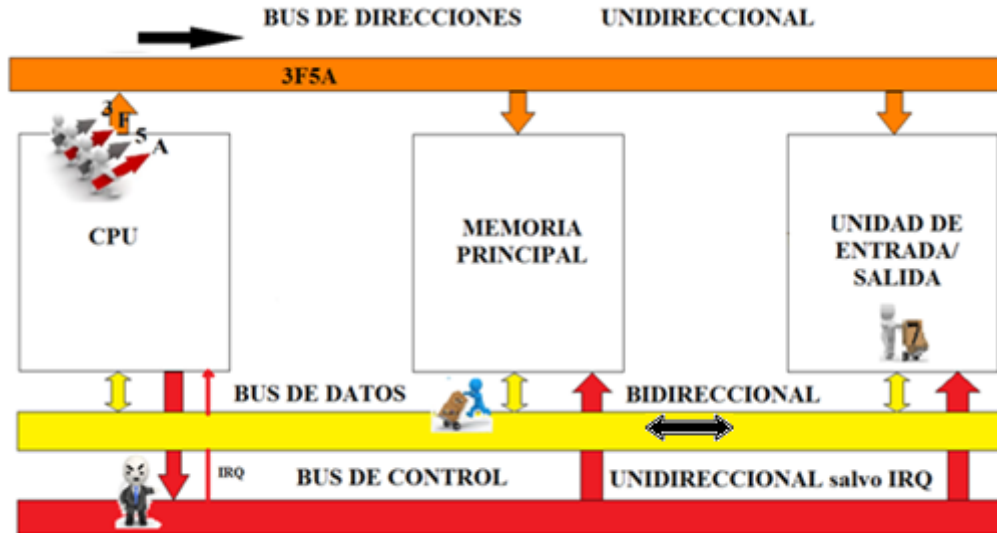
La CPU tiene que indicarle tanto a la MP como a la unidad de E/S las direcciones de dónde leer o escribir. Pero, ¿cómo se distingue entre direcciones para MP o de E/S? Se pueden utilizar buses de direcciones *aislados* (independientes) o *mapeados*. Este tema se desarrollará en secciones posteriores.

**c) Bus de Control:** Es un conjunto de cables que transmite los *microcomandos* desde la CPU hasta el resto de los componentes. Cada línea del bus transmite una señal específica. Cada línea es unidireccional. Las señales de control dependen del modelo de CPU y suelen incluir: *leer*, *escribir en la memoria* y *clock*.

Dentro de las líneas de control se encuentra la línea de IRQ (Interrupt Request), que significan *petición de interrupción*. Esta señal se utiliza para que los dispositivos puedan interrumpir el procesamiento de la CPU, para que sean atendidos ante algún evento. Por ejemplo, cuando un módem recibe datos a través de la red, éste interrumpe a través de la IRQ; el procesador deja lo

que estaba haciendo y ejecuta un programa para atender la interrupción.

En síntesis, el bus de control transporta las órdenes emanadas de la UC de la CPU, llamadas “microcomandos”, que llegan a la MP y E/S. Es unidireccional. Sin embargo existe el caso de las llamadas INTERRUPTACIONES (IRQ = Interrupt Request), donde MP o E/S pueden enviar un pedido de interrupción para ser atendido por CPU. Por ejemplo, la MP interrumpe para indicar lectura efectivizada, o la E/S lo hace para avisar que la impresora se quedó sin tinta.



**Figura 15:** El bus de control transmite microcomandos emitidos siempre por la Unidad de Control (UC) situada en la CPU.

Recordemos que la UC gobierna el funcionamiento del sistema. El bus de control es, por este motivo, unidireccional. Sin embargo, como excepción, existen señales (como por ejemplo: IRQ) que permiten interrumpir a la CPU.

### 3. El concepto de Memoria en Informática

La función de cualquier tipo de memoria es **ALMACENAR** datos, programas e información.

Se puede pensar como un gran contenedor, armario, mochila, cuaderno, libro, etc. La memoria **NO PROCESA** información, sólo la almacena. Por lo tanto, las acciones que se pueden realizar con cualquier memoria son: **LEER** (CARGAR) y/o **ESCRIBIR** (GUARDAR, GRABAR, ALMACENAR).

En todos los casos, la **LECTURA**, produce la acción de “duplicar” el contenido leído, está en el libro y en la mente del lector. Lo contrario ocurre con la **ESCRITURA**, ya que ésta es destructiva: “pisa”, sobreescribe, borra lo anteriormente escrito y queda la última versión.

La Memoria de una computadora se clasifica, de acuerdo a la forma de manejar a la información, en: (1) **ROM** (*Read Only Memory*): Memoria de Sólo Lectura. (2) **RWM** (*Read and Write Memory*): Memoria de Lectura Escritura.

#### 3.1. Memoria ROM

La Memoria ROM (*Read Only Memory*) es la Memoria de Sólo Lectura. Es decir, es una memoria creada para que el sistema (y el usuario) **SÓLO PUEDA LEERLA, NO ESCRIBIRLA.**

Se pueden pensar ejemplos de la vida cotidiana donde se utilizan memorias tipo ROM. Un cartel con el nombre de una calle, comercio, escuela; una frase escrita en un monumento, una

inscripción en una remera, un tatuaje. Todos estos ejemplos fueron creados para ser solamente leídos, pero NO para volver a escribirlos.

Las computadoras poseen su propia Memoria ROM. Ésta se utiliza para almacenar la información esencial necesaria para el **arranque** de la computadora, acción denominada *booteo* o *bootear* (bootstrap) y realizar los diagnósticos iniciales. *Bootstrap* es, literalmente, una presilla de cuero ubicada en la parte interna superior de la caña para ayudar a ponerse las botas. En inglés se lo toma metafóricamente como una serie de procesos ejecutados por uno mismo sin ayuda externa.

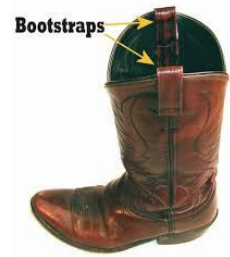


Figura 16: Booteo

No es posible guardar esta información imprescindible para la inicialización en el disco rígido, ya que dentro de esta información contenida en la memoria ROM, se encuentran los *propios parámetros* del disco rígido. Es por ello que al proceso de arranque se conoce con el nombre de booteo, el cual entre otras cosas permitirá acceder a leer en el disco rígido u otra memoria secundaria (por ejemplo, un pendrive) el núcleo del sistema operativo, lo carga en la memoria Memoria Principal y lo ejecuta. La información y los programas contenidos en la memoria ROM son:

- a) BIOS (Sistema Básico de Entrada/Salida)
- b) Configuración de parámetros del sistema
- c) POST (Autoprueba de Encendido)

A continuación se explican estos tres elementos:

a) El BIOS (Basic Input Output System = Sistema Básico de Entrada Salida), es un conjunto de programas que permite controlar las principales interfaces de entrada-salida. Se lo denomina BIOS ROM y físicamente está almacenado en un chip que se puede visualizar insertado en la placa madre o placa principal (*motherboard*) de la computadora. Al encender la computadora, el BIOS lee la información almacenada en el CMOS y utiliza los valores leídos para configurar el equipo.



Figura 17: Distintos tipos de BIOS.

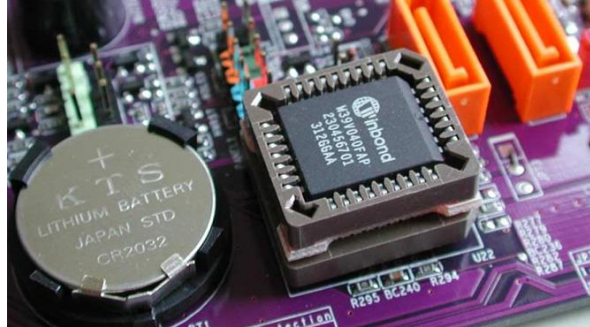
b) **Configuración de parámetros del sistema - CMOS (Complementary Metal–Oxide–Semiconductor)**<sup>7</sup> alude a la pantalla que se visualiza al iniciarse el ordenador. Se utiliza para modificar los parámetros del sistema (erróneamente llamada *BIOS*). Para utilizarla se accede a una memoria de acceso aleatorio de bajo consumo de corriente llamada RAM CMOS que funciona

<sup>7</sup> En castellano, *semiconductor complementario de óxido metálico*: es el tipo de material usado en la construcción del chip.



con una batería de larga duración incorporada, lo que le permite conservar su contenido incluso con el equipo apagado. En el CMOS se encuentra la siguiente información:

- Información básica de configuración, que incluye una utilidad de configuración automática que detecta los parámetros correctos de la unidad de disco rígido.
- Configuración para ajustar el rendimiento del sistema.
- Selección de características de ahorro de energía.
- Configuración de la contraseña para el hardware.
- Configuración de los periféricos integrados.



**Figura 18:** CMOS (en socket removible) con su batería.

c) **Auto-prueba de Encendido** (POST = *Power On Self Test*) es un programa que se ejecuta automáticamente cuando arranca el sistema. Se encarga de verificar e inicializar los componentes de entrada/salida y configurar y diagnosticar el estado del hardware (por ejemplo, chequeo u operación de “contar” la RAM en el inicio, verificar el disco rígido, entre otras funciones).



Debido a que las memorias ROM son mucho más lentas que las RAM, las instrucciones suministradas en la ROM a veces se copian a la RAM en el inicio; proceso denominado *respaldo*, aunque a menudo se le llama *memoria de respaldo*). La misma batería ejecuta el reloj en tiempo real incorporado cuando el equipo está apagado; habitualmente este reloj está incrustado en el chip del CMOS.

## Tipos de ROM

Existen distintos tipos de ROM, que van evolucionando continuamente<sup>8</sup>.

<sup>8</sup> <http://miguel301sistemasoperativos.blogspot.com.ar/2012/10/unidad-central-de-proceso.html>



DENOMINACIÓN DE LA ROM	DESCRIPCIÓN
ROM (propiamente dicha)	Memorias de sólo lectura originales que escribían información binaria sobre una placa. Ya no se utilizan.
PROM ( <i>Programmable Read Only Memory</i> ) o Memoria Programable de Sólo Lectura	Fueron desarrolladas por TI (Texas Instruments). Se pueden programar una sola vez. Están formadas por miles de fusibles que se pueden quemar (representan así el 0), aplicando una alta tensión. Los fusibles que no se queman representan el 1.
EPROM ( <i>Erasable Programmable Read Only Memory</i> ) o Memoria Programable y Borrable de Sólo Lectura	Son memorias ROM Programables que se pueden borrar. Este proceso de borrado se realiza utilizando luz ultravioleta que reconstituye los fusibles (se trata, en este caso, de componentes especiales) y los vuelve a poner todos en 1.
EEPROM ( <i>Electrically Erasable Programmable Read Only Memory</i> ) o Memoria Programable de Sólo Lectura, Borrable Eléctricamente	Son memorias PROM borrables, pero a diferencia de las anteriores, se pueden borrar mediante el pasaje de corriente eléctrica, aun cuando se encuentren en el ordenador. Utilizan dos o tres transistores por bit. Puede escribirse y leerse por palabra.
FLASH ROM o FLASH EPROM	Son una variante de las EEPROM que utilizan un solo transistor por bit, lo que las convierte en memorias con mayor capacidad para un mismo tamaño. Se borran por página. El tema será ampliado en secciones posteriores.

### 3.2. Memoria de Lectura Escritura (RWM)

RWM significa READ WRITE MEMORY, es decir Memoria de Lectura y Escritura. Como ejemplos conceptuales, podríamos pensar en el pizarrón, un cuaderno, etc. En este tipo de memoria es posible LEER (READ, LOAD) lo que está escrito Y **también** ESCRIBIR (WRITE, SAVE) en ella. Las computadoras poseen una memoria con estas características (RWM) cuya función es ALMACENAR PROGRAMAS, DATOS E INFORMACIÓN MIENTRAS SE ESTÁ PROCESANDO. Este tipo de Memoria (llamada popularmente Memoria RAM), tiene como característica que al apagarse la computadora, pierde su contenido. Por este motivo se dice que es VOLÁTIL. Será explicada en secciones posteriores.

### 3.3. Modos de acceso a la información

Imagine la siguiente situación: El docente pide a los alumnos que cada uno escriba su apellido en una tarjeta. Luego recoge todas las tarjetas en una bolsa y dice... “– Voy a extraer el nombre del alumno que va a pasar a resolver el ejercicio”.

Analicemos la situación:

1. ¿Se sabe, a priori, qué alumno será seleccionado? NO. Todos tienen igual chance de ser elegidos.
2. ¿Se tarda el mismo tiempo en extraer cualquier papelito? SÍ. El tiempo de “acceso” a cualquier papelito es el mismo.

Por lo tanto, el alumno cuyo apellido comienza con A y el alumno cuyo apellido comienza con Z tienen la misma chance de ser seleccionados y el mismo tiempo de extracción de su apellido.

A este tipo de acceso a la información se lo denomina RAM (Random Access Memory), que significa *Memoria de Acceso Aleatorio o Al Azar*.

Este tipo de acceso es el utilizado en los juegos de azar como el bingo, la ruleta, los juegos de naipes.

Por el contrario, si un Profesor “toma lista”, los alumnos cuyos apellidos comienzan con las primeras letras del abecedario, saben que serán nombrados antes que aquellos cuyos apellidos comienzan con las últimas letras. En este caso, el tiempo de acceso al apellido, no es el mismo. Este tipo de acceso se denomina SECUENCIAL. La memoria principal de la computadora tiene ACCESO ALEATORIO (RAM), ya sea para la memoria de sólo lectura ROM, como para la memoria de lectura y escritura RWM.

### 3.4. La Memoria Principal

La Memoria Principal de una computadora, también conocida como *memoria de usuario* o simplemente RAM, está compuesta por un conjunto de dispositivos electrónicos capaces de almacenar bits. Dichos dispositivos se agrupan de a 8 bits, denominados BYTE. 1 BYTE = 8 bits.



Figura 20: Integrado de Memoria RAM<sup>9</sup>

Cada posición de memoria, posee una **dirección** (expresada usualmente en hexadecimal) y un **contenido** que son los bits almacenados en esa dirección.

A la memoria principal se accede por DIRECCIÓN para leer o grabar su contenido.

Se puede pensar a la memoria principal como un casillero, donde cada casilla posee una dirección y tendrá algo almacenado en ella. Es de destacar que, si bien en la figura 21 se muestran las direcciones de memoria, en la práctica esas direcciones NO SE ALMACENAN, sin embargo, se conoce su ubicación y se accede a su contenido.

DIRECCIÓN	CONTENIDO
3F5A	11001100
3F5B	11110000
3F5C	10000001
3F5D	10111111
3F5E	00001111
3F5F	00001001
3F60	00000111
....	.....

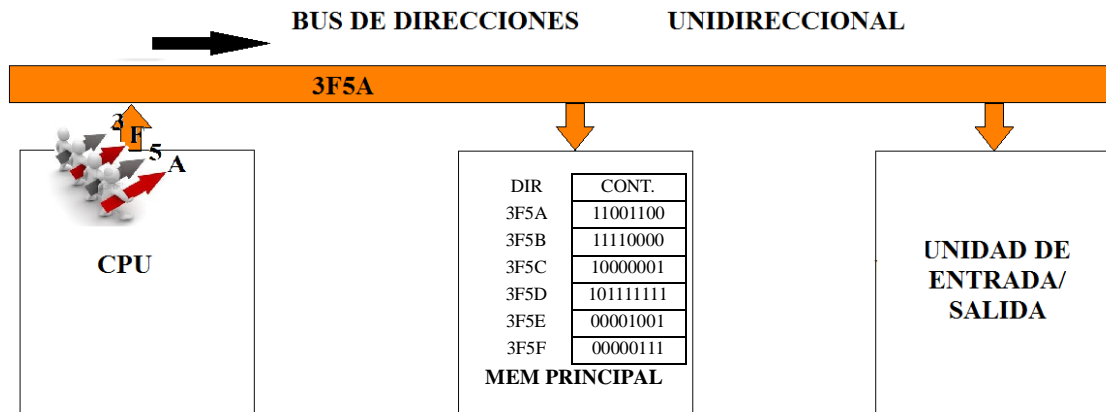
Figura 21: Una parte de la memoria principal, con sus direcciones y contenidos.

La cantidad de bits almacenados en cada dirección depende del diseño del microprocesador. La mayoría de las computadoras direccionan por BYTE, es decir, el contenido de cada posición de memoria es 1 BYTE (8 bits). Retomando el concepto de “palabra de memoria”: cantidad de bytes que se leen o graban en una sola operación de lectura o escritura. Por ejemplo, dependiendo del procesador, existen palabras de 8 bits (1byte), 16 bits (2 bytes), 32 bits (4 bytes), 64 bits (8bytes) ya que existen otras computadoras que direccionan por palabra, en este caso cada posición de memoria contiene 2 bytes (palabra de 16 bits), 4 bytes (palabra de 32 bits), 8 bytes (palabra de 64 bits). Existen computadoras de palabra fija, las cuales utilizan una cantidad de bits (bytes) para

<sup>9</sup> <http://es.slideshare.net/patan005/memoria-ram-9818549?related=1>

cada palabra, como en los ejemplos dados. Estas computadoras son más eficientes a la hora de realizar cálculos, si bien “desperdician” bits si el dato es de poca longitud (ej: con palabras de 16 bits, para almacenar el número 9, se utilizan 16 bits, cuando sólo se necesitan 4 bits: 1001). Otra organización son las computadoras de “palabra variable”, que almacenan cada carácter (8 bits), en posiciones de memoria distintas. Utilizan más eficientemente la memoria disponible, pero es más difícil realizar cálculos.

En la figura 22 se muestra la memoria principal ubicada en la arquitectura de Von Neuman.



**Figura 22:** Ubicación de la Memoria Principal dentro del Sistema.

Las memorias RAM se las puede clasificar en dos grupos:

- SRAM. Estáticas, aquellas que conservan su contenido mientras estén conectadas a una fuente de energía eléctrica, mencionamos como ejemplos los registros de la CPU y la memoria CACHE. Esta última se trata de una memoria mucho más rápida que la principal, construida con elementos de muy alta velocidad, de alto costo, por lo cual hablamos de pocos componentes (y cuya capacidad instalada es inferior a la RAM Dinámica). La CPU obtiene de ella partes de programas utilizados frecuentemente en los procesos, para no perder tiempo accediendo a Memoria Principal.
- DRAM. Dinámicas: aquellas en las cuales se “refresca la información” para no perder la misma. Esta acción se realiza en la lectura de memoria. Ejemplo: la memoria principal.

Las Memorias DRAM son económicas que las SRAM.

Dentro de los tipos de Memoria disponibles dentro de una computadora existen:

- Registros de CPU: Son una memorias de muy alta velocidad y muy poca capacidad (por ejemplo 32 bits), que se encuentran integrados en el microprocesador y mantienen estable un valor en forma transitoria (por ejemplo el Registro Acumulador del Modelo de Von Neumann). Serán explicados en apartados posteriores.
- Memoria Caché: Es una memoria de acceso muy rápido (SRAM) y de tamaño más pequeño que la Memoria Principal<sup>10</sup> (que se sitúa entre la Unidad Central de Proceso y la Memoria Principal (DRAM)). Almacena temporalmente los datos recientemente procesados para reducir el tiempo de acceso a la Memoria Principal.
- Memoria auxiliares, masivas o secundarias que serán tratadas en la sección de Periféricos de Entrada/ Salida.

<sup>10</sup> Por ejemplo: INTEL i7 con 25 M de cache, [https://ark.intel.com/es/products/94456/Intel-Core-i7-6950X-Processor-Extreme-Edition-25M-Cache-up-to-3\\_50-GHz](https://ark.intel.com/es/products/94456/Intel-Core-i7-6950X-Processor-Extreme-Edition-25M-Cache-up-to-3_50-GHz)

### 3.5 Jerarquías de Memorias

Debido a las diferentes tecnologías utilizadas en la construcción de los diferentes tipos de memorias que forman parte de una computadora, puede establecerse una Jerarquía de Memorias de acuerdo a distintos parámetros: Tiempo de acceso (tiempo transcurrido desde el momento que se presenta una dirección en Memoria hasta que su contenido sea leído o grabado), Capacidad de Almacenamiento, Costo, etc.

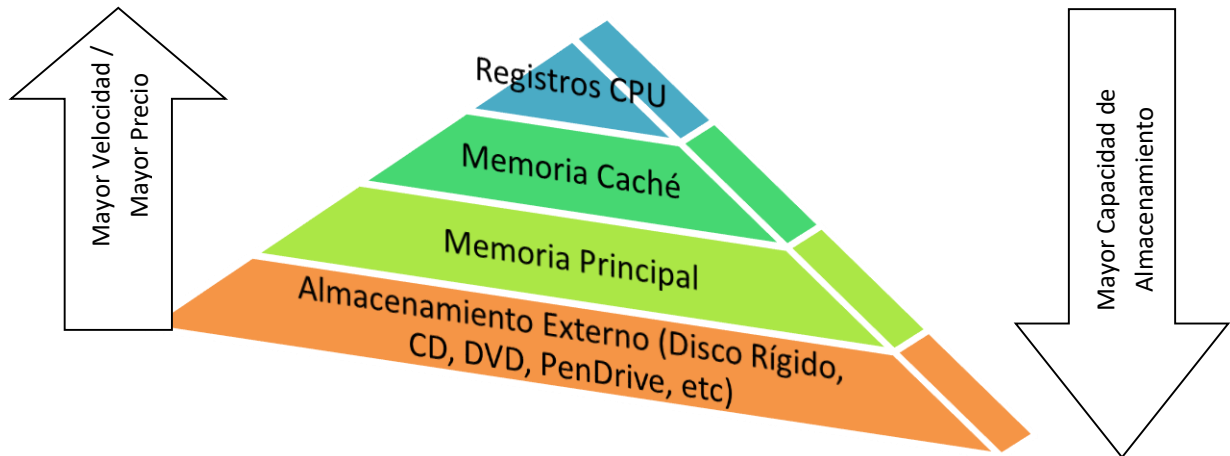


Figura 23: Jerarquía de Memorias.

### 3.6. Organización de Direccionamiento de Memoria y Entrada / Salida

Según hemos visto, la Memoria Principal y la Unidad de Entrada/ Salida poseen direcciones que se transmiten por el bus de direcciones. Pero... ¿Cómo se diferencia una dirección de Memoria Principal de una dirección de Entrada/ Salida?

Existen dos formas: (1) Sistema de Memoria Mapeada; (2) Sistema de Memoria Aislada o independiente (con buses físicos o buses lógicos). Los cuales se explican a continuación:

**a) Sistema de Memoria Mapeada:** En el modo mapeado, de todo el rango de direcciones posibles a conformar con los bits del bus de direcciones, parte de ellas serán destinadas a la memoria principal y otras para direccionar a los dispositivos de E/S (ver figura 24). Así, nos quedarán unos subrangos de direcciones para MP y otros para E/S.

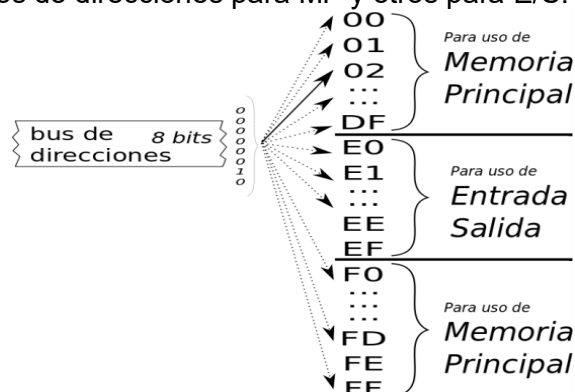
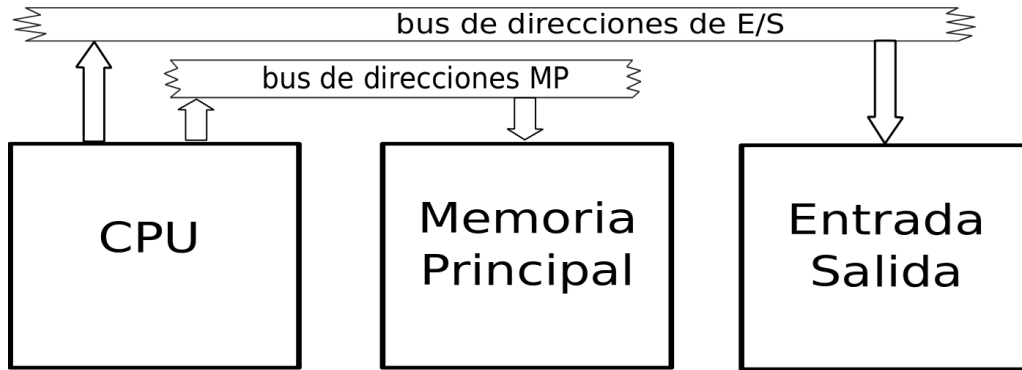


Figura 24: Mapa de Memoria para Memoria Mapeada.

Al colocarse una dirección el procesador envía la orden de lectura o escritura a MP o E/S dependiendo del valor de esa dirección, según el mapa de memoria del sistema.

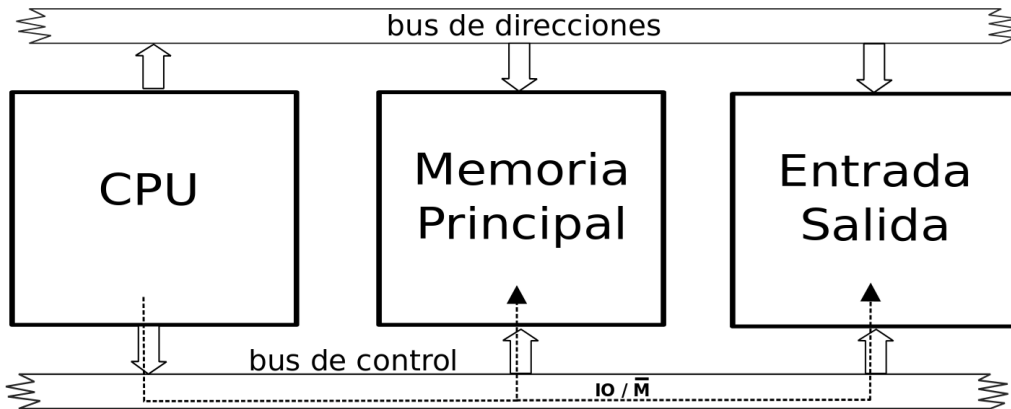
### c) Sistema de Memoria Aislada o Independiente:

- Con dos buses físicos: Un bus de direcciones para la MP y otro para E/S. En este caso de dos buses físicos, existe un conjunto de cables para la MP y otro conjunto de cables para E/S. La existencia de más cables es una evidente desventaja que incrementa el tamaño de la solución y sus costos.



**Figura 25:** Sistema aislado o independiente, con dos buses físicos.

- *Con dos buses lógicos:* En vez de tener dos buses físicos, podemos usar un solo bus de direcciones y una línea del bus de control para indicar *si la dirección es para la MP o para E/S*. En este caso tendremos *dos buses lógicos*. La línea del bus de control utilizada se suele llamar  $IO/\bar{M}$  (Input Output / Not Memory), como se muestra en la parte inferior de la figura 26. Si en esta línea se pone un “1”, indica que la dirección enviada a través del bus de direcciones es para E/S; y si se pone un “0”, es para Memoria Principal.



**Figura 26:** Sistema aislado o independiente, con dos buses lógicos.

## 4. La Unidad Central de Proceso (C.P.U.)

La Unidad Central de Proceso (*Central Processing Unit*, por sus siglas: CPU), es un microprocesador que se encarga de ejecutar las instrucciones de programa y controlar todo el sistema. Está compuesta por los siguientes elementos:

(1) Unidad de Control (UC); (2) Unidad Aritmética y Lógica (ALU); (3) Registros; (4) Buses Internos.

### 4.1. Registros físicos

Un **REGISTRO FÍSICO** es un conjunto de circuitos (en algunos casos, Flip Flops) que permite **mantener estable** en forma temporaria un conjunto de bits (dato, dirección, código de operación, índice, etc.) hasta que sea reemplazado (“pisado” o sobrescrito) por otro conjunto de bits. Este conjunto de bits puede ser leído o grabado en el registro. La cantidad de bits que almacena el registro depende del tipo del mismo.

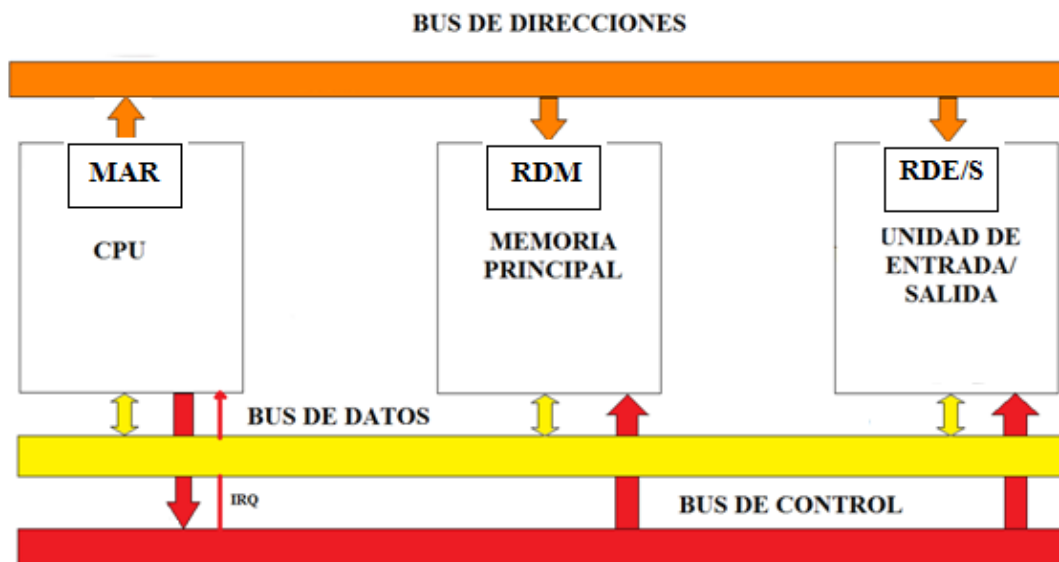
• **Registros que almacenan direcciones:**

– *En la CPU:* **MAR** (Memory Address Register). Es un registro que contiene y mantiene estable la **dirección** de MP o de E/S, a la cual se desea acceder para leer o grabar.

– *En la MP:* **RDM** (Registro de Direcciones de Memoria). Contiene y mantiene estable la **dirección** de Memoria Principal en la cual se desea leer o grabar.

– *En la Unidad de E/S:* **RDE/S** (Registro de Direcciones de Entrada/Salida). Contiene y mantiene estable la **dirección** de Entrada/Salida (recordemos que esta dirección representa un dispositivo periférico) en la cual se desea leer o grabar.

Las direcciones contenidas en estos registros viajan, en todos los casos, por el **bus de direcciones**. Siempre parten de la CPU y llegan a MP y a E/S (como se ha explicado, este bus es unidireccional).



**Figura 27:** Los registros de direcciones: MAR, RDM y RDE/S. Se ubican en la parte superior de cada bloque del esquema, porque pertenecen al circuito de direcciones.

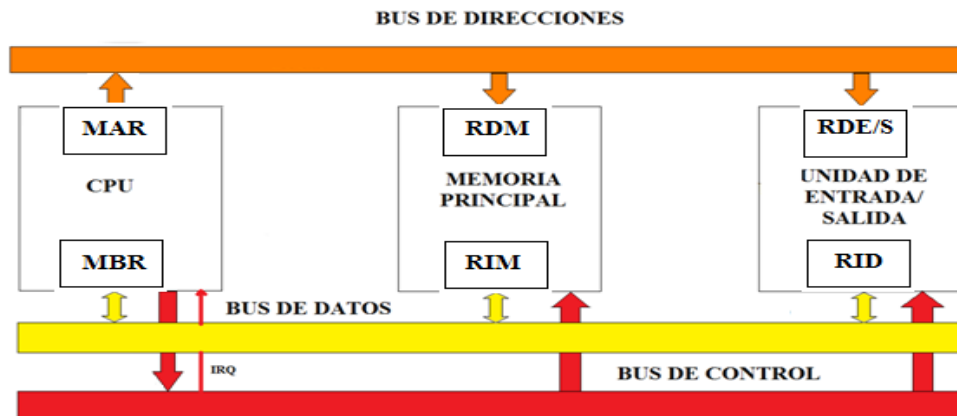
• **Registros que almacenan datos o instrucciones:**

– *En la CPU:* **MBR** (Memory Buffer Register). Es un registro que contiene y mantiene estable el dato o la instrucción que entra a la CPU para ser procesado, o bien el dato que sale de la CPU y se desea almacenar en MP o almacenar y/o mostrar en los dispositivos de E/S.

– *En la MP:* **RIM** (Registro de Intercambio de Memoria). Es un registro que contiene y mantiene estable el dato o la instrucción que se lee en MP, o bien el dato o instrucción que se desea grabar, en la dirección indicada en el RDM.

– *En la Unidad de E/S:* **RID** (Registro de Intercambio de Datos). Es un registro que contiene y mantiene estable el dato o la instrucción que se lee en un dispositivo de E/S o bien, el dato o instrucción que se desea grabar o mostrar en el dispositivo cuya dirección es la dirección indicada en el RDE/S.

Los datos o instrucciones contenidas en estos registros viajan, en todos los casos, a través del **bus de datos**, en ambos sentidos (como se ha mencionado, este bus es bidireccional).



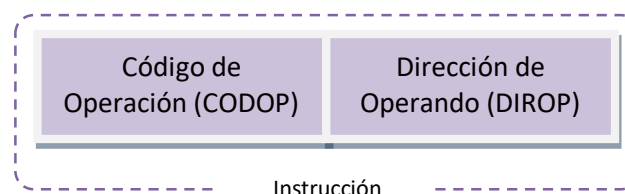
**Figura 28:** Los registros de datos e instrucciones: MBR, RIM y RID. Se ubican en la parte inferior de cada bloque del esquema, porque pertenecen al circuito de datos e instrucciones.

- **Registro ACUMULADOR (ACUM):** Almacena y mantiene estables los **datos** de entrada, que posteriormente serán procesados; y también almacena el resultado de la operación de la ALU que se está ejecutando.
- **Registro de ESTADOS (REst):** Contiene las SEÑALES de ESTADO o FLAGS (Signo, Carry, Overflow, Zero, Paridad, etc.) que determinan el estado del resultado, almacenado en la ALU (en el acumulador); y son utilizadas por la Unidad de Control para diversas funciones, entre ellas, saltos condicionales, finalización abrupta de programas (ejemplo: división por cero, etc.). Recordemos que dichas señales pueden tomar los valores 0 (ausencia del evento en cuestión) y 1 (si se produjo el evento). Ejemplo: la señal de Overflow estará en 1 si el resultado de la operación produjo overflow y 0 en caso contrario.

Nombre	S Signo	Cy Carry	Of Overflow	Z Zero	P Paridad
Valor de la señal	0	0	0	0	0

**Figura 29:** Representación esquemática del registro de estados y las señales de estado o *flags*.

- **Registro CONTADOR DE PROGRAMA (CP):** Contiene la dirección de la Próxima Instrucción de Programa que se va a ejecutar, por ese motivo, también es denominado RPI. Por sus siglas en inglés (Program Counter), puede llamárselo asimismo PC. Su funcionamiento será explicado más adelante, en el tema: Ciclo de Instrucción.
- **Registro de INSTRUCCIÓN (RI):** Está conectado a la Unidad de Control. Como su nombre lo indica almacena la instrucción que se está procesando. En la figura 30, se muestran las dos partes que contiene una instrucción.



**Figura 30:** Una instrucción está formada por el Código de Operación y la Dirección de Operando.



- **Registro Temporal de Operando (RTOp):** Contiene el dato (operando), con que se va a ejecutar la instrucción que se está procesando.

Existen otros registros que apuntan a direcciones de estructuras de datos en memoria, como los siguientes:

- **Registro PUNTERO DE PILA (SP, por su nombre en inglés: Stack Pointer):** Contiene la dirección del tope de la pila. Los datos que están almacenados en la pila, se encuentran almacenados en la MP (en definitiva pilas y colas son formas de organizar la memoria RAM).
- **Registro PUNTERO DE MEMORIA (MEMORY POINTER):** Contiene la dirección de memoria del elemento más alto de un arreglo de memoria (vector). Se utiliza cuando se desea acceder a un *conjunto de datos consecutivos* en memoria. Por ejemplo: un VECTOR.

Las “pilas” y los “vectores” son estructuras de datos que se utilizan generalmente para simplificar ciertas operaciones de programación. La PILA (en inglés, STACK), es una estructura de la información que se utiliza para guardar datos. Funciona como una “pila de platos”, en este caso, de datos. En ella, los datos se sitúan uno “arriba” de otro y la pila va creciendo. Cuando se desea quitar un dato, se extrae el último que se puso. La organización responde a la idea UEPS = “Último que Entra Primero que Sale” o LIFO = “Last In First Out”. El Registro PUNTERO DE PILA (STACK POINTER), contiene la dirección de memoria del TOPE ACTUAL de la pila.

En la figura 31, el tope de la pila está en la dirección de Memoria Principal C7. Para APILAR un dato se INCREMENTA el TOPE:  $C7 + 1 = C8$ . El nuevo dato apilado ocupa la dirección de memoria C8, que se convierte en el nuevo TOPE DE LA PILA.

Para DESEMPILAR un dato se DECREMENTA el TOPE:  $C7 - 1 = C6$ . La dirección de Memoria del TOPE DE LA PILA es ahora C6.

Ejemplo: Lectura de un Vector de 5 posiciones.

En este caso se agregan elementos al final del arreglo.

Puede pensarse como una lista, donde los datos se van agregando al final de la misma.



**Figura 31:** Representación de la estructura y las funciones de una pila.

## 4.2. Unidad de Control (UC)

Es el dispositivo alojado dentro de la CPU que es el verdadero cerebro del sistema, se encarga de **controlar** todo el sistema y emitir las órdenes denominadas “microcomandos”, necesarios para ejecutar cualquier programa y funcionamiento interno de la computadora.

Para realizar su tarea, la UC debe secuenciar las acciones que deben realizar los circuitos necesarios para ejecutar cada instrucción de un programa. Por ejemplo: leer o guardar datos en Memoria Principal, cargar, incrementar o decrementar registros de CPU, etc.

Clock (Ck) o reloj es el encargado de:

- Sincronizar las operaciones que realizan los diferentes componentes del sistema informático.
- Marcar la hora.

El Clock es un cristal piezo-eléctrico de cuarzo que genera millones de pulsos eléctricos por segundo que llegan a la UC que emite una cantidad de pulsos por segundo constantemente. El número de pulsos que emite el reloj cada segundo se denomina frecuencia y se mide en pulsos por segundo donde cada pulso es un Ciclo. La unidad de medida es el Herzio (Hertz). Como el Clock tiene una frecuencia de millones de pulsos por segundo, se utiliza el Megahertz para medir dicha frecuencia. La función del clock es marcar el ritmo de ejecución de los microcomandos, generando una señal periódica que es utilizada por todas las partes de la computadora para sincronizar y coordinar todas las operaciones dentro de la misma. Cuanto mayor frecuencia tiene el clock, el procesamiento es más rápido, ya que se ejecutan más microcomandos por segundo. Con cada pulso de reloj, van apareciendo microcomandos en el bus de control para procesar cada instrucción. Puede hacerse una analogía entre el clock y el patrón de una galera o un timonel de un bote. Ambos marcan el ritmo de remar. Cuanto más frecuente es la señal (BONG!, BONG!), más rápido reman los hombres (ver figura 32).



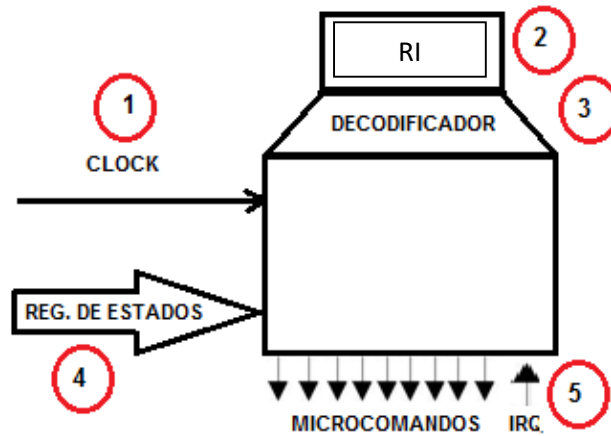
Figura 32: Analogías del CLOCK<sup>11</sup>

La UC no gobierna directamente a los periféricos mediante buses que llegan directamente a ellos. Para esta función, la CPU ejecuta un subprograma especial para cada periférico, que llega a la interfaz del mismo y le ordena que debe hacer.

La Unidad de Control funciona de la siguiente forma:

- 1) Está conectada al Registro de Instrucción, que contiene la instrucción que se está ejecutando.
- 2) Posee un **decodificador**, que, de acuerdo al CodOp (contenido en la instrucción), habilita el microprograma o circuito necesario para ejecutar la operación.
- 3) Recibe las señales del registro de estados (REst), que informan características del último resultado obtenido. Por medio de estas señales, la UC “toma decisiones”, ejecutando instrucciones condicionales, disparando distintos programas de acuerdo con el contenido del registro de estados. Por ejemplo, “*si la señal de Carry del REst está en 1, pasar a la dirección A123; si no, continuar*”.
- 4) Recibe la solicitud de interrupción (IRQ) que envían los periféricos y la señal de Reset.

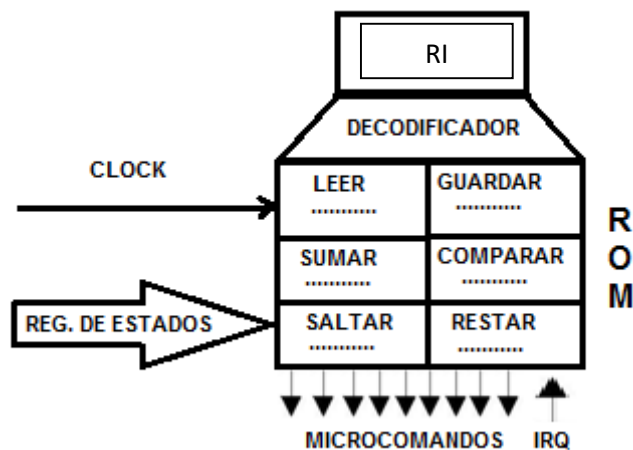
<sup>11</sup> Imagen extraída de <http://diccionarioasterix.blogspot.com.ar/2010/02/g.html>



**Figura 33:** La Unidad de Control

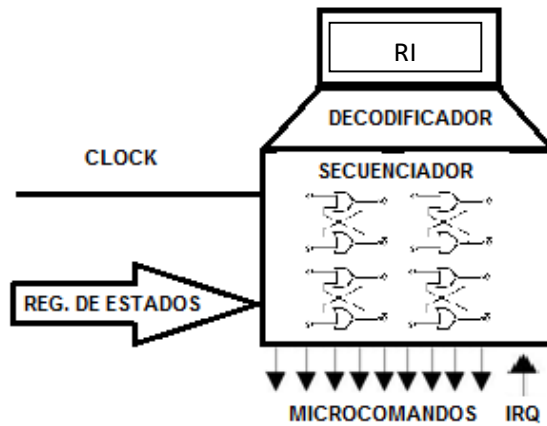
La construcción interna de la UC presenta dos tipos tradicionales: (a) *microprogramada* y (b) *cableada*. De acuerdo a cómo estén almacenados o implementados los microcomandos, la UC puede estar construida de varias formas, pero dentro de las más tradicionales se encuentran, como se mencionó, la microprogramada y la cableada.

- **Unidad de Control Microprogramada:** En este tipo de Unidad de Control, los microcomandos necesarios para procesar cada código de instrucción con que cuenta el procesador, se hallan grabados en una memoria tipo ROM o inclusive una PROM. El código de instrucción que será ejecutado, es decodificado y de acuerdo a ese código se accede a la zona de la ROM que contiene una secuencia de los microcomandos necesarios para procesarla, que se transmitirán en orden por el bus de control a las unidades o registros que intervendrán en el procesamiento. Si se trata de una PROM, se puede alterar el comportamiento de la Unidad de Control, y por lo tanto la forma de procesar. Este concepto se denomina FIRMWARE.



**Figura 34:** La Unidad de Control *microprogramada* está constituida, principalmente, por una ROM.

- **Unidad de Control Cableada:** Las UC cableadas están diseñadas con técnicas de optimización y construidas en un circuito integrado con circuitos secuenciales (flip flops), que permiten obtener alta velocidad y eficiencia.



**Figura 35:** La Unidad de Control *cableada* está formada por circuitos secuenciales.

Son más veloces, eficientes y se utilizan para sistemas en los que se requieren estas características. Son, por tal motivo, más costosas.

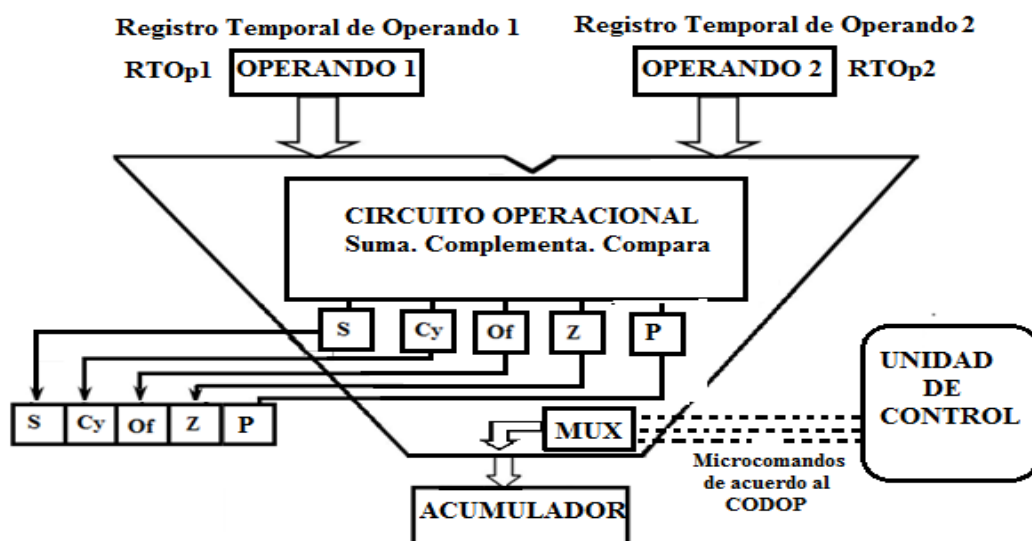
### 4.3. Unidad Aritmética y Lógica (ALU)

Es la unidad encargada de realizar las operaciones matemáticas y las comparaciones; y determinar las características del último resultado calculado. Se encuentra en la CPU.

Está compuesta por los siguientes módulos:

- Circuito Operacional.
- Circuitos para calcular las señales de estado y de usos múltiples.
- Multiplexor (MUX).
- Registro Acumulador
- Registros Temporales de Operandos RTOp.

El Circuito Operacional realiza las operaciones de suma, resta (por medio del complemento, Unidad 2) y complemento. Por medio de ellas realiza la función lógica de comparación.



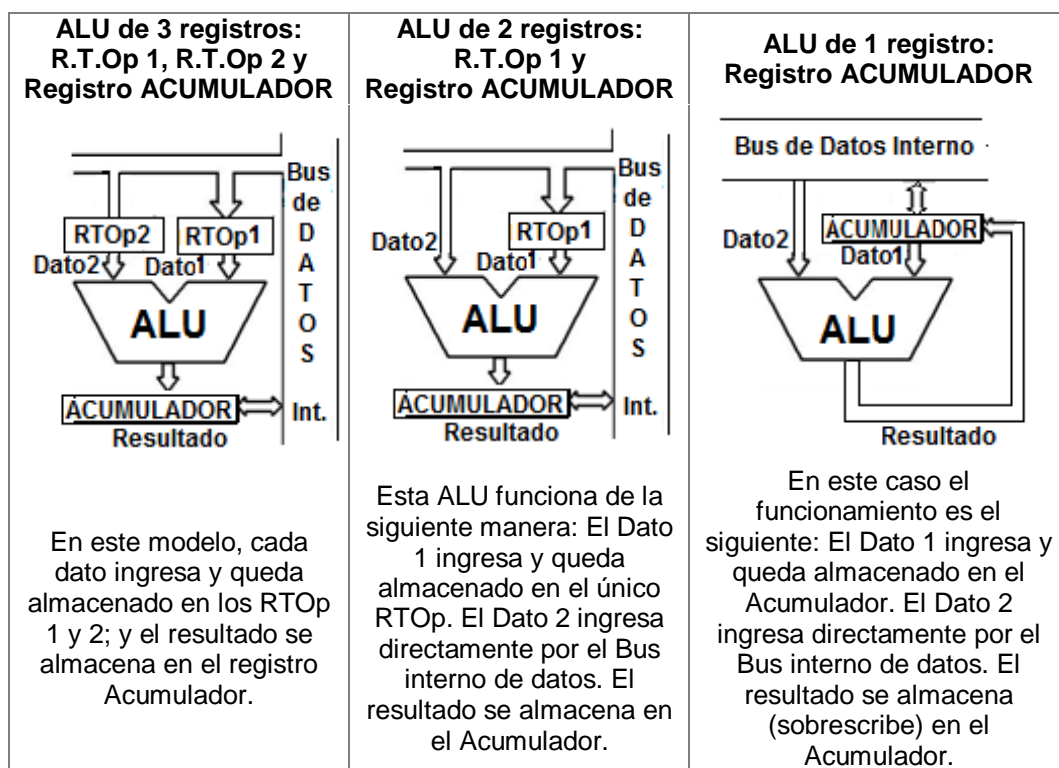
**Figura 36:** Constitución interna de la Unidad Aritmético Lógica (ALU).

La ALU recibe como entrada dos operandos: Operando 1 y Operando 2, que llegan a todos los bloques internos encargados de realizar las operaciones, a este componente de la ALU constituye el Circuito Operacional.

La Unidad de Control (UC) selecciona el tipo de operación requerida por la instrucción de programa que se está procesando, por medio de microcomandos de selección a las entradas del Multiplexor. De esta forma, sólo el resultado de la operación requerida se almacena en el registro ACUMULADOR.

Existen tres modelos de ALUs de acuerdo con la cantidad de registros que la componen.

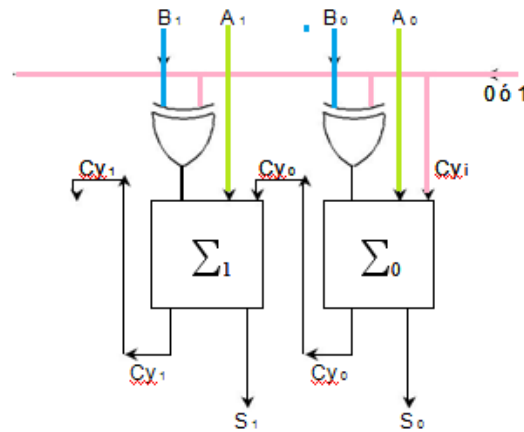
En dos de ellos se utiliza el Registro Temporal de Operando (R.T.Op.), que mantiene estable el dato que se ha de operar. En todos los casos, el resultado de la operación se almacena en el registro ACUMULADOR. Este registro también recibe operandos en el modelo de ALU de un solo registro.



**Figura 37:** Modelos de ALU.

• **Circuito operacional:** Este circuito es el encargado de realizar las operaciones de Suma, Resta (por medio de suma de complemento, Unidad 2), Complemento y Comparación.

Como es posible observar en la figura 38, el circuito operacional está compuesto por tantos sumadores totales, como bits tenga la palabra de memoria. Como fue estudiado anteriormente, cada sumador total tiene como entrada los dos bits que se desean sumar y el arrastre Carry (Cy) de la suma anterior. Es de destacar que el primer sumador correspondiente a los bits menos significativos de ambos operandos, también es un sumador completo, con tres entradas. ¿Por qué, si no hay Carry?



**Figura 38:** Sumador – restador del circuito operacional de la ALU

En realidad este circuito cuenta con algunas diferencias.

Se puede observar un nuevo (cable denominado 0 ó 1) en el diagrama de la figura 38; que funciona como *entrada de control*, que va a permitir seleccionar la operación deseada. Si se desea sumar, se envía un “0” por dicho cable. Si se desea restar, se envía un “1”; en este caso, se realiza la resta por medio de la suma del Complemento a la Base del sustraendo, como se estudió en capítulos anteriores.

Esa señal se transmite a cada sumador e ingresa a una compuerta XOR.

La otra entrada del XOR es un bit de uno de los operandos (en el diagrama  $B_0, B_1 \dots$ ).

Recordemos que una compuerta XOR, si recibe un “0” como entrada, la salida es igual a la otra entrada. Por el contrario, si una entrada es “1”, la salida es la otra entrada invertida.

Cada sumador, entonces, recibe las siguientes entradas:

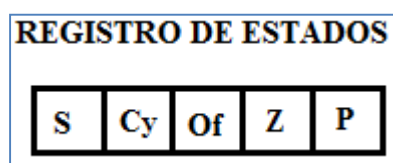
- Un bit de uno de los operandos (denominado  $A_0$  y  $A_1$  en el diagrama).
- La salida de la compuerta XOR.
- El Carry del sumador anterior.

Asimismo, se puede observar que el sumador *menos significativo* recibe las siguientes 3 entradas:

- Un bit de uno de los operandos ( $A_0$  y  $A_1$  en el diagrama).
- Un bit de la entrada de control (0 ó 1 en el diagrama).
- La salida de la compuerta XOR.

El Carry correspondiente al *sumador más significativo* genera la señal de Cy del Registro de Estados.

La ALU actualiza las señales de estado de acuerdo con el valor del resultado de la operación efectuada. Estas señales son almacenadas en el REGISTRO DE ESTADOS. El contenido de cada señal de estado sólo toma dos valores: 0 ó 1.



**Figura 39:** Señales de Estado del Registro de Estados.

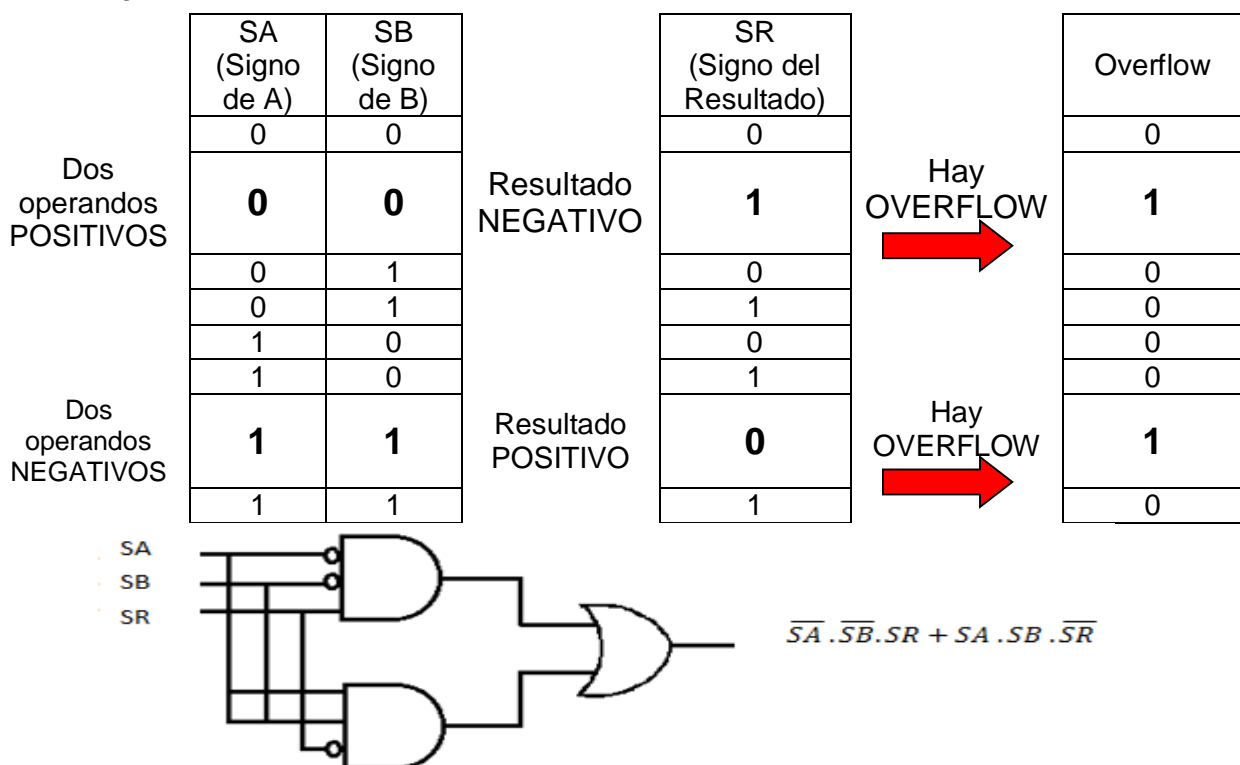


Las señales de estado que pertenecen a esta ALU son:

Señal	Nombre	Valor:	
<b>S</b>	Signo del resultado de la operación	0: SIGNO POSITIVO	1: SIGNO NEGATIVO
<b>Cy</b>	Carry, acarreo o arrastre	0: NO HUBO Carry	1: HUBO CARRY
<b>Of</b>	Overflow, desborde	0 : NO HUBO Overflow	1 : HUBO OVERFLOW
<b>Z</b>	Cero (en inglés Zero)	0: el resultado NO ES CERO	1: el resultado ES CERO
<b>P</b>	Paridad	0: No hay paridad PAR EN LOS 1	1: Hay paridad PAR EN LOS 1

La Unidad de Control (UC) recibe las señales de estado y las emplea para realizar saltos condicionales. Al llegar a una instrucción condicional, la UC continúa procesando según las condiciones previstas por el programador de acuerdo a los valores del registro de estados (REst). *Ejemplo: El resultado rebalsa la capacidad del Acumulador (detectado porque el signo del resultado es opuesto al de los sumandos). La UC de control consulta constantemente las señales de estado. La señal de Overflow(Of) pasó a valer 1. Si eso sucede, el programa emite un mensaje de error de OVERFLOW.* Cada señal de estado es generada por los circuitos aritméticos, lógicos o circuitos especialmente implementados para ese fin.

- **Señal de arrastre o Carry (Cy):** El arrastre del resultado de una operación ( $Cy_o$ ), se obtiene del último sumador del circuito sumador/restador/comparador/complementador, es decir, coincide con el " $Cy_n$ ". En el diagrama corresponde a  $Cy_1$ .
- **Señal Signo (S):** El signo del resultado se obtiene tomado del último bit (MSB) del resultado.
- **Señal de Overflow (Of):** El desborde (Overflow, abreviadamente "Of") se observa cuando al sumar dos números de igual signo, el resultado tiene signo opuesto, como se observa en la tabla siguiente.

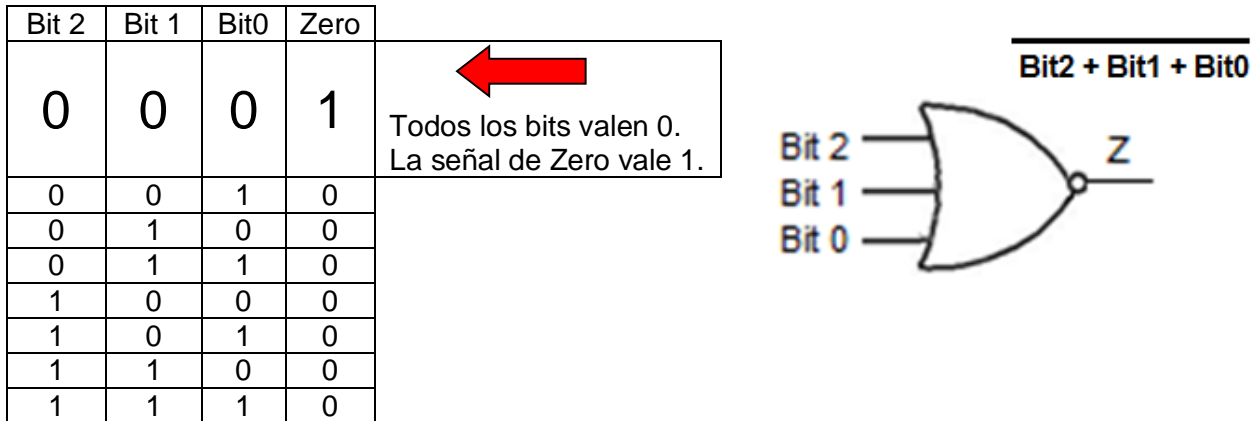


**Figura 40:** Overflow (Of) y una forma de implementarlo con compuertas lógicas.

- **Señal de Cero o Zero (Z):** Resultado cero (en inglés Zero), cuando cada uno y todos los bit



del resultado son cero. Puede implementarse con una compuerta NOR a la que se hacen ingresar todos los bits del resultado obtenido. Recordemos que la compuerta OR, da como resultado un 1, cuando por lo menos una de sus entradas es 1. Esto significa que la única forma de que la salida sea 0, es que todas las entradas sea 0. Entonces si queremos que la Señal de Zero se active (valga 1) cuando todos los bits sean 0, es necesario negar la salida. El ejemplo muestra 3 bits de entrada.



**Figura 41:** Circuito para Flag Zero

- **Señal Paridad (P):** La paridad par en 1 puede verificarse simplemente utilizando una XOR negada; sobre los bits resultantes del circuito operacional. Notese que cuando los bits del resultado, en este ejemplo: S1, S2... estén uno en 0 y otro en 1, la XOR dará 1 por resultado al estar negada P valdrá 0, es decir la cantidad de “unos” no es par, por ello la señal P tendrá un cero. Es decir P en 0 indicará paridad par en unos y P en 1 será paridad impar de unos.



**Figura 42:** Circuito para Flag Paridad

## 5. Fundamentos de la Unidad de Entradas y Salidas

Dentro del Modelo de Von Neumann, la Unidad de Entrada/Salida (E/S) es la encargada de manejar las transferencias entre el exterior de la computadora y el interior de ésta. Conecta a la CPU y a la MP con los dispositivos de E/S, ya sean éstos periféricos o memorias secundarias masivas (como un disco rígido). La Unidad de E/S tiene como función *controlar* el funcionamiento de los dispositivos periféricos y de memoria secundaria masiva; y *adecuar* los datos e información que entran y/o salen de los mismos a las características de los buses del sistema.

### 5.1. Generalidades

Las Unidades de E/S correspondientes a una computadora personal, donde la actividad de E/S no es muy abundante, están constituidas por un conjunto de registros y buses que se describen a continuación.

- Registro de Direcciones de E/S (RDE/S)
- Selector
- Registro de Intercambio de Datos (RID)

- Registros intermediarios de dispositivos (Reg. A, Reg. B, Reg. C, etc.). Son registros que se definen como de Entrada, Salida o Entrada/Salida, para recibir o emitir el flujo de bits desde o hacia un periférico.
- Registro de Control: permite que el programador defina el funcionamiento de cada uno de los registros intermediarios de dispositivos (Reg. A, Reg. B, etc.).
- Registro de Estados: permite que el periférico o el dispositivo de almacenamiento masivo informe al sistema su estado actual (por ejemplo si una impresora está lista para imprimir).

Los buses internos de la Unidad de Entrada/Salida están representados, en la figura, en color gris.

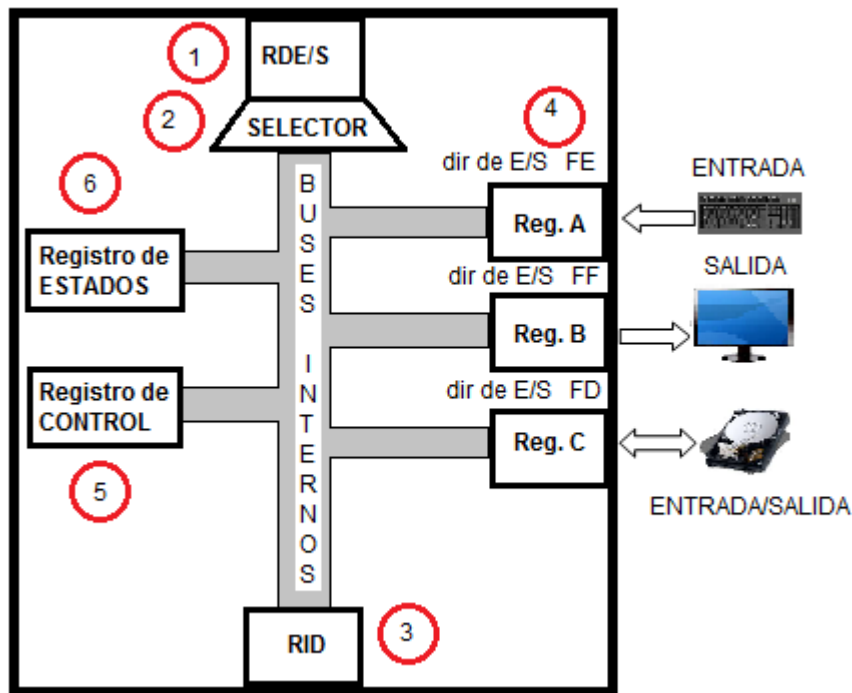


Figura 43: Unidad de Entradas y Salidas (E/S).

## 5.2. Funcionamiento de la Unidad de E/S

Para comprender el funcionamiento, nos basaremos en la Lectura de datos desde un periférico: En este ejemplo el programador decide MOSTRAR (imprimir) una información por medio del MONITOR. Este sistema cuenta con Memoria Mapeada, donde las direcciones desde F0 a FF se destinan a E/S.

Por medio del Registro de Control (5), se determina que:

- a) La dirección de E/S FD corresponde al Registro C, y servirá de conexión con un disco rígido (almacenamiento secundario masivo) que realizará E/S.
- b) La dirección de E/S FE corresponde al Registro A, y servirá de conexión con un teclado que realizará sólo Entrada.
- c) La dirección de E/S FF corresponde al Registro B, y servirá de conexión con un monitor que realizará sólo Salida.

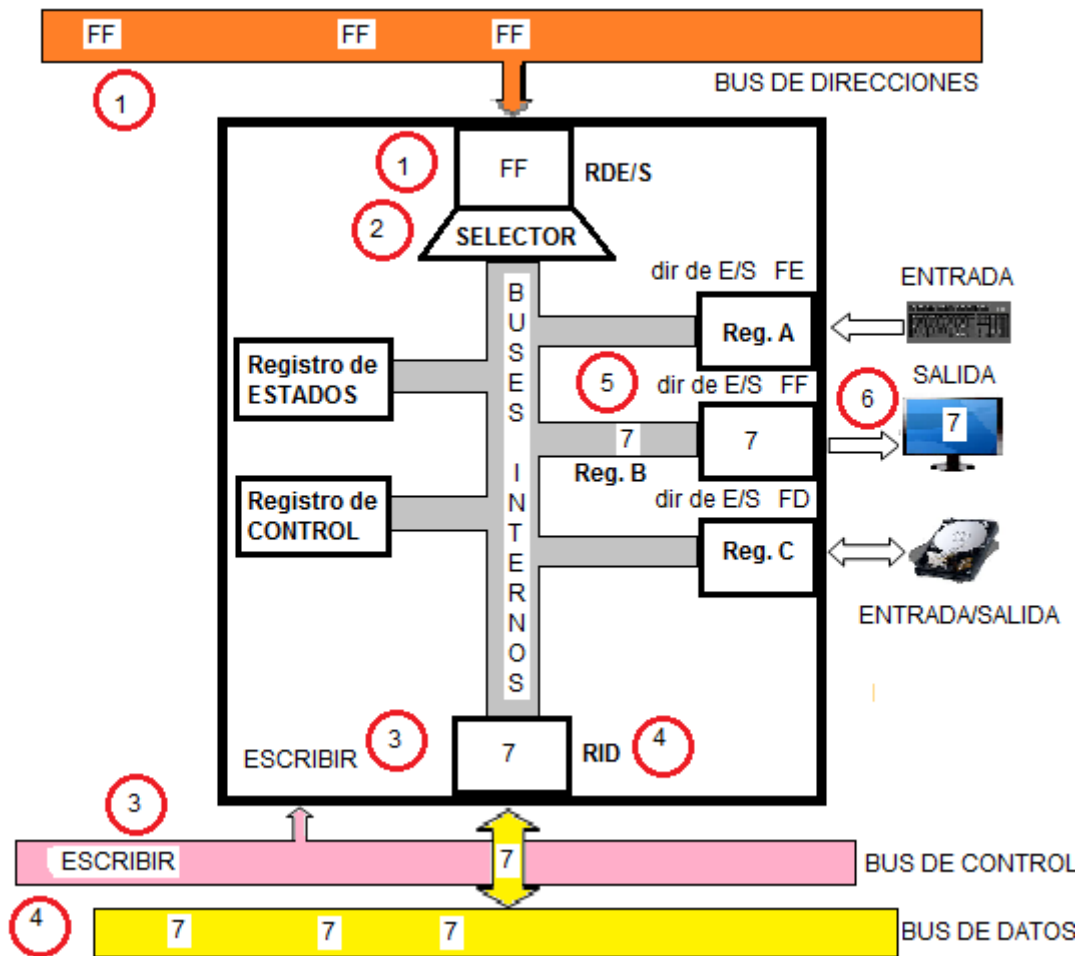
Para ejecutar la instrucción de MOSTRAR por monitor se siguen los siguientes pasos:

- 1.- El MAR contiene la dirección FF a la cual se quiere acceder para mostrar la información. Esta dirección se transmite por el bus de direcciones hasta el RDE/S (1).
- 2.- El Selector (2) habilita la dirección FF, que corresponde al Registro B.
- 3.- Simultáneamente, llega la orden de "Escribir" (en este caso Mostrar) por el bus de control.
- 4.- La información que estaba en el ACUMULADOR de la ALU, se transmite por el bus de datos

hasta el RID (3).

5.- La información se transmite internamente al Registro B, donde se escribe el valor a mostrar.

6.- Éste la envía al Monitor donde el usuario puede verla.



**Figura 44:** Ejemplo de funcionamiento de la Unidad de Entradas y Salidas (E/S).

Los gráficos y el funcionamiento presentados en esta sección corresponden a una arquitectura de sistemas simple que emplea un solo bus de datos para transferir datos e instrucciones entre CPU y MP y/o E/S, ya que el movimiento de bits es escaso. En este tipo de arquitectura, una placa madre (*motherboard*) contiene la CPU, la MP, la memoria caché y los controladores de E/S (por ejemplo, el del teclado) en forma “*on board*”, es decir, soldados a la placa madre. Otros controladores (placa de video, placa de sonido, etc.) pueden insertarse en *slots* o ranuras de la misma placa madre. Así, los controladores tienen acceso a los buses de direcciones, datos y control. A su vez, los periféricos y memorias secundarias masivas, se conectan a los controladores mediante puertos (*ports*), por ejemplo, el puerto USB (Universal Serial Bus).

### 5.3 Organización de E/S

Las distintas unidades que componen una computadora poseen características distintas de diseño y de componentes electrónicos, mecánicos, etc. Dichas características imponen velocidades de procesamiento también distintas. Por ejemplo: la CPU, que es la unidad más veloz, es 10 veces más rápida que la Memoria Principal. Asimismo la Memoria Principal es 200.000 veces más rápida que los dispositivos de E/S encargados de la comunicación del usuario con la computadora llamados “periféricos”.

La velocidad de todo el proceso, está marcada por el proceso más lento, y este es la E/S.

Los dispositivos periféricos, son mucho más lentos que la CPU y poseen características particulares de acuerdo al periférico. Por esa razón, los periféricos se conectan a “Controladores” que son circuitos encargados de adaptar las señales procedentes de los periféricos para que sean transmitidos por el bus de datos. En nuestro esquema inicial de 3 unidades, los Controladores constituyen la denominada “Unidad de E/S”.

En este esquema, la CPU es la que permite el uso del único bus, ya sea a Memoria Principal o a E/S. Recordemos que el Bus de Datos es “bidireccional” y sin el control y supervisión de la CPU, se producirían numerosas “colisiones”, que harían más lento el procesamiento.

Durante el tiempo que la CPU autoriza el uso del Bus de Datos a un periférico, éste NO puede ser utilizado por la Memoria Principal u otro periférico, ya que se producirían choques entre bits (colisiones). Existen diferencias entre los distintos tipos de computadoras (personales y mainframes), de acuerdo a la organización de E/S.

#### 5.4. Atención a Dispositivos de Entradas y Salidas. Interrupciones y DMA.

Existen tres maneras principales en que puede realizarse la atención de los dispositivos de E/S: (1) Interrupciones; (2) Programada o por programa; (3) DMA (Acceso Directo a Memoria).

Para comprender los tres tipos de interrupciones, se mostrarán los siguientes ejemplos.

a) Imagine que está en una clase, por ejemplo de esta Materia. El Profesor está explicando un tema y usted desea hacerle una pregunta. El profesor permite que un alumno pueda interrumpir su explicación. Entonces usted puede “interrumpir” la explicación del Profesor preguntando en voz alta su duda. El Profesor deja de desarrollar su clase, se fija hasta dónde ha explicado y le responde; aclara su duda y continúa con la explicación a partir del momento y del tema en que fue interrumpido.

b) En otra Materia, en cambio, el Profesor aclaró desde la primera clase que no permite interrupciones mientras explica. También aclara que cada hora, él va a preguntar a los alumnos si desean consultar alguna duda.

c) En una tercera Materia, el Profesor no da clase, simplemente proporciona a los alumnos una dirección web con la clase del día y los alumnos acceden a esa clase sin la participación del profesor.

• **Interrupciones:** Corresponden al primer ejemplo (a). La CPU se encuentra procesando un programa principal y recibe un pedido (señal) de interrupción (IRQ, por sus siglas en inglés: Interrupt Request) emitido por un dispositivo de E/S.

Si la CPU acepta el pedido de interrupción:

- Primero finaliza la instrucción en curso.
- Interrumpe el procesamiento del programa que estaba ejecutando y guarda los contenidos de los registros necesarios para retomar el procesamiento sin comenzar todo nuevamente.
- Ejecuta el programa necesario para atender la interrupción.
- Cuando finaliza la atención, restablece los contenidos de los registros y continúa procesando el programa principal.

Todo este proceso se realiza en tiempo real.

• **Programada o por programa:** El programa principal verifica el estado de los dispositivos de E/S por medio de secuencias de instrucciones que verifican el estado de los dispositivos de

entrada salida y, prestarán atención a las necesidades de estos últimos recién en el momento en que se haga la verificación.

• **Acceso Directo a Memoria (DMA):** El DMA se produce cuando un controlador realiza una operación de lectura o escritura de memoria SIN INTERVENCIÓN de la CPU. Recordamos que hasta ahora, para acceder a MP, se debía pasar forzosamente por la CPU, con la consabida pérdida de tiempo. Sin embargo, existen casos; como cuando un dispositivo es muy veloz o no se puede detener la transferencia de datos sin pérdidas, donde es necesario recurrir a DMA.

Algunos dispositivos de entrada y salida tienen prioridad sobre la CPU, ya que los discos y otros artefactos en movimiento no pueden detenerse y, forzándolos a esperar, se podría perder información.

Los pasos del DMA son los siguientes:

- a) Un circuito especial denominado “*Controlador de DMA*” se encarga de administrar la lógica necesaria para llevar a cabo los procesos de acceso directo a memoria, vinculando la CPU y los controladores que pueden realizar este tipo de acceso.
- b) El controlador comienza solicitando a la CPU un *Pedido de Acceso Directo a Memoria* (DRQx o *Direct Access Request*).
- c) El Sistema debe identificar el periférico conectado al controlador solicitante, y le informa al controlador la dirección y la cantidad de bytes de la MP a la cual acceder.
- d) La CPU le otorga el permiso y entonces el controlador se convierte en el “*master*” (el que toma el control) del bus de datos y de la MP.
- e) La CPU pasa a un estado de desconexión.
- f) El sistema le dará al controlador la orden de comenzar el DMA.

Por ejemplo, al tratarse de la lectura de un disco rígido, la CPU le debe indicar el bloque que se desea leer a la controladora y esta última, deberá esperar a obtener el primer byte de la cabeza lectora para solicitar convertirse en maestro del bus por medio de una “solicitud de acceso directo a memoria” DRQx.

Respecto de la organización de DMA, existen tres métodos para realizar el proceso de DMA:

(1) Ráfagas; (2) Robo de ciclos; (3) Transparente., desarrollados en ANEXO 2.

## 6. Fundamentos del Procesamiento de Datos

Un PROGRAMA, es un conjunto finito de instrucciones que se ejecutan en forma secuencial, salvo que se indique una bifurcación.

### 6.1 Instrucciones

Un programa está compuesto por INSTRUCCIONES. El procesamiento de un PROGRAMA, significa que la CPU va a ejecutar cada una de las instrucciones que componen el programa.

Las instrucciones sufrieron una evolución desde los principios de la Informática.

Cada cambio, significó la creación de un nuevo registro dentro de la CPU.

Las instrucciones primitivas, contenían las siguientes partes, llamadas CAMPOS.

Código de operación (CODOP) ¿qué hacer?	Dirección del operando 1 ¿dónde está?	Dirección del operando 2 ¿dónde está?	Dirección del resultado ¿dónde se guarda?	Dirección de la próxima instrucción ¿dónde se sigue?
9BH	20H	21H	22H	78H

Como es posible observar, cada instrucción ocupaba muchos bytes y los programas resultaban muy extensos (ocupaban una gran cantidad de bytes). Por ese motivo, el diseño de las operaciones fue haciéndose más pequeño. Para lograr ese objetivo, se crearon nuevos registros (CP, RTOp, etc.) y operaciones (MOVE) en la CPU, hasta llegar a un diseño de instrucción mínimo que es el que se emplea en esta unidad.

Código de Operación (Codop) ¿Qué Hacer?	Dirección del Operando 1 ¿Dónde Está?
9BH	20H

Los códigos de instrucción indican distinto tipo aritméticas (sumar, restar, multiplicar, dividir), de E/S (leer, guardar), lógicas (comparación), de salto o ruptura de secuencia, etc.

Existen también algunas instrucciones sin dirección de operando, algunas instrucciones de control (NOP: no operar) y otras cuyo operando se ubica en un registro de CPU.

## 6.2 Ciclo de instrucción

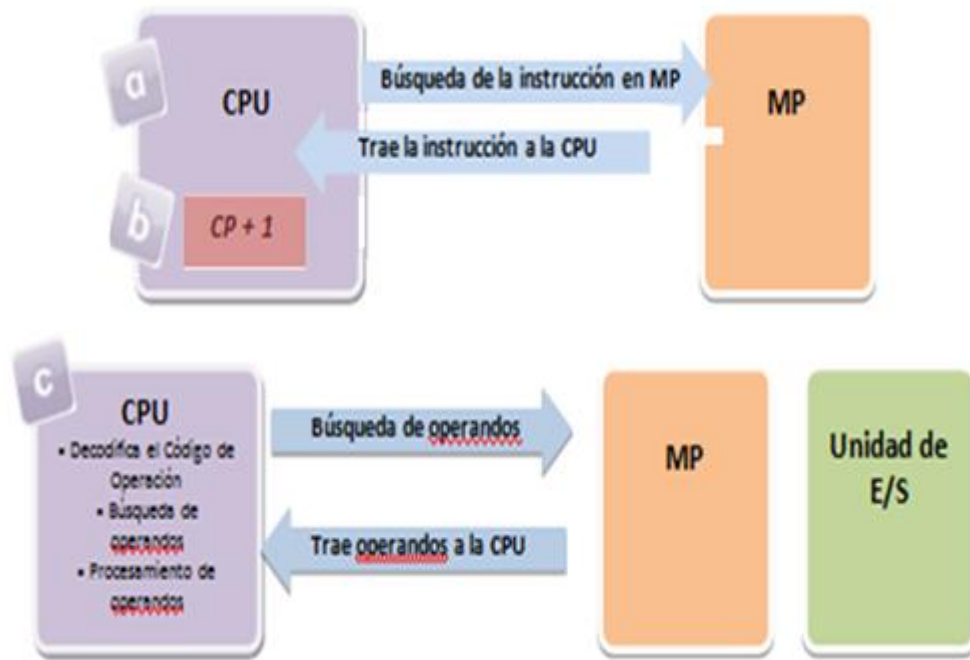
El Ciclo de Instrucción comprende los pasos necesarios para que la CPU **busque en memoria y ejecute una instrucción**. Para ello deben cumplirse las siguientes condiciones:

- 1.- El programa que se desea ejecutar (DEBE ESTAR ALMACENADO EN LA MEMORIA PRINCIPAL (NUNCA se ejecuta desde disco, pen drive, CD, DVD, etc.).
- 2.- El Sistema Operativo asignó previamente una dirección a partir de la cual se almacenará el programa.
- 3.- Las instrucciones del programa se ubican en forma consecutiva en memoria. Ocupan, cada una, una *posición* de memoria; y tienen, cada una, una *dirección*.
- 4.- El programa se procesa en forma SECUENCIAL (una instrucción tras otra en forma ordenada), salvo que existan instrucciones de salto dentro del programa.
- 5.- Se comienza por la PRIMERA instrucción, luego la SEGUNDA y la TERCERA, etc.

El Ciclo de Instrucción se compone de 2 (dos) Fases:

- Fase de Búsqueda
- Fase de Ejecución

Al comenzar la ejecución de un programa el primer paso de la fase de búsqueda será cargar la posición de la primer instrucción de ese programa en el CP (esto lo realiza el sistema operativo). Luego tal como lo muestra la figura 47, se procede a otros pasos de la fase de búsqueda representados por (a) y (b); a continuación habiendo incrementado el CP (para que apunte a la próxima instrucción) se procede a continuar con la fase de ejecución representado por la parte (c). Al concluir la ejecución de la instrucción comenzará nuevamente la fase de búsqueda leyendo la próxima instrucción que estará por la posición apuntada por el CP.



**Figura 47:** (a) y (b) Fase de Búsqueda que incluye Incremento del contador de programa CP; (c) Fase de Ejecución.

Este proceso se repite para cada instrucción del programa y finaliza al terminar éste. El próximo ejemplo permitirá comprender en detalle lo que sucede en cada fase.

### 6.3 Ejemplo de procesamiento

Se posee una computadora que tiene un teclado como unidad de entrada cuya dirección es FE y un monitor, cuya dirección es FF, como salida. La longitud de palabra es de 16 bits, 8 para el código de operación y 8 bits para el campo dirección de memoria. Tanto las instrucciones como las direcciones de memoria se presentan expresadas en hexadecimal.

El CP se incrementa de a 1 por palabra, ya que cada instrucción ocupa una sola posición de memoria.

La ALU posee un solo registro: "A" (ACUMULADOR). Las instrucciones a utilizar en este ejercicio son:

<b>Código de la instrucción</b>	<b>Descripción</b>	<b>En símbolos</b>
96 <sub>H</sub>	Cargar el acumulador con el contenido de la posición "m"	$A \leftarrow m$
97 <sub>H</sub>	Almacenar en la dirección "m" el contenido del acumulador	$m \leftarrow A$
9B <sub>H</sub>	Sumar al contenido del acumulador, el contenido de la dirección "m"	$A \leftarrow A + m$

El programa fue cargado a partir de la posición 19<sub>H</sub> de memoria y los datos se almacenarán a partir de la posición A1<sub>H</sub> de memoria.



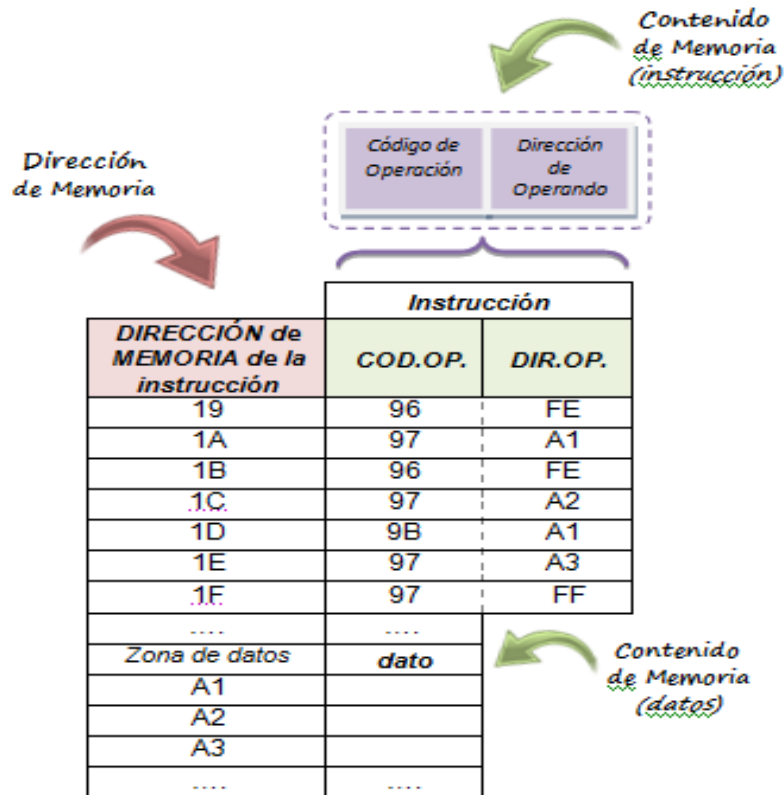


Figura 48: Situación de Memoria.

## • Procesamiento

El ejemplo comienza con:

### – INICIALIZACIÓN DE REGISTROS

- INICIALIZACIÓN O RESETEO DE REGISTROS
- CP ← DIRECCIÓN DE LA 1ra. INSTRUCCIÓN DEL PROGRAMA

En nuestro ejemplo la *primera instrucción* del programa está almacenada en la dirección 19<sub>H</sub> de Memoria Principal.

### – FASE DE BÚSQUEDA

**Paso 1:** El Contador de Programa (CP) se carga con la dirección 19<sub>H</sub>.

#### • ACCESO A MP

Es necesario acceder a la dirección que indica el CP para ir a buscar y leer la instrucción que se desea procesar y copiarla en CPU.

**Paso 2:** Para realizar el acceso a MP, el contenido del CP, en este caso 19<sub>H</sub>, pasa al MAR.

**Paso 3:** Desde allí “viaja” por el bus de direcciones hasta el RDM ubicado en la MP.

**Paso 4:** Una vez que la dirección 19<sub>H</sub> llega al RDM, el decodificador de direcciones de memoria selecciona y habilita la posición indicada (en este caso, selecciona y habilita la dirección 19<sub>H</sub>).

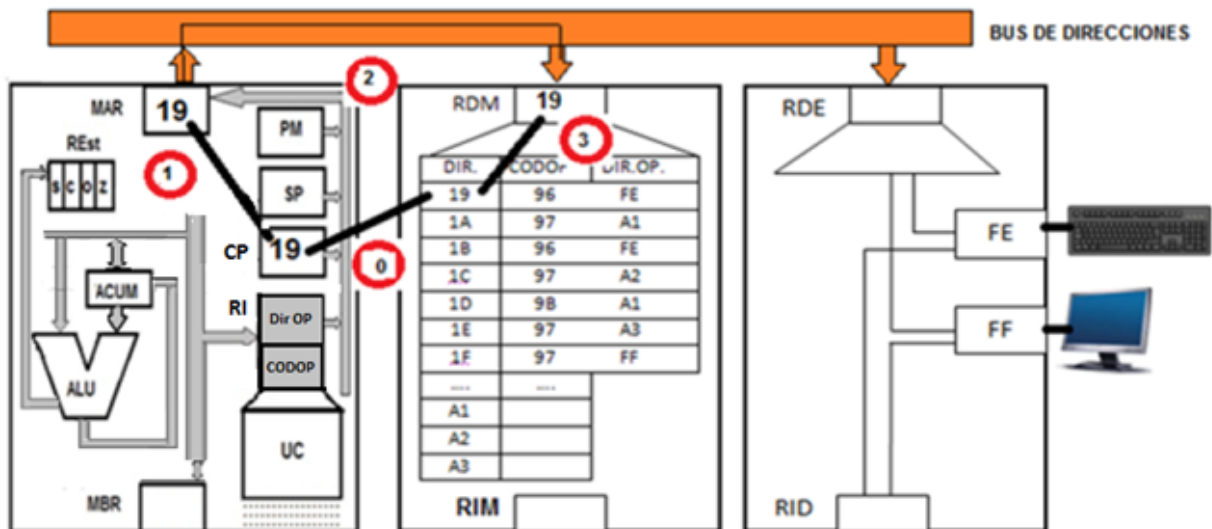


Figura 49. Pasos de la Fase de Búsqueda – Acceso a MP

### • LECTURA DE LA INSTRUCCIÓN

**Paso 5:** Llega por el bus de control la orden de *lectura* de memoria (en la fase de búsqueda se recupera la instrucción por ello se lee el contenido de la posición de memoria).

**Paso 6:** Esta orden produce que una copia de la instrucción llegue al RIM.

**Paso 7:** De allí, la instrucción “viaja” por el bus de datos, rumbo a la CPU y llega al MBR.

**Paso 8:** Una vez en el MBR, LA INSTRUCCIÓN COMPLETA pasa al RI (Registro de Instrucción).

**Paso 9:** La parte correspondiente al CODOP (Código de Operación) está conectada a la Unidad de Control y la parte de dirección de operando pasa al MAR para acceder al operando.

**Paso 10:** Inmediatamente se incrementa el CP y apunta a la próxima instrucción.

*Aquí finaliza la Fase de Búsqueda.*

### - FASE DE EJECUCIÓN

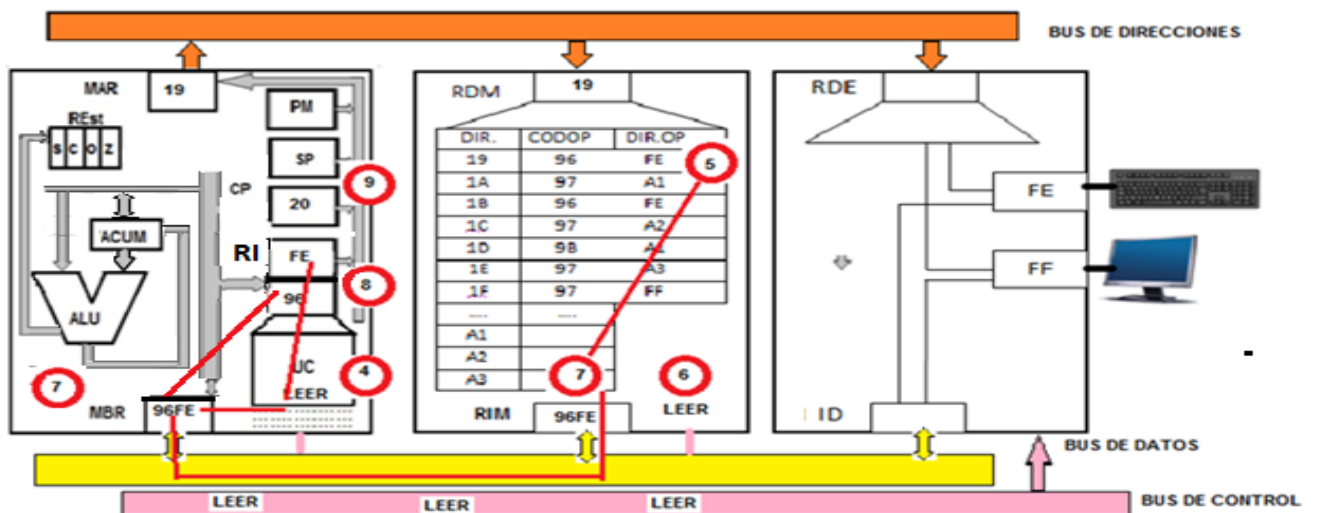


Figura 50: Pasos de la Fase de Búsqueda, “Lectura de la instrucción”

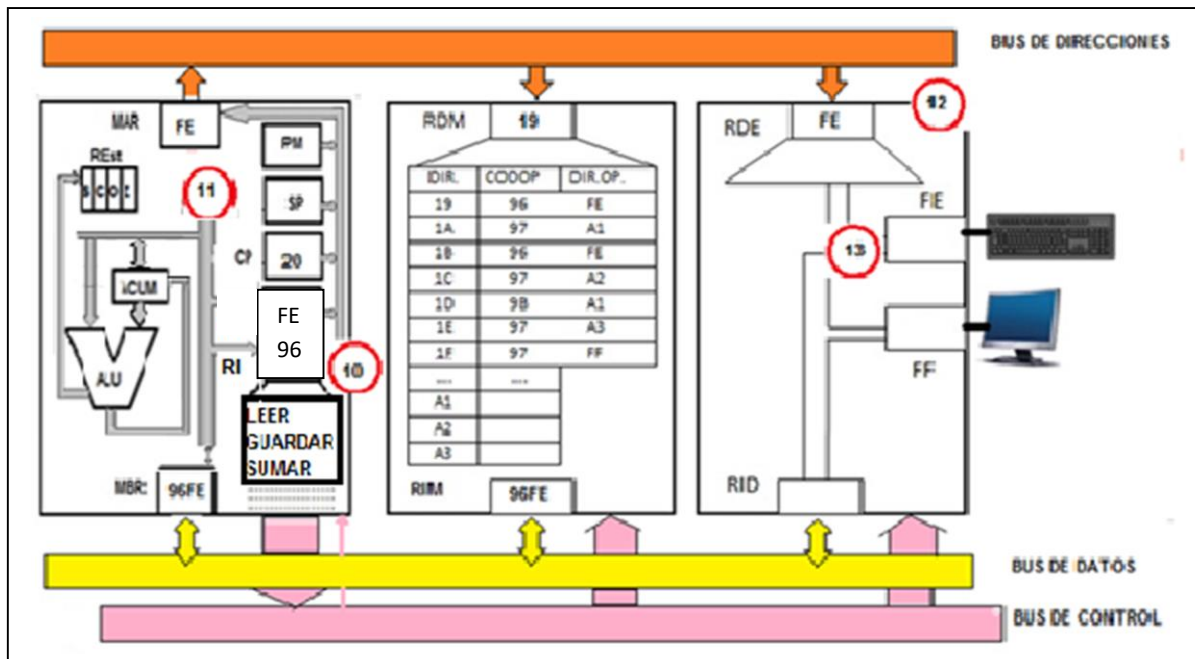
**Paso 10.** Decodificar el CODOP y emitir los microcomandos para su ejecución.

**Paso 11.** Como se trata de una lectura de E/S, es necesario acceder al registro FE de la Unidad

de E/S para conectarse con el Teclado y permitir el ingreso del dato. Para ello, la dirección del dato, contenida en el RI (Registro de Instrucción), es copiada en el MAR.

**Paso 12.** Desde allí se transmite por el BUS de direcciones hasta en RDE/S.

**Paso 13.** El Decodificador de direcciones de E/S habilita el registro de dirección FE.



**Figura 51:** Pasos de la Fase de Búsqueda, "Decodificar la instrucción"

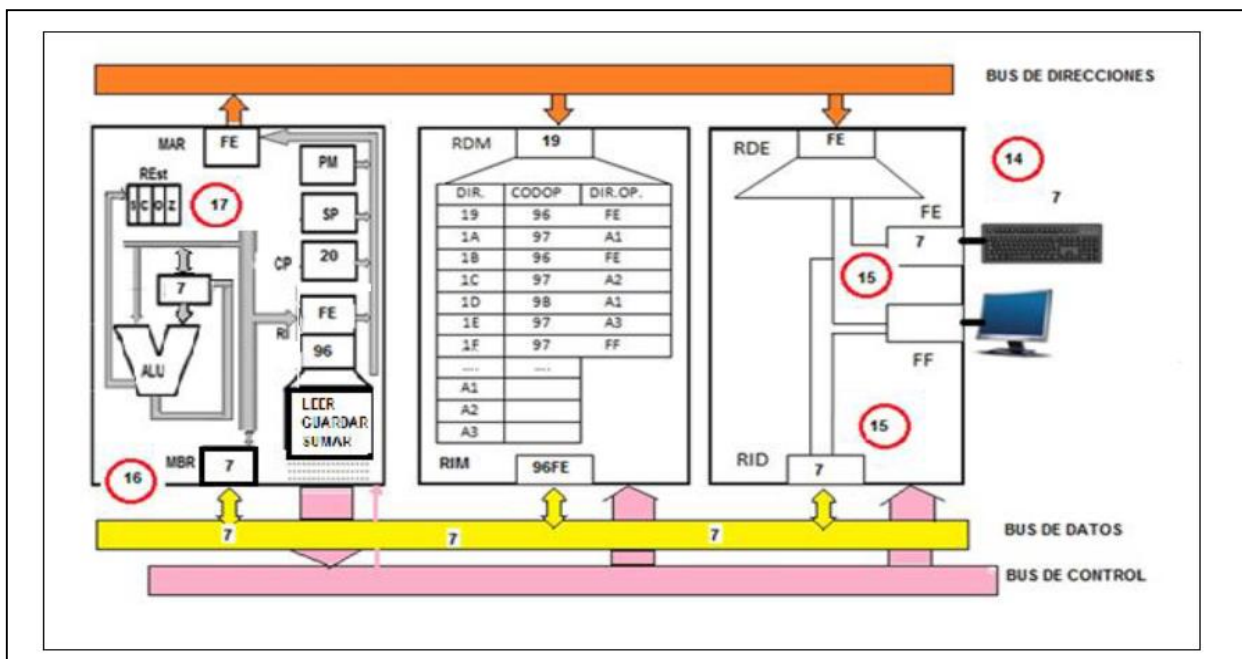
**Paso 14.** El usuario pulsa las teclas correspondientes al dato (en este ejemplo el número 7).

**Paso 15.** El dato pasa por el Registro FE y se almacena en el RID.

**Paso 16.** El dato se transmite por el bus de datos hasta el MBR.

**Paso 17.** El dato se almacena en el ACUMULADOR.

Aquí finaliza el Ciclo de Instrucción de la 1ra. Instrucción de Programa.



**Figura 52:** Pasos de la Fase de Búsqueda, "Ingreso del Dato"

## 7. Modos de direccionamiento

Los *modos de un direccionamiento* de un procesador son las diferentes formas de transformación del campo de dirección de la operación en la dirección del operando.

Un modo de direccionamiento *especifica la forma de establecer la dirección*, es decir el lugar físico verdadero, correspondiente a un operando, mediante el uso de la información contenida en registros, o en una instrucción de la máquina. Debe recordarse que:

- La DRO es la Dirección Real de Operando.
- Es decir, es la DIRECCIÓN, donde se encuentra el dato.

Recordemos que una instrucción está formada por dos partes:

CÓDIGO DE OPERACIÓN "CODOP"	DIRECCIÓN DE OPERANDO
¿QUÉ HAY QUE HACER? Leer, grabar, sumar, comparar, etc.	¿EN QUÉ DIRECCIÓN DE MEMORIA ESTÁ ALMACENADO EL OPERANDO?

### 7.1 Direccionamiento Absoluto

En este modo de direccionamiento, la dirección de memoria del operando, se *informa de manera completa* como parte de la instrucción. Se proporciona directamente la dirección real del operando.

CÓDIGO DE OPERACIÓN "CODOP"	DIRECCIÓN DE MEMORIA DEL OPERANDO
--------------------------------	--------------------------------------

- La DRO indica exactamente el lugar donde se encuentra el dato.
- Para conocer la dirección real del operando (DRO), no es necesario hacer ninguna operación, la dirección está en la propia instrucción. Por ejemplo: Leer un dato que se encuentra en la dirección DE02.

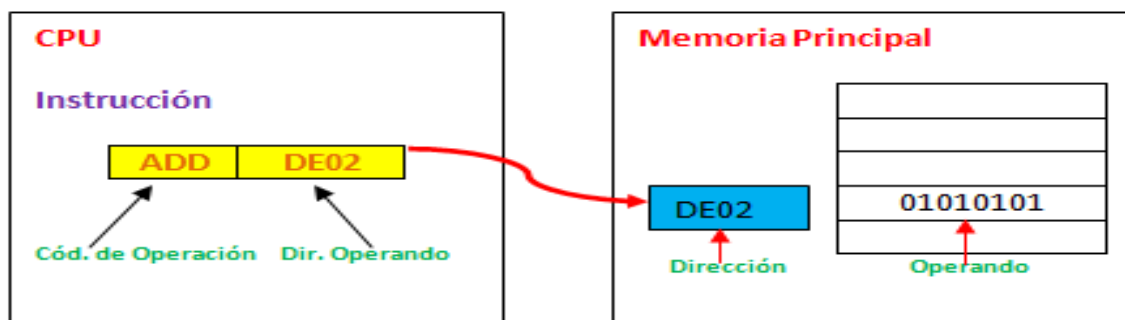


Figura 53: Direccionamiento Absoluto

### 7.2 Direccionamiento Relativo

En este modo de direccionamiento, la posición absoluta del operando, se expresa mediante un *desplazamiento* a partir de una posición prefijada. La Dirección Real del Operando (DRO) se obtiene como el **desplazamiento** ("offset") a partir de una posición **base** que se encuentra almacenada en algún registro de CPU que funciona como Registro Base.

La DRO se calcula sumando la dirección que aparece en la instrucción (desplazamiento u OFFSET) más el contenido de un registro de CPU llamado genéricamente REGISTRO BASE.

<b>DRO =</b>	<b>CONTENIDO DE REGISTRO BASE</b>	<b>+</b>	<b>DESPLAZAMIENTO (OFFSET)</b>
--------------	-----------------------------------	----------	--------------------------------

Para conocer la DRO es necesario *realizar una operación* con el desplazamiento que está en la instrucción y a partir de una posición base que está en un registro, según la forma de obtener la DRO, los modos de direccionamiento relativos se clasifican en:

- Indexado
- Relativo Propiamente dicho
- Inmediato
- Paginado

Todos estos modos de direccionamiento son, por ende, relativos; y se detallan en el ANEXO 1.

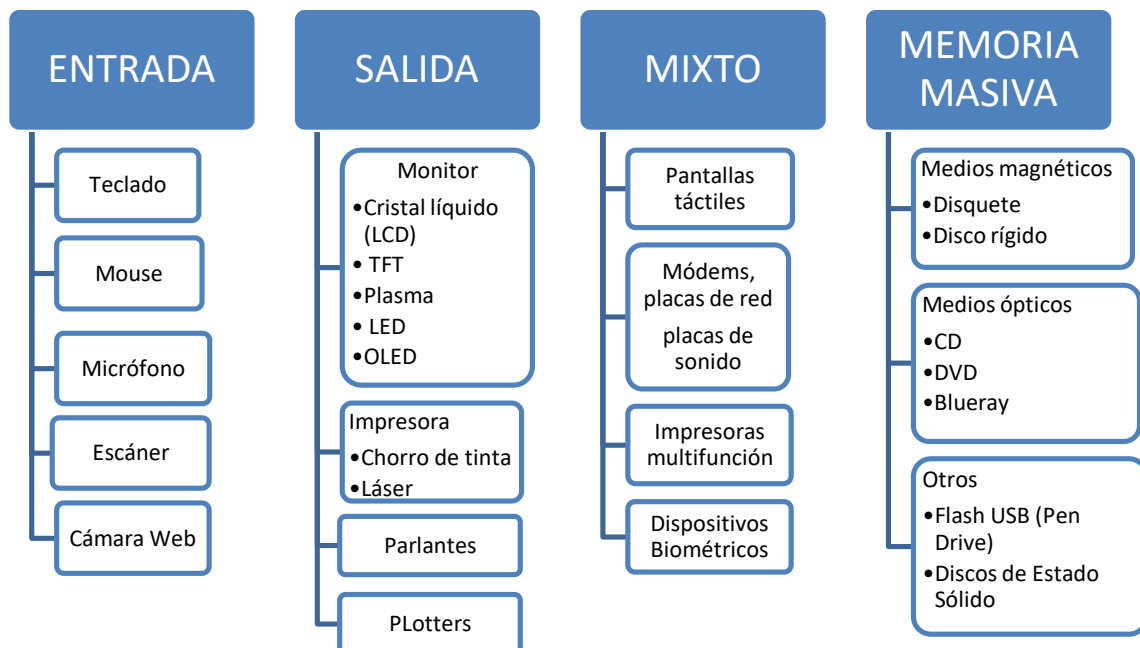
## 8. Periféricos y dispositivos de entrada / salida

Los periféricos son el hardware que nos permite interactuar con la computadora; es decir, poder ingresar datos y obtener información. Asimismo, los medios de almacenamiento secundario, auxiliar o masivo, permiten almacenar grandes volúmenes de información permanentemente. Todos estos dispositivos, se convierten en los intermediarios entre el usuario y la computadora.

Los periféricos se pueden clasificar según su función en: *periféricos de entrada*, permiten al usuario el ingreso de los datos, y *periféricos de salida*, permiten obtener la información procesada. Hoy en día también podemos hablar de los *periféricos mixtos*, son los periféricos que nos permiten tanto el ingreso de los datos como la obtención de la información.

Las *memorias masivas* o *memorias secundarias*, son los medios de almacenamiento que nos permiten guardar la información, a través de un medio físico, en forma permanente.

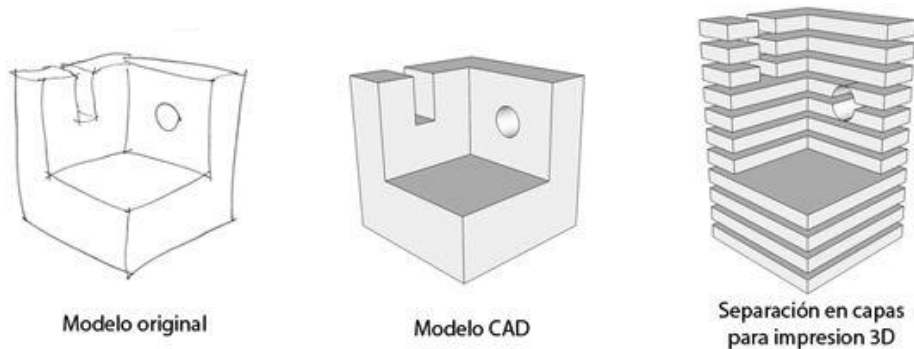
### 8.1 Clasificación De Periféricos Y Dispositivos De Entrada / Salida



**Figura 62.** Clasificación de Periféricos y Dispositivos de E/S. Impresoras 3D

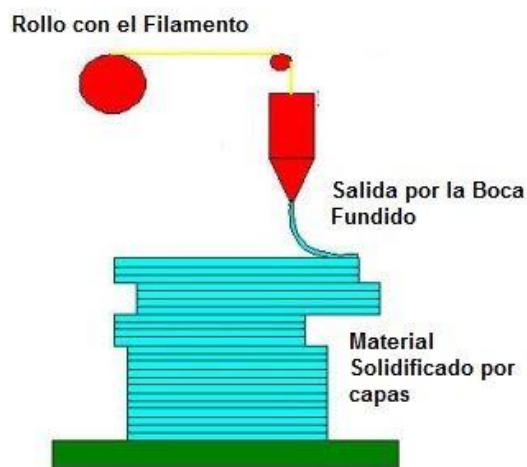
Son dispositivos que permiten el ingreso de datos a la computadora de manera diferente.

Las impresoras 3D son capaces de crear un objeto con sus 3 dimensiones, es decir las impresoras 3D crean volúmenes, objetos tridimensionales largo, ancho y profundidad). Este resultado lo realizan construyendo capas que van superponiendo hasta lograr el objeto, esta forma de trabajo se denomina “proceso aditivo”. En la siguiente figura se puede observar el dibujo realizado a mano alzada, el dibujo hecho con CAD (programa de Diseño Apoyado por Computadora) y el diseño fraccionado en capas. Actualmente existen scanners 3D que permiten escanear un objeto, guardar su imagen y luego imprimirla con la impresora 3D, logrando un objeto tridimensional.



**Figura 75:** Impresora 3D.<sup>12</sup>

La forma de impresión más difundida, debido a su menor costo, trabaja de la siguiente forma, la impresora va fundiendo un hilo de polímero, capa a capa, formando el objeto sólido.



**Figura 76:** Impresora 3D. Láser fundiendo hilo de polímero.<sup>13</sup>

## Disco magnéticos

### Disco rígido (o disco duro)

El disco rígido (*Hard Disk Drive*, HDD en su sigla en inglés) está generalmente instalado en el interior del gabinete de la computadora, pero también existen discos externos. El disco está

<sup>12</sup> Imagen extraída de: <http://www.areatecnologia.com/informatica/impresoras-3d.html>

<sup>13</sup> Imagen extraída de: <http://www.areatecnologia.com/informatica/impresoras-3d.html>

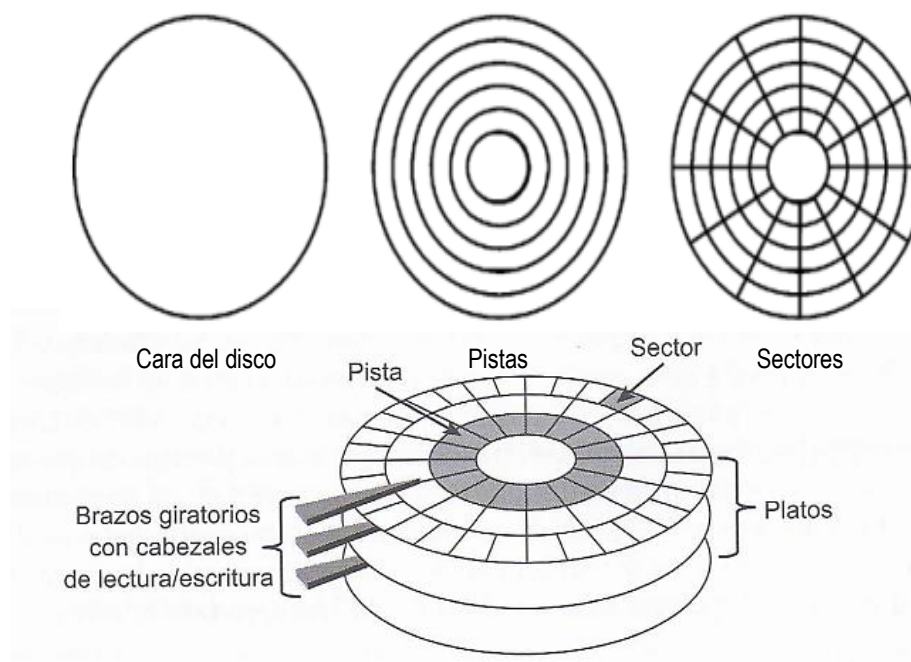


formado por las *pistas* de datos son **círculos concéntricos** invisibles. Están enumeradas y se dividen en pequeños bloques denominados *sectores*, que contienen (por ejemplo) 512 bytes cada uno.

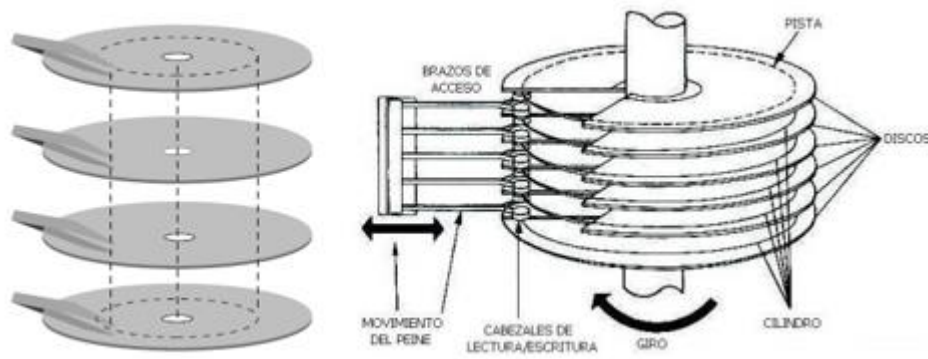
Un *clúster* es la agrupación de varios sectores; mientras que la unión de varios clústers conforma una pista. El número total de pistas y sectores depende del tipo de disco que se esté utilizando y del tipo de formato (formateo) que se le dé. La capacidad de almacenamiento de información en un disco depende de los bits por pista y el número de pistas. Los componentes básicos de un disco rígido son: un conjunto de discos rígidos (por ejemplo, dos, cuatro o más) comúnmente llamados platos, sellados dentro de una caja metálica para evitar el ingreso de impurezas; un eje en el cual se montan todos ellos; una serie de cabezales de lectura / escritura (al menos uno para cada lado de cada plato); un motor para hacer girar el eje con sus discos, otro para posicionar los cabezales en el cilindro que se desea leer o escribir y los circuitos electrónicos integrados que permiten controlar todos estos procesos.

Los platos son de metal (aleaciones de aluminio) y se encuentran recubiertos en ambos lados con una capa delgada de óxido de hierro (el aluminio y sus aleaciones no poseen propiedades magnéticas).

El conjunto de pistas que son leídas simultáneamente por los cabezales, se denomina **cilindro** (ver figura 80 y 81).



**Figura 80.** Conceptos de Cara, Pista, Sector y Plato en un conjunto de discos rígidos



**Figura 81.** Concepto de **cilindro** en un conjunto de discos rígidos

Conociendo el número de **caras, pistas, sectores y bytes por sector**, es posible determinar la capacidad de almacenamiento aplicando la siguiente fórmula:

$$Cap = Nc . Np . Ns . Cs$$

- $Cap$  = capacidad del disco
- $Nc$  = número de caras
- $Np$  = número de pistas
- $Ns$  = número de sectores
- $Cs$  = capacidad del sector

Quando el software de aplicación indica al sistema operativo que debe leer o escribir a un archivo, solicita que el controlador del disco rígido traslade los cabezales de lectura/escritura a la tabla de asignación de archivos. El sistema operativo lee para determinar en qué punto comienza un archivo en el disco, o qué partes del disco están disponibles para guardar un nuevo archivo.



**Figura 82.** Aspecto exterior e interior de un HDD (unidad de disco rígido).

## Almacenamiento masivo de estado sólido

## Memorias flash

La memoria *flash* – derivada de la memoria EEPROM – permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Gracias a ello, la tecnología flash, siempre mediante impulsos eléctricos, permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM original, que sólo permitía actuar sobre una única celda de memoria en cada operación de programación.

Se denominan “flash” debido a sus tiempos rápidos de borrado y escritura<sup>14</sup>. La mayoría de estos chips usan una operación de borrado masivo. Las memorias tipo flash más modernas ofrecen un

<sup>14</sup> El borrado no se realiza a nivel de byte (como en las EEPROM) sino a nivel de bloques, borrando de una sola acción un conjunto importante de posiciones de memoria, permitiendo a partir de allí la programación

modo de borrado por sector. El tiempo de escritura en una memoria flash es de aproximadamente 10  $\mu$ s (microsegundos), mientras que en una memoria EEPROM ronda los 5 ms (milisegundos).

Esta sectorización también simplifica la reprogramación, pues se puede alterar un solo bloque de memoria sin modificar los demás; y facilita, asimismo, la operación de lectura.

Las tecnologías de tipo flash han permitido incrementar la cantidad de componentes y circuitos integrados por unidad de superficie, por lo que poseen capacidad de almacenamiento y velocidad de acceso mayores; y un menor costo. Se trata de la tecnología empleada en los dispositivos llamados “memoria USB”.

Económicamente, el precio en el mercado cumple la Ley de Moore; aumentando su capacidad y disminuyendo el precio a medida que transcurre el tiempo.

Entre sus ventajas, presentan gran resistencia a los golpes, alta velocidad, bajo consumo de energía y un funcionamiento silencioso, ya que **no contiene partes móviles**. No presentan, por ello, desgaste mecánico. Su pequeño tamaño y peso liviano también son ventajas importantes.

No obstante, todos los tipos de memoria flash sólo permiten un número limitado de escrituras y borrados, generalmente entre 10.000 y un millón, dependiendo de la celda, de la precisión del proceso de fabricación y de la tensión eléctrica necesaria para su borrado.

La mayor capacidad y el reducido costo han convertido a las memorias flash en el medio de almacenamiento no volátil más utilizado en dispositivos móviles.

Este tipo de memoria está fabricado con puertas lógicas NOR y NAND para almacenar los ceros o unos correspondientes.

Existen numerosos modelos con diferentes prestaciones orientadas al mercado industrial, y también en forma de tarjetas externas diseñadas para el mercado de consumo (cámaras digitales, reproductores portátiles de .mp3, teléfonos móviles). Algunos de estos modelos son: Secure Digital, Compact Flash, SD-Multi Media Card, Smart Media, Memory Stick, entre otros.

El acceso a estas memorias se realiza mediante protocolos de “conexión – comunicaciones” potentes y fáciles de implementar, lo cual ha extendido su empleo.

### Flash USB

La memoria USB (*Universal Serial Bus*), llamada también lápiz USB, memoria externa o **pendrive** es un dispositivo de almacenamiento de datos que utiliza memoria flash para guardar datos e información. Ofrecen una excelente combinación de alta capacidad de almacenamiento, elevadas velocidades de transferencia de datos y gran flexibilidad.



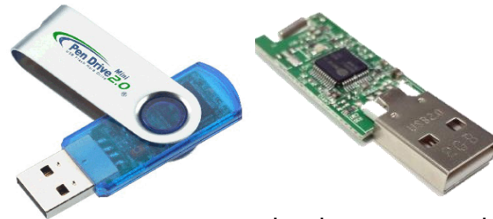
**Figura 86.** Diferentes modelos de memorias flash.

---

individual de cada byte. Justamente el nombre “flash” proviene del borrado “instantáneo” de un bloque de la memoria, permitiendo economizar tiempo al no tener que borrar individualmente cada byte.

Los primeros modelos requerían una batería, pero los actuales utilizan la energía eléctrica procedente del puerto USB. Estas memorias son resistentes a los golpes, al polvo, y algunos hasta al agua; factores que afectan a los medios ópticos como los CD, DVD y BD.

Se han convertido en el sistema de almacenamiento y transporte personal de datos más utilizado, desplazando en este uso a los antiguos disquetes y a los CD. Se pueden encontrar en el mercado fácilmente memorias de 1 a 512 GB (en diversas capacidades de  $2^n$ ) y hasta 1 TB. Las memorias con capacidades más altas pueden encontrarse todavía, a causa de su precio, fuera del rango de un consumidor doméstico. Esto supone, como mínimo, el equivalente a 180 CD de 700 MB, aproximadamente.



**Figura 87.** Exterior e interior de una memoria flash USB.

Los sistemas operativos actuales pueden leer y escribir en las memorias con sólo conectarlas a un puerto USB del equipo encendido, recibiendo la tensión de alimentación a través del propio conector, de 5 V y un consumo de 2,5 W como máximo. Al conectar el dispositivo al puerto USB de la computadora, el sistema detecta automáticamente el dispositivo y lo considera “*disco extraíble*”. La forma de trabajar con esta unidad de disco es igual a la que utilizamos con el “disco rígido”. En equipos antiguos (como los que utilizaban Windows 98) se necesita instalar un controlador proporcionado por el fabricante. Las diversas distribuciones GNU/Linux también tienen soporte para estos dispositivos de almacenamiento.

### **Evolución de las memorias flash al disco rígido de estado sólido (SSD)**

Las unidades de estado sólido o discos de estado sólido (SSD, del inglés *Solid-State Drive*) son dispositivos de almacenamiento que usan memoria de estado sólido (tipo flash o SDRAM) en lugar de platos giratorios magnéticos como los discos rígidos tradicionales.

Asocian las ventajas de una memoria de estado sólido (menor tiempo de acceso y menor latencia) con las ventajas de un disco rígido tradicional (gran capacidad de almacenamiento), aunque todavía tienen costos elevados.

Sus capacidades típicas son 128, 256 y 512 GB. También existen unidades más pequeñas, de entre 20 y 40 GB, que se utilizan en dispositivos híbridos (el sistema operativo se instala en el SSD para aprovechar su velocidad y rapidez en el arranque, y los datos y aplicaciones se instalan en una unidad de disco rígido convencional).

Entre las **ventajas** de los SSD se pueden citar:

- Arranque más veloz.
- Altas velocidades de lectura y escritura.
- Menor consumo de energía, que reduce el consumo de batería en equipos portátiles. También es menor la emisión de calor en momentos de alta transferencia de datos.
- Resistencia a golpes, vibraciones y caídas.
- Menor peso y tamaño.
- No generan ruido, por no poseer partes móviles.

Las desventajas de los SSD pueden ser:

- Precio elevado.
- Mayor pérdida de datos ante una falla física.
- Reducción del rendimiento luego de un uso prolongado e intensivo en las memorias NAND.



**Figura 88.** Exterior e interior de un SSD.