



Academia de preparación de oposiciones
<http://www.uni10.es>

Tema 2 - Elementos funcionales de un ordenador digital.

Índice

1.	Introducción	2
2.	Arquitectura y funcionamiento de un ordenador digital	2
2.1.	Arquitectura Von Neumann del computador digital.....	2
2.2.	Funcionamiento de un computador digital.....	3
3.	La unidad central de proceso (CPU)	4
3.1.	Unidad aritmética.....	5
3.1.1.	Estructura.....	5
3.1.2.	Operaciones.....	6
3.2.	Unidad de control.....	8
3.2.1.	Operaciones elementales.....	8
3.2.2.	Diseño de la unidad de control.....	9
4.	Memoria interna	11
4.1.	Jerarquía de memoria.....	11
4.2.	Memoria interna.....	12
4.3.	Memorias de semiconductores.....	13
4.3.1.	RAM estáticas.....	13
4.3.2.	RAM dinámicas (DRAM).....	13
4.3.3.	ROM, PROM, EPROM, EEPROM y Flash.....	14
4.3.4.	VRAM.....	14
4.3.5.	Módulos de memoria.....	14
4.4.	Mapa de memoria de un computador.....	14
5.	Unidad de entrada/salida	15
5.1.	Controlador de E/S.....	16
5.2.	Direccionamiento de puertos de entrada y de salida.....	18
5.3.	Técnicas de transferencias de entrada/salida.....	18
5.3.1.	Control por medio de E/S programada.....	18
5.3.2.	Control por medio de interrupciones.....	19
5.3.3.	Control por medio de controlador DMA.....	20
6.	Conclusiones	21

1. Introducción

Un ordenador es una máquina diseñada con el objetivo de automatizar el tratamiento de la información en todas sus fases, con el fin de resolver un problema determinado.

Los ordenadores digitales utilizan un sistema binario para representar la información. Esto es así debido a que el principio de funcionamiento de éstos reside en el paso o no de una corriente eléctrica a través de un determinado componente electrónico: en la actualidad, un transistor.

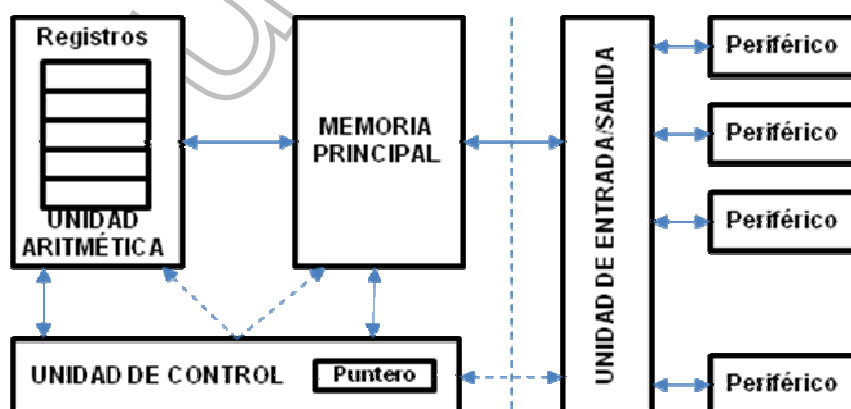
De este modo, los ordenadores digitales reconocen únicamente dos estados posibles: 0 y 1. Cada uno de estos dígitos binarios recibe el nombre de bit (del inglés, *binary digit*).

2. Arquitectura y funcionamiento de un ordenador digital.

2.1. Arquitectura Von Neumann del computador digital.

El modelo básico de arquitectura empleada en los computadores digitales fue establecida en 1945 por Von Neumann. Esta arquitectura es todavía, aunque con pequeñas modificaciones, la que emplean la gran mayoría de los fabricantes.

La siguiente figura muestra la estructura general de un computador tipo Von Neumann. Esta máquina es capaz de ejecutar una serie de instrucciones u órdenes elementales llamadas instrucciones de máquina, que deben estar almacenadas en la memoria principal para poder ser leídas y ejecutadas.



Puede observarse de la figura que esta máquina está compuesta por las siguientes unidades: memoria principal, unidad aritmética, unidad de control, y unidad de entrada/salida.

La memoria principal está compuesta por un conjunto de celdas idénticas, esto es, que tienen el mismo tamaño (mismo número de bits). En un instante determinado se puede seleccionar una de estas celdas, para lo que se emplea una dirección que la identifica. Sobre la celda seleccionada se puede realizar una operación de lectura, que permite conocer el valor almacenado previamente en la celda, o de escritura, que permite almacenar un nuevo valor. Estas celdas se emplean tanto para almacenar los datos como las instrucciones de máquina.

La unidad aritmética permite realizar una serie de operaciones elementales tales como suma, resta, AND, OR, etc. Los datos sobre los que opera esta unidad provienen de la memoria principal, y pueden estar almacenados de forma temporal en algunos registros de la propia unidad aritmética.

La unidad de control se encarga de leer, una tras otra, las instrucciones de máquina almacenadas en la memoria principal, y generar las señales de control necesarias para que todo el computador funcione y ejecute las instrucciones leídas (la figura indica estas señales mediante flechas de trazos). Para conocer en todo momento la posición de memoria en la que está almacenada la instrucción que corresponde ejecutar, existe un registro apuntador llamado contador de programa que contiene esta información.

La unidad de entrada/salida realiza la transferencia de información con unas unidades exteriores llamadas periféricos, lo que permite, entre otras cosas, cargar datos y programas en la memoria principal y sacar resultados impresos.

Finalmente, conviene resaltar que existen unos caminos o buses cuyo objetivo es hacer que las instrucciones y los datos circulen entre las distintas unidades del computador. Estos caminos están representados por flechas de trazo continuo en la figura.

2.2. Funcionamiento de un computador digital.

Un computador es capaz de ejecutar una serie de instrucciones u órdenes llamadas instrucciones de máquina. Estas instrucciones suelen ser sorprendentemente elementales, tales como sumar o restar dos números, mover un dato de una posición de memoria a otra, etc.

Las fases de ejecución de una instrucción son las siguientes:

1. La unidad de control envía a la memoria principal la dirección de la instrucción a ejecutar, que está almacenada en el contador de programa, y activa las señales de control necesarias para que ésta le entregue la mencionada instrucción.
2. La unidad de control recibe la instrucción, la analiza y, en caso necesario, lee los operandos de la memoria principal, enviando su dirección y activando las correspondientes señales de control.

3. Seguidamente, bajo las directrices de la unidad de control, se realiza la operación sobre los operandos y, si es necesario, se salvaguarda el resultado en la memoria principal o en un registro.
4. Una vez ejecutada la instrucción, se incrementa el contador de programa, con lo que se puede pasar a ejecutar la instrucción siguiente.

Este funcionamiento muestra que el computador solamente es capaz de ir ejecutando una secuencia consecutiva de instrucciones de máquina. Estas secuencias de instrucciones reciben el nombre de programas. Para que se pueda romper la secuencia lineal (o consecutiva) de ejecución de instrucciones, es necesario que existan instrucciones de máquina que permitan modificar el contenido del contador de programa, esto es, que permitan hacer lo que llamaremos bifurcaciones.

Para que el computador pueda realizar una función determinada, es necesario que previamente se realice la descomposición de esa función en su correspondiente conjunto de instrucciones de máquina, operación que recibe el nombre de programación. Además, para que el computador realice una función determinada, o lo que es equivalente, para que ejecute un programa, se tienen que dar las siguientes condiciones:

- Debe existir el correspondiente programa en lenguaje máquina.
- El programa, así como sus datos, deben encontrarse en memoria principal. Esto exige generalmente una operación de transferencia, desde un periférico de almacenamiento secundario hasta la memoria principal, operación que deberá hacerse mediante un programa que llamaremos cargador.
- El contador de programa de la unidad de control debe ser actualizado con la dirección correspondiente a la primera instrucción del mencionado programa.

Una vez satisfechas estas tres condiciones, el computador irá leyendo y ejecutando las instrucciones que forman el programa.

3. La unidad central de proceso (CPU).

Se denomina Unidad Central de Proceso al conjunto de la unidad de control, los registros y la unidad aritmética de un computador, esto es, al bloque que ejecuta las instrucciones. Muchas veces se utilizan las siglas UCP o CPU {Central Processing Unit} para denominar este conjunto. Para formar un computador hay que añadir la memoria principal, la unidad o unidades de entrada/salida y los periféricos. Con frecuencia se denomina procesador a una CPU, aunque a veces se extrapola este término de procesador al conjunto formado por una CPU y una pequeña memoria, como, por ejemplo, cuando se habla de un procesador de entrada/salida.

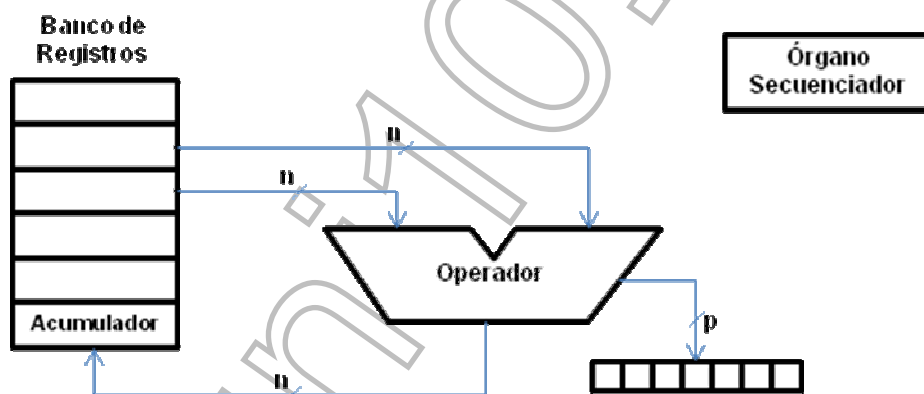
3.1. Unidad aritmética.

La unidad aritmética es la encargada de tratar los datos, ejecutando las operaciones requeridas, de acuerdo al programa en curso. Aunque las operaciones que realiza son tanto de tipo aritmético como lógico, se empleará el término de unidad aritmética para el conjunto, puesto que la parte aritmética es, con mucho, la más importante y compleja. No obstante, existe mucha literatura en la que se le denomina unidad aritmético-lógica o ALU por sus siglas en inglés.

La unidad de control del computador se encargará de enviarle los datos correspondientes y de indicarle qué operación ha de realizar.

3.1.1. Estructura

La siguiente figura representa la estructura de una unidad aritmética. En general se compone de uno o varios operadores, de un conjunto de registros, de unos biestables de estado y, en algunos casos, de un órgano secuenciador.



Un operador es una unidad funcional capaz de realizar una o varias operaciones aritméticas o lógicas, tales como suma, resta, AND, OR, etc.

La unidad de control lleva a cabo el control de las operaciones de los algoritmos de los operadores. Sin embargo, algunas veces el operador se construye con su órgano secuenciador correspondiente.

El banco de registros sirve para que el usuario almacene temporalmente datos y resultados intermedios. En muchas máquinas, uno de estos registros, llamado acumulador, se utiliza como depositario de los resultados del operador y, sobre su contenido, se realizan muchas operaciones que no se pueden hacer sobre otros registros.

La unidad aritmética está generalmente dotada de unos biestables de estado que almacenan ciertas condiciones relativas a la última operación realizada por ella. Típicamente suelen encontrarse los siguientes biestables de estado: cero

(se pone a 1 si el resultado es cero), negativo (se pone a 1 si el resultado es negativo), acarreo (se pone a 1 si el resultado tiene acarreo), desbordamiento (se pone a 1 si el resultado tiene desbordamiento).

3.1.2. Operaciones

Operaciones de desplazamiento.

Las operaciones de desplazamiento consisten en correr los bits de una palabra hacia la izquierda o derecha. Llamando A al operando origen, que se considerará compuesto por n bits ($a_0, a_1, a_2, \dots, a_n$), y D al resultado o destino, compuesto también por n bits, la operación de desplazamiento consiste en hacer $d_{i+k}=a_i$ para $i = 0, 1, \dots, n-1$. El parámetro k indica el número de posiciones del desplazamiento, mientras que el signo indica si el desplazamiento es a derecha (signo -) o a izquierda (signo +).

La forma en que se tratan los extremos del desplazamiento, en los que no se puede aplicar directamente la expresión $d_{i+k}=a_i$, permite diferenciar tres tipos de desplazamientos: los lógicos, los circulares y los aritméticos.

Operaciones lógicas.

Las operaciones lógicas, que generalmente se encuentran en los computadores, son: negación NOT, suma OR, producto AND, y suma exclusiva XOR. La realización de estas operaciones es inmediata, englobándose generalmente en un solo operador combinacional. en unión a las operaciones aritméticas elementales.

Operaciones de cambio de extension de signo.

El cambio de signo en complemento a uno consiste en hacer la negación lógica de todos los bits. Una etapa de puertas XOR, gobernada por una señal de control E, permite cambiar el signo del operando, haciendo $E = 1$, o transmitirlo sin modificarlo, haciendo $E = 0$. Estas etapas negadoras se suelen emplear como etapa previa de los sumadores, para convertirlos en Testadores.

En representación de complemento a 2, se debe hacer la negación lógica y sumar un 1 al resultado. Dado que uno de los sumandos de esta última operación es fijo e igual a 1, no es necesario emplear para ello un sumador completo.

En representación de signo magnitud bastará con cambiar el bit de signo, lo que se puede hacer con una sola puerta XOR que afecte a este bit.

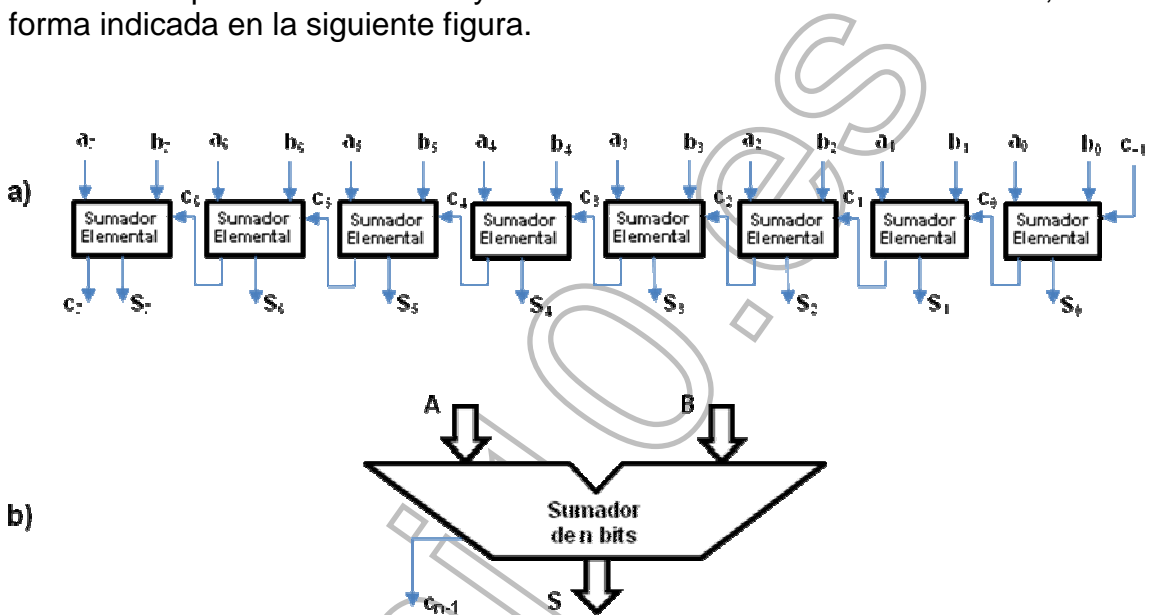
Por su lado, la extensión de signo debe realizarse cuando se transfiere un operando de n bits a un elemento que tenga m bits, de forma que $m > n$. Si no se rellenan convenientemente los bits adicionales, se cambiará el valor numérico del operando.

Operaciones de suma y resta

La operación de suma es el pilar sobre el que se construyen las unidades aritméticas de la mayoría de los computadores. Generalmente, se consideran unidades sumadoras-restadoras, puesto que la realización de la resta se puede obtener simplemente añadiendo una etapa de negación aritmética en el término sustraendo.

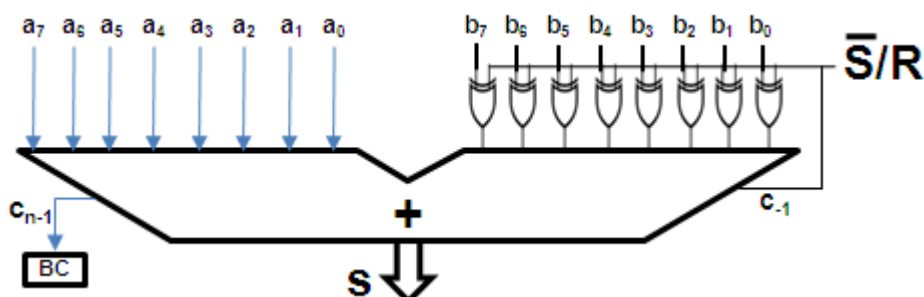
El sumador elemental es un circuito combinacional capaz de sumar dos dígitos binarios, más el posible acarreo de la etapa anterior, produciendo el dígito de suma y el de acarreo a la etapa siguiente. Se emplea normalmente como bloque constructivo para formar sumadores paralelo.

El sumador paralelo se construye mediante n sumadores elementales, de la forma indicada en la siguiente figura.



El restador paralelo puede hacerse siguiendo el procedimiento empleado con el sumador, mediante n circuitos restadores elementales. Ahora bien, puesto que es muy fácil modificar el circuito sumador anterior para producir una resta, resulta mucho más económico emplear un único circuito que permita hacer la suma y la resta, seleccionando la operación a realizar mediante una señal de control que llamaremos S/R.

La siguiente figura representa el esquema del circuito. Con la señal S/R = 0 el circuito suma y con S/R = 1 resta.



Operaciones de multiplicación y división.

Se suelen realizar utilizando un operador de suma/resta y un algoritmo adecuado. Es bastante raro que un computador disponga de un operador combinacional para multiplicar y/o dividir. Solamente las máquinas muy potentes o de aplicación específica están dotadas de ellos.

3.2. Unidad de control.

La unidad de control tiene como función básica la ejecución de la siguiente secuencia:

- Leer de la memoria principal la instrucción apuntada por el contador de programa e incrementarlo.
- Decodificar la instrucción leída.
- Hacer que se ejecute la instrucción.

La unidad de control emplea el contador de programa para realizar el secuenciamiento de las instrucciones. Ella se encarga de ir incrementándolo y de cargarlo con nuevos valores para realizar bifurcaciones, salvaguardando su contenido anterior si éstas son con retorno.

El código de operación indica qué operación se debe realizar y cuáles son los modos de direccionamiento a aplicar. La unidad de control deberá localizar los operandos, mandados, en su caso, a la unidad aritmética y almacenar el resultado.

El registro de estado contiene información sobre determinadas condiciones de los resultados de operaciones anteriores, así como sobre posibles situaciones anómalas o especiales como son desbordamientos aritméticos, interrupciones, errores de paridad, etc., que exigen acciones especiales por parte de la unidad de control.

La ejecución de cada instrucción requiere realizar una secuencia de pequeños pasos, que llamaremos operaciones elementales. La ejecución de cada una de estas operaciones elementales requiere la activación de las correspondientes señales de control. Por tanto, el objetivo de la unidad de control es la generación de las secuencias de señales de control que permiten realizar las distintas operaciones elementales de cada instrucción.

3.2.1. Operaciones elementales

Las operaciones elementales que puede realizar todo sistema digital síncrono se pueden clasificar en: operaciones de transferencia y operaciones de proceso.

Las operaciones de transferencia requieren dos elementos de almacenamiento, uno origen y otro destino. Para hacer esta operación, primero hay que establecer un camino físico entre la salida del origen y la entrada del destino. Una vez está estable el camino, hay que enviar una señal al destino para que se "cargue" con la información que tiene a su entrada.

Las operaciones de proceso tienen un planteamiento básico idéntico a las operaciones de transferencia. La diferencia fundamental es que la información origen se transforma al pasar a través de un operador en su camino hacia el destino.

3.2.2. Diseño de la unidad de control

La unidad de control toma como entrada el código de operación así como del periodo en que se encuentra la instrucción. Con estas informaciones (a las que ocasionalmente se debe añadir la salida del comparador de estado) quedan perfectamente definidas las señales de control que deben activarse durante ese periodo. En este sentido, la unidad de control es un mero traductor que convierte la combinación [código de operación + periodo + estado], en las señales de control especificadas en el correspondiente cronograma.

El diseño de la unidad de control exige, por tanto, haber definido previamente las señales que hay que activar en cada una de las instrucciones de máquina que debe ser capaz de interpretar. Esta definición puede hacerse mediante cronogramas, mediante operaciones elementales o usando un lenguaje simbólico, que exprese en todo detalle las operaciones a realizar y el orden en que deben llevarse a cabo.

El secuenciamiento de instrucciones se encuentra incluido en cada una de las instrucciones. En efecto, toda instrucción se encarga de iniciar la siguiente. Por tanto, el secuenciamiento de instrucciones no implica ninguna condición adicional de diseño sobre la unidad de control.

Por otro lado, la duración de las señales viene determinada por el reloj u oscilador general, debiendo acomodarse a un número entero de periodos.

Existen dos métodos de construir y diseñar una unidad de control: mediante lógica cableada, y mediante una memoria (lógica almacenada).

Unidad de control en lógica cableada.

La unidad de control cableada se construye mediante puertas lógicas, y se diseña siguiendo alguno de los métodos clásicos de diseño lógico. No entraremos en describir estas técnicas de diseño. Solamente haremos unos comentarios sobre las ventajas e inconvenientes de este tipo de solución:

- Dada la complejidad del circuito considerado, es muy laborioso y costoso su diseño y puesta a punto.

- Además, es muy difícil de modificar, puesto que suele exigir un rediseño completo.
- Sin embargo, en igualdad de condiciones, este circuito es más rápido que el diseñado mediante lógica almacenada, por lo que los computadores rápidos utilizan lógica cableada.

El diseño de unidades de control cableadas ha tomado un nuevo auge, especialmente en los computadores de tipo RISC. Esto se debe a las modernas técnicas de diseño por computador para circuitos VLSI, que resuelven automáticamente la mayor parte de las dificultades de diseño de lógica cableada. A ello hay que añadir la carrera de los fabricantes hacia máquinas cada vez más rápidas, en las que la lógica cableada es la única solución válida.

Unidad de control almacenada. Microprogramación.

La idea básica de la unidad de control almacenada consiste en emplear una memoria para almacenar el estado de las señales de control en cada periodo de cada instrucción. Por tanto, para generar el cronograma de una instrucción bastará con ir leyendo las sucesivas palabras de esta memoria correspondientes a ésta. Esta memoria recibe el nombre de memoria de control o micromemoria.

A cada palabra, que define un periodo de una instrucción, se le llama microinstrucción, por lo que estas unidades de control también reciben el nombre de unidades de control microprogramadas. En principio, las microinstrucciones tienen un bit por cada señal de control.

Se emplea el término firmware para referirse globalmente al conjunto de los microprogramas de una máquina. Es un término de difícil equivalencia castellana, que suele traducirse por microcódigo.

La unidad de control microprogramada puede ser demasiado compleja para máquinas sencillas y demasiado lentas para máquinas grandes. La ventaja principal es su simplicidad conceptual, y el hecho de que la información de control reside en una memoria que suele ser de tipo ROM.

Para que una unidad de control microprogramada pueda llevar a cabo su cometido, ha de cumplir las tres condiciones básicas siguientes:

1. Ha de tener una memoria de control con capacidad suficiente para almacenar todos los microprogramas correspondientes a todas las instrucciones de máquina.
2. Ha de tener un procedimiento para hacer corresponder a cada instrucción de máquina su microprograma, o lo que es lo mismo, ha de tener un procedimiento para convertir el código de operación de la

instrucción en la dirección de la memoria de control donde empieza su microprograma.

3. Ha de tener un mecanismo para ir leyendo las sucesivas microinstrucciones del microprograma en curso y para bifurcar a un nuevo microprograma, cuando termina el que se está ejecutando.

4. Memoria interna

La memoria es el órgano del computador que almacena la información que éste posee, es decir, sus datos y programas.

Las operaciones básicas que permite son dos: escritura o almacenamiento y lectura. En la escritura o almacenamiento la memoria necesita que le suministren una información y una dirección. La operación consiste en grabar la información en la dirección especificada. En la lectura la memoria recibe la dirección de la información deseada. La operación consiste en suministrar la información previamente escrita en esa dirección.

Los dispositivos de memoria pueden verse como cajas negras, a las que hay que suministrar una dirección y unas señales de control que especifican la operación que se desea realizar. Además, hay que enviar o recibir el dato o bloque de datos correspondiente.

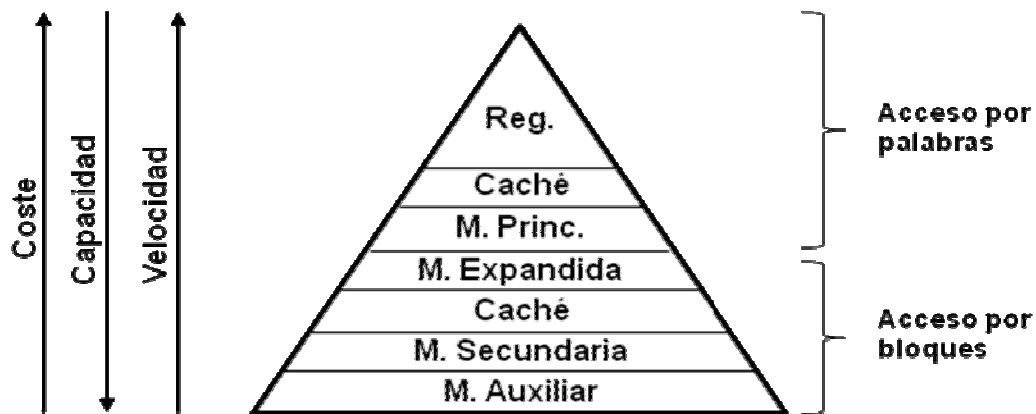
La capacidad de las memorias se puede expresar en unidades de bits, octetos o palabras. Lo más frecuente es expresarla en octetos (en la literatura es muy corriente encontrar el término de *byte* en vez del octeto). Se emplean los prefijos de:

K = kilo (1 K = 2^{10} = 1024 unidades)
 M = mega (1 M = 2^{20} = 2^{10} K)
 G = giga (1 G = 2^{30} = 2^{20} K = 2^{10} M)
 T = tera (1 T = 2^{40} = 2^{30} K = 2^{20} M = 2^{10} G)
 P = peta (1 P = 2^{50} = 2^{40} K = 2^{30} M = 2^{20} G = 2^{10} T)

4.1. Jerarquía de memoria.

En los computadores se desea que la memoria tenga una gran capacidad de almacenamiento, un tiempo de acceso pequeño y un precio reducido. Para ello, la memoria se suele estructurar en varios niveles. Existirá un nivel rápido, de pequeña capacidad, y niveles sucesivos, de menor velocidad, pero mayor capacidad. La información se ubicará en uno de los niveles, de acuerdo a su probabilidad de uso. Si en un momento determinado se necesita esta información, se transfiere al nivel superior más rápido para ser utilizada.

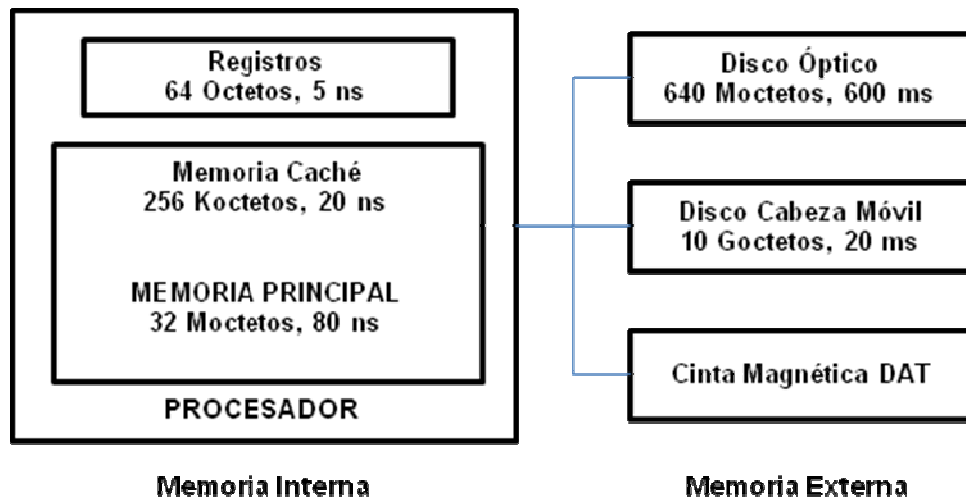
La siguiente figura muestra los niveles que pueden existir en la jerarquía de memoria, indicando algunas de las características de los mismos.



4.2. Memoria interna.

La memoria interna del computador la forman los tres primeros niveles de la jerarquía, esto es:

1. Los registros, entre los que cabe destacar los registros de uso general incluidos en la unidad aritmética, forman el nivel más rápido y de menos capacidad.
2. La memoria principal, que es la memoria donde tienen que residir los programas y sus datos para poder ser ejecutados por el computador, forma el tercer o segundo nivel, según que el computador disponga o no de memoria caché.
3. La memoria caché es una memoria auxiliar que se emplea para acelerar los accesos a la memoria principal, pero que es transparente al usuario, esto es, funcionalmente el computador se comporta como si esta memoria no existiese, pero trabaja más rápido que sin ella. La siguiente figura muestra estos tres niveles, así como la relación de la CPU con el resto de los dispositivos de memoria.



Hoy en día, la memoria interna de los computadores es de semiconductores. Los registros se construyen con la misma lógica que el resto del computador y la memoria principal con las pastillas de memoria.

4.3. Memorias de semiconductores.

La primera memoria comercial de semiconductor tenía una capacidad de 64 bits y fue fabricada por IBM en 1968. Poco más tarde, en 1972, aparecen los primeros EBM con memoria principal de semiconductores. Desde entonces, es el tipo de memoria empleada en la construcción de las memorias principales de los computadores.

Se puede establecer la siguiente clasificación:

- De lectura y escritura o RAM: estáticas y dinámicas o con refresco (llamadas DRAM).
- De sólo lectura: ROM (read only memory), PROM (*programmable read only memory*), EPROM (*erasable programmable read only memory*), y EEPROM (*electrically erasable programmable read only memory*)

4.3.1. RAM estáticas.

Las memorias RAM son memorias de semiconductor basadas en puntos de memoria de tipo biestable, por lo que mantienen su estado siempre que no se interrumpa la alimentación.

4.3.2. RAM dinámicas (DRAM).

Estas memorias usan puntos de memoria de un solo transistor. El estado se almacena en un condensador que tiende a descargarse, por lo que hay que refrescarlas para que no pierdan su información.

4.3.3. ROM, PROM, EPROM, EEPROM y Flash.

Consideramos en este apartado las memorias permanentes de semiconductor. Las memorias ROM y PROM sólo permiten la operación de lectura. La información contenida se graba en el proceso de fabricación (ROM) o bien en un proceso posterior irreversible (PROM).

Las memorias EPROM, EEPROM y Flash son memorias permanentes (de sólo lectura), pero que se pueden borrar mediante un proceso especial, como es radiación ultravioleta, en el caso de las EPROM, o elevadas corrientes, en el caso de las EEPROM y Flash. Las EPROM y Flash se borran en su totalidad, mientras que las EEPROM lo hacen selectivamente, a nivel de palabra.

Las memorias EPROM han sustituido a las PROM. Por otro lado, las EEPROM y Flash son más caras que las anteriores, pero son interesantes cuando hay probabilidad de tener que hacer cambios en su contenido.

Un uso importante de estas memorias estriba en almacenar el programa de arranque de los computadores.

4.3.4. VRAM

La VRAM es un nuevo tipo de memoria RAM dinámica, especialmente diseñada para construir memorias de refresco de terminales gráficos, de donde deriva su nombre: Vídeo-RAM. Estas pastillas incluyen, además de una memoria DRAM, unos registros de desplazamiento, que permiten acceder en serie a la información de la misma. El acceso paralelo normal de la DRAM se puede simultanear con el acceso a los registros de desplazamiento, por lo que la VRAM es una memoria de dos puertas, una serie y otra paralelo.

4.3.5. Módulos de memoria.

Las pastillas de memoria DRAM se suelen emplear empaquetadas en módulos de memoria llamados SIMM (*Single Inline Memory Module*). Existen básicamente dos modelos: de bus de 8 bits o de bus de 32 bits. La siguiente figura esquematiza la estructura de estos dispositivos.

4.4. Mapa de memoria de un computador.

Para que un computador pueda ejecutar un programa necesita que éste, así como sus datos, se encuentren en memoria principal. Para referenciar o direccionar estos datos e instrucciones, el computador genera y manipula direcciones de memoria principal. Por la propia construcción del computador, estas direcciones se ven limitadas a un cierto tamaño, correspondiente al número de bits que es capaz de manejar ese computador en las operaciones

de direccionamiento. Por ejemplo, si el contador de programa es un registro que tiene m bits, las direcciones de las instrucciones que puede direccionar ese computador se limitan al rango $[0, 2^m - 1]$. Por su lado, las direcciones de los operandos vendrán también limitadas a m bits.

Se llama mapa de memoria a todo el espacio direccionable por un computador. Este espacio viene determinado por el tamaño de las direcciones, puesto que, decir que se dispone de m bits de dirección, equivale a decir que se tiene un mapa de memoria de 2^m . Por otro lado, las m líneas de dirección forman lo que se llama el bus de direcciones. Por extensión, muchas veces se dice que un computador tiene un mapa de memoria de m bits, refiriéndose al tamaño de las direcciones que maneja.

5. Unidad de entrada/salida.

El problema de conexión de periféricos a una CPU se debe, principalmente, a que las características de los dispositivos de E/S suelen diferir considerablemente de las de la CPU; entre otras:

- La velocidad de transmisión de los periféricos es notablemente menor que la velocidad con que opera la CPU, y además es muy variable.
- La longitud de las palabras de datos suele variar.
- Los códigos y niveles eléctricos para representar los datos suelen diferir.

Para comprender adecuadamente las funciones de los elementos electrónicos de un periférico conviene distinguir dos niveles en la realización de una entrada/salida:

- Transferencias elementales de información. Sirven para la recepción o envío de una información individual (*byte* o palabra). Estas informaciones pueden ser datos propiamente dichos o información de control para el periférico, la CPU o información sobre el estado del propio periférico. Las transferencias elementales de información se realizan físicamente a través de puertos de entrada o salida, que son registros que se conectan directamente a uno de los buses del computador. Cada puerto tiene asociado una dirección o código, de forma que la CPU "ve" a un periférico como si fuese un puerto o conjunto de puertos.
- Operaciones de entrada/salida. Una operación de entrada consiste en la transferencia de un conjunto de datos (usualmente denominado bloque o registro físico) que constituye una línea del monitor de pantalla, un sector de un disco, o un bloque de una cinta magnética, por ejemplo. Las operaciones de E/S se componen, por tanto, de transferencias elementales que son monitorizadas por la CPU o por circuitos incluidos en el controlador del periférico.

5.1. Controlador de E/S.

Cada periférico necesita disponer de su propio controlador. Un controlador está constituido por un conjunto de circuitos de adaptación del formato de señales y características de temporización entre la CPU y los dispositivos de E/S. El controlador también se encarga de llevar a cabo las transferencias de datos entre la CPU y el periférico, para ello se conecta entre su periférico y un bus del computador, recibiendo y generando señales de control para el periférico y señales de estado para la CPU.

Desde la CPU se actúa sobre los periféricos a través de las instrucciones de los programas. Para poder utilizar eficazmente un periférico, su sistema operativo debe contener una rutina especial denominada gestor del periférico para controlar las transferencias de información. Sin estas rutinas del sistema operativo sería imposible, o extremadamente complejo, utilizar un periférico desde un lenguaje de alto nivel.

Los controladores de periféricos se interconectan al bus a través de los puertos de E/S. Los controladores suelen contener cinco elementos: decodificador de direcciones, puerto de datos, puerto de estado, puerto de control y circuitos que generan las secuencias de control interno (secuenciador del controlador). Los controladores cubren básicamente estos objetivos:

1. Selección o direccionamiento del periférico. La CPU sitúa en el bus de direcciones el código o dirección del puerto que debe intervenir en el tráfico de datos. En una transmisión concreta, sólo uno de los puertos debe estar conectado eléctricamente al bus de datos, para transmitir a través de él. El selector de direcciones se encarga de identificar cuándo la dirección del bus de direcciones corresponde a su código (dirección), para dar paso al intercambio de información con el periférico. De esta forma se evitan posibles conflictos de acceso al bus.
2. Almacenamiento temporal. La comunicación física entre el periférico y la CPU se hace a través del controlador, por lo que éste contiene un puerto de datos para almacenar temporalmente los datos a transferir.
3. Sincronización. La velocidad operativa del computador es mucho mayor que la de los periféricos. El controlador regula el tráfico de información para que no se den problemas de sincronización o de pérdidas de información. El controlador suele actuar con unas señales de control y estado que intercambia con la CPU, indicando situaciones tales como que está preparado o listo para recibir o transmitir, que ha reconocido la llegada de unos datos, que desea ser atendido por la CPU (señal de petición de interrupción) para transmitir a ésta unos datos, etc. En definitiva, el secuenciador del controlador en función del contenido del puerto de estado y control debe generar señales de estado y control para iniciar o terminar la transferencia.
4. Control del periférico. La CPU debe ser capaz de interrogar al controlador, para conocer su estado, leyendo el contenido del puerto de

estado, o de enviar órdenes a éste, escribiéndolas en el puerto de control. La operación de lectura suele realizarse siempre inmediatamente antes de transferir o captar datos del periférico. Usualmente, se puede acceder individualmente a los bits del puerto de estado y control, pudiendo unos bits ser activados por la CPU y otros por el propio controlador. Por ejemplo, un bit puede indicar si el dispositivo se encuentra conectado/desconectado a la red de suministro eléctrico; obviamente este bit será activado por el periférico.

5. Conversión de datos. Consiste en la adaptación de las características eléctricas y lógicas de las señales empleadas por el dispositivo de E/S y por el bus: niveles de tensión, impedancias, etc.; lógica (positiva o negativa); conversión paralelo/serie y serie/paralelo, etc.
6. Detección de errores, mediante la inclusión/detección de paridad, transformación según códigos polinómicos, etc. En caso de detectarse un error de este tipo, el secuenciador cambiaría el bit del puerto de estado indicador de error de transmisión, y se procedería a una repetición de la transferencia de información.
7. Gestión de transmisión de bloques de información. En los periféricos que intercambian la información en forma de registros físicos, el controlador puede disponer de un contador que controle el número de palabras recibidas/transmitidas.

Los controladores pueden estar físicamente ubicados en el chasis del computador central o en el dispositivo propiamente dicho (controlador IDE de disco duro, por ejemplo), o distribuidos entre ambos.

Las operaciones de entrada/salida son de una gran complejidad. En los primeros computadores la CPU controlaba minuciosamente todas las operaciones elementales que efectuaba el periférico. Poco a poco estas funciones, para descargar de trabajo a la CPU, se han ido descentralizando y pasando a los controladores, resultando éstos cada vez más complejos.

La mayor parte de los controladores pueden considerarse computadores de uso específico, ya que contienen un microprocesador, ROM con programas de gestión del periférico, una memoria intermedia (*buffer*) o memoria caché para datos, etc. La CPU, a través de los puertos de salida correspondientes, da las órdenes oportunas al controlador para realizar un conjunto de operaciones de entrada o salida.

En los computadores grandes es normal que se incluyan procesadores de periféricos o canales de datos. Estos son computadores dedicados a funciones de entrada/salida, que suelen contener un microprocesador con instrucciones especializadas y una memoria local. Cada procesador de periféricos puede atender a varios periféricos simultáneamente, encargándose de operaciones de transferencias específicas de E/S, tal como intercambio de datos entre dos periféricos conectados a él. Como en el caso de los controladores inteligentes, siempre es la CPU quien supervisa su funcionamiento, aquí la CPU, una vez

que ha cargado en los puertos o en una zona de memoria principal accesible por ambos, los parámetros correspondientes, lanza a ejecutar los programas residentes en los procesadores de periféricos. Cuando el procesador finaliza la ejecución del programa de E/S, indica este hecho a la CPU por medio de una interrupción.

5.2. Direccionamiento de puertos de entrada y de salida.

Con referencia a las operaciones de entrada/salida, los repertorios de instrucciones de los computadores suelen optar por una de las dos siguientes posibilidades:

- Disponer de instrucciones de E/S específicas.
- No disponer de instrucciones específicas de E/S; en cuyo caso los puertos de E/S se codifican dentro del mapa de memoria, diciéndose entonces que las instrucciones de E/S están mapeadas en memoria. En este caso se seleccionan un conjunto de direcciones de memoria para codificar las direcciones de los puertos. El decodificador de direcciones de cada puerto se encarga de comprobar si la dirección (en principio de memoria) depositada en el bus de direcciones corresponde a su código, en cuyo caso el controlador actúa. Para evitar fragmentar el mapa de memoria se agrupan las direcciones para E/S en una zona bien definida, usualmente al principio o fin del mapa de memoria.

Conviene notar que, aunque el procesador disponga de instrucciones específicas de E/S, siempre se pueden codificar los periféricos dentro del mapa de memoria.

5.3. Técnicas de transferencias de entrada/salida.

Hay, fundamentalmente, tres métodos para establecer y supervisar las operaciones de transferencias de datos de E/S: E/S programada, E/S controlada por interrupciones y E/S con controlador de acceso directo a memoria (DMA).

5.3.1. Control por medio de E/S programada.

En la E/S programada el inicio de la operación se efectúa a iniciativa de la ejecución por la CPU de instrucciones de un programa. Existen dos variantes: E/S programada sin, o con, consulta de estado.

E/S programada sin consulta de estado o incondicional.

El programa, por medio de la CPU, decide el momento en que se realiza la transferencia, por lo que el dispositivo de E/S debe estar siempre dispuesto a

recibir datos (en el caso de una salida) o debe tener siempre datos disponibles (en el caso de una entrada).

E/S programada con consulta de estado.

Antes de realizar una operación de E/S de un dato de un determinado periférico, el programa incluye instrucciones para leer el puerto de estado del controlador del periférico, y comprueba si está listo o no para realizar la transferencia. Obviamente, en el puerto de estado existirá un bit dedicado a almacenar si el periférico está preparado para efectuar la operación de salida o, en el caso de una entrada, si en el puerto de entrada de datos se encuentra el dato listo para ser transferido.

La iniciativa de la transferencia la sigue teniendo el programa, pero para transmitir cada dato es necesario leer los bits de estado. Con este método se adapta la velocidad de transferencia de la CPU a la del periférico. Hay dos formas de efectuar la consulta de estado: con bloqueo de programa y con consulta periódica. La consulta de estado con bloqueo de programa, implica que cuando hay que hacer una operación de entrada o salida y el periférico no está preparado, el programa entra en un lazo de espera hasta que el dispositivo llegue a la situación de "listo". En el procedimiento con consulta periódica o de escrutinio, periódicamente se consulta el periférico o conjunto de periféricos para ver si están preparados para recibir datos o si tienen datos para la CPU, caso de no ser así el programa pasa a realizar otras operaciones, que no necesiten del dato a captar (caso de una entrada).

5.3.2. Control por medio de interrupciones.

En el método de E/S controlada por interrupciones, el periférico toma la iniciativa y solicita ser atendido, activando una línea de petición de interrupción. En las salidas, el periférico puede provocar una petición de interrupción si no está listo para realizar la transferencia. Cuando la CPU reconoce la interrupción, suspende el programa en curso y ejecuta un programa de gestión de interrupción para transferir información con los puertos del controlador del periférico en cuestión. Como en el caso de E/S programada, los pasos de transferencia de datos quedan bajo el control de un programa, en este caso, del programa gestor de la interrupción.

La rutina de gestión de la interrupción debe identificar el periférico que solicitó la interrupción (caso de que varios de ellos compartan la misma línea de petición de interrupción) y debe efectuar la transferencia de un dato o de un bloque de datos.

Básicamente se plantean tres problemas para atender una interrupción: determinación de la dirección de la rutina de servicio de la interrupción, identificación del origen de la interrupción y gestión de prioridades de peticiones, ya que en un momento dado pueden existir varias peticiones de interrupción sin haber sido atendidas. Un computador, por lo general, tiene conectados una gran variedad de dispositivos con poder de interrupción, y la

acción requerida dependerá de su causa. Es, por tanto, necesario identificar la causa o dispositivo que produjo la interrupción.

Uno de los principales problemas que se suele plantear es la gestión de interrupciones simultáneas. Cuando se solicita una interrupción, ésta no se acepta hasta que la instrucción (o ciclo de bus) que esté en ejecución termine. Durante ese breve tiempo pueden haberse generado otras interrupciones, que requieren ser atendidas. El procedimiento de prioridad debe determinar qué instrucción se atiende primero.

Básicamente se suelen considerar cuatro tipos de soluciones al problema planteado:

- Gestión de prioridades centralizada por la propia CPU. Si sólo hay un dispositivo por cada línea de interrupción, la CPU acepta la solicitud que llegue por la línea de mayor prioridad.
- Gestión de prioridades centralizada en circuitos externos. En este caso se utilizan circuitos especiales externos a la CPU que resuelven por *hardware* el problema.
- Gestión de prioridades por sondeo o consulta (*polling*). La entrada/salida se controla por interrupción, pero la rutina de servicio lanza un programa de sondeo para ver qué dispositivo solicitó la interrupción.
- Gestión de prioridades por encadenamiento (*daisy-chain*). El diseño del sistema de interrupciones se hace de forma que tiene mayor prioridad el dispositivo que esté "eléctricamente" más cercano a la CPU. Los distintos elementos del computador (CPU, módulos de memoria, etc.) y controladores de periféricos, suelen insertarse en distintas ranuras del chasis del computador, y cada ranura tiene asociada una prioridad.

5.3.3. Control por medio de controlador DMA.

Debido a lo rutinario de la programación de las operaciones de entrada/salida y a la gran frecuencia de su utilización, se ha ideado la técnica de acceso directo a memoria o DMA (*Direct Access Memory*) que permite descargar a la CPU de las operaciones más elementales de E/S, y que implica la utilización de circuitos especializados (controladores DMA o CDMA).

El CDMA actúa como controlador del bus y supervisa las transferencias de información entre memoria principal y uno o más controladores de E/S, sin intervención directa de la CPU, salvo en la inicialización y final. Tanto en los casos de E/S programadas como controladas por interrupción, debe ejecutarse por la CPU una rutina gestora del periférico que controla minuciosamente la operación de entrada/salida; la ejecución de estas instrucciones es muy frecuente y consume un considerable tiempo de CPU. El controlador DMA persigue resolver este problema y con él se pueden lograr velocidades de transferencia sólo limitadas por la velocidad de los buses y los tiempos de

acceso a memoria y al periférico. Este incremento de la velocidad es posible ya que la mayoría de los pasos para efectuar una E/S se realizan por hardware: mantenimiento de la dirección de memoria a acceder en un contador especializado, etc. y es especialmente notable cuando se transfieren bloques de información entre memoria principal y un periférico y viceversa.

Cuando va a actuar el CDMA, la CPU le cede el control del bus del sistema (salvo algunas líneas del bus de control) encargándose aquel de generar tanto las direcciones de memoria y del dispositivo implicado en la transferencia, como las señales de control necesarias para llevar a buen término las mismas. En definitiva, el CDMA actúa como controlador o maestro del bus del sistema.

Una vez inicializado el CDMA, cada transferencia de palabra de datos se realiza en sólo un ciclo de escritura (o lectura) de memoria (uno o pocos ciclos de reloj). Por contra, una E/S programada o realizada con una rutina de servicio de interrupción necesita la ejecución de varias instrucciones (numerosos ciclos de reloj).

La técnica DMA se utiliza principalmente con periféricos rápidos, tales como discos, controladores gráficos, impresoras de alta velocidad, sistemas de adquisición de datos, etc.

6. Conclusiones.

El modelo propuesto por Von Neumann define la arquitectura elemental de los ordenadores digitales modernos, formados por las siguientes unidades: unidad aritmético-lógica, unidad de control, unidades de entrada/salida y memoria principal.

La CPU contiene tres elementos principales para poder efectuar su trabajo: la unidad de control, la unidad aritmético-lógica y el banco de registros. La unidad de control es la encargada de leer las instrucciones de la memoria principal, decodificarlas y ejecutarlas. En su ejecución puede ser necesario emplear la unidad aritmético-lógica, que se encarga de transformar los datos de entrada en datos de salida.

Los registros, la memoria caché y la memoria principal forman lo que se conoce como memoria interna del ordenador. En la actualidad, la naturaleza de la señal que se emplea para almacenar información en esta memoria es eléctrica.

Las memorias eléctricas pueden clasificarse en memorias de lectura y escritura: RAM; y memorias de sólo lectura: ROM, PROM, EPROM y EEPROM. Si el elemento que se emplea para almacenar un bit de información en una memoria RAM es el transistor, hablaremos de memorias estáticas o SRAM; mientras que si es un condensador, hablaremos de memorias dinámicas o DRAM.

Las unidades de entrada/salida permiten que la CPU pueda comunicarse con los diferentes dispositivos de entrada/salida, ofreciendo una abstracción de las

características de éstos. Generalmente, para poder utilizar un dispositivo de entrada/salida es necesario disponer del controlador adecuado que permita la comunicación entre una determinada aplicación y el propio dispositivo.

Uni10.es