Unidad 1

Introducción a los sistemas informáticos

En esta Unidad aprenderemos a:

- Analizar las características de un sistema informático.
- Diferenciar entre hardware y software.
- Identificar y describir los elementos funcionales de un sistema informático.
- Conocer los datos que maneja un sistema informático.
- Conocer los sistemas de numeración utilizados por un sistema informático.
- Codificar y relacionar la información en los diferentes sistemas de representación.

Y estudiaremos:

- El hardware.
- El software.
- El procesador, la memoria y los dispositivos de entrada/salida.
- Datos numéricos, alfabéticos y alfanuméricos.
- El código binario, octal y hexadecimal.
- Los códigos alfanuméricos.
- Las medidas de la información.

1. Introducción

La vida en sociedad del ser humano ha originado la necesidad de transmitir y tratar la información de una forma continuada. Con este fin, a lo largo del tiempo se han ido perfeccionando diferentes técnicas y medios. El gran avance tecnológico en las dos últimas décadas del siglo XX y en la primera del siglo XXI ha desarrollado herramientas cada vez más complejas capaces de cubrir esta necesidad con gran precisión y rapidez.

Es el **ordenador** la herramienta que actualmente nos permite el tratamiento automático de la información, facilitándonos en gran medida su organización, proceso, transmisión y almacenamiento.

El término **informática** ha ido evolucionando a lo largo del tiempo, pero en la actualidad se considera la ciencia que estudia el tratamiento automático de la información. Procede de la fusión de dos palabras: ***información*** y ***automática.***

Su desarrollo ha sido espectacular en las dos últimas décadas del siglo XX, siendo una herramienta imprescindible en comunicaciones, telefonía, medicina, aeronáutica, vigilancia, control de tráfico, etc.

Ampliación

Analiza el significado de la palabra informática a lo largo del tiempo.

2. El sistema informático, software y hardware

El ordenador se puede definir como una máquina compuesta de elementos físicos **(hardware),** en su mayoría de origen eléctrico-electrónico, capaz de realizar una gran variedad de trabajos a gran velocidad y con gran precisión.

Un ordenador está formado por un conjunto de componentes electrónicos que por sí mismos no son capaces de realizar demasiadas funciones. Estos componentes electrónicos necesitan de otros componentes no físicos que los pongan en funcionamiento; nos estamos refiriendo a programas **(software).** Los programas nos servirán para nuestro fin: procesar datos **(información).**

Para que los componentes electrónicos de un ordenador sean capaces de funcionar y realizar un proceso determinado, es necesario ejecutar un conjunto de órdenes o instrucciones. Estas instrucciones, ordenadas y agrupadas de forma adecuada, constituyen un **programa.** El conjunto de varios programas se denomina **aplicación informática.**

Pero un programa no funciona por sí solo. Es decir, tenemos los componentes electrónicos; tenemos los programas que incluyen los datos necesarios que se tienen que procesar, pero sigue faltando algo. El componente que falta, que también es un componente software y es el objeto del presente libro, es el **sistema operativo.** El sistema operativo es el componente software de un sistema informático capaz de hacer que los programas **(software)** procesen información **(datos)** sobre los componentes electrónicos de un ordenador o sistema informático **(hardware).**

Vocabulario

Un **programa,** también llamado programa informático, programa de computación o programa de ordenador, es simplemente un conjunto de instrucciones para un ordenador.

Veamos un ejemplo. Elegimos un coche; el motor, las ruedas, la amortiguación, la gasolina, los cilindros, etc., constituirían el hardware. Como un coche no puede circular por sí mismo, hace falta alguien que lo ponga en marcha, que lo arranque, que pise el acelerador, etc. Pongamos por caso que la persona sentada al volante es el equivalente a un programa informático, ya que dispone de toda la información necesaria para hacer funcionar el vehículo. Tiene órdenes precisas de cómo arrancar, de cómo circular, etc., pero es evidente que le falta la forma de interactuar con el vehículo. Es por ello por lo que los fabricantes de vehículos ponen a disposición

del usuario una serie de instrumentos, como palancas de intermitentes, llave de contacto, palancas de cambios, pedales, etc. Comparemos estos elementos con el sistema operativo o medio de comunicación del software (conductor) con el hardware (vehículo).

Cuando un programa está compuesto por varios forma lo que se denomina una **aplicación informática.** Pero una aplicación informática puede estar formada por un único programa. En este caso no se llamaría aplicación informática sino simplemente programa. Sin embargo, un programa bancario, por ejemplo, es normal que conste de varios programas. Cada programa que forma el complejo programa bancario tiene una finalidad concreta; un programa sirve para hacer nóminas, otro para gestionar préstamos hipotecarios, otro para realizar la contabilidad, etc. En este caso, el programa bancario no es un simple programa, es una aplicación informática. En definitiva, una aplicación es un macroprograma que consta de varios programas independientes aunque interrelacionados; es decir, programas que funcionan de forma autónoma, pero que pueden necesitar información procesada por otros programas dentro del macroprograma.

Instrucciones, programas y aplicaciones informáticas, en general, quedan definidos bajo el término **software.**

Un sistema informático es el conjunto de elementos físicos o hardware que son necesarios para la explotación de las aplicaciones informáticas o software.

El sistema informático o hardware es tangible, es decir, se puede ver y tocar (monitor, teclado, procesador, memoria). Los programas o aplicaciones informáticas, así como el propio sistema operativo, son intangibles; son software, pero no se puede tocar ni ver el conjunto de instrucciones del que están formados.

Entre software y hardware existe otro concepto importante dentro de un sistema informático: el ***firmware.*** Es la parte intangible (software) de componentes del hardware. Es el caso del software con el que están programadas las memorias ROM, que son hardware. Sirva de ejemplo el software pregrabado que incorporan los teléfonos móviles para realizar todas las funciones a las que están destinados. El *firmware* no es fácilmente modificable. Una vez que se introduce o se graba en un componente hardware, queda prácticamente invariable a lo largo de la vida del ordenador. El *firmware* es, por tanto, software introducido en componentes electrónicos o hardware.

Vocabulario

El ***firmware*** es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

Actividades

1. **¿Todos los programas están formados por instrucciones?**
2. **¿Hay programas que están formados por otros programas?**
3. **¿El *firmware* es hardware o software?**

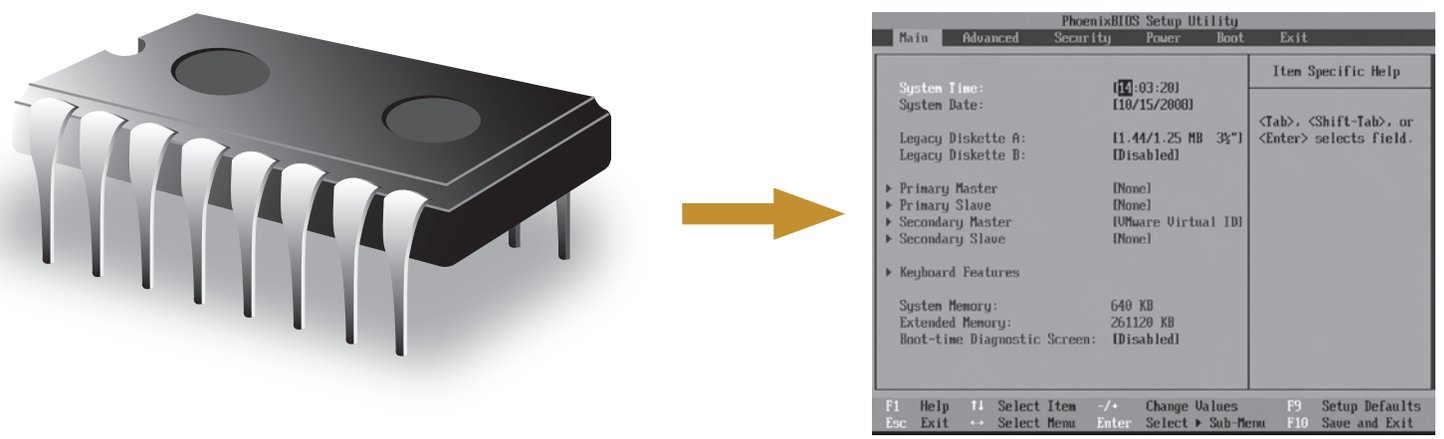


Fig. 1.1. Firmware: *ROM-BIOS.*

3. Componentes software. Sistema operativo y aplicaciones

Ten en cuenta

El **software** se clasifica en software básico (el sistema operativo) y software de aplicaciones (los programas estándar o a medida).

El software se compone de dos partes fundamentales:

- El software básico.
- El software de aplicaciones.

Se define como software básico aquella parte del software sin la cual el ordenador no puede funcionar. También recibe el nombre de **sistema operativo.**

El sistema operativo es el alma del ordenador. Sirve de comunicación entre el usuario y el hardware de la máquina. Controla los recursos hardware de la máquina según las necesidades, los programas de aplicación, el lugar donde se almacenan los datos, el momento en que hay que imprimir, el momento en que se pulsa un botón del ratón, etcétera.

El **software de aplicaciones** es la parte del software que sirve para procesar la información de forma personalizada. Lo integran los programas y los datos. Los programas permiten editar textos, extraer información, editar gráficos, realizar cálculos numéricos, etcétera.

Otra clasificación del software de aplicación se hace según este sea **estándar** o **a medida.** El estándar es el que encontramos en el mercado y está a disposición del usuario con unas características predeterminadas. Este software lo utiliza el usuario adaptado a su forma de trabajo y a las características del propio software. Por el contrario, el software a medida es el que diseñan analistas e implementan (codifican en un lenguaje de programación) programadores atendiendo a las necesidades concretas de cada usuario. En este caso, el software se adapta al usuario.

4. Componentes físicos. El hardware

Ya sabemos que el hardware es la parte física del ordenador. Son elementos tangibles. Algunos componentes hardware son la memoria, la fuente de alimentación, los cables, la tarjeta gráfica, etcétera.

Los componentes físicos del ordenador se pueden clasificar en los siguientes:

1. **Unidad central de proceso** (UCP). Consta de:
   - **Unidad aritmético-lógica** (UAL).
   - **Unidad de control** (UC).
2. **Memoria central** (MC) o RAM.
3. **Controladores.**
4. **Unidad de entrada/salida** (E/S).
5. **Buses.**
6. **Unidades periféricas** o periféricos de entrada/salida.

La Figura 1.2 recoge la integración global de estos componentes, interconectados entre sí.

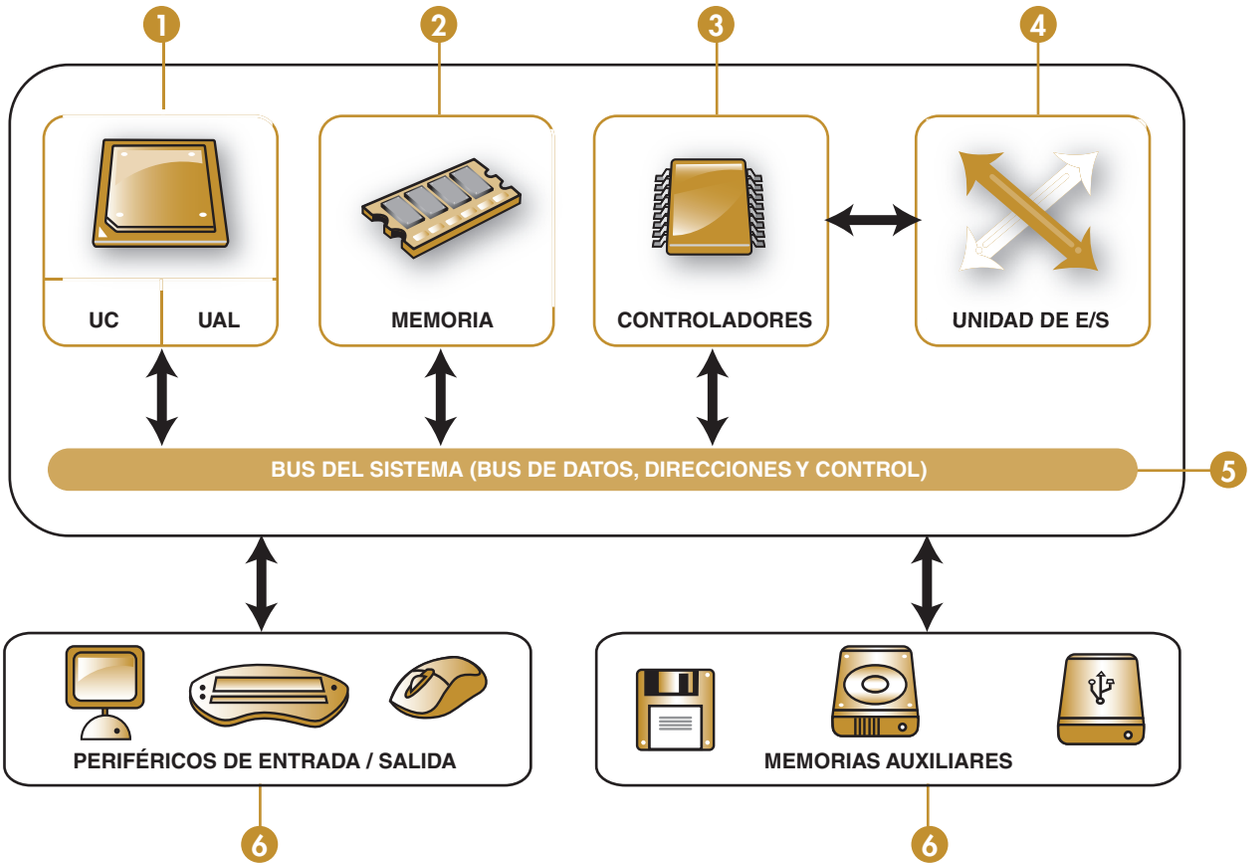


Fig. 1.2. *Componentes de un ordenador.*

4.1. Unidad Central de Proceso. Funciones, componentes, tipos y características

La unidad central de proceso o UCP, también denominada **procesador,** es el elemento encargado del control y ejecución de las operaciones que se efectúan dentro del ordenador con el fin de realizar el tratamiento automático de la información.

Ten en cuenta

La unidad central de proceso gestiona y controla todo lo que ocurre dentro de un ordenador.

Es la parte fundamental del ordenador. Se encarga de controlar todas las tareas y procesos que se realizan dentro de él. Está formado por la **unidad de control** (UC), la **unidad aritmético-lógica** (UAL) y su propia memoria, que no es la RAM. El procesador es la parte pensante del ordenador; se encarga de todo: controla los periféricos, la memoria, la información que se va a procesar, etc.

Como ya hemos anticipado, el procesador consta de dos partes fundamentales:

- Unidad de control (UC).
- Unidad aritmético-lógica (UAL).

Para que el procesador pueda trabajar necesita, además, otros componentes hardware del sistema informático: la memoria principal o central del ordenador (RAM), la unidad de entrada/salida, los periféricos de entrada/salida, los controladores y los buses.

El procesador gestiona lo que hay en memoria desde o hacia los periféricos gracias a la unidad de entrada salida, buses y controladores del sistema. Recordemos la Figura 1.2.

Veamos los componentes de la UCP:

A. Unidad de control (UC)

La **unidad de control** o **UC** es la parte pensante del ordenador; es como el director de una orquesta, ya que se encarga del gobierno y funcionamiento de los aparatos que la componen. La tarea fundamental de la UC es recibir información para interpretarla y procesarla después mediante las órdenes que envía a los otros componentes del ordenador (véase Fig. 1.3).

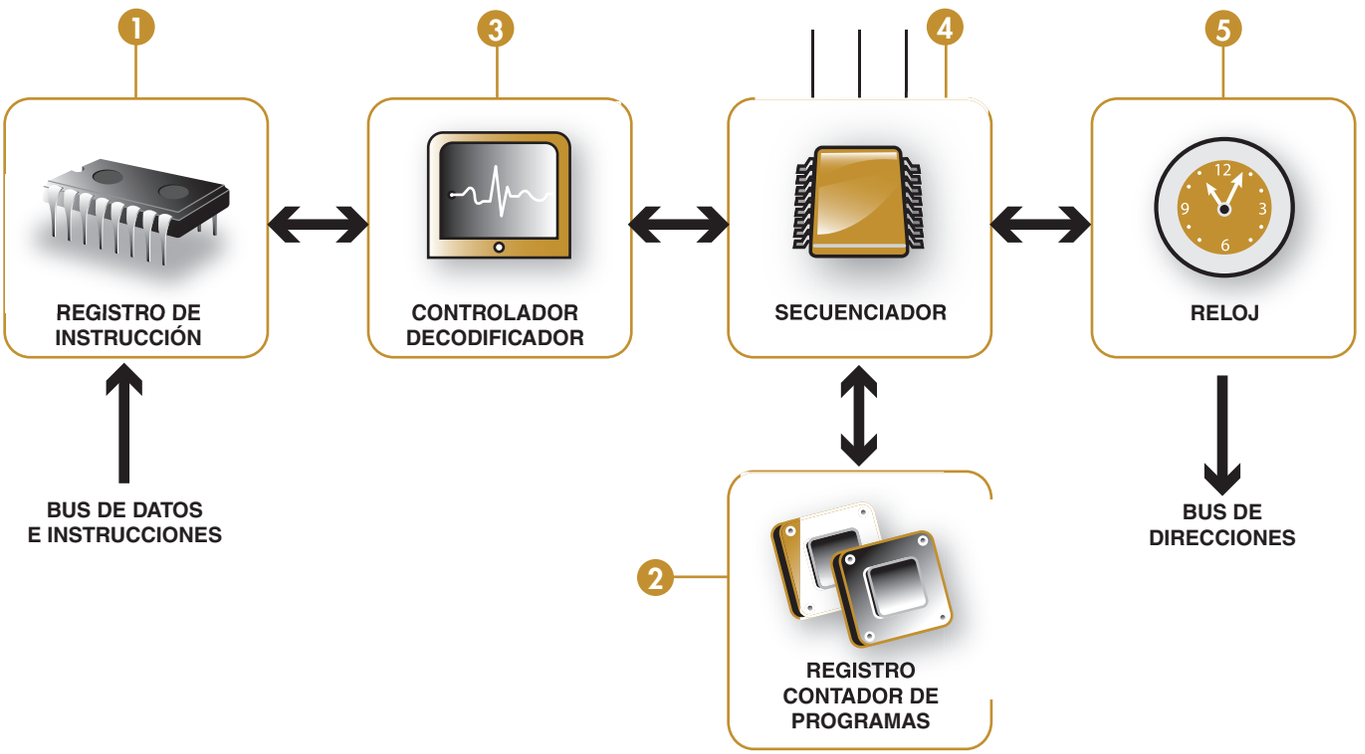


Fig. 1.3. *La unidad de control o UC.*

Se encarga de traer a la memoria interna o central del ordenador (RAM) las instrucciones necesarias para la ejecución de los programas y el procesamiento de los datos. Estas instrucciones y datos se extraen, normalmente, de los soportes de almacenamiento externo. Además, la UC interpreta y ejecuta las instrucciones en el orden adecuado para que cada una de ellas se procese en el debido instante y de forma correcta.

Para realizar todas estas operaciones, la UC dispone de algunos aliados, pequeños espacios de almacenamiento que son su esencia. Estos espacios de almacenamiento se denominan **registros.** Además de los registros, tiene otros componentes. Todos ellos se detallan a continuación:

1. **Registro de instrucción.** Es el encargado de almacenar la instrucción que se está ejecutando.
2. **Registro contador de programas.** Contiene la dirección de memoria de la siguiente instrucción a ejecutar.
3. **Controlador y decodificador.** Se encarga de interpretar la instrucción para su posterior proceso. Es el encargado de extraer el código de operación de la instrucción en curso.
4. **Secuenciador.** Genera las microórdenes necesarias para ejecutar la instrucción.
5. **Reloj.** Proporciona una sucesión de impulsos eléctricos a intervalos constantes.

B. Unidad aritmético-lógica (UAL)

La **unidad aritmético-lógica** o **UAL** es la parte de la UCP encargada de realizar operaciones aritméticas y lógicas sobre la información. Las operaciones aritméticas pueden ser suma, resta, multiplicación, división, potenciación, etc. Las lógicas son normalmente de comparación, para las que se emplean los operadores del álgebra de Boole. Algunas de estas operaciones se detallan en la Tabla 1.1.

Los elementos más importantes que componen la UAL, cuya disposición puede observarse en la Figura 1.4, son los siguientes:

Operación	Operador
Mayor que	>
Menor que	<
Mayor o igual	>=
No mayor	*NOT > (<=)*
Y lógico	*AND*
O lógico	*OR*

Tabla 1.1. *Álgebra de Boole.*

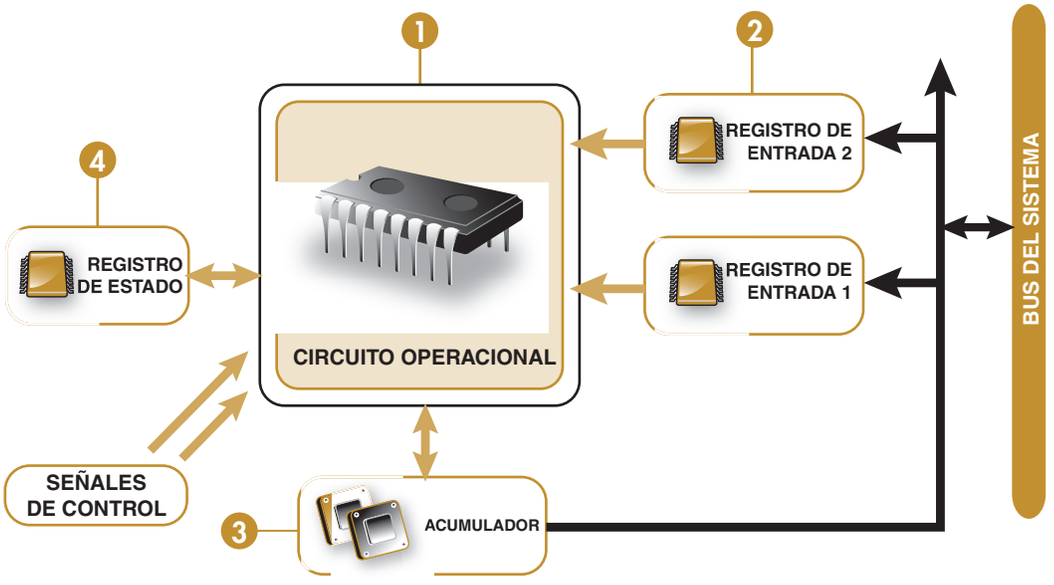


Fig. 1.4. *La unidad aritmético-lógica.*

Vocabulario

La **memoria RAM,** también llamada memoria de acceso aleatorio o memoria de acceso directo, se compone de uno o más chips y se utiliza como memoria principal para programas y datos.

1. **Operacional o circuito operacional.** Realiza las operaciones con los datos de los registros de entrada.
2. **Registros de entrada.** Contienen los operandos de la operación.
3. **Acumulador.** Almacena los resultados de las operaciones.
4. **Registro de estado.** Registra las condiciones de la operación anterior.

4.2. La memoria. Funciones, tipos y características

El ordenador almacena dentro de su memoria interna todos los programas y datos con los que se va a trabajar y que van a ser procesados. Los dos tipos de memoria esenciales con los que puede trabajar el ordenador son:

- **Memorias de almacenamiento externo.** Se les da esta denominación a los soportes de almacenamiento, ya que son capaces de almacenar información. Son memorias externas: discos duros, disquetes, cintas DAT, *pen drives,* etc., y aunque estén físicamente dentro de la carcasa del ordenador, como es el caso de los discos duros, la denominación de externas es para diferenciarlas precisamente de la propia RAM.

Actividades

4. **¿Qué componente de la UCP es el que indica la instrucción que hay que procesar?**
5. **¿Qué componente la procesa?**
6. **¿Tiene memoria la UC?**

Estas memorias son más lentas que la propia memoria principal, ya que constan de componentes electrónicos y mecánicos. Son no volátiles, de tal forma que la información permanece en ellas incluso después de quitar el suministro de energía eléctrica al ordenador.

- **Memoria interna.** Dentro del ordenador existen varios tipos de memorias que no son consideradas externas. Son las siguientes:
  - **RAM** *(Random Access Memory)*. En ella es posible almacenar y modificar información y es lo que se conoce como memoria principal, memoria central o memoria de acceso directo.
  - **ROM** *(Read Only Memory)*. Es una memoria de solo lectura, cuya información no puede ser modificada y que sirve básicamente para poder inicializar el sistema informático.

La memoria interna, principal o central **(MC)** es la que está situada físicamente dentro de la carcasa del ordenador.

A. Memoria RAM

La **memoria RAM** es un componente necesario para que se pueda procesar la información. Casi todo, por no decir todo, lo que se tiene que procesar dentro del ordenador debe pasar tarde o temprano por la memoria central.

Los elementos que componen la memoria central o principal, como se aprecia en la Figura 1.5, son los siguientes:

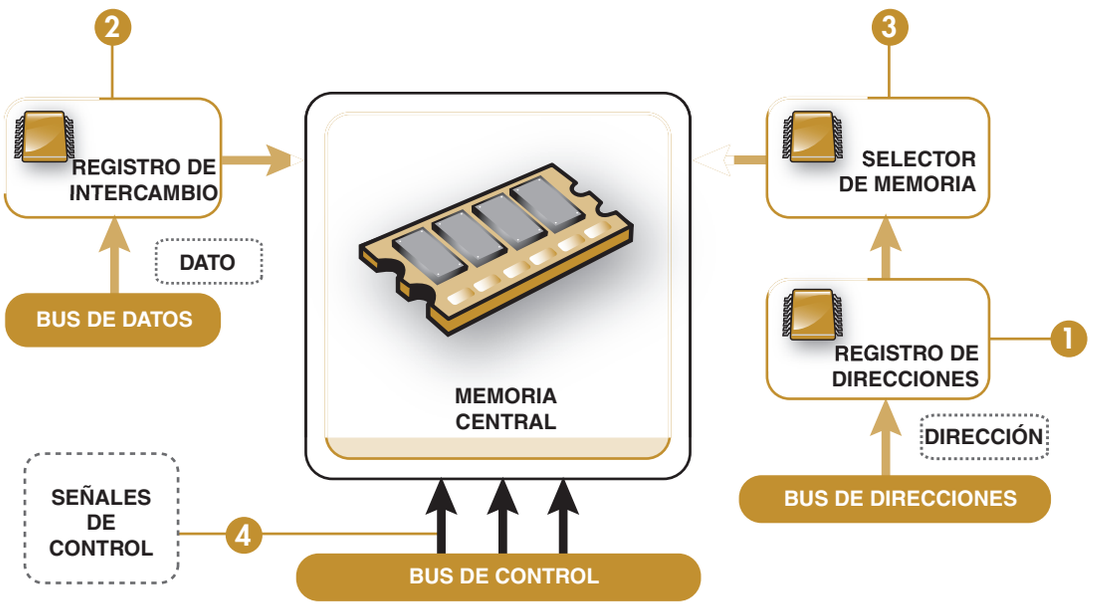


Fig. 1.5. *La memoria central.*

1. **Registro de direcciones.** Contiene la dirección de la celda o posición de memoria a la que se va a acceder.
2. **Registro de intercambio.** Recibe los datos en operaciones de lectura y almacena los datos en las operaciones de escritura.
3. **Selector de memoria.** Se activa cada vez que hay que leer o escribir conectando la celda o posición de memoria con el registro de intercambio.
4. **Señales de control.** Indica si una operación es de lectura o escritura.

Investigación

Analiza en la Web los dos tipos de **memoria caché** más comunes.

La memoria central está formada por componentes electrónicos **(biestables)** capaces de almacenar información en forma de ceros y unos (sistema binario). Cada información de este tipo recibe el nombre de **bit.**

En la RAM tienen que estar físicamente ubicados los programas y los datos que se tienen que procesar. Cuando ejecutamos un programa, como por ejemplo Word, Writer, etc., este pasa del soporte de almacenamiento o memoria externa en el que está almacenado de forma permanente, a cargarse en memoria central (operación de lectura).

Además de la memoria central, lo normal es que los ordenadores incorporen otro tipo de memoria para agilizar los cálculos que realizan los programas. Suelen ser memorias intermedias colocadas entre la RAM y el procesador, que almacenan temporalmente la información a procesar. Este tipo de memorias reciben el nombre de **memoria caché** y no son RAM propiamente dicha, sino otro tipo de memorias internas que almacenan la información que se utiliza con más frecuencia.

Por otro lado, no hay que confundir los soportes de almacenamiento con la memoria interna; es decir, un disco duro no es memoria interna. El disco duro se considera memoria externa o auxiliar. Los disquetes, CD-ROM, cintas, *pen drives,* etc., son soportes de almacenamiento. Se denominan memorias externas y, al igual que la memoria interna, todas almacenan información. Lo que ocurre es que la memoria interna almacena la información solo temporalmente, para procesarla, mientras que los soportes de almacenamiento externo tienen la función principal de almacenar la información de forma permanente.

Físicamente hablando, los componentes electrónicos que forman la MC son las denominadas **celdillas** o **biestables,** que actúan como pequeños condensadores, de tal forma que la presencia de energía dentro de ellas puede traducirse como un uno (1) lógico y la ausencia de energía como un cero (0) lógico (Figura 1.6).

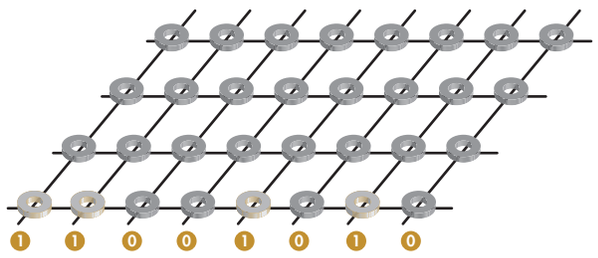


Fig. 1.6. *Esquema de las celdillas de memoria.*

La información en memoria se suele almacenar en **bloques.** Estos bloques suelen ser de ocho celdillas; es decir, equivalen a 8 bits y se denominan **byte** (combinación de ceros y unos). Cada conjunto de ellos representa un carácter, es decir, cualquier letra o número como combinación de 8 bits.

Estos condensadores, como tales, transcurrido cierto tiempo, se van descargando. Evidentemente, para no perder la información de la memoria, el propio sistema informático tendrá que proceder a recargarlos antes de que se descarguen definitivamente. Este proceso es conocido como **refresco de memoria.**

- **DRAM** *(**D**ynamic **R**AM).* Es un tipo de memoria RAM electrónica construida mediante condensadores. Cuando un condensador está cargado se dice que almacena un **BIT a uno.** Si está descargado, el valor del **BIT es cero.** Para mantener las celdillas cargadas, este tipo de memoria necesita refrescarse cada cierto tiempo: el refresco de una memoria RAM consiste en recargar nuevamente con energía los condensadores que tienen almacenado un uno para evitar que la información se pierda (de ahí lo de *Dynamic*). La memoria DRAM es más lenta que la memoria SRAM, pero mucho más barata de fabricar.

- **SRAM** *(**S**tatic **R**AM).* Es un tipo de memoria RAM alternativa a la DRAM que no necesita refrescarse. SRAM y DRAM son memorias volátiles, lo que significa que cuando se corta el suministro de corriente, los datos almacenados se pierden. Debido al alto coste de fabricación de la SRAM y a su alta velocidad, suele utilizarse como **memoria caché.**

Actividades

7. **¿Qué memoria pierde la información cuando se deja de suministrar corriente al equipo?**
8. **¿Qué memoria es más rápida, la RAM o la caché?**

- **SDRAM** (***S**ynchronous **D**ynamic **RAM***). Es una memoria que incorpora la capacidad de la DRAM y la velocidad de la SRAM; es decir, necesita refresco de sus celdas, pero en un intervalo superior de tiempo. Esta memoria es la que incorporan en la actualidad la mayoría de los ordenadores personales.
- **DDRAM** *(**D**ouble **D**ata **R**ate)* o memoria de doble recarga o memoria de doble tasa de transferencia. Compuesta por memorias SDRAM, tiene la característica de que se refresca dos veces por impulso de reloj. Es una memoria de funcionamiento muy complejo, pero tiene la ventaja de ser prácticamente el doble de rápida que cualquiera de las anteriores.

En la actualidad, una de las características fundamentales de las memorias RAM es la velocidad con que la información se puede almacenar en ellas. Esta velocidad es mayor cuanto menos se tarde en acceder a la posición de memoria requerida en cada instante. La velocidad se mide en nanosegundos (60, 70, 80, 100,...). Cuanto menor sea el tiempo de acceso, más rápido será el acceso que se pueda realizar a cualquier posición de memoria para poder grabar o leer su información.

B. Memoria ROM

La **memoria ROM** o memoria de solo lectura contiene programas especiales que sirven para cargar e iniciar el arranque del ordenador. En ella se encuentra almacenada toda la información referente a los componentes hardware de los que consta nuestro equipo.

Posteriormente, será labor del sistema operativo realizar el resto de operaciones para poder empezar a utilizar el ordenador.

Investigación

Analiza en la Web los tipos de BIOS más comercializadas.

El software que integra la ROM forma el **BIOS** del ordenador *(**B**asic **I**nput **O**utput **S**ystem)* o sistema básico de entrada/salida.

El BIOS se encuentra físicamente en varias partes del ordenador. El componente principal está en la placa base. Inicialmente, los BIOS se programaban sobre memorias de tipo ROM, lo que implicaba que cualquier modificación en el sistema no podía realizarse a menos que lo hiciese el fabricante. Había que sustituir el componente electrónico para modificar la configuración del BIOS. Por eso, posteriormente, el BIOS se montó en memorias de tipo **PROM** *(**P**rogrammable **R**ead **O**nly **M**emory)*, que son programables una sola vez y después de haber sido montadas en la placa.

CEO

En la Web del Centro de Enseñanza Online encontrarás más información de configuración de los diferentes tipos de BIOS más comunes.

El BIOS es un código que localiza y carga el sistema operativo en la RAM; es un software elemental instalado en una pequeña ROM de la placa base que permite que esta comience a funcionar. Proporciona las órdenes básicas para poner en funcionamiento el hardware indispensable para empezar a trabajar. Como mínimo, maneja el teclado y proporciona salida básica (emitiendo pitidos normalizados por el altavoz del ordenador si se producen fallos) durante el arranque.

En la actualidad, se utilizan las memorias de tipo **EPROM** *(**E**rasable **P**rogrammable **R**ead **O**nly **M**emory)*, que permiten cambiar la configuración asignada. Este proceso es complejo, pero no implica realizar operaciones físicas sobre los componentes que están montados.

Actividades

9. **¿Todas las memorias ROM son de solo lectura?**
10. **¿Cómo se accede al BIOS del equipo?**

Todas estas memorias son no volátiles, y la información que contienen no desaparece nunca debido a que están programadas de fábrica. No necesitan ningún suministro de energía para mantener su configuración.

La **CMOS** *(**C**omplementary **M**etal **O**xide **S**emiconductor)* es un tipo de memoria interna del ordenador que se caracteriza por consumir muy poca energía eléctrica, lo que la hace idónea para almacenar datos del BIOS.

El BIOS es un tipo de memoria que en los ordenadores se utiliza para guardar los datos básicos de hardware y de configuración. Por ejemplo, en él se guarda la información sobre los discos duros (cuántos y de qué características) y otras informaciones como la fecha y la hora. Para que toda la información que mantiene no se borre, es necesario que la CMOS siempre tenga corriente eléctrica. Cuando el ordenador está apagado, obtiene energía de una pequeña pila o batería ubicada en la placa base.

La configuración del BIOS se puede modificar si instalamos un nuevo disco duro, si queremos cambiar la fecha, la hora del sistema, etc. Para acceder al BIOS y poder modificar sus valores, hay que pulsar las teclas **F2** o **Supr** durante el proceso de inicio del equipo, dependiendo del BIOS de cada equipo. Así se accede al **SETUP** del equipo, en el que se configuran las opciones de inicio básicas del ordenador.

Otros tipos de memorias internas que no son RAM como tal, pueden ser las memorias que incorporan las tarjetas gráficas, que liberan a la RAM de las tareas de procesamiento gráfico. Así, la memoria **VRAM** o memoria de vídeo se utiliza para almacenar las imágenes que se quieren visualizar, en vez de hacerlo directamente sobre la RAM. Actualmente, este tipo de memoria es fundamental debido a la evolución de la tecnología multimedia.

En la actualidad, la mayoría de los ordenadores incorporan en la propia tarjeta o adaptador gráfico la llamada **SGDRAM** *(**S**uper **G**raphics **D**ynamic **R**andom **A**ccess **M**emory)*. Se trata de una memoria de elevada capacidad, a veces verdaderamente elevada, que se caracteriza por su alta velocidad y bajo consumo.

La **CDRAM,** por ejemplo, es un tipo de memoria que actúa entre el procesador y el periférico correspondiente. En algunos casos, estas memorias actúan como memorias caché (memoria intermedia de alta velocidad). Suelen ir asociadas a determinados dispositivos, como unidades de CD-ROM y dispositivos de entrada/salida, para liberar a la RAM de operaciones innecesarias.

En cuanto a la estructura de la memoria, desde un principio, e independientemente del sistema operativo, la memoria se ha estructurado en varios niveles (véase Fig. 1.7).

1. **Memoria convencional.** De 0 a 640 Kb.
2. **Memoria superior.** De 641 Kb hasta 1024 Kb (1 Mb).
3. **Memoria extendida.** De 1 025 Kb hasta el límite de la placa base del equipo.

Los fabricantes de ordenadores han dividido desde un principio la estructura en esas tres partes fundamentales, que son gestionadas de forma diferente por cada sistema operativo. Así, por ejemplo, MS-DOS gestiona la memoria dividiéndola realmente en estas tres capas. Windows 9X, XP, Vista 2003 Server, 2008 Server, UNIX o Linux gestionan la memoria de forma diferente, olvidándose casi por completo de esta estructura. Ya veremos qué gestión se hace de la memoria en cada uno de los sistemas operativos.

En cualquier caso, esta cuestión es fundamental, pues una cosa es el modo en que los ordenadores reconocen la memoria después de ser fabricados (tres capas) y otra muy distinta el modo en que el sistema operativo gestiona las diferentes capas de la RAM.

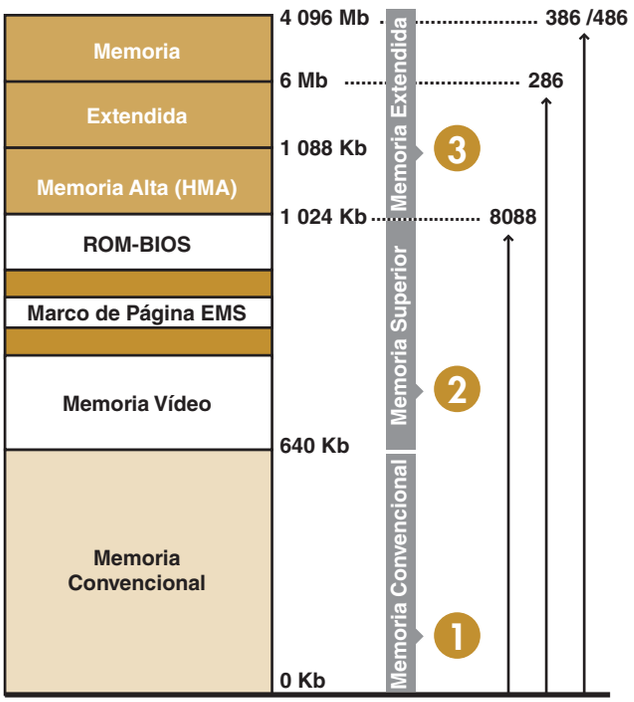


Fig. 1.7. *Esquema de la estructura de la memoria.*

Para acceder a las celdillas de memoria hay que atender al concepto de **dirección de memoria.** Esta dirección es la situación del componente electrónico dentro del conjunto de componentes de la memoria. De esta forma, cuando se accede a una dirección de memoria, lo que se hace es acceder a un conjunto de biestables (condensadores). Cada uno de estos biestables físicos referencia un **bit** lógico (0,1). El bit se define como la mínima unidad de información.

Ten en cuenta

A la información que hay en la memoria se accede a través de su dirección, que es la posición que ocupa dentro de la misma.

El conjunto de 8 bits a los que se accede se denomina byte, carácter o palabra. A partir de aquí, la información se mide como conjunto de bytes, es decir, bloques de 8 bits.

El direccionamiento es una operación que se realiza cuando el procesador ejecuta o interpreta una instrucción. Toda instrucción está compuesta por un **código de operación** y un **operando.** El código de operación es la instrucción en sí (suma, resta, multiplicación, etc.) y el operando es el dato o información que se va a procesar. Según el método utilizado, la rapidez de ejecución de un programa será mayor o menor. Los llamados **modos de direccionamiento** son las diferentes formas de acceder a los operandos en cada instrucción. La Figura 1.8 muestra los modos de direccionamiento:

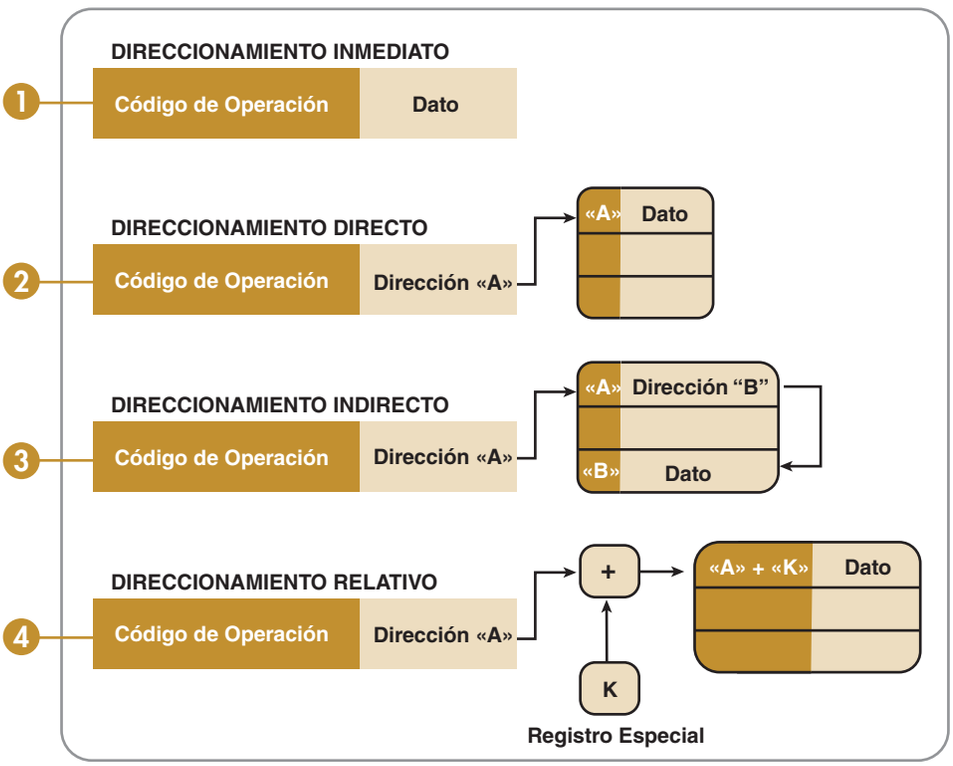


Fig. 1.8. *Esquema de los diferentes direccionamientos.*

1. **Direccionamiento inmediato.** En la instrucción está incluido directamente el operando o dato.

2. **Direccionamiento directo.** En la instrucción, el campo del operando contiene la dirección en memoria donde se encuentra el operando.

3. **Direccionamiento indirecto.** El campo del operando contiene una dirección de memoria en la que se encuentra la dirección efectiva del operando.

4. **Direccionamiento relativo.** La dirección del dato que interviene en la instrucción se obtiene sumando a la dirección de la propia instrucción una cantidad fija, que normalmente está contenida en un registro de tipo especial.

Ejemplo

Tipo de direccionamiento que se utiliza según la instrucción a ejecutar:

a) Si la instrucción a ejecutar es ***SUMA 56,*** el procesador accede directamente al dato, en este caso, a la posición 56 de memoria en la que se encuentra el dato, y realiza la operación. El resultado de sumar se queda en la misma posición de memoria. Es lo que en programación se denomina un acumulador. Si en la posición 56 de memoria hay, por ejemplo, un **8,** el resultado de la operación ***SUMA 56*** genera un valor de **16,** que se almacena en la misma posición de memoria, es decir, en la 56. Ahora en la posición 56 hay un **16.**

b) Si la operación a ejecutar es ***SUMA 10 20,*** el procesador accede a la posición 10 y a la posición 20 de memoria para extraer el dato que hay en cada una de ellas y almacena el resultado en la posición 20 de memoria. Si la posición 10 de memoria contiene un **3** y la posición 20 un **2,** el resultado es la suma de **2** y **3,** y se almacena en la segunda posición de memoria. De esta forma, la posición 10 seguirá teniendo un **3** y en la 20 habrá un **5** y no el **2** que había al principio.

c) Si la operación a ejecutar es ***RESTA 32 12 45,*** el procesador resta el contenido de la posición 32 de memoria y el contenido de la posición 12, y deja el resultado en la posición 45. Si, por ejemplo, la posición 32 contiene un **5** y la posición 12 un **3,** independientemente de lo que contenga la posición 45, el resultado quedará así:

- La posición 32 contiene un **5.**
- La posición 12 contiene un **3.**
- La posición 45 contendrá el resultado de la resta, es decir, **2.**

En los tres casos el direccionamiento utilizado es el directo, que es el más habitual en este tipo de operaciones aritméticas.

4.3. Unidades de entrada/salida y buses

La **unidad de entrada/salida** sirve para comunicar el procesador y el resto de componentes internos del ordenador con los periféricos de entrada/salida y las memorias de almacenamiento externo o auxiliares.

Recordemos que actualmente las denominaciones UC y UAL han dejado de tener importancia aisladamente. Hoy en día, el conjunto de estos dos componentes se denomina procesador. La memoria RAM y la unidad de entrada/salida no forman parte, como tales, del procesador, sino que son parte del hardware sin las que este no es capaz de realizar prácticamente ninguna operación.

Hay otro componente importante dentro del ordenador que está relacionado directamente con el procesador: el **bus.**

El bus es el elemento responsable de establecer una correcta interacción entre los diferentes componentes del ordenador. Es, por lo tanto, el dispositivo principal de comunicación. En un sentido físico, se define como un conjunto de líneas de hardware (metálicas o físicas) utilizadas para la transmisión de datos entre los componentes de un sistema informático. En cambio, en sentido figurado es una ruta compartida que conecta diferentes partes del sistema.

La evolución de los buses a lo largo de la historia ha sido determinante para la evolución de los sistemas operativos. Al fabricarse buses mucho más rápidos y con más líneas, los sistemas operativos han ido mejorando y aportando nuevas funcionalidades que antiguamente no podían ser implementadas por falta de velocidad en los buses. Hoy por hoy, el bus sigue determinando en gran medida la velocidad de proceso de un equipo, ya que sigue siendo uno de los componentes hardware que mayores limitaciones tiene.

Actividades

11. **¿Cómo se accede al contenido de una celda de memoria?**
12. **¿Se puede modificar el contenido de una celda o posición de memoria?**

En lo referente a la estructura de interconexión mediante los buses, existen de dos tipos:

- **Bus único.** Considera a la memoria y a los periféricos como posiciones de memoria, y hace un símil de las operaciones E/S con las de escritura/lectura en memoria. Estas equivalencias consideradas por este bus hacen que no permita controladores DMA ***(Direct Access Memory)*** de acceso directo a memoria.
- **Bus dedicado.** Este, en cambio, al considerar la memoria y periféricos como dos componentes diferentes, permite controladores DMA.

El bus dedicado (véanse Figs. 1.2, 1.3, 1.4 y 1.5) contiene varias subcategorías más que son las siguientes:

- **Bus de datos.** Transmite información entre la CPU y los periféricos.
- **Bus de direcciones.** Identifica el dispositivo al que va destinada la información que se transmite por el bus de datos.
- **Bus de control o de sistema.** Organiza y redirige hacia el bus pertinente la información que se tiene que transmitir.

@ Investigación

Analiza en la Web los tipos de procesadores de última generación.

La capacidad operativa del bus depende del propio sistema, de la velocidad de este, y la «anchura» del bus (número de conductos de datos que operan en paralelo) depende de los bits que se pueden transmitir simultáneamente, según el tipo de procesador que incorpore el equipo.

El bus es como una autopista en la que el tráfico es muy intenso. Por eso, el tipo de bus que incorpore el ordenador determinará que este sea más rápido o más lento.

El bus se caracteriza por el número y la disposición de sus líneas (cada una de ellas es capaz de transmitir un bit, que es la unidad mínima de transmisión de la información). Concretamente, en los primeros PC era de 8 bits; es decir, solo contaban con ocho líneas de datos. En la actualidad, los más extendidos son los de 16, 32, 64, 128 bits o superiores (véase Tabla 1.2).

Procesador	BUS datos/BUS direcciones
8086	16 / 8
8088	16 / 8
80286	16 / 16
80386	32 / 16
80486	32 / 66
PENTIUM	50 / 66
AMD K-6 2	32 / 100
PENTIUM II	32 / 100
CELERON	32 / 66
PENTIUM III	32 /100
AMD ATHLON	32 / 100
PENTIUM IV	32 / 64
PENTIUM CORE DUO	64 / 133

Tabla 1.2. *Características del bus de datos y direcciones.*

El número de bits que circulan define el número de líneas de que se dispone para realizar el paso de información de un componente a otro. Son como los carriles de una autopista: cuantos más carriles haya, más vehículos podrán circular por ella.

La Figura 1.9 ilustra cómo podría representarse físicamente el bus del sistema, relacionando el procesador y la memoria, aunque hay que tener en cuenta que el bus relaciona básicamente todos los componentes del ordenador.

La estructura es la siguiente:

1. **Procesador.**
2. **Buses.**
3. **Memoria RAM.**

También es muy importante la velocidad con la que los bits circulan por el bus. Esta velocidad se mide en megahercios, y de ello depende el rendimiento global del equipo. Hay buses a 66 Mhz, pasando por toda una gama que va hasta más de 1 066 Mhz en los ordenadores que montan procesadores de última generación. Comparémoslo con una autopista o carretera: no es lo mismo que exista una limitación de 90 km/h que otra de 130 km/h. Si un bus tiene muchas líneas y son muy rápidas, mejor para el rendimiento del ordenador.

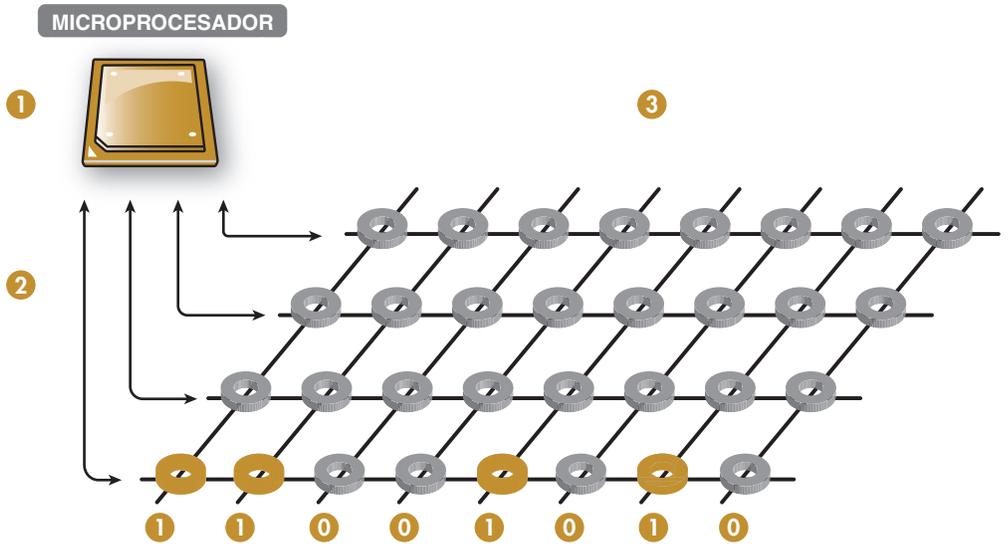


Fig. 1.9. *Esquema del bus del sistema.*

La frecuencia o velocidad del bus queda determinada por los impulsos de reloj. El reloj es, por tanto, el componente que determina la velocidad, ya que a mayor frecuencia en Mhz, más rápida es la circulación de bits por las líneas del bus.

Y no solo eso. El bus determina la arquitectura y, por tanto, su tamaño determina el del registro de instrucción. Así, el código de operación puede ser mayor, siendo posible ejecutar un mayor número de operaciones. Esto produce un aumento de potencia, no por mayor rapidez, sino por mayor complejidad de las instrucciones.

4.4. Los periféricos

Los **periféricos** son dispositivos hardware con los cuales el usuario puede interactuar con el ordenador (teclado, ratón, monitor), almacenar o leer datos y/o programas (dispositivos de almacenamiento o memorias auxiliares), imprimir resultados (impresoras), etcétera.

Se denominan periféricos, por ejemplo, los dispositivos que sirven para introducir datos y programas en el ordenador desde el exterior hacia su memoria central para que puedan ser utilizados. Son los llamados periféricos de entrada: teclados, ratones, etc.

También hay periféricos que sirven para extraer información desde el ordenador hacia el exterior, como impresoras o monitores.

Los hay que sirven para ambas cosas, como discos duros, CD-ROM regrabables, disquetes, etc.

Los periféricos se conectan con el ordenador, es decir, con la UCP y sus componentes, a través de los denominados **puertos** o conectores externos. Esta gestión la lleva a cabo otra parte esencial del ordenador: la **unidad de entrada/salida,** componente hardware usado para la gestión de periféricos.

En una primera aproximación podemos hacer una clasificación de los periféricos teniendo en cuenta desde o hacia dónde envían información. Es decir, la clasificación se hace atendiendo a que la información que circula a través del bus de datos lo haga desde el periférico a la memoria central **(periférico de entrada)** o viceversa **(periférico de salida).**

Conectado el periférico al ordenador a través del cable o conector correspondiente, la información que se envía o transmite circula dentro del ordenador a través de los buses vistos anteriormente.

Actividades

13. **¿Puede ser más rápido un equipo con un bus de 16 bits que otro con un bus de 32 bits?**
14. **El bus de direcciones de un equipo, ¿qué indica exactamente?**

Truco

Para diferenciar si un periférico es de entrada, salida o de entrada/salida, solamente tienes que fijarte en si el periférico envía información a la memoria del ordenador, en cuyo caso es de entrada. Si es la memoria la que envía información al periférico, entonces es de salida. Si se envía o recibe información simultáneamente desde la memoria, el periférico es de entrada/salida.

Actividades

15. **¿Cuántos tipos de *drivers* puede tener un dispositivo periférico?**
16. **¿Puede funcionar un periférico sin *driver*?**

Muchos de los periféricos de entrada/salida necesitan un tipo de software especial para ser configurados; en otras palabras, para utilizar una impresora, por ejemplo, primero hay que instalarla. Esto significa que es necesario introducir dentro de la configuración del ordenador y acorde con nuestro software básico unos programas específicos que permitan al sistema operativo reconocer el periférico y utilizarlo de forma correcta. Estos programas se denominan ***drivers*** o **controladores.**

5. Componentes lógicos. El software

Una vez vistos los componentes hardware de un sistema informático, hay que tener en cuenta que para que un sistema informático sea útil es necesario que procese información.

La información que se procesa en un sistema informático puede ser de diferentes tipos: textos, gráficos, música, etc.

A continuación exploraremos de forma general los tipos de datos que procesa un sistema informático.

5.1. Los datos. Tipos de datos

Por sentido común, para que el ordenador funcione, necesita información con la que trabajar. Esta información es de varios tipos dependiendo de su función. Básicamente, el ordenador sirve para procesar información en forma de datos, datos que pueden ser textos, imágenes, tablas de hojas de cálculo, etc.

El ordenador también maneja información que servirá para procesar esos datos. En este caso, nos estamos refiriendo a programas o aplicaciones informáticas, como los procesadores de textos, que se utilizan para procesar datos en formato texto, las herramientas de diseño gráfico que se utilizan para procesar datos en formato de imagen, etc.

Por último, un sistema informático necesita otro tipo de software fundamental. Este software está compuesto de programas y datos que ponen en funcionamiento las aplicaciones informáticas, las cuales procesan a su vez sus propios datos. En este caso nos referimos al **sistema operativo,** definido como el componente software que sirve para que la información pueda ser procesada por las aplicaciones informáticas mediante la utilización de todos los componentes hardware del sistema informático. El sistema operativo consta de programas propios que sirven para realizar otras funciones.

La primera clasificación que podemos hacer de los tipos de datos la observamos en la Figura 1.10 y es la siguiente:

1. **Datos de entrada.** Son los que se suministran al ordenador desde los periféricos de entrada (teclado, ratón, módem, escáner, etc.) o desde los diferentes soportes de información (disquetes, discos duros, CD-ROM, etc.). Forman la primera fase del tratamiento automático de la información: ***entrada.***

2. **Datos intermedios.** Son aquellos que se obtienen en la segunda fase del tratamiento automático o de la información: ***proceso.***

3. **Datos de salida.** También llamados resultados, completan el proceso del tratamiento automático de la información: ***salida.***

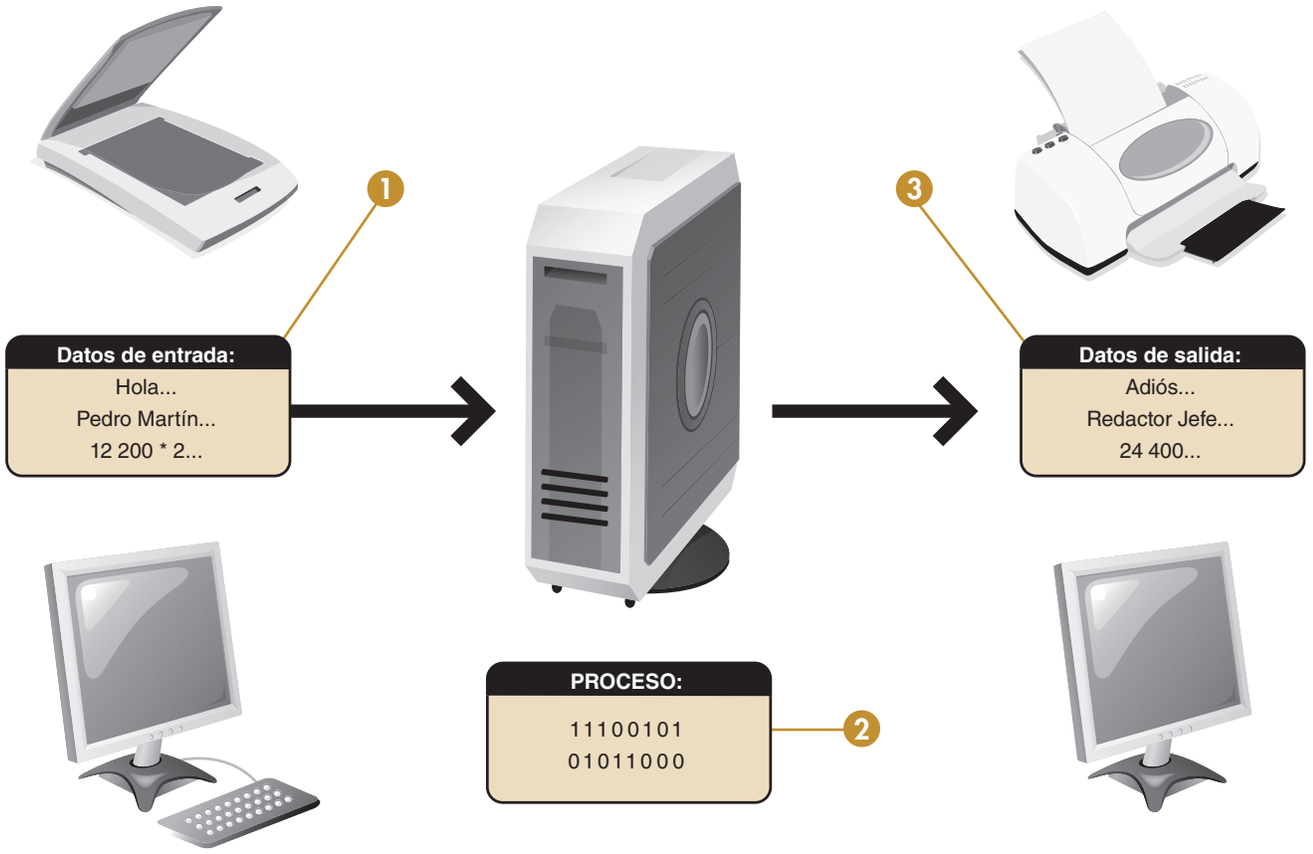


Fig. 1.10. *Tratamiento automático de la información. Tipos de datos.*

Puede observarse a través de las diferentes unidades periféricas de salida (monitor, impresora, plotter, etc.), que con su posterior distribución y análisis, completan el proceso.

Otra clasificación que podemos hacer de los datos, según varíen o no durante el proceso, es la siguiente:

- **Datos fijos.** Son los que permanecerán constantes durante el proceso o programa que se les aplique. Los datos fijos reciben el nombre de **constantes.** Un ejemplo es un programa que emita facturas en euros y pesetas; es evidente que el cambio del euro será el mismo en todo el proceso.
- **Datos variables.** Son aquellos que sí se modifican a lo largo del proceso según sucedan determinadas condiciones o acciones realizadas por los programas.

Según la forma de ser utilizados por el ordenador, otra clasificación es:

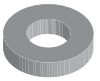
- **Datos numéricos.** Son los dígitos del 0 al 9.
- **Datos alfabéticos.** Son las letras mayúsculas y minúsculas de la A hasta la Z.
- **Datos alfanuméricos.** Son una combinación de los anteriores, más una serie de caracteres especiales (*, /, -, %, etc.).

En general, todos los sistemas operativos, salvo algunos de los considerados antiguos, trabajan con los datos de la misma forma y con los mismos tipos de datos.

Los datos son procesados por los diferentes programas que maneja el sistema operativo o por los programas que ejecutan los usuarios. Como veremos a continuación, los datos que se procesan en un sistema informático se implementan en códigos numéricos o alfanuméricos para poder utilizarlos.

Ten en cuenta

Las **celdillas de memoria** pueden tomar los dos estados siguientes:


Indica ausencia de corriente eléctrica.

Indica presencia de corriente eléctrica.

5.2. Los sistemas de codificación

A. Introducción a los sistemas de codificación

Los sistemas de codificación se utilizan para procesar la información que el usuario entiende y el ordenador no. Es evidente que el usuario y el sistema informático trabajan en lenguajes diferentes.

Centrémonos en la memoria por un momento. La memoria no puede almacenar la letra A o el carácter *.

La memoria del ordenador, y por extensión el resto de componentes internos, no entiende de letras o números. Solamente entiende de corriente eléctrica.

Por eso, cuando el usuario quiere almacenar una letra en memoria, por ejemplo, la primera letra de su documento de texto, el ordenador, gracias al sistema operativo y a los componentes de hardware, se encarga de transformar la letra y de almacenarla en un conjunto (normalmente 8 bits) de impulsos eléctricos.

Si, por el contrario, leemos de una posición de memoria, primero se analizan las celdillas correspondientes. Cuando se han analizado ocho de ellas, se sabe, por diseño del propio sistema operativo y gracias a la equivalencia del código, que se ha leído un byte o carácter como conjunto de ocho bits.

Cada posición magnetizada se convierte en un uno y cada posición no magnetizada en un cero. Se busca en la tabla de códigos y se compara la combinación de esos ocho bits, obteniendo la equivalencia con el carácter concreto.

En ese caso se visualiza, por ejemplo, el carácter equivalente al byte leído y no se visualizan los ocho bits.

Ejemplo

Cómo se almacena el carácter \ en memoria

El sistema operativo y el resto de componentes hardware tienen que transformar ese carácter en alguna combinación válida de impulsos eléctricos para almacenarlo. En este caso, las ocho celdillas de memoria correspondientes se magnetizarán o no de la siguiente forma (Fig. 1.11):

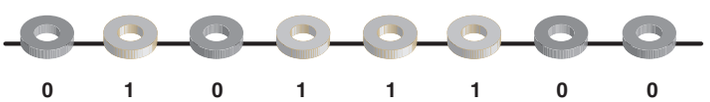


Fig. 1.11. *Ejemplo de magnetización eléctrica.*

Podemos meditar en qué se basa el sistema informático para saber qué celdilla tiene que magnetizar o no. Sencilla y llanamente, se basa en un CÓDIGO. Cuando tecleamos el carácter, se busca dentro de una tabla (código ASCII o UNICODE que veremos más adelante) la correspondencia adecuada.

Esta tabla de códigos la introduce el fabricante del sistema operativo dentro del conjunto de instrucciones y datos que lo forman.

Es un estándar internacional y todos los fabricantes de software y hardware lo conocen y lo utilizan.

De esta forma, todos ellos tienen las mismas equivalencias y a cada uno de ellos le resulta fácil interpretar la información que procesan otros programas o componentes hardware.

En el ejemplo anterior, el conjunto de ocho celdillas de memoria se han magnetizado al introducir el carácter \, ya que el sistema operativo ha leído en su tabla de códigos el siguiente valor:

0 1 0 1 1 1 0 0

Lo ha interpretado y ha magnetizado la celdilla (1) de memoria o no (0) posicionalmente, para obtener la combinación de esos ocho valores, dígitos binarios o bits.

B. Sistemas de numeración

Se define un **sistema de numeración** como el conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.

Estos sistemas se caracterizan por la **base** a la que hacen referencia y que determina el distinto número de símbolos que lo componen. Nosotros utilizamos el sistema de numeración en base 10, compuesto por 10 símbolos diferentes (del 0 al 9).

Los sistemas de numeración que utilizamos son sistemas posicionales, es decir, el valor relativo que cada símbolo representa queda determinado por su valor absoluto y por la posición que ocupe dicho símbolo en un conjunto.

$$283 = 2 \cdot 10^2 + 8 \cdot 10^1 + 3 \cdot 10^0 = 200 + 80 + 3$$

Todos los sistemas posicionales están basados en el **Teorema Fundamental de la Numeración (TFN),** que sirve para relacionar una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal.

En él, X es el valor absoluto del dígito en cuestión, i es la posición que ocupa el dígito con respecto al punto decimal y B es la base. El símbolo Σ (sumatorio) indica que para obtener el valor del número (NÚM), será necesario sumar todos los productos.

Esta misma fórmula también se puede expresar de la siguiente forma:

$$NÚM = X_n \cdot 10^n + \ldots + X_2 \cdot 10^2 + X_1 \cdot 10^1 + X_0 \cdot 10^0 + X_{-1} \cdot 10^{-1} + X_{-2} \cdot 10^{-2} \ldots + X_{-N} \cdot 10^{-N}$$

Ten en cuenta

El **Teorema Fundamental de la Numeración (TFN)** queda determinado por la fórmula siguiente:

$$NÚM = \Sigma X_i \cdot B^i$$

C. Codificación numérica

Son tres los sistemas de codificación que utiliza habitualmente un sistema informático:

- **Binario.** Este sistema utiliza dos símbolos diferentes: el cero y el uno (0,1). Es el sistema que maneja el ordenador internamente, ya que lo utilizan sus componentes electrónicos.

  Cada uno de estos símbolos recibe el nombre de **bit,** entendiendo por tal la mínima unidad de información posible.

  Los símbolos del sistema decimal pueden representarse (codificarse) en binario mediante el TFN. Cada símbolo decimal puede representarse con una combinación de cuatro bits.

- **Octal.** Es un sistema en base 8 que utiliza los símbolos del 0 al 7 para representar las cantidades, las cuales quedan reproducidas posicionalmente por potencias de 8. El sistema de numeración en base 8 tiene una correspondencia directa con el binario, ya que cada símbolo en base 8 puede representarse mediante una combinación de 3 bits.

- **Hexadecimal.** Es un sistema de numeración en base 16. Utiliza 16 símbolos diferentes, del 0 al 9 y los dígitos valores (o letras) A, B, C, D, E y F. Estas letras representan, respectivamente, los dígitos 10, 11, 12, 13, 14 y 15 del sistema decimal. Este sistema también tiene una correspondencia directa con el sistema binario, ya que cada símbolo en base 16 se puede representar mediante una combinación de 4 bits.

Decimal	Binario	Base 8	Base 16
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

Tabla 1.3. *Sistemas decimal, binario, octal y hexadecimal.*

El sistema que maneja internamente un ordenador es el binario, pero, en ocasiones, por comodidad en el manejo de los datos, se suele utilizar el octal y el hexadecimal, ya que mucha de la información que nos muestra el sistema operativo, como direcciones de memoria, está expresada en hexadecimal.

El sistema de numeración binario tiene una gran importancia en el funcionamiento del ordenador. Ya se ha señalado que la memoria del ordenador es un conjunto de biestables. En ellos puede haber o no corriente eléctrica.

En la Tabla 1.3 podemos ver los primeros 20 símbolos decimales y sus correspondencias en binario, base 8 y base 16.

D. Cambios de base de numeración

El sistema informático trabaja en el sistema de numeración binario. Nosotros trabajamos en el sistema de numeración decimal. El ordenador no entiende el sistema de numeración decimal para realizar sus cálculos, pero nosotros no entendemos el binario para realizar los nuestros.

Es necesario saber interpretar el código binario para poder entender las operaciones que en muchas ocasiones se realizan dentro del ordenador. Para ello, debemos aprender a pasar números binarios a decimales y a la inversa. Por extensión, el ordenador utiliza los sistemas de numeración de base 8 y base 16 (por ser múltiplos del sistema binario) para mostrarnos información relativa a algunos procesos que realiza.

En primer lugar, veamos cómo se pasa un número de base 10 a base 2. Esta operación se realiza dividiendo el número de base 10 (dividendo) por 2 (divisor). El cociente obtenido de la división se convertirá en dividendo, para volver a dividirlo por 2 (divisor). Al nuevo cociente obtenido se le aplica la misma operación, y así sucesivamente hasta que aparezca un cociente igual a 0.

En resumen, tendremos que dividir sucesivamente entre 2 el número en base 10, hasta que resulte un cociente 0. El número en binario se obtiene uniendo todos los restos en orden inverso de aparición.

Caso práctico 1

Pasar a base 2 el número 90 que está en base 10

Primero dividimos el número por 2 (base destino) y el cociente que obtenemos lo dividimos de nuevo por 2. El nuevo cociente lo volvemos a dividir por 2, y así sucesivamente hasta que aparezca un cociente igual a 0.

90 : 2 = 45. Resto 0.

45 : 2 = 22. Resto 1.

22 : 2 = 11. Resto 0.

11 : 2 = 5. Resto 1.

5 : 2 = 2. Resto 1.

2 : 2 = 1. Resto 0.

1 : 2 = 0. Resto 1.

Ordenamos los restos sucesivos que aparecen en las divisiones, pero en orden inverso, y obtenemos la nueva codificación en base 2:

Resultado: $\mathbf{90_{(10} = 1011010_{(2}}$

Si queremos pasar el mismo número a base 8 y 16, la forma de proceder sería la misma, teniendo en cuenta que ahora el divisor es el 8 o el 16, respectivamente.

Paso a base 8:

90 : 8 = 11. Resto 2.

11 : 8 = 1. Resto 3.

1 : 8 = 0. Resto 1.

Resultado: $\mathbf{90_{(10} = 132_{(8}}$

Paso a base 16:

90 : 16 = 5. Resto 10 (A).

5 : 16 = 0. Resto 5.

Resultado: $\mathbf{90_{(10} = 5A_{(16}}$

Como se puede ver en este último caso, el primer resto ha sido 10. Pero este símbolo en hexadecimal no existe; existe la A como símbolo décimo de la base.

Si lo que queremos hacer es la operación contraria, es decir, pasar de base 2 a base 10, procederemos multiplicando por potencias sucesivas de 2, empezando por 20 cada dígito binario de izquierda a derecha. Sumaremos los valores obtenidos y tendremos pasado el número.

Caso práctico 2

Pasar el número 1001 de binario a base 10

Primero se toman los dígitos binarios, 4 en total, y se van multiplicando por potencias de 2 de izquierda a derecha. El último exponente que pondremos en base 2 y con el que multiplicaremos el último dígito será el *n*–1, siendo *n* el número de dígitos que tiene la cifra de base 2.

$1001_{(2} = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = 9_{(10}$

Truco

Cuando pasemos de base 10 a cualquier base, nunca podremos obtener un resto de división superior o igual a la base a la que estamos pasando.

Los cambios de base entre bases equivalentes, como son las bases 2, 8 y 16, se pueden realizar de forma directa, teniendo en cuenta la equivalencia de bits con la que se puede representar cada dígito de estas bases en binario. En base 8, un dígito octal queda representado por una combinación de 3 bits ($2^3 = 8$, siendo 3 el número de bits). En hexadecimal, la asociación es de 4 bits ($2^4 = 16$).

Como cada número en base 8 y en base 16 tiene una correspondencia directa con el número en binario mediante un conjunto de 3 y 4 bits respectivamente, si queremos transformar un número en base 8 o en base 16 a un número en base 2 o viceversa, bastará con formar grupos de 3 o 4 bits respectivamente. Esta **transformación** se llama **directa.**

Truco

Para saber si varios sistemas de numeración son equivalentes, solo tienes que analizar si todos ellos se pueden representar como potencia del más pequeño.

Caso práctico 3

Pasar el número 132 de base 8 a base 16

En primer lugar, pasamos el 132 que está en octal a binario de forma directa. Como cada dígito octal se puede expresar con 3 dígitos binarios, tenemos:

$132_{(8} = \mathbf{001}\ \ 011\ \ \mathbf{010}_{(2} = 001011010_{(2}$

Así, transformamos directamente el dígito 1 en 001, el 3 en 011 y el 2 en 010. Mirando la equivalencia de la Tabla 1.3, vemos que cada dígito en base 8 tiene su correspondencia con 3 dígitos binarios.

Ahora, para pasar a base 16, basta con hacer grupos de 4 bits empezando por la derecha. Si faltan dígitos por la izquierda, los completamos con 0, aunque en nuestro caso no son significativos, ya que como en cualquier sistema de numeración, los 0 a la izquierda no tienen valor.

$001011010_{(2} = \mathbf{0000}\ \ 0101\ \ \mathbf{1010}_{(2} = \mathbf{5A}_{(16}$

El bloque de 4 bits de más a la derecha tiene su equivalencia con el dígito 10 en hexadecimal, pero como este símbolo no existe en este sistema de numeración, lo hacemos corresponder con su símbolo correspondiente que es la letra A. El bloque del centro se corresponde con el dígito 5 y, evidentemente, los cuatro 0 de la izquierda representan un 0, y como tal, no tiene valor precisamente por estar a la izquierda. En este caso, el resultado sería el siguiente:

Resultado: $\mathbf{132_{(8} = 5A_{(16}}$

El mismo caso sería el paso de base 16 a base 8. Para ello bastaría pasar a binario el número en hexadecimal y hacer bloques de 3 bits.

Ten en cuenta

Los símbolos del sistema hexadecimal, a partir del décimo, se representan con las letras A, B, C, D y F.

También podemos realizar un cambio de base por el **método indirecto,** que consiste en pasar el número de base *n* a base 10, y posteriormente pasarlo a base *m*. En nuestro ejemplo, n = 8 y m = 16. Este método se utiliza siempre que las bases de numeración no tengan correspondencia posicional. Así, si queremos transformar un número de base 6 a base 5, es evidente que siempre necesitaremos pasar por base 10. Ahora bien, si las bases son binario, octal y hexadecimal, al ser potencias de 2 y ser equivalentes, el paso puede ser directo.

Caso práctico 4

Pasar el número 132 en octal a base 16, pasando por base 10

La forma de convertir un número de base n a base 10 consiste en utilizar el Teorema Fundamental de la Numeración.

$132_{(8} = 1 \cdot 8^2 + 3 \cdot 8^1 + 2 \cdot 8^0 = 1 \cdot 64 + 3 \cdot 8 + 2 \cdot 1 = 64 + 24 + 2 = 90$

A partir de aquí se procede como se dijo antes para transformar el número 90 en base 10 a base 16.

90 : 16 = 5. Resto 10 (A).

5 : 16 = 0. Resto 5.

Resultado: $\mathbf{132_{(8} = 5A_{(16}}$

Caso práctico 5

Realizar los siguientes cambios de base:

a) **Pasar el número 0111 1011 1010 0011 que está en binario a base 16 y base 8.**

b) **Pasar el número 100 101 100 que está en binario a base 8 y base 16.**

c) **Pasar el número 1274 de base 8 a base 2 y a base 16.**

d) **Pasar el número ABF de base 16 a base 8 y base 2.**

a) Primero, hacemos el cambio de base 16. Agrupamos los bits de 4 en 4 empezando por la derecha. El resultado es el siguiente: **0111 1011 1010 0011**$_{(2.}$ Localizamos los dígitos equivalentes en base 16 y el resultado que obtenemos es el siguiente: $\mathbf{7\ B\ A\ 3_{(16}}$

Del mismo modo, pero realizando agrupaciones de 3 en 3 bits, obtendremos el número equivalente en base 8.

000 111 101 110 100 $011_{(2}$ = 7 5 6 4 $3_{(8}$

b) Procediendo de forma similar al caso a), los resultados obtenidos son los siguientes:

100 101 $100_{(2}$ = 4 5 $4_{(8}$

0001 0010 $1100_{(2}$ = 1 2 $C_{(16}$

c) Aquí, el procedimiento es a la inversa. Tomamos de derecha a izquierda cada dígito del número de base 8 y escribimos sus equivalentes en binario. Cada dígito en base 8 corresponde a 3 dígitos binarios. Este es el resultado:

1 2 7 $4_{(8}$ = 001 010 111 $100_{(2}$

(Continúa)

Caso práctico 5

Obtenido el número en binario, podremos agrupar los dígitos de 4 en 4 de derecha a izquierda para obtener así el correspondiente número en base 16.

$\underline{0010}\ 1011\ \underline{1100}_{(2} = 2\ B\ A_{(16}$

d) De forma similar, lo primero es pasar el número de base 16 a binario, buscando su equivalencia de 4 bits por cada dígito hexadecimal.

$A\ B\ F_{(16} = 1010\ 1011\ 1111_{(2}$

Luego, se agrupan los dígitos binarios de 3 en 3 de derecha a izquierda para obtener el equivalente en base 8. Así:

$101\ \underline{010}\ 111\ \underline{111}_{(2} = 5\ 2\ 7\ 7_{(8}$

E. Otros tipos de codificación numérica

Coma o punto fijo. El punto fijo se usa para la representación de números enteros. Hay tres formas de representar los números en coma fija: **binario puro, decimal desempaquetado** y **decimal empaquetado.**

Para el binario puro se utiliza una combinación de 32 bits en la que el bit de la izquierda sirve para representar el signo: 0 para el signo + y 1 para el signo –. Los restantes 31 bits sirven para representar el valor del número.

El decimal desempaquetado representa cada número decimal de forma que cada una de sus cifras ocupa un byte u octeto.

En primer lugar, para hablar de la codificación en decimal desempaquetado, hay que ver cómo se representan los números decimales en **DCB** (**D**ecimal **C**odificado en **B**inario) o **BCD** *(**B**inary **C**oded **D**ecimal).*

En este sistema, cada dígito decimal se representa con una combinación de 4 bits. La Tabla 1.4 muestra cómo se representan en BCD las cifras decimales del 0 al 9.

Cada número en decimal desempaquetado lleva en los 4 bits de la izquierda cuatro 1 denominados **bits de zona.** El cuarteto de la derecha se utiliza para codificar el número en DCB. El signo se representa en el cuarteto de bits de la izquierda correspondiente al último octeto: 1100 para el signo positivo y 1101 para el signo negativo.

El decimal empaquetado representa cada cifra con un conjunto de 4 bits. El conjunto de 4 bits de la derecha se usa para representar el signo con la misma combinación que en el caso anterior.

Coma flotante. Se utiliza para representar números reales y enteros con un rango de representación mayor que el que ofrece el punto fijo. Con eso conseguimos que el ordenador pueda tratar números muy grandes o muy pequeños.

La representación de números en coma flotante se puede hacer de dos formas:

- **Simple precisión.** Se utilizan 32 bits para representar cualquier cantidad numérica.
- **Doble precisión.** Se utiliza una combinación de 64 bits para representar una cifra.

Actividades

17. ¿Se puede pasar directamente un número de base 3 a base 5?

18. Si existiera el sistema de numeración de base 32, ¿se podría pasar un número de base 8 a 32 de forma directa?

Ejemplo

Así se representa el número –10 en coma o punto fijo.

10000000000000000000000000001010

Decimal	DCB
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Tabla 1.4. *Esquema del código DCB o BCD.*

Ejemplo

Así se representa el número 2371 decimal en decimal desempaquetado:

1111 0010 1111 0011 1111 0111 1100 0001

signo +

Y si se trata del –2371:

1111 0010 1111 0011 1111 0111 1101 0001

signo –

El número 2371 en decimal empaquetado se representaría de la siguiente forma:

0010 0011 0111 0001 1100

signo +

F. La codificación alfanumérica

Ya sabemos que los datos, además de numéricos, pueden ser alfabéticos o alfanuméricos. Normalmente, con los datos alfanuméricos podemos construir instrucciones y programas. Por otro lado, es lógico pensar que el ordenador no solamente procesará datos numéricos, sino también datos alfabéticos y combinaciones de los anteriores, como datos alfanuméricos.

Los sistemas de codificación alfanumérica sirven para representar una cantidad determinada de símbolos en binario. A cada símbolo le corresponderá una combinación de un número de bits.

Los sistemas de **codificación alfanumérica** más importantes son:

- **ASCII** *(**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)*. Este sistema utiliza una combinación de 7 u 8 bits, dependiendo del fabricante, para representar cada símbolo. Es el más utilizado y el que emplea símbolos diferentes (28). Con este código se pueden representar dígitos del 0 al 9, letras mayúsculas de la A a la Z, letras minúsculas, caracteres especiales y algunos otros denominados de control.

  En la Tabla 1.5 se recogen 128 de los 256 símbolos diferentes que se pueden representar con el código ASCII de 8 bits.

- **EBCDIC** *(**E**xtended **BCD** **I**nterchange **C**ode)*. Cada símbolo se representa por una combinación de 8 bits agrupados en dos bloques de cuatro. Es el formato extendido del BCD.

- **UNICODE.** Es un código internacional utilizado hoy por hoy en la mayoría de los sistemas operativos. Permite que un producto software o página Web específica se oriente a múltiples plataformas, idiomas o países sin necesidad de rediseño. Concretamente, el código ASCII tiene una tabla específica para cada país, ya que los diferentes símbolos de todos los países no cabrían en una tabla.

  UNICODE define la codificación de caracteres, así como las propiedades y los algoritmos que se utilizan en su aplicación. Proporciona un número único para cada carácter, sin importar la plataforma (hardware), el programa (software) o el idioma.

La mayoría de líderes del mercado como Apple, HP, IBM, Microsoft, Oracle, Sun, Unisys y otros, han adoptado la norma UNICODE, permitiendo crear aplicaciones y hardware estándar con XML, Java, etc. Es compatible con muchos sistemas operativos actuales, así como con la mayoría de los exploradores de Internet, permitiendo que un producto software se oriente a varias plataformas o idiomas sin necesidad de rediseño.

Actividades

19. ¿Por qué la palabra España se escribe Espana en muchas facturas y justificantes bancarios?

Caracteres no imprimibles				Caracteres imprimibles								
Nombre	**Dec**	**Hex**	**Car.**	**Dec**	**Hex**	**Car.**	**Dec**	**Hex**	**Car.**	**Dec**	**Hex**	**Car.**
Nulo	0	00	NUL	32	20	Espacio	64	40	@	96	60	`
Inicio de cabecera	1	01	SOH	33	21	!	65	41	A	97	61	a
Inicio de texto	2	02	STX	34	22	"	66	42	B	98	62	b
Fin de texto	3	03	ETX	35	23	#	67	43	C	99	63	c
Fin de transmisión	4	04	EOT	36	24	$	68	44	D	100	64	d
enquiry	5	05	ENQ	37	25	%	69	45	E	101	65	e
acknowledge	6	06	ACK	38	26	&	70	46	F	102	66	f
Campanilla (beep)	7	07	BEL	39	27	'	71	47	G	103	67	g
backspace	8	08	BS	40	28	(	72	48	H	104	68	h
Tabulador horizontal	9	09	HT	41	29	)	73	49	I	105	69	i
Salto de línea	10	0A	LF	42	2A	*	74	4A	J	106	6A	j
Tabulador vertical	11	0B	VT	43	2B	+	75	4B	K	107	6B	k
Salto de página	12	0C	FF	44	2C	,	76	4C	L	108	6C	l
Retorno de carro	13	0D	CR	45	2D	-	77	4D	M	109	6D	m
Shift fuera	14	0E	SO	46	2E	.	78	4E	N	110	6E	n
Shift dentro	15	0F	SI	47	2F	/	79	4F	O	111	6F	o
Escape línea de datos	16	10	DLE	48	30	0	80	50	P	112	70	p
Control dispositivo 1	17	11	DC1	49	31	1	81	51	Q	113	71	q
Control dispositivo 2	18	12	DC2	50	32	2	82	52	R	114	72	r
Control dispositivo 3	19	13	DC3	51	33	3	83	53	S	115	73	s
Control dispositivo 4	20	14	DC4	52	34	4	84	54	T	116	74	t
neg acknowledge	21	15	NAK	53	35	5	85	55	U	117	75	u
Sincronismo	22	16	SYN	54	36	6	86	56	V	118	76	v
Fin bloque transmitido	23	17	ETB	55	37	7	87	57	W	119	77	w
Cancelar	24	18	CAN	56	38	8	88	58	X	120	78	x
Fin medio	25	19	EM	57	39	9	89	59	Y	121	79	y
Sustituto	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
Escape	27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
Separador archivos	28	1C	FS	60	3C	<	92	5C	\	124	7C	|
Separador grupos	29	1D	GS	61	3D	=	93	5D	]	125	7D	}
Separador registros	30	1E	RS	62	3E	>	94	5E	^	126	7E	~
Separador unidades	31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Tabla1.5. *Esquema de la tabla de código ASCII de 8 bits hasta el carácter 127.*

5.3. Medidas de la información

Ampliación

Bit = mínima unidad de información.

4 Bits = Nibble o cuarteto.

8 Bits = 1 Byte.

1 024 Bytes = 1 Kilobyte.

1 024 Kilobytes = 1 Megabyte (Mb).

1 024 Megabytes = 1 Gigabyte (Gb).

1 024 Gigabytes = 1 Terabyte (Tb).

1 024 Terabytes = 1 Petabyte (Pb).

1 024 Petabytes = 1 Exabyte (Eb).

1 024 Exabytes = 1 Zettabyte (Zb).

1 024 Zettabytes = 1 Yottabyte (Yb).

1 024 Yottabytes = 1 Brontobyte (Bb).

1 024 Brontobytes = 1 Geopbyte (Geb).

Ya sabemos que el bit es la mínima unidad de información. Este queda representado por un 0 o un 1.

En este sentido, se puede establecer una equivalencia de medidas en múltiplos de bits utilizados para designar cada medida.

De esta forma, por ejemplo, un Tb se corresponde con 2^{40} bytes.

Veamos algunas de estas equivalencias de forma más detallada en la Tabla 1.6.

El número 1 024 es una potencia de 2 (2^{10}). Su uso está justificado, ya que el ordenador utiliza internamente el sistema de codificación binario para todas sus operaciones.

El byte se suele emplear para reasentar un carácter alfanumérico.

Antes vimos las codificaciones ASCII y EBCDIC, que usaban 8 bits para representar cada símbolo.

Actualmente, la capacidad de la memoria RAM se mide en Mb o Gb, y la capacidad de los discos duros en Gb o Tb.

Unidad	Abreviatura	Se habla de	Representa
1 Kilobyte	Kb	kas	1 024 bytes
1 Megabyte	Mb	megas	1 024 Kb (1 048 576 bytes)
1 Gigabyte	Gb	gigas	1 024 Mb (1 073 741 824 bytes)
1 Terabyte	Tb	teras	1 024 Gb (un billón de bytes)

Tabla 1.6. *Equivalencias de medidas de información.*

Ejemplo

Para realizar la equivalencia de unas medidas de información con otras, tendremos siempre en cuenta la medida a la que queremos llegar y la medida de la que partimos.

En la siguiente tabla podemos ver algunas transformaciones ya realizadas.

Cantidad a transformar	Kb	Mb	Gb	Tb
160 000 Kb	**160 000 Kb**	160 000 Kb/1 024= **156,25 Mb**	160 000 Kb/1 024/1 024 = 156,25 Mb/1 024= **0,152587 Gb**	160 000 Kb/1 024/1 024/1 024 = 156,25 Mb/1 024/1 024 = 0,152587 Gb/1 024 = **0,000149 Tb**
2 000 Mb	2 000 Mb · 1 024 = **2 048 000 Kb**	**2 000 Mb**	2 000 Mb/1 024 = **1,953125 Gb**	2 000 Mb/1 024/1 024 = 1,953125 Gb /1 024 **0,001907 Tb**
550 Gb	550 Gb/1 024/1 024 = 563 200 Mb/1 024 = **576 716 800 Kb**	550 Gb/1 024 = **563 200 Mb**	**550 Gb**	550 Gb / 1 024 = **0,537109 Tb**
0,1 Tb	0,1 Tb · 1 024 · 1 024 · 1 024 = 102,4 Gb · 1 024 · 1 024 = 104 857,6 Mb · 1 024 = **107 374 182,4 Kb**	0,1 Tb · 1 024 · 1 024 = 102,4 Gb · 1 024 = **104 857,6 Mb**	0,1 Tb · 1 024 = **102,4 Gb**	**0,1 Tb**

Comprueba tu aprendizaje

1. **La información contenida en disquete y discos duros, ¿es software o hardware?**
2. **Dibuja un esquema de cómo se representaría la palabra HOLA en las celdillas de memoria.**
3. **Clasifica los siguientes periféricos y soportes según su tipo: impresora, escáner, módem, monitor, disco duro, *pen drive*, tarjeta de sonido.**
4. **Indica las partes y funciones de un sistema informático.**
5. **Completa las siguientes tablas de códigos:**

Binario				11111111
Decimal		123		
Octal	16			
Hexadecimal			CAE	

Binario	10101001			
Decimal		987		
Octal			701	
Hexadecimal				FEA

Binario				111001001
Decimal		110		
Octal	621			
Hexadecimal			ADAD	

6. **En el ejercicio anterior se han manejado códigos numéricos que pueden corresponder, por ejemplo, a una dirección de memoria, etc. Pero en el ordenador se maneja todo tipo de información, no únicamente numérica. No sólo existen códigos numéricos como BCD (4 bits), sino que, como sabemos, existen códigos alfanuméricos, ASCII (7 u 8 bits), EBCDIC (8 bits), FIELDATA (6 bits), que hacen corresponder cada carácter con una cadena binaria de un número de bits.**

   a) Usando ASCII de 8 bits y EBCDIC, transcribe a una cadena binaria la palabra CADENA.

   b) Usando ASCII de 8 bits y EBCDIC, transcribe a una cadena binaria la frase «HOY es 3-10-01».

   c) ¿Tiene alguna ventaja o inconveniente usar un código u otro?

   d) ¿Cuántos caracteres distintos pueden representarse con estos códigos: BCD, FIELDATA, ASCII, EBCDIC?

7. **¿Puede funcionar un ordenador sin software básico? ¿Y sin unidad de disco duro?**
8. **Explica qué tipo de método de direccionamiento siguen las siguientes instrucciones para ejecutarse:**

   SUMA 45

   RESTA 34 12

   PROD 32 12 45

9. **Realiza el seguimiento del siguiente programa informático y especifica qué resultados se obtienen al final del mismo:**

   SUMA 20 12 15

   RESTA 15 14 13

   SUMA 13 17

   MOVER 17 18

   Contesta las siguientes preguntas:

   a) ¿Cuál es el contenido de la posición 18 de memoria? ¿Y de la posición 13?

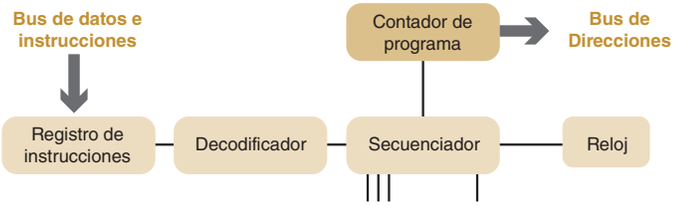
   b) ¿En qué posición se almacena el resultado final de la operación?

   c) Realiza todo el seguimiento de la ejecución del programa suponiendo que en todas las posiciones de memoria hay un 1.

Nota. En los ejercicios 8 y 9, los números indicados son posiciones de memoria.

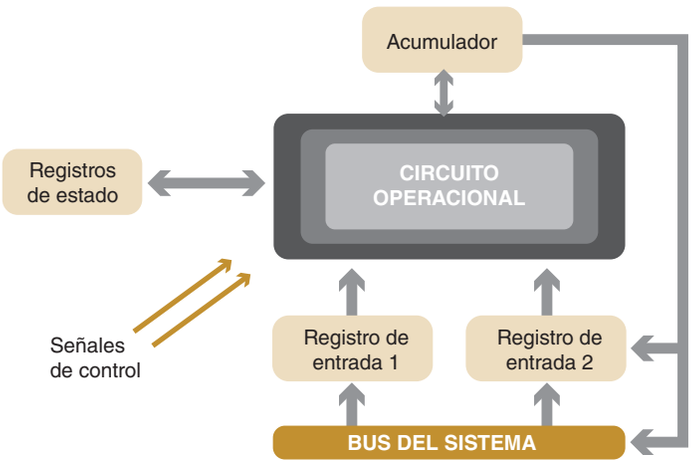
Comprueba tu aprendizaje

10. Explica cada uno de los componentes del siguiente esquema:



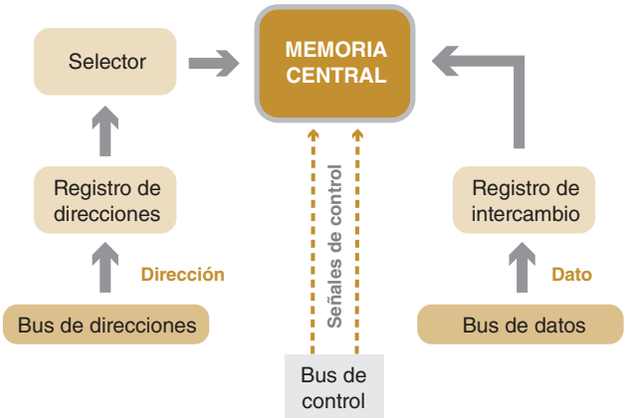
- Registro de instrucción.
- Registro contador de programas.
- Controlador y decodificador.
- Secuenciador.
- Reloj.

11. Explica cada uno de los componentes del siguiente esquema:



- Operacional o circuito operacional.
- Registros de entrada.
- Registro acumulador.
- Registro de estado.

12. Explica cada uno de los componentes del siguiente esquema:



- Registro de direcciones.
- Registro de intercambio.
- Selector de memoria.
- Señal de control.

13. Teniendo en cuenta la siguiente tabla, sigue la representación de los números hasta el 31 decimal:

Decimal	Binario	Base 8	Base 16
0	00000	0	***0***
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13