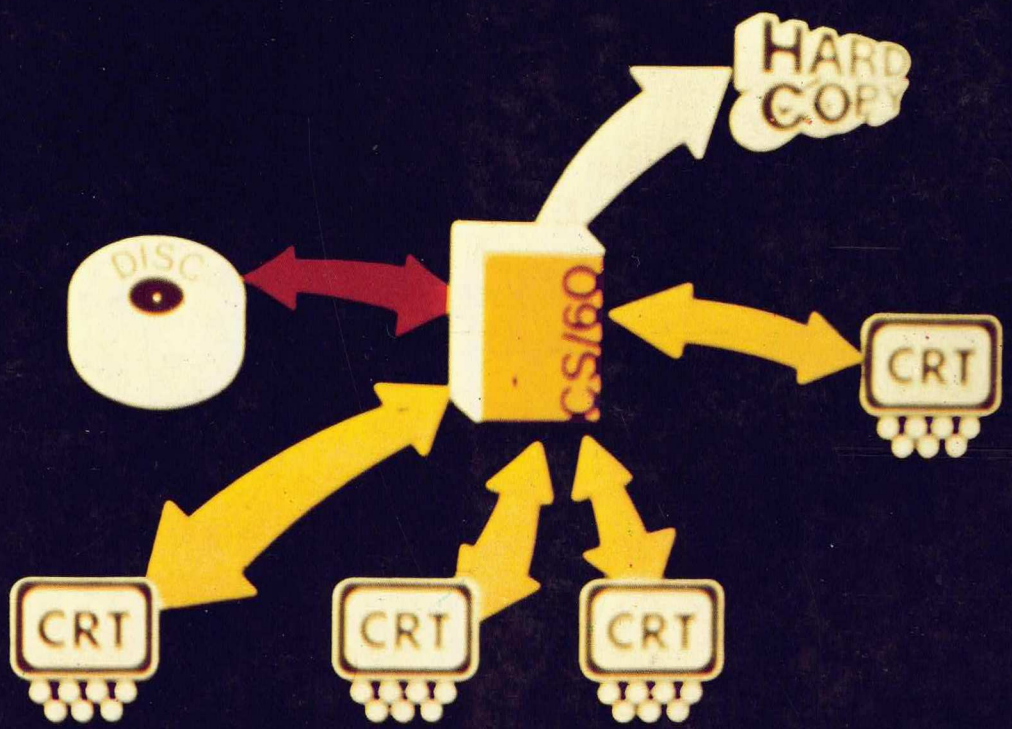
PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS



EDITORIAL tesys

TOMO 4

El presente tratado de **Programación de Ordenadores** está compuesto por un conjunto de temas que constituyen la estructura imprescindible para todo aquel que desee un amplio conocimiento del campo de la Programación.

Dichos temas vienen englobados en volúmenes diversos que a continuación se detallan; constituyéndose por separado como unidades monográficas susceptibles de un estudio particular, al margen del conocimiento conjunto y global de todos ellos, como compendio completo para el logro final de programar.

La agrupación por tomos se desarrolla de la siguiente forma:

Tomo 1. **Estructura Básica del Proceso de datos.**

Tomo 2. **Estructura Básica del Proceso de datos.**

Tomo 3. **Estructura Básica del Proceso de datos.**

Tomo 4. **Estructura Básica del Proceso de datos.**

Tomo 5. **Estructura Básica del Proceso de datos.**

Tomo 6. **Lenguaje Basic**

Tomo 7. **Lenguaje Cobol**

Tomo 8. **Lenguaje Cobol**

Tomo 9. **Leyes de Construcción de programas.**

Tomo 10. **Diccionario de Proceso de datos**

El tratamiento de los temas comprende tanto los aspectos teóricos como las aplicaciones al diseño de estructuras básicas de posible uso inmediato, reagrupándose en los siguientes conjuntos específicos que por sí mismos pasan a ser unidades independientes.

Son los siguientes:

Grupo I: Tomos 1, 2, 3, 4 y 5. **(Estructura Básica del Proceso de Datos).**

Grupo II: Tomo 6. **(Lenguaje Basic).**

Grupo III: Tomos 7 y 8. **(Lenguaje Cobol).**

Grupo IV: Tomo 9. **(Leyes de Construcción de Programas).**

PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS

PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS

AUTOR

JOSE LUIS MORALES PASCUAL
Licenciado en Informática

COLABORADORES

CARLOS ANTONIO MARTINEZ TREVIN
Ingeniero de Telecomunicación

ANDRES J. ERNESTO RUANA GIRONES
Ingeniero Industrial

BARCELONA (ESPAÑA)

© E.C.C. - Barcelona, 1981

Editado por:
EDITORIAL TESYS - Bruch, 20 - Barcelona-10 (España), 1981

Primera edición: Marzo, 1981

Depósito legal: B. 25.869-1980 (IV)

I.S.B.N. 84-85838-00-9 (Obra completa)
I.S.B.N. 84-85838-04-1 (Tomo 4)

Impreso por:
GRAFOS, Arte sobre papel.
Paseo Carlos I, 157 - Barcelona-13.

Impreso en España - Printed in Spain

CONTENIDO

Introducción 9

1. Funcionamiento del ordenador (I). Transmisión y almacenamiento de la información 11

2. Funcionamiento del ordenador (II). Unidad aritmética. Unidad de control. Unidades de E/S 73

3. Funcionamiento del ordenador (III). Los Sistemas operativos 115

4. Funcionamiento del ordenador (IV). Métodos de proceso. Teletratamiento 131

Soluciones de los ejercicios 167

Indice general 168

INTRODUCCION

En este tomo de la obra de «Programación de Ordenadores» se estudia principalmente el funcionamiento de los equipos físicos (*hardware*) y los programas y métodos de funcionamiento (*software*) y aplicaciones de un sistema de proceso de datos.

En los primeros capítulos se ofrece una visión general de las funciones internas del ordenador y de la organización lógica de los mecanismos de funcionamiento del sistema de proceso de datos.

Para comprender con mayor facilidad la aplicación de los ordenadores, así como su utilización y programación es necesario entender el principio de su funcionamiento.

Junto con los *sistemas operativos* (conjunto de programas encargados del control y funcionamiento del sistema de proceso de datos) y los diferentes sistemas de funcionamiento del ordenador, se exponen, por último, unos conceptos elementales sobre *sistemas de teletratamiento* (proceso de datos a distancia), puesto que el ordenador constituye cada vez más el coordinador de las técnicas de telecomunicación.

1.

FUNCIONAMIENTO DEL ORDENADOR (I)

Transmisión y almacenamiento de la información

CUADRO RESUMEN DE LA LECCION

FUNCIONAMIENTO DEL ORDENADOR I

1.1. Transmisión de la información

- Trans. en serie
- Trans. en paralelo
- Trans. mixta
- Circuitos de conmutación
  - Circuito Y
  - Circuito O
  - Combinación de circuitos
- Puertas lógicas
  - Puerta Y
  - Puerta O
  - Inversor
  - Puertas NO-O
  - Puertas NO-Y
  - Combinaciones de puertas
  - Puerta de autorización
  - Puerta de prohibición
  - Selector de dirección
  - Decodificador

FUNCIONAMIENTO DEL ORDENADOR I

1.2. Almacenamiento de la información

- Unidades de información
- Representación de los datos en la M.C.
  - Datos numéricos
  - Datos alfanuméricos
- Acceso a la M.C.
  - Memoria de acceso aleatorio
  - Memoria de solo lectura
  - Memoria asociativa
- Constitución de la M.C.
  - Mem. de núcleos magnéticos
  - Mem. de agujas magnéticas
  - Mem. de hilos magnéticos
- Memorias de gran capacidad
- Memorias flip-flop
- Registros
  - Circuito de mantenimiento
  - Registro acumulador
  - Registro de desplazamiento
- Clasificación de los soportes de almacenamiento

FUNCIONAMIENTO DEL ORDENADOR

La finalidad del ordenador es la de ejecutar programas. Para ello, cada ordenador se sirve de la Unidad Central de Proceso (UCP), de las unidades de Entrada y de las unidades de Salida.

La UCP recibe y almacena el programa y ejecuta las instrucciones del mismo. Las unidades de entrada (lectora de fichas por ejemplo) leen los soportes de información de entrada (fichas, cintas perforadas, cintas magnéticas, etc.) e introducen los datos y las instrucciones en la memoria de acuerdo con las instrucciones del programa. Las unidades de salida (impresora, perforadora de fichas, etc.) reciben de la memoria central (MC) los resultados del proceso y los escriben en los soportes de salida.

El ordenador, por medio del programa, actúa sobre los datos almacenados siguiendo un proceso lógico y alterando, convenientemente, las instrucciones del programa durante su ejecución.

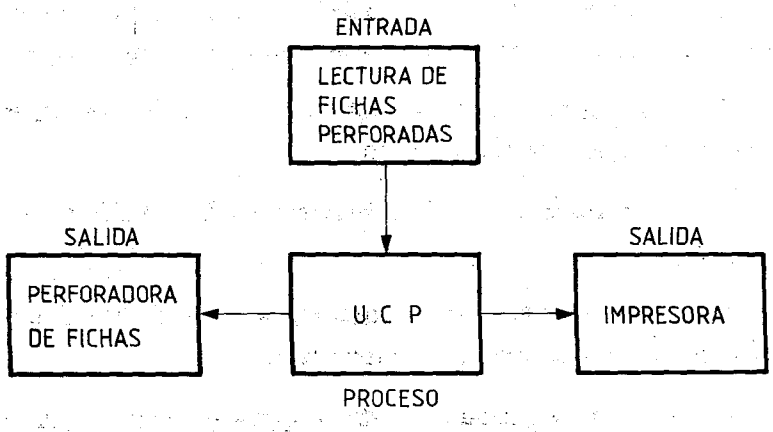


Fig. 1.1. Esquema del funcionamiento de un ordenador.

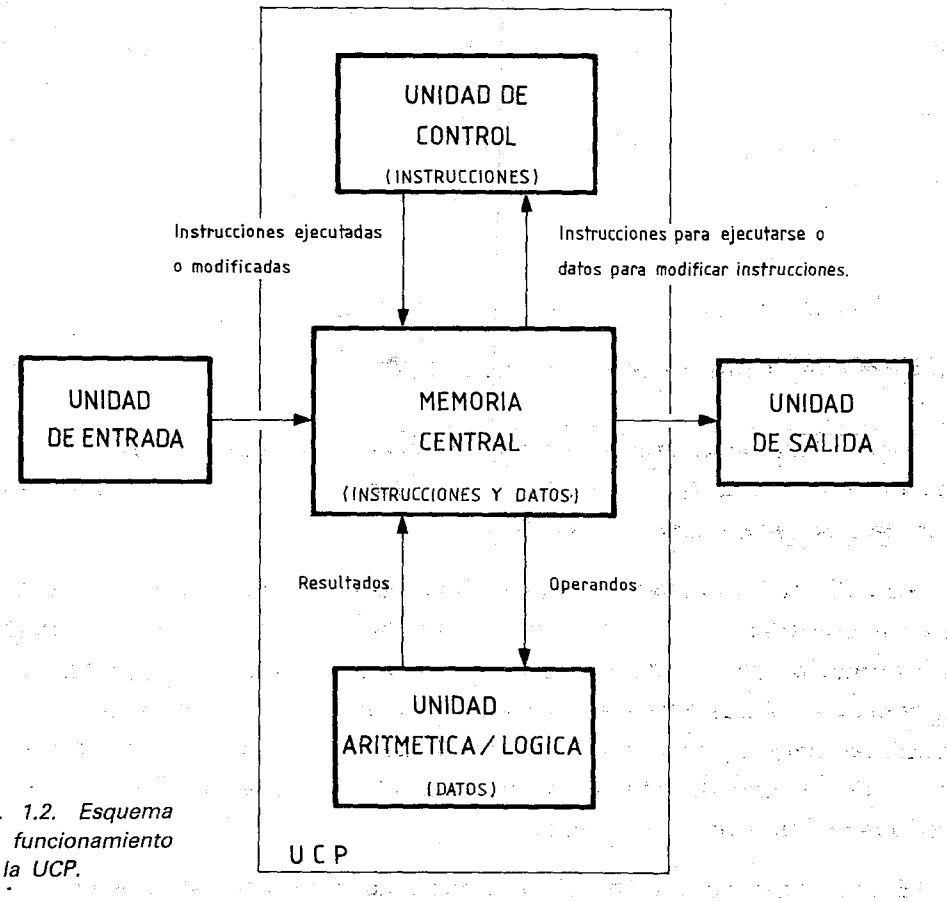


Fig. 1.2. Esquema del funcionamiento de la UCP.

La unidad central de proceso (UCP) está constituida por la memoria central (MC), la unidad de control (UC) y la unidad aritmética/lógica (UAL).

En la MC se almacenan los datos y las instrucciones de los programas necesarios para realizar un proceso.

Las instrucciones se transmiten desde la MC a la UC siguiendo la secuencia del programa.

Asimismo, la MC puede proporcionar a la UC los datos necesarios para posibles modificaciones de instrucciones.

La UC controla el desarrollo del programa ejecutando las instrucciones que recibe de la MC, en donde vuelven las instrucciones una vez ejecutadas o modificadas junto con los datos de las modificaciones.

La UC, en el momento de recibir las instrucciones de la MC, realiza la carga de los datos desde la MC a la UAL y controla las operaciones de cálculo que contienen las instrucciones y que se realizan en la UAL. El resultado de estas operaciones se almacena en la MC.

La UC, además, se encarga de controlar las operaciones de entrada y salida de los datos e instrucciones hacia la MC y desde la MC.

1.1 TRANSMISION DE LA INFORMACION

La transmisión de la información se efectúa en forma de *impulsos* de corriente aplicados a conductores eléctricos situados entre los distintos elementos que componen el sistema de proceso de datos.

Un impulso consiste en una corriente o una tensión eléctrica que se mantiene constante durante un cierto tiempo (*duración del impulso*).

Los impulsos de señales eléctricas son sucesiones de valores elevados o de *nivel alto* (1) durante un tiempo muy corto, y valores nulos o de *nivel bajo* (0) durante un tiempo mayor. En la práctica no es posible conseguir la forma ideal del impulso.

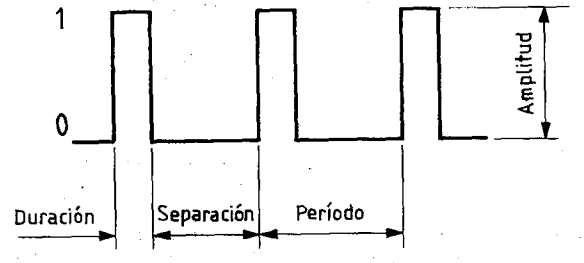


Fig. 1.3. Factores determinantes de los impulsos.

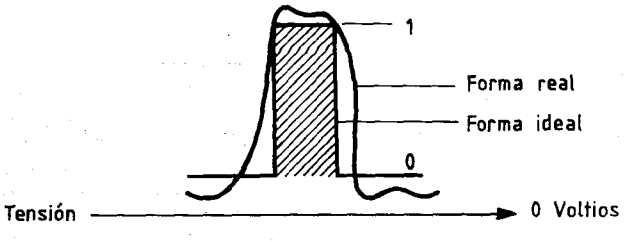


Fig. 1.4. Comparación entre impulso real e impulso ideal.

Un impulso genera el bit 1 y la ausencia de impulsos produce el bit 0.

Para que esto pueda suceder correctamente es necesario que en la transmisión de informaciones exista una sincronización en el tiempo entre los elementos que envían la información y los que la reciben. Esto se consigue por medio de impulsos a frecuencias determinadas.

El generador mediante el cual se obtienen los impulsos se denomina *reloj* ya que determina las relaciones en el tiempo entre los distintos elementos del ordenador.

Por ejemplo, si se desea transmitir una cifra decimal desde un punto a otro del ordenador deben transmitirse por lo menos 4 bits, ya que para representar una cifra decimal en binario (código BCD) son necesarios 4 bits. Dichos bits pueden transmitirse de dos formas distintas:

1) Utilizando una sola línea y transmitir sucesivamente (en *serie*) los 4 bits.
2) Utilizando cuatro líneas y transmitir simultáneamente (en *paralelo*) los 4 bits.

Por lo tanto, existen dos sistemas de transmisión de la información:

— Transmisión en serie

— Transmisión en paralelo

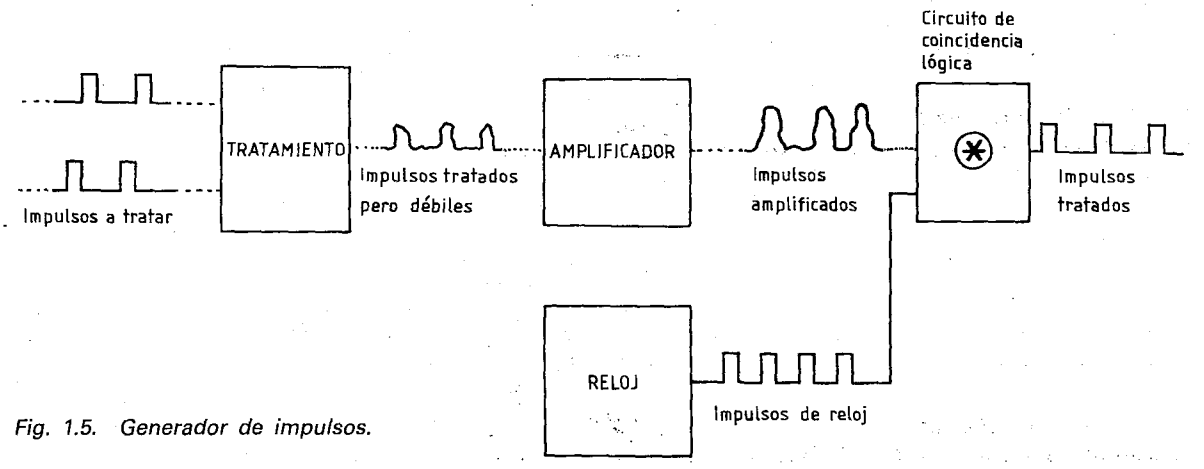


Fig. 1.5. Generador de impulsos.

1.1.1. Transmisión en serie

En el sistema de transmisión en serie, las informaciones codificadas en binario se transmiten a través de una línea o conductor bit a bit, uno a continuación de otro.

Antes de la transmisión se almacena la información en un registro (emisor) de la memoria. En la memoria se encuentra también el registro receptor correspondiente.

La transmisión de la información se efectúa de forma que cada impulso se desplaza un lugar binario a la derecha, tanto las informaciones del emisor como las del receptor.

La duración de los impulsos depende de la velocidad de transmisión, que a su vez viene determinada por la frecuencia del reloj.

Los impulsos del reloj son los que determinan en qué instante se debe leer la información en el receptor para no confundir un bit 0 con un tiempo de separación entre bits (ya que en el tiempo de separación no existen impulsos).

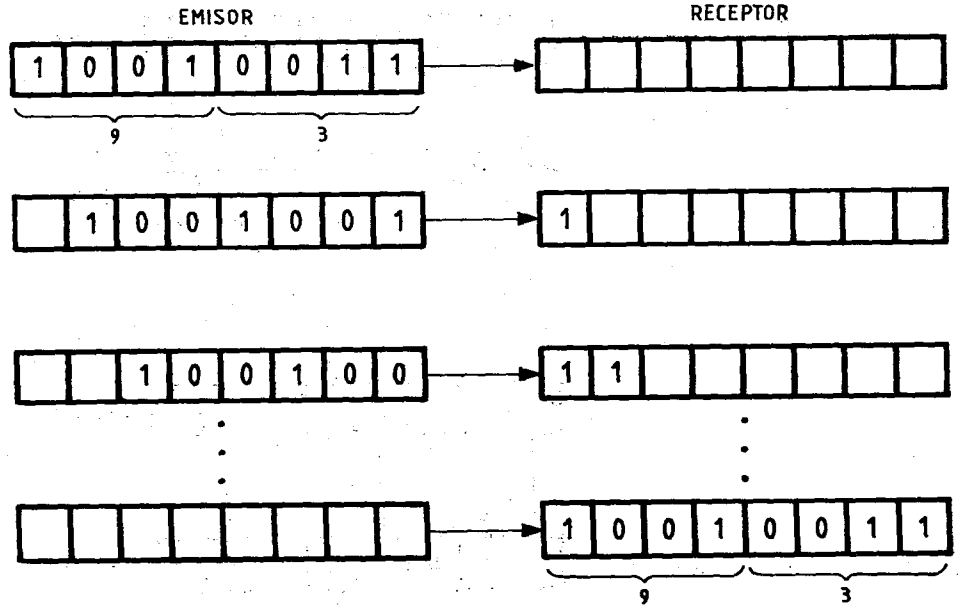


Fig. 1.6. Transmisión en serie.

1.1.2. Transmisión en paralelo

Para transmitir información en *paralelo* se necesitan tantos conductores como bits tengan que transmitirse.

Cuando se transmite en paralelo, cada conductor recibe el impulso correspondiente al nivel alto si transmite un bit 1, o quedará en el nivel bajo si es un bit 0.

Así pues, en la transmisión en paralelo debe tenerse en cuenta que cada bit sigue un conductor particular y que todos los bits se transmiten al mismo tiempo.

En la transmisión en paralelo un impulso, en cada uno de los bits, produce la transmisión de todos los bits del registro emisor al registro receptor. Por lo tanto, la velocidad de transmisión es muy superior a la del sistema en serie, ya que la transmisión en serie el tiempo se descompone en tantas fases como bits haya que transmitir.

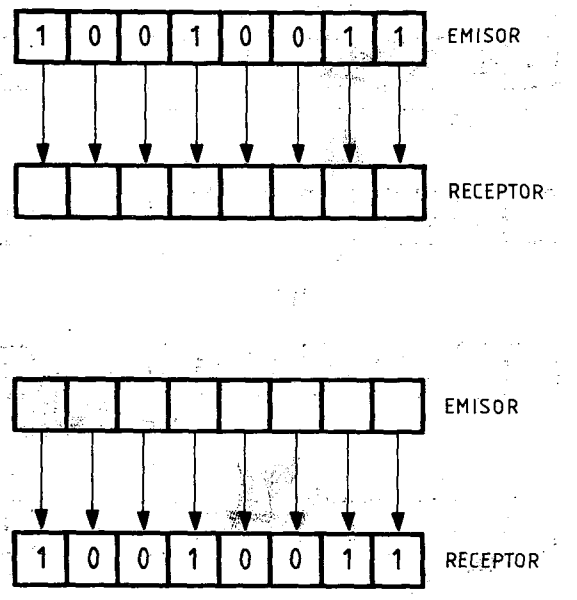


Fig. 1.7. Transmisión en paralelo.

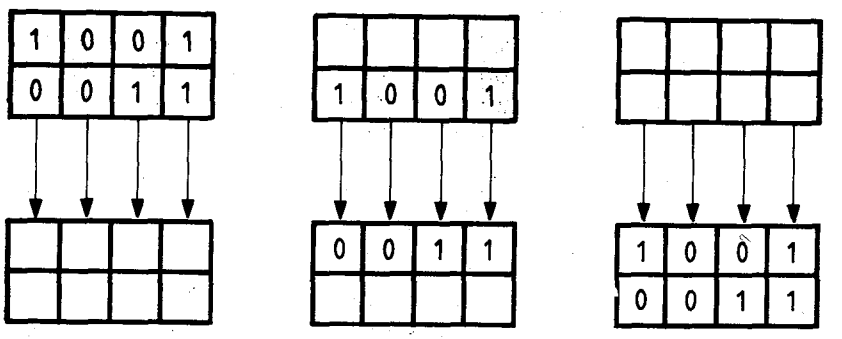


Fig. 1.8. Transmisión mixta (serie-paralelo).

1.1.3. Transmisión mixta

En el sistema de transmisión mixto (serie-paralelo) se transmiten simultáneamente (en paralelo) todos los bits de un grupo del registro emisor (por ejemplo, los cuatro bits que representan una cifra decimal), y, por otro lado, se transmiten sucesivamente (en serie) cada uno de los grupos del registro.

Por ejemplo, en la transmisión de cifras decimales, cada cifra queda definida con 4 bits. Las cifras 9 (en binario 1001) y 3 (en binario 0011) se transmiten en serie, es decir, primero se trasmite el 3 y a continuación el 9. Pero la transmisión de cada cifra es en paralelo, es decir, los cuatro bits que forman cada cifra se transmiten al mismo tiempo.

1.1.4. Circuitos de conmutación

Para conocer cómo se realizan las funciones del ordenador es conveniente el estudio de los circuitos de conmutación (un *conmutador* es un dispositivo eléctrico que sirve para que una corriente cambie de conductor).

En un ordenador existe un conjunto de elementos lógicos digitales que realizan las operaciones elementales de conmutación.

Estos circuitos eléctricos se encuentran en el ordenador formando redes de conmutación.

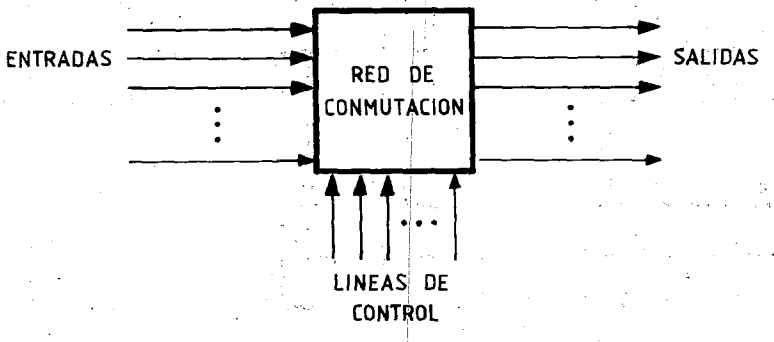


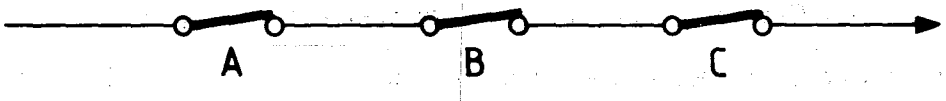
Fig. 1.9. Esquema de un circuito de conmutación.

Una red de conmutación está provista de un conjunto de entradas y un conjunto de salidas. Además existe otro conjunto de entradas formado por las *líneas de control* que controlan la secuencia de los conductores de conmutación y hacen que se abran o se cierren en un momento determinado.

Los contactos de un circuito sólo pueden encontrarse en dos estados distintos: abiertos o cerrados. El álgebra de Boole se utiliza para determinar el mínimo número necesario de contactos para obtener una secuencia dada.

Circuito Y

En un circuito de conmutación Y sólo pasará corriente eléctrica cuando todos los contactos del circuito (conmutadores o interruptores) estén cerrados de forma que el circuito estará también cerrado. En un circuito Y los contactos están en serie.



Por ejemplo, supongamos tres interruptores en serie, A, B y C. Para que pase la corriente por este circuito han de estar los tres interruptores cerrados (en estado 1).

Cuando uno, al menos, de los interruptores está en estado 0 la salida del circuito es 0.

La tabla de la verdad correspondiente al circuito Y es la siguiente:

A	*B*	*C*	✳
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

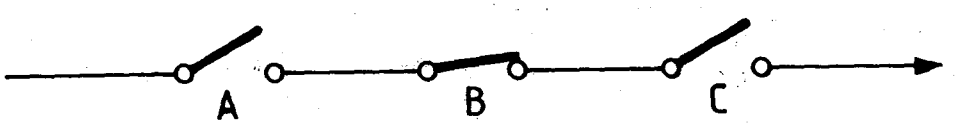
En la tabla de la verdad, se representa un interruptor cerrado o activado por un 1, y un interruptor abierto o no activado por un 0.

Otra forma de representar las dos condiciones en que pueden encontrarse los interruptores de un determinado circuito es mediante ecuaciones. Por ejemplo:

$$y = A \ast B \ast C$$

Si el valor de esta variables, A, B y C, es 1, la salida *y*, será 1.

Supongamos ahora un circuito en el que no están cerrados todos los interruptores que lo componen.



Para que la salida *y* sea 1, es decir, la condición de cerrado de este circuito se obtiene con la ecuación:

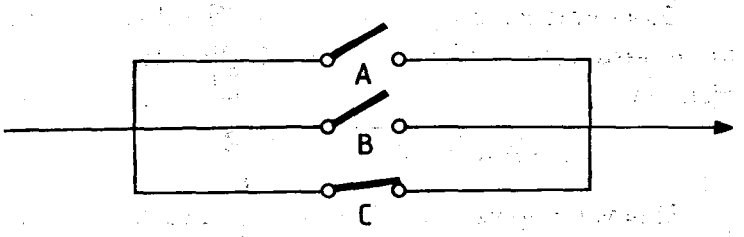
$$y = \overline{A} \ast B \ast \overline{C}$$

donde A y C representan interruptores abiertos, y B un interruptor cerrado. La tabla de la verdad para este caso será:

A	B	C	×
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Circuito O

Los interruptores que componen un circuito O se encuentran en paralelo. Por lo tanto, un circuito O estará cerrado (es decir, pasará corriente) mientras uno de los interruptores por lo menos esté cerrado.



Por ejemplo, en un circuito con tres interruptores, A, B y C, la salida será 0 solamente cuando A, B y C sean 0 (estén abiertos).

La tabla de la verdad correspondiente al circuito 0 es la siguiente:

A	B	C	+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Si los interruptores A y B están abiertos y C cerrado, la condición de abierto para este circuito se tiene con la siguiente ecuación:

$$y = A + B + \overline{C}$$

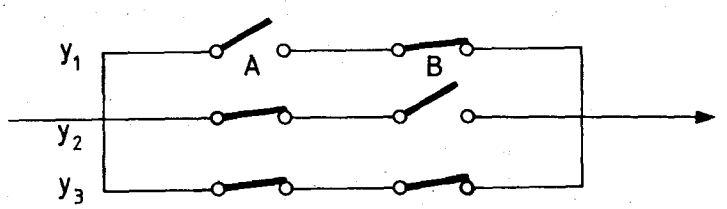
La tabla de la verdad de este circuito es la siguiente:

A	*B*	*C*	+
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Combinaciones de circuitos

Generalmente, los circuitos Y y O no se utilizan solos, sino combinaciones de ambos.

Supongamos el siguiente circuito, compuesto a su vez por tres circuitos simples:



Para que la salida de cada circuito en particular sea 1, se tienen las siguientes ecuaciones:

$$y_1 = \overline{A} \times B$$

Estos circuitos pueden combinarse como circuito O obteniendo la ecuación

$$y_1 + y_2 + y_3 = \overline{A} \times B + A \times \overline{B} + A \times B$$

La tabla de la verdad correspondiente a esta ecuación es:

A	B	$\bar{A} * B$	$A * \bar{B}$	$A * B$	+
0	0	0	0	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	1

1.1.5. Puertas lógicas

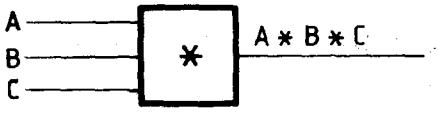
Para obtener la conmutación de señales eléctricas se aplica cierta tensión a las entradas de un circuito lógico llamado *puerta.* Esta tensión se representa por un 1. El símbolo 0 representa la ausencia de tensión eléctrica.

Los operadores o circuitos lógicos Y (AND), O (OR), NO (NOT) pueden considerarse como puertas lógicas.

Puertas Y

Si todas las entradas de una puerta Y son 1, se dice que la puerta está *abierta* y entonces es posible la transmisión de información a través de esta puerta. Es decir, en una puerta Y sólo hay salida cuando todas las entradas están en nivel 1.

Por lo tanto, el circuito Y puede considerarse como una puerta abierta (1) cuando recibe una señal y cerrada (0) cuando no recibe esta señal.



La tabla de la verdad de la puerta Y es:

A	B	C	*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Puertas O

Un circuito lógico OR puede considerarse como una puerta abierta (1) si una cualquiera de las entradas o una combinación de ellas está en nivel 1

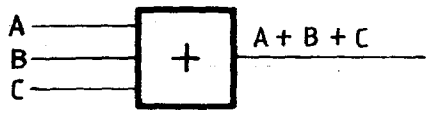
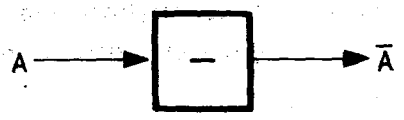


Tabla de la verdad de la puerta:

A	B	C	+
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Inversor

El circuito *inversor* o complemento está constituido por un transistor cuya salida es la inversa de la entrada.

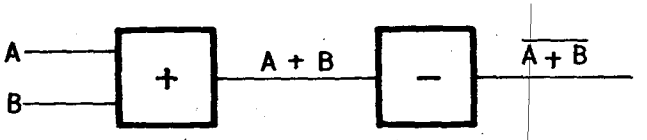


El inversor convierte la entrada A en Ā. Debe estar diseñado de forma que se obtenga a la salida un nivel opuesto al de la entrada. Por ejemplo, si se representa el nivel alto, 1, por una tensión de 10 voltios, y el nivel bajo, 0, por 0V; cuando se aplique a la entrada un 1, la salida debe ser 0V; mientras que si se aplica a la entrada un 0, la salida será 10V.

Puertas NO-O

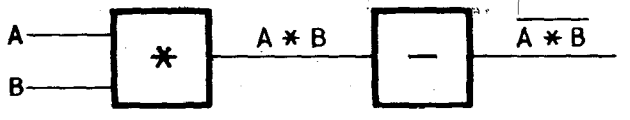
En el circuito que realiza la función NO-O se obtiene a la salida una tensión correspondiente al nivel 1 cuando no se aplica señal en ninguna de las entra-

das. En caso contrario, es decir, si se aplica a cualquier entrada un nivel 1 aparece un 0 a la salida del circuito.



Puertas NO-Y

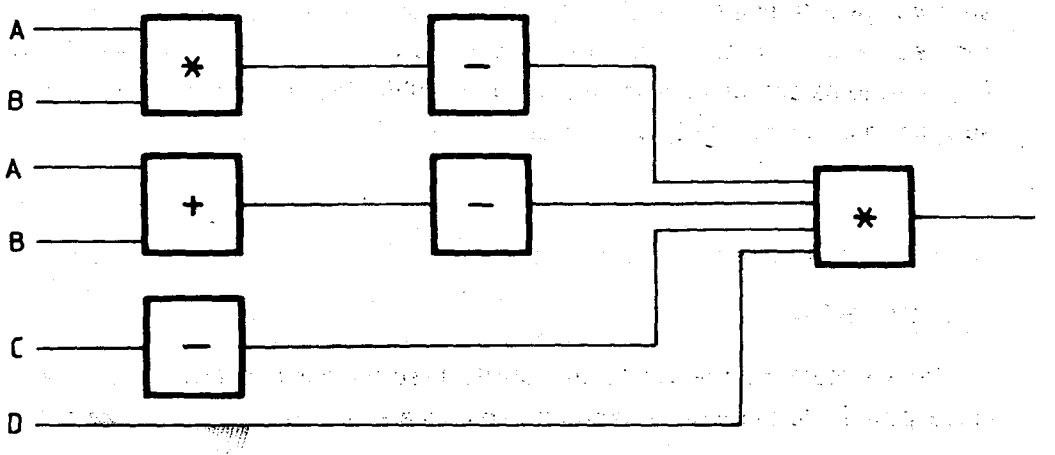
Los transistores de una puerta NO-Y tienen un funcionamiento inverso al de los transistores de una puerta NO-O. La salida será 1 cuando exista algún nivel 1 en la entrada. Cuando todas las entradas se encuentren en nivel 1, la salida será 0.



Combinaciones de puertas

Las puertas lógicas Y y O junto con los inversores se utilizan frecuentemente formando combinaciones de puertas. Estas combinaciones pueden expresarse como ecuaciones del álgebra de Boole.

Por ejemplo, supongamos el siguiente circuito:



A partir de este circuito puede obtenerse la ecuación correspondiente:

$$y = \overline{(A * B)} * \overline{(A + B)} * \bar{C} * D$$

El problema inverso consiste en hallar el diagrama por bloques o lógico a partir de una ecuación dada.

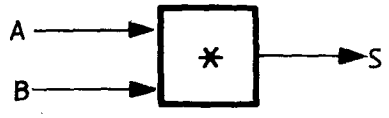
Sea, por ejemplo, la ecuación:

$$y = (A + B) * C * D + \bar{A} * \overline{(B + C)} * D$$

En las figuras 1.10 y 1.11 se muestran dos formas diferentes de representar el diagrama de esta ecuación.

Puerta de autorización

Una puerta de autorización es un circuito Y de dos entradas. Por una entrada recibe una señal portadora de información para ser transmitida; y por la otra entrada recibe una señal que autoriza o prohíbe la transmisión.



La entrada A representa la señal portadora de la información a transmitir. Siendo B la señal que autoriza o prohíbe dicha transmisión.

B	*A*	*S*
0	0	0
0	1	0
1	0	0
1	1	1

En la tabla de la verdad puede comprobarse que cuando la señal de autorización es 1, la salida S es igual a la entrada A. Así pues, cuando la señal de autorización es 0, la salida también es 0 cualquiera que sea el valor de la entrada A. Este circuito Y es una puerta de autorización. La señal B autoriza o transmite la información de A hasta S.

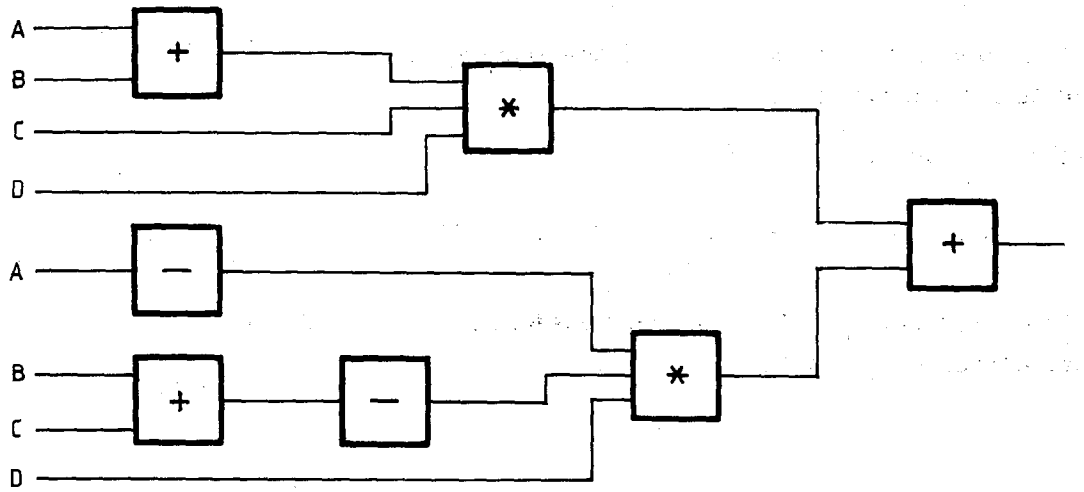


Fig. 1.10. Diagrama correspondiente a la ecuación

$y = (A + B) * C * D + \overline{A} * \overline{(B + C)} * D.$

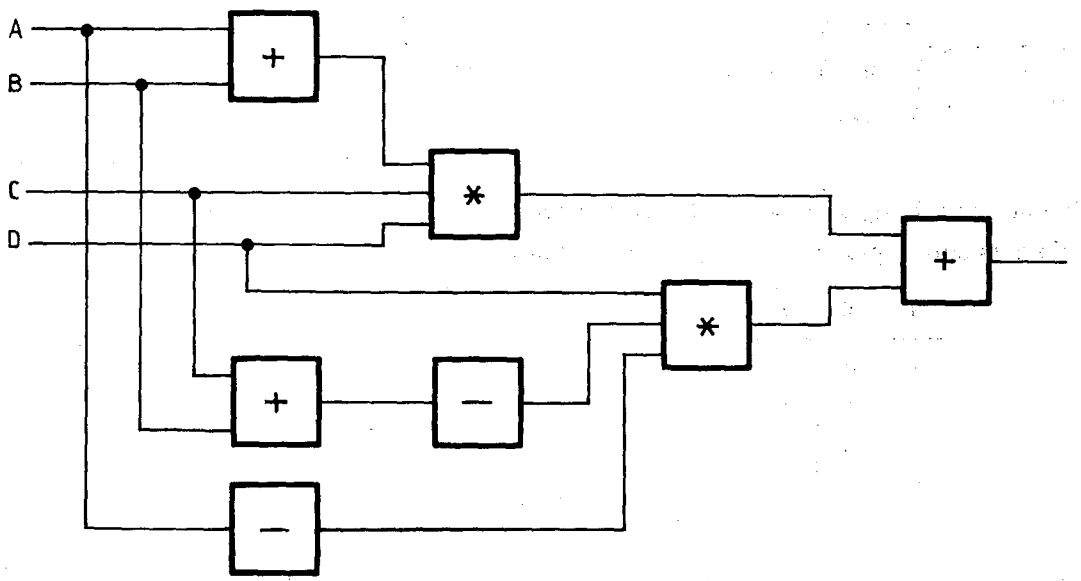
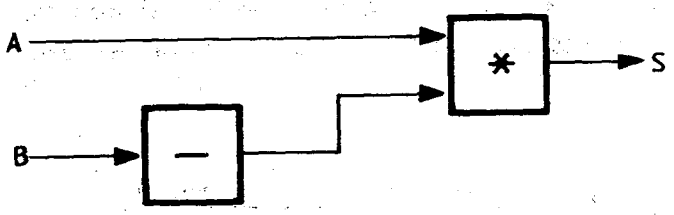


Fig. 1.11. Otra forma de representar el diagrama de la figura anterior.

Puerta de prohibición

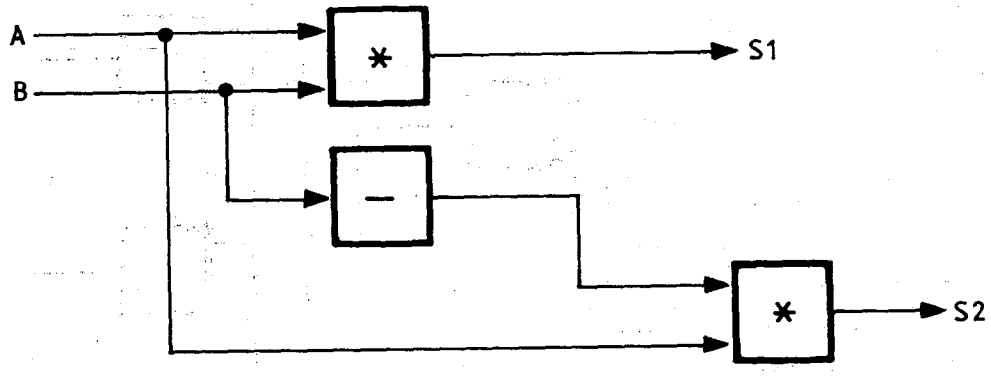
Una *puerta de prohibición,* llamada también *circuito de inhibición,* no permite el paso de la información de entrada hacia la salida. Es decir, cuando la señal de prohibición, B, es 1, la salida, S, siempre será O, cualquiera que sea el valor de la señal de entrada, A.



B	$\overline{B}$	A	S
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

Selector de dirección

La combinación de las puertas de autorización y prohibición permite el paso de la información por una vía o por otra, prohibiendo o invalidando la vía no utilizada.



Si la señal de entrada, A, es 1 y la señal B es 1, entonces la salida S1 será 1 y la salida S2 del circuito será 0.

En el caso de que A sea 1 y B sea 0, la salida S1 será 0 y S2 será 1, puesto que $\overline{B} = 1$.

Por lo tanto, con la combinación de las puertas de prohibición y de autorización se tiene un *selector de dirección.* (B) que permite el paso de la información desde A hasta S1 o desde A hasta S2 según sea el valor (0 o 1) de la señal B.

Decodificador

Mediante la combinación de selectores puede obtenerse un *decodificador.* Se trata de colocar un nuevo selector en cada una de las salidas del primero, a continuación otro selector sobre cada una de las nuevas salidas y así sucesivamente.

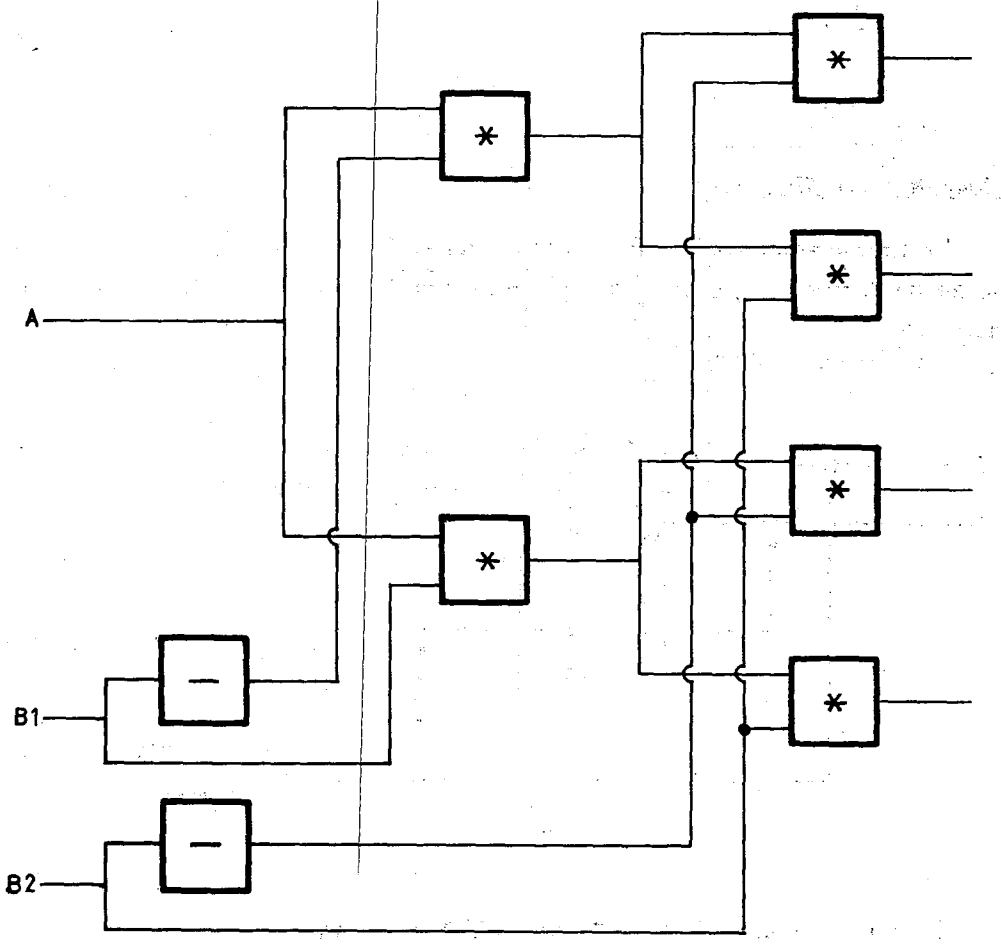


Fig. 1.12. Diagrama de un codificador.

Una dirección de dos bits es suficiente, por tanto, para seleccionar una de las 4 (es decir, 2^2) vías de salida. Asimismo, para una dirección de tres bits se tienen 8 vías, para una de 4 bits se tienen 16 y, en general, para *n* bits se tienen 2^n vías.

Otra forma de representar un decodificador es mediante circuitos.

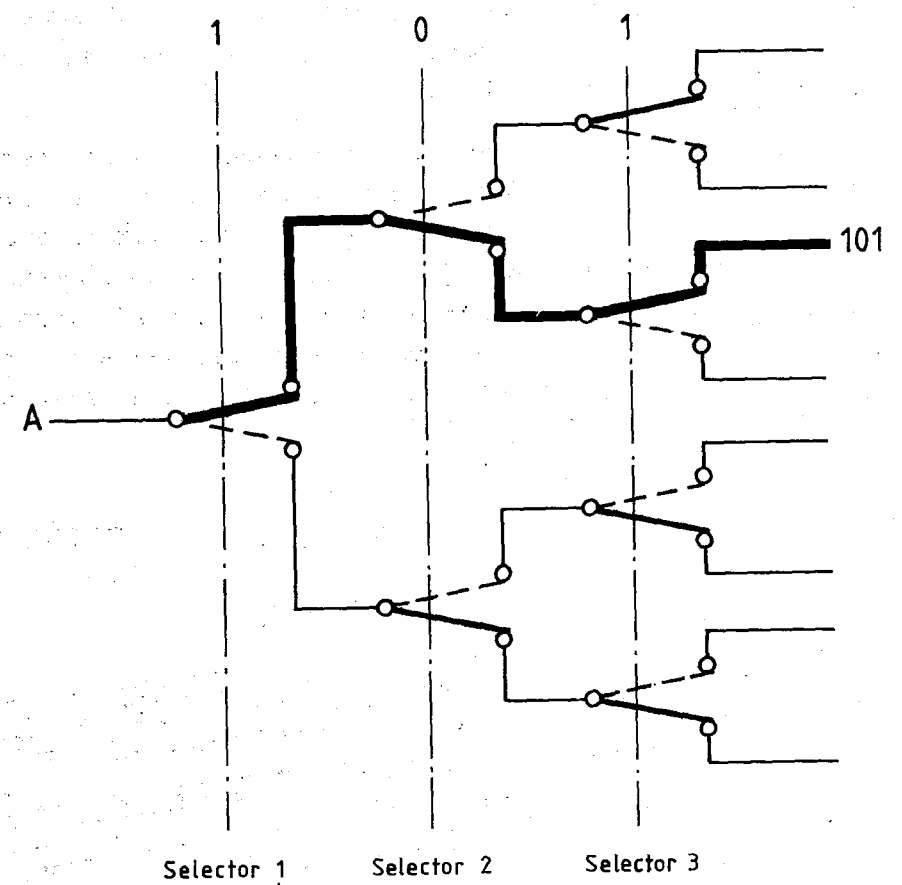


Fig. 1.13. Decodificador representado mediante circuitos.

1.2. ALMACENAMIENTO DE LA INFORMACION

La memoria central es el dispositivo de almacenamiento básico para el ordenador. En ella se almacenan los datos y las instrucciones del programa.

La MC dispone de un conjunto de zonas o posiciones de almacenamiento. Cada una de estas *posiciones de memoria* se identifica por un número llamado *dirección* de la posición de memoria.

Sencillamente la *dirección* indica el lugar donde se encuentra la zona de memoria. En esa zona de memoria se almacena una información que se llama *contenido* de la posición de memoria.

La memoria se estructura a partir de elementos capaces de almacenar la unidad básica: el bit. En una posición de memoria se almacenan varios bits formando un carácter o una palabra.

La memoria puede representarse gráficamente como una estructura cúbica. El cubo tiene un plano formado por una cuadrícula. La cuadrícula está formada por líneas que se cortan en ángulo recto. En cada punto de corte de las líneas se almacena un bit. Se coloca debajo de éste otro plano idéntico. Los puntos donde se almacenan los bits en un plano y en el siguiente están en una misma vertical y la unión de todos ellos forma una posición de memoria. El conjunto de todos los planos forma el ortoedro que constituye toda la memoria.

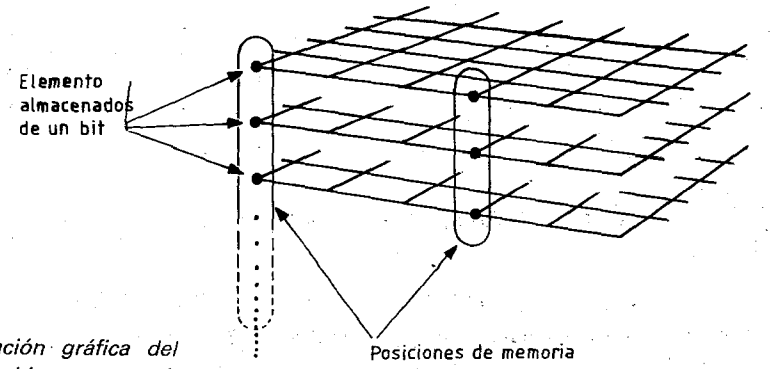


Fig. 1.14. Representación gráfica del almacenamiento de un bit en una posición de memoria.

Según este esquema una posición de memoria estaría formada por un bit del primer plano, por otro bit del segundo plano en la misma vertical, por un bit del tercer plano en la misma vertical y así sucesivamente hasta llegar al total de bits que use el ordenador para una posición de memoria.

La MC tiene la función de almacenar las instrucciones y los datos necesarios para la ejecución de los programas.

Los datos e instrucciones se almacenan en posiciones de memoria direccionables de manera que se puede acceder directamente a estas posiciones por medio de una dirección de memoria sin necesidad de seguir un orden determinado. Este tipo de acceso se llama *acceso aleatorio* o *acceso al azar* (random access).

Para almacenar información se utilizan otros dispositivos de acceso aleatorio (tambor y disco magnético) y de acceso secuencial (cuando el acceso a las posiciones de la memoria está sometido a un orden preestablecido como, por ejemplo, las cintas magnéticas).

Estos dispositivos se utilizan generalmente como memorias auxiliares o externas. Su misión es almacenar gran cantidad de datos que pueden ser transferidos a la MC cuando se necesiten para un proceso.

Las memorias auxiliares también se utilizan para almacenar los programas que se han utilizado o se van a utilizar en procesos posteriores y los programas que proporcionan los constructores de ordenadores, constituyendo lo que se llama *biblioteca de programas.*

Otros dispositivos para el almacenamiento de la información son los *registros.*

Los registros están formados por células de almacenamiento electrónicas. En cada célula puede almacenarse un bit.

1.2.2. Unidades de información

El ordenador realiza operaciones con los datos, según las directrices de un programa, para obtener resultados. Tanto los datos como las instrucciones de los programas deben ser almacenados internamente en el ordenador.

Se repasan en esta lección las estructuras básicas que puede tomar la información en el ordenador.

La información que almacena el ordenador (ya sean datos o instrucciones) presenta los formatos siguientes:

— Bit.
— Byte.
— Palabra.
— Campo.
— Registro.
— Fichero.

Bit

Se denomina *bit* a la información más elemental que puede almacenar un ordenador.

Puesto que el ordenador trabaja internamente con datos binarios, los únicos dos valores que puede tomar un *bit* son el cero y el uno.

La palabra *bit* proviene de la contracción de los términos ingleses *bi*nary digi*t.* (Dígito binario).

Todos los demás conjuntos de información proceden de esta célula básica. Reuniendo varios *bits* se tienen las demás estructuras de la información.

Byte

Los caracteres que se almacenan en la memoria están formados por un conjunto de bits (4, 6, 8, etc.). Al conjunto de 8 bits tratados por el ordenador como unidad se le denomina *byte* u octeto. Ejemplo: 10010101.

El *byte* constituye una unidad para la medida de la capacidad de almacenamiento de una memoria. Se dice, por ejemplo, que una memoria puede almacenar 1024 bytes. Puesto que las actuales memorias puede almacenar cantidades muy grandes de datos, se utiliza como medida de dicha capacidad un múltiplo de esta cantidad. El Kbyte (leído como kilobyte). La transformación de una unidad en la otra se realiza sabiendo que

1 Kbyte = 1024 bytes.

Ejemplo:

Calcular la capacidad de una memoria de 64 Kbytes.

64×1024 bytes = 65536 bytes.

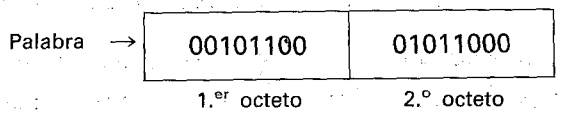
Palabra

Se denomina *palabra* al conjunto de octetos (o caracteres, en general) que trata la máquina en bloque.

En definitiva, la palabra es una medida de la potencia de la máquina. Existen máquinas que utilizan palabras de un octeto, de dos, de tres o de cuatro. Así, las máquinas trabajan con 8, 16, 24 ó 32 bits. Aumentando su potencia a medida que aumenta el número de bits que manejan.

La clasificación entre ordenadores y miniordenadores se basa en su potencia y, por lo tanto, en el número de octetos que usan para la palabra.

Ejemplo de una palabra de 2 octetos:



La relación existente entre la potencia y los octetos que se manejan se muestra con el siguiente ejemplo:

Supóngase un ordenador hipotético que manejara palabras de dos bits. Las posibilidades distintas de codificación que se tienen con dos bits son de $2^2 = 4$.

Estas posibilidades son:

00
01
10
11

Si se utiliza este ordenador, las instrucciones tendrán una longitud de dos bits (incluyendo los datos que tengan que manejar esas instrucciones) y, por lo tanto, sólo habrán 4 instrucciones diferentes. El ordenador sólo podría hacer, como máximo, cuatro acciones diferentes. La potencia de este ordenador sería muy limitada.

Si se compara este hipotético ordenador con uno real, por ejemplo, de 16 bits en cada palabra se ve la gran diferencia entre ambos: El ordenador

de 16 bits por palabra podría realizar como máximo 65536 (2^{16}) acciones diferentes (suponiendo que, se han usado los 16 bits para codificar la instrucción).

Una palabra podría usarse para la representación del conjunto de números. La capacidad del ordenador nos daría el rango de variación, es decir, el mayor y menor de los números que pueden almacenarse. Ordenadores potentes, con palabras de 32 bits, podrían almacenar cantidades inferiores o igual a 4294967295 (es decir, $2^{32} - 1 = 429496795$).

Campo

Se llama campo a una zona de memoria (no sólo de la MC sino de cualquier dispositivo de almacenamiento) a la que puede accederse en un programa.

El concepto de campo está muy ligado al concepto de variables. La variable es el nombre que se le da a una posición de memoria en el programa. Cualquier operación que el programador quiera hacer en esa zona de memoria se realizará utilizando dicha variable.

El contenido de la posición de memoria varía según las operaciones que se van efectuando a lo largo del programa.

Por ejemplo, se crea un contador en una posición de memoria (Fig. 1.15). A esa posición de memoria se le hace referencia mediante la variable I. En primer lugar, la posición de memoria que se representa por la variable I contendrá el valor 1. Se dice que I vale 1 o que I = 1. A continuación, después de incrementar el contador, la posición de memoria contendrá el valor 2, o lo que es lo mismo I = 2. Y así sucesivamente, I = 1, 2, 3, 4 ...

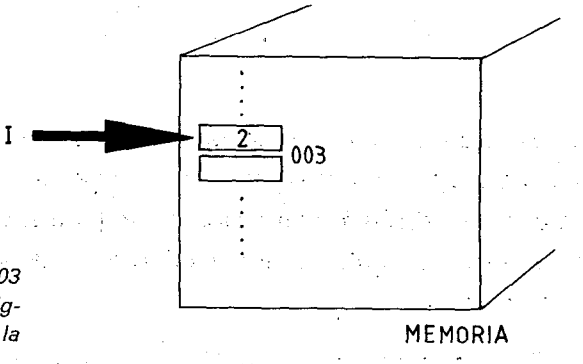


Fig. 1.15. La posición de memoria 003 contiene el dato 2. La variable que designa a esa posición de memorias es I. *En la figura,* I = 2.

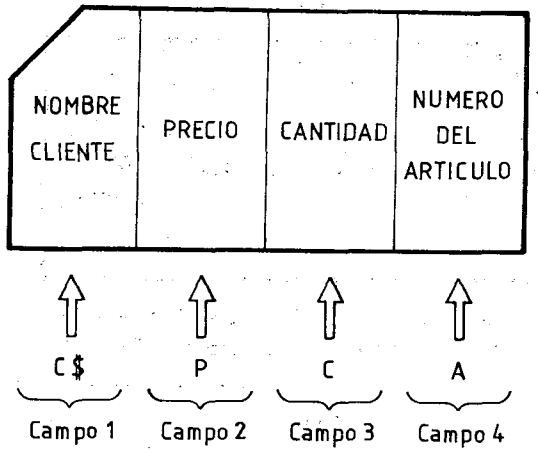


Fig. 1.16. Un campo, en una tarjeta perforada, sería el número de columnas sucesivas destinadas a representar una información. A cada uno de los campos se le da un nombre de variable diferente.

La variable, I, representa a una posición de memoria en la que va variando su contenido.

Un ejemplo de campo en un soporte es el siguiente: en una tarjeta perforada se escribe el nombre de un cliente, el código del artículo que pide, la cantidad y el precio unitario.

Cada una de estas informaciones constituye un campo diferente. Estos campos tendrán nombres de variables diferentes (Fig. 1.16).

Registro

Un *registro* es un conjunto de campos de datos que se trata como una unidad.

Este agrupamiento de campos se clasifica en:

— Registros lógicos.
— Registros físicos.

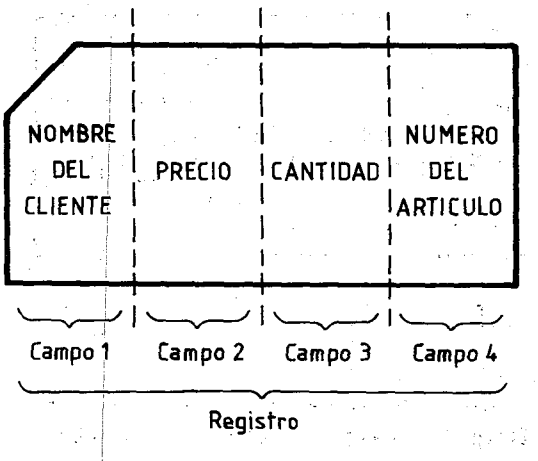


Fig. 1.17. El registro está formado por 4 campos C1, C2, C3, C4. Todos ellos hacen referencia a una misma información: un pedido de un cliente.

Registro lógico

Se denomina *registro lógico* a un conjunto de campos de datos relacionados y tratados por la lógica del programa como una unidad (es decir, como un *bloque*).

Los datos de un registro lógico están contenidos en un soporte de información de entrada o salida para ser procesados.

La longitud de un registro lógico puede ser desde uno a cualquier número de caracteres.

Un registro lógico es el que trata el programador como unidad; está definido por la dirección de la posición inicial y por su longitud.

Por ejemplo, un registro lógico puede estar contenido en una sola tarjeta perforada, o puede estar contenido en varias tarjetas. Es decir, puede ocurrir que un registro lógico esté formado por varias tarjetas perforadas o que una tarjeta perforada contenga varios registros lógicos.

Así, por ejemplo, en una ficha de 80 caracteres pueden representarse uno, dos, tres o cuatro registros. Toda la tarjeta que hace referencia a un pedido de un cliente, sería un registro lógico, estaría compuesto por cuatro campos cuya característica común es el hacer referencia a un mismo cliente (Fig. 1.17). Otra tarjeta con los datos de otro cliente constituiría otro registro lógico, poseyendo los mismos campos que en la anterior tarjeta pero, naturalmente con otra información debida a este nuevo cliente.

Los registros lógicos se agrupan en forma de *bloques,* ahorrándose espacio de memoria y tiempo de transmisión.

Registro físico

Se llama *registro físico* a un conjunto de caracteres contiguos almacenados en un soporte de información.

Un registro físico puede contener uno o varios registros lógicos. Cuando en un registro físico hay varios registros lógicos, éstos se agrupan en bloques para reducir el espacio de memoria ocupado por los registros físicos y para reducir el tiempo en las operaciones de entrada/salida. Para cada operación de entrada/salida se transmite un registro físico completo.

Si un registro físico contiene un número determinado, N, de registros lógicos, se dice que los registros lógicos se encuentran agrupados en N bloques.

Un registro físico está definido por su comienzo y final físicos, y es tratado, por los dispositivos empleados como unidad.

Por ejemplo, en un lote de tarjetas perforadas, cada una de las tarjetas representa un registro físico, ya que está separada físicamente de las restantes tarjetas del lote.

La longitud de un registro físico depende del dispositivo utilizado. Así, una tarjeta perforada es un registro físico de 80 caracteres (longitud fija); una cinta magnética y una cinta perforada disponen de registros de longitud variable; los discos y tambores magnéticos poseen pistas de longitud fija, de manera que debe intentarse que cada registro ocupe, aproximadamente, una pista.

Se define un registro físico como el lugar de almacenamiento temporal de una información que está tratando el ordenador. Se dice temporal para significar que el ordenador usa de estas células de almacenamiento para la

información mientras está en fase de tratamiento. Una vez se ha obtenido un resultado, éste se lleva a la memoria, ya sea el resultado parcial o el total, pero siempre que no vaya a usarse inmediatamente, en cuyo caso se quedaría en el registro.

Registros físicos se encuentran en cualquier módulo componente del ordenador. En la unidad de control porque tiene que almacenar las instrucciones para su análisis, en la unidad aritmética porque tiene que manejar los datos operando y guardar resultados intermedios. En las unidades de E/S porque los datos se almacenan hasta que el periférico esté listo.

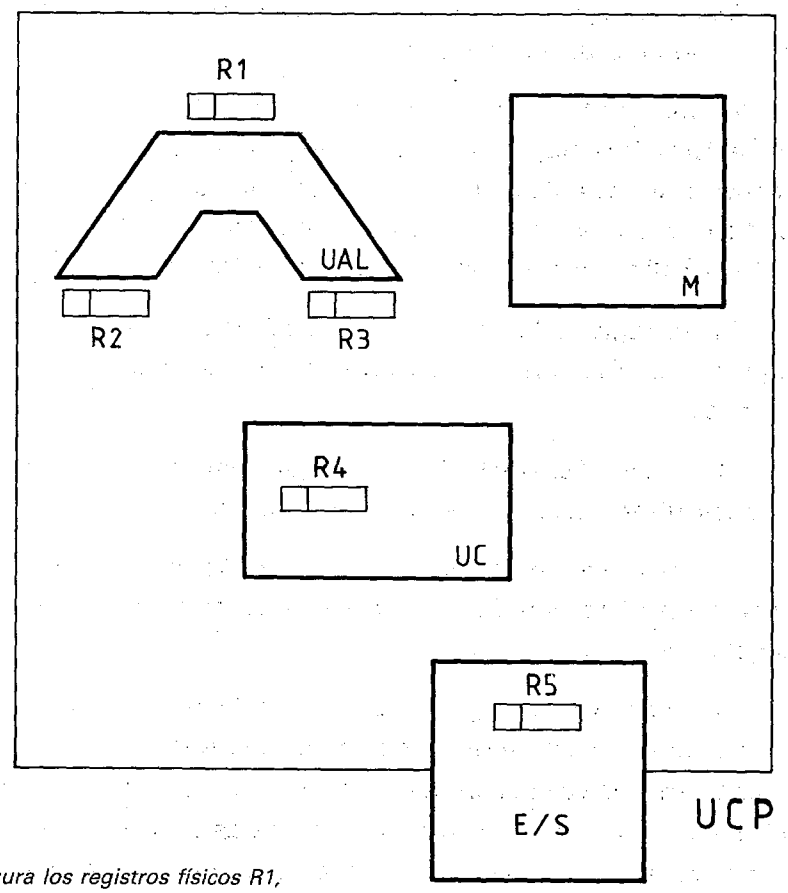


Fig. 1.18. En la figura los registros físicos R1, R2, R3 están en la UAL, el R4 en la UC y el R5 en las unidades de E/S.

Fichero

Un *fichero* está constituido por un conjunto de registros lógicos, con características comunes.

La cantidad de información que se almacena en un fichero suele ser tan grande que precisa un soporte para su almacenamiento, no pudiendo usarse la memoria central para esta operación.

El fichero es un conjunto de fichas y el ordenador tiene un fichero compuesto por un conjunto de registros. La información guardada en una ficha es análoga a la guardada en un registro. Dicha información también puede ser separada en campos, que en una ficha llamaríamos zonas o conceptos.

Para aclaración se emplea igualmente el ejemplo que se viene considerando (Fig. 1.19.). Cuando se reciben los pedidos de todos los clientes se abren registros para cada uno de ellos. Ese conjunto de registros constituye el fichero.

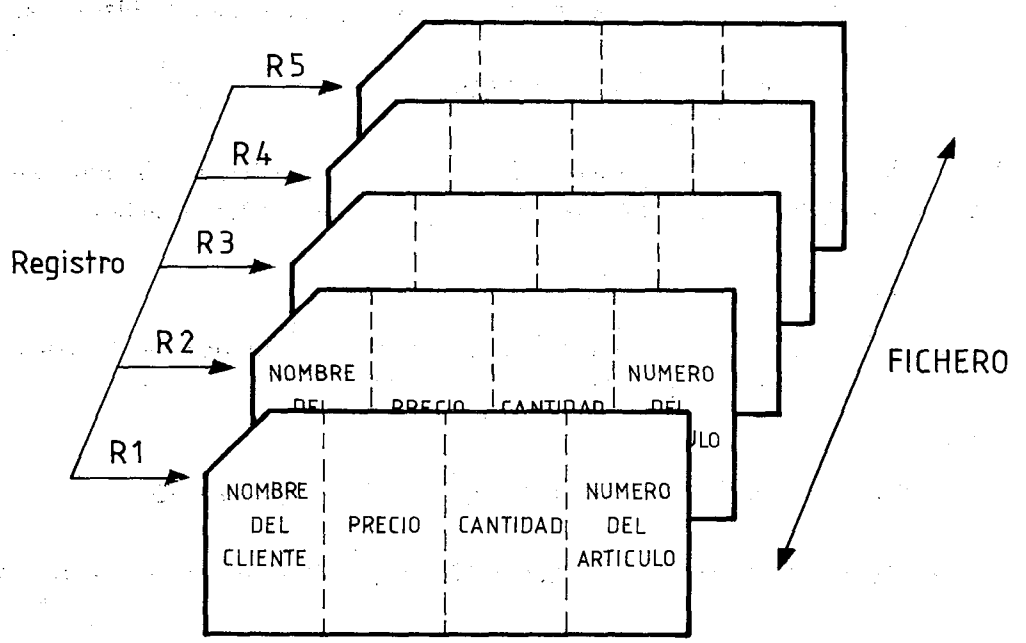


Fig. 1.19. Ejemplo de un fichero o un conjunto de registros con características comunes.

1.2.2. Representación de los datos en la MC

Una vez estudiados los conceptos básicos de la información debe conocerse cómo se almacenan en el ordenador los datos o informaciones.

Los datos que se almacenan en la MC del ordenador pueden ser de dos tipos: numéricos y alfanuméricos.

Datos numéricos

Los datos numéricos están formados por caracteres numéricos. La representación de una cantidad, sea cual sea su valor, será un dato numérico.

Cualquier cantidad o número puede ser considerado bajo dos aspectos:

Primero.

Como resultado de una fracción positiva o negativa, que como tal resultado estará formado por: un signo, una parte entera y una parte decimal. Que tanto la parte entera, como la decimal, o ambas a la vez, pueden ser 0. Ejemplos:

$$+ 75{,}63; \quad - 0{,}040; \quad + 18{,}00; \quad - 0{,}0004; \quad - 0{,}000$$

En la práctica se omite la representación de los signos positivos. Asimismo, en Proceso de Datos, se sustituye la coma por el punto (0.75; − 64.36)

Segundo.

Como resultado del producto de un número por una potencia de 10 (o de 2 en el sistema binario). Ejemplos:

$$+ 75{,}63 = (+ 7563) \times 10^{-2} = (+ 7{,}563) \times 10^{1} = (+ 0{,}7563) \times 10^{2}$$

$$- 0{,}0004 = (- 4) \times 10^{-4} = (- 0{,}4) \times 10^{-3} = (- 40) \times 10^{-5} = (- 0{,}0004) \times 10^{0}$$

En estos ejemplos puede verse que cada uno de los productos tiene dos factores. El primer factor es el número en cuestión, con o sin coma decimal. El segundo factor es siempre la constante 10 elevada a un exponente variable.

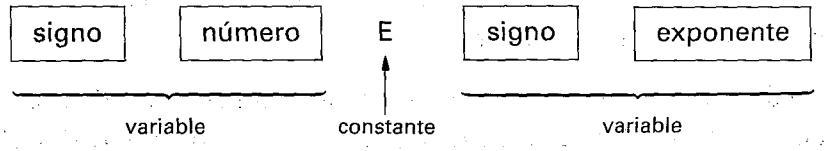
En matemáticas es normal representar un valor numérico constante

mediante una letra. En este caso concreto se puede representar la constante 10 por la letra E. Los ejemplos anteriores quedarían representados así:

$$+\ 75{,}63 = (+\ 7563)\ E^{-2} = (+\ 7{,}563)\ E^{1} = (+\ 0{,}7563)\ E^{2}$$

$$-\ 0{,}0004 = (-\ 4)\ E^{-4} \text{ ... etc.}$$

Esquemáticamente podría representarse uno de estos productos así:



Según este esquema, los ejemplos citados se representarían:

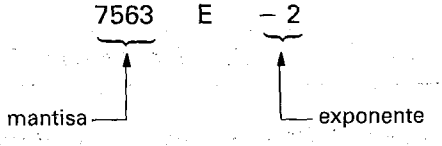
+ 75.63 = 7563E−2= 756.3E−1 = 0.7563E2

− 0.0004 = − 4E−4 = − 0.4E−3 = − 40E−5 = − 0.000004E2

En esta transcripción se han omitido los signos positivos y se ha sustituido la coma por el punto.

Esta manera de representar una cantidad se conoce con el nombre de *notación científica.*

En la notación científica, a la parte que queda a la izquierda de la E se le denomina *mantisa* y a lo que queda a la derecha, *exponente.*



Mediante el empleo de la notación científica puede conseguirse una mantisa que sea siempre entera o siempre decimal. El exponente será quien determine la situación del punto o coma decimal.

Otra facultad de la notación científica es la de que permite representar macro o microcantidades con un número de cifras (o dígitos) mucho más reducido que la notación matemática convencional.

Los datos numéricos pueden representarse en la MC bajo dos tipos de formatos:

— En coma fija.
— En coma flotante.

Coma fija

Para la representación de un número en *coma fija* (denominada también *punto fijo*), debe preestablecerse previamente cuántas serán las cifras decimales con las que se desea operar.

El hecho de escoger un mayor o menor número de cifras decimales dependerá de la exactitud con la que se desee operar, o de la aproximación que se necesite. En el campo comercial, por ejemplo, suele operarse con dos cifras decimales.

Una vez determinada la extensión del campo destinado a las cifras decimales, el ordenador considerará, solamente, las cifras (o dígitos) que quepan en dicho campo. Si el dato numérico introducido tiene menos cifras decimales que las establecidas, el ordenador le añadirá ceros hasta completar el campo. Si el dato es un número entero, el ordenador ocupará todo el campo decimal con ceros. Y finalmente, si el número introducido tiene más cifras decimales que la extensión del campo, el ordenador considerará solamente todas las primeras cifras decimales que pueda admitir, y despreciará las restantes.

En la representación por coma fija, el ordenador ocupa siempre la totalidad del campo establecido, aunque para un dato concreto no le hubiera hecho falta el empleo de tanta extensión.

	2 Posiciones decimales		*4 Posiciones decimales*	
Número	*Representación*	*Significado*	*Representación*	*Significado*
176.34	17634	176.34	1763400	176.3400
81.9	08190	081.90	0819000	081.9000
12.	01200	012.00	0120000	012.0000
126.078549	12607	126.07	1260785	126.0785
0.0003	00000	000.00	0000003	000.0003

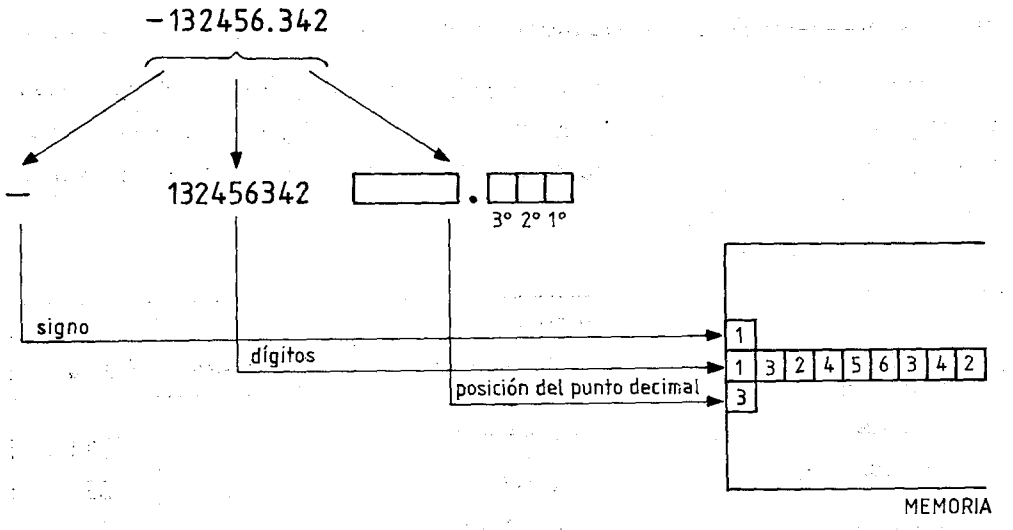


Fig. 1.20. Representación en la memoria de un número en coma fija.

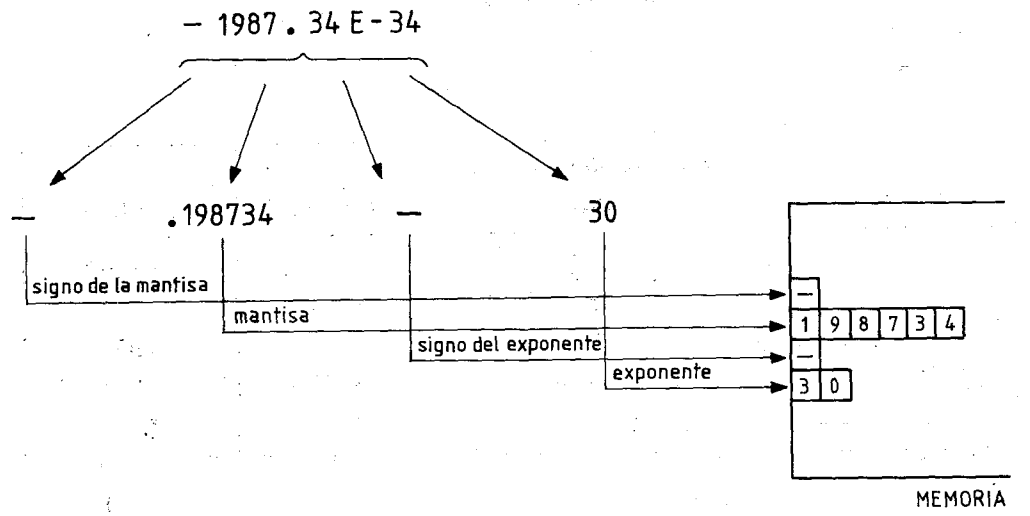


Fig. 1.21. Representación en la memoria de un número en coma flotante.

Coma flotante

Para la representación de datos en *coma flotante* se recurre a la notación científica. Se ha visto que en notación científica puede conseguirse una *mantisa* que sea totalmente entera o totalmente decimal haciendo variar el exponente.

Esto permite la supresión de todo cero innecesario, incluso de la coma o punto decimal. En cierto modo puede decirse que el punto decimal «flota» hasta el exponente, siendo éste el que da el valor exacto a la mantisa.

Número	*Representación en coma flotante*	*Exponente*	*Mantisa*
674.52	0.67452E3	3	67452
0.0000032	0.32E−5	− 5	32
− 0.0000798	− 0.798E−4	− 4	− 798
25000	0.25E5	5	25
0.5	0.5E0	0	5

La coma decimal (o binaria) no ocupa lugar en el ordenador. El exponente suele ocupar dos posiciones decimales. Así por ejemplo los números anteriores pueden escribirse de la siguiente forma:

Número	*Representación flotante*	*Exponente*	*Mantisa*
674.52	.67452 03	3	67452
0.0000032	.32 −05	5	32
− 0.0000798	.798 −04	4	− 798
25000	.25 05	5	25
0.5	.5 00	0	5

Cuando debe tratarse gran cantidad de dígitos decimales suele recurrirse al formato de coma flotante.

Como caso particular de los números en coma flotante están los números en precisión doble. En este caso se emplea el doble de palabras para almacenar la mantisa. Como resultado de esta ampliación el número de cifras de la mantisa que puede almacenarse en la máquina es más grande y la precisión es mayor.

En otros sistemas, la especificación de las variables lleva implícita el número de palabras que han de usarse para la representación.

Por ejemplo, Real * 4 significa en un lenguaje que usa cuatro veces más palabras de las que usa normalmente para almacenar los números reales.

Entero * 2 usa el doble de palabras que las usuales en el sistema para almacenar los números enteros. Cualquier otro número podría almacenarse según estas reglas.

Los números con exponentes grandes en valor absoluto (sin considerar el signo), son los que tienen menor precisión, pues el número de ceros que habría que añadir para pasar a la representación matemática convencional es muy elevado.

Por ejemplo el número: 1E 34 tiene una representación numérica con un 1 seguido de 34 ceros, eso significa que sería necesario el conocimiento de las 34 cifras para que la precisión fuera exacta. El ordenador no va a ser capaz de almacenar esas 34 cifras de la mantisa.

Por el contrario el número 1E 2 tiene la representación de un uno seguido de dos ceros, (100) por lo que son únicamente necesarias dos cifras para dar todo su margen de variación con la exactitud correcta.

Datos alfanuméricos

Se consideran datos alfanuméricos los que se componen de caracteres alfabéticos. Los dígitos del 0 al 9 considerados como caracteres, no como número, y los caracteres especiales, que servirán para construir cualquier expresión: paréntesis, signos de puntuación, &, etc.

Se puede demostrar que el número de bits necesarios para codificar todos los caracteres es 6, ya que $2^6 = 64$. Es decir con este código pueden representarse 64 caracteres diferentes, que son suficientes.

Conociendo este código una palabra basta para almacenar uno de estos caracteres. Una forma de representación es la que utiliza más de una palabra y

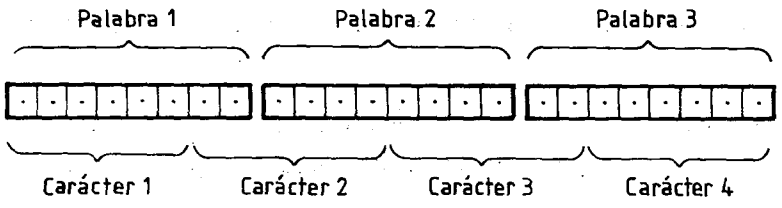


Fig. 1.22. Con tres palabras se consigue un conjunto de 24 bits en el que caben 4 caracteres alfanuméricos.

considera este número de palabras globalmente para almacenar un conjunto de caracteres alfanuméricos.

Sea un ordenador que trabaje con palabras de 32 bits. Una palabra de este ordenador puede dar cabida a 5 caracteres alfanuméricos, (32/6 = 5) cuando se ha usado la codificación de 6 bits.

A un ordenador que posea palabras de 8 bits le sería conveniente juntar tres palabras para guardar caracteres alfanuméricos. Con esas tres palabras consigue un conjunto de almacenamiento de 24 bits ($8 \times 3 = 24$) en el que caben 4 ($24/6 = 4$) caracteres alfanuméricos (siempre considerando una codificación de 6 bits). Este mismo ordenador si hubiera usado una palabra para almacenar un caracter alfanumérico habría desaprovechado dos bits de ella ($8 - 6 = 2$).

1.2.3. Acceso a la MC

El proceso de *lectura* de una memoria significa conocer el contenido de una posición particular de la memoria.

Para la ejecución de esta tarea, el ordenador precisa conocer cuál es la dirección de la posición de memoria de la que se quiere obtener el contenido. Esta dirección le es proporcionada por el programa. Todos estos pasos no tienen que tenerse en cuenta especialmente cuando se está programando en

un lenguaje de alto nivel. En este lenguaje con dar el nombre de la variable por la que se llama a esa posición de memoria ya es suficiente, el programa compilador ya se ha encargado de desmenuzar esa operación en varias instrucciones más elementales y averiguar el número de la posición de memoria a la que se quiere acceder.

Esta operación de lectura no borra el contenido de la posición de memoria. Después de la lectura el contenido de la posición de memoria sigue allí.

El proceso de *escritura* consiste en colocar en una cierta posición de memoria un dato, que se constituirá en contenido de esa posición de memoria.

El ordenador, para poder realizar esta operación, debe conocer:

— el valor del dato que se va a colocar en la memoria.
— la dirección de la posición de memoria donde se va a colocar.

Al igual que en la operación de lectura, en un programa de alto nivel únicamente es necesario decir el nombre de la variable que representa a esa posición de memoria para que se realice la operación.

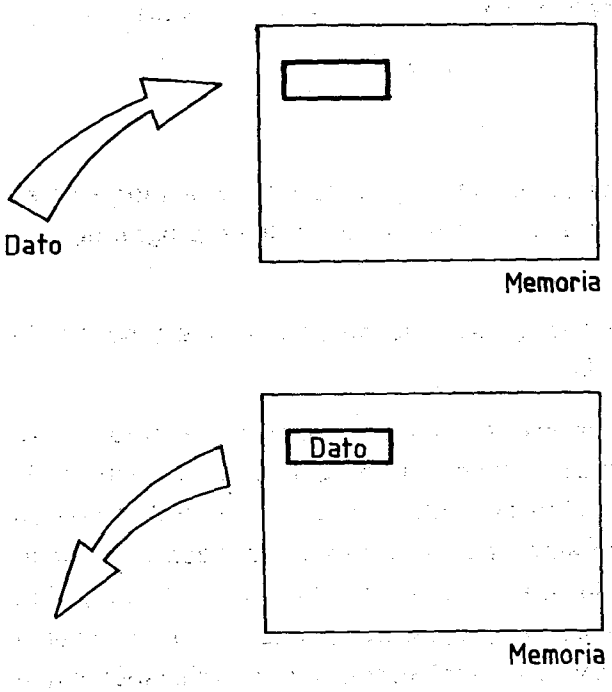


Fig. 1.23. En la figura podemos apreciar las dos operaciones que se efectúan en la memoria: lectura y escritura.

Inversamente a lo que ocurre con la lectura, la operación de escritura es un proceso destructivo, es decir, que al escribir en una posición de memoria se borra el contenido de dicha posición. Por lo tanto, al efectuar un programa el programador debe asegurarse, siempre que se vaya a escribir, que el contenido actual (antes de la escritura) de esa posición no es necesario que se conserve.

Pueden considerarse tres tipos de memorias según las posibilidades de acceso a ellas.

— Memoria de acceso aleatorio.
— Memoria de sólo lectura.
— Memoria asociativa.

Memoria de acceso aleatorio

La característica principal de la *memoria de acceso aleatorio,* RAM (del inglés Random Acces Memory), consiste en que puede realizarse la lectura y la escritura de cualquier contenido en cualquier posición. Otra de las características es su rápido acceso (entendiendo por acceso el tiempo que tarda en colocarse un dato en la posición de memoria o de extraerse de la misma).

Este tipo de memoria se usa para guardar los datos que van a utilizarse a menudo y rápidamente, y para almacenar resultados intermedios.

Memoria de solo lectura

En la *memoria de sólo lectura,* ROM (del inglés, Read Only Memory), no se tiene posibilidad de escribir en ella durante la ejecución de un programa, sólo puede leerse su contenido.

El fabricante de estas memorias, graba de forma permanente el contenido de dicha memoria durante su fabricación.

La utilización de esta memoria es pues para la información que sólo precise ser leída, por ejemplo para almacenamiento de programas permanentes. Es importante resaltar que las máquinas que sólo tuvieran este tipo de memoria sólo podrían efectuar los programas que estuvieran grabados allí, serían máquinas de programa único , sin poder modificarse. Por lo tanto, la utilización de estas memorias es la de almacenar programas que no han de cambiarse nunca, como el programa monitor, los compiladores, o para ordenadores que sólo ejecutan una tarea específica, por ejemplo los ordenadores encargados del control de máquinas-herramientas.

Existen diversos tipos de memorias de sólo lectura. Cada uno de ellos se acerca más en sus características a las de acceso aleatorio proporcionando al usuario una mayor facilidad para la escritura.

En primer lugar está la PROM, o memoria ROM programable. El usuario puede programar el contenido de la memoria con una máquina especial, llamada *programador de memorias PROM.* Durante la ejecución del programa el ordenador no puede escribir en ella, pero si las características del contenido tuvieran que cambiarse se puede hacer con cierta facilidad volviendo a programar la memoria. Los otros tipos de memoria permiten una programación más cómoda como las EAROM, programables eléctricamente.

Memoria asociativa

La particularidad de esta memoria es que es capaz de direccionarse por el contenido. Esto es, que la posición donde se va a almacenar un dato se localiza según el valor de ese dato.

Lo que se consigue con esto es que sea el propio ordenador quién se encargue de elegir la posición de memoria en la que se va a almacenar, efectuando un cálculo con el dato que va a almacenarse.

1.2.4. Constitución de la memoria central

La MC está compuesta por ceros y unos, de forma que toda la información necesaria para la ejecución de un problema se encuentra almacenada en la MC en forma de bits. Estos bits están agrupados formando *posiciones de memoria.* Cada posición de memoria contiene un carácter (*byte*), es decir, ocho bits (*octeto*) o seis bits (*sexteto*). También hay ordenadores capaces de almacenar varios caracteres en una posición de memoria formando lo que se llama una *palabra.* De ahí que la capacidad de algunas memorias se mida por caracteres y la de otras por palabras.

Así pues, aunque la unidad más pequeña de información que puede almacenarse en la MC es el bit, la unidad más pequeña que puede extraerse o introducirse desde o hacia la memoria es el *carácter* en unas máquinas y la *palabra* en otras.

A las máquinas que utilizan la palabra como unidad básica se les llama *máquinas palabra,* y a las máquinas que utilizan el carácter se les llama *máquinas carácter.*

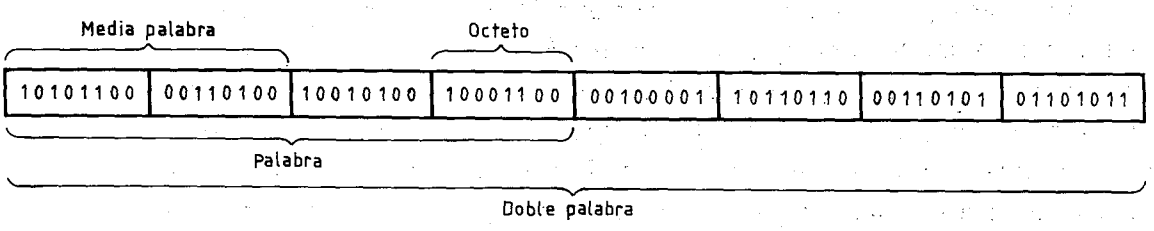


Fig. 1.24. Unidades básicas de almacenamiento.

En la mayoría de los casos, las palabras son de longitud fija. El número de bits de una palabra depende del tipo de ordenador, sin embargo, las más usuales son las de 32 bits (4 octetos) ó 36 bits (6 sextetos). Además se utilizan longitudes de palabras de 12, 16, 24 ó 48 bits (múltiplos de 6 u ocho).

También pueden tratarse como unidad la *media palabra* (2 octetos) y la *doble palabra* (8 octetos).

Otro concepto que se utiliza para el almacenamiento de datos es el de *campo de memoria* que está formado por un conjunto de posiciones de memoria. Un campo de memoria puede almacenar variables o constantes.

Para una determinada variable se pueden reservar varias posiciones de memoria. Este tipo de variables se denominan *tablas.* Estas variables se pueden dividir, mediante un *subíndice,* en varias posiciones de memoria de manera que un mismo nombre de una variable puede referirse a varios valores numéricos.

Por ejemplo, la variable X puede representar a las posiciones de memoria: X_1, X_2, X_3, ..., de forma que cada X_i, siendo *i* el subíndice, representa un sólo valor numérico, es decir, la posición dentro de la tabla X.

Así por ejemplo, se puede construir la tabla clasificada por categorías de las ciudades de un país según su número aproximado de habitantes. La variable X representa el nombre de la tabla, siendo X_i la categoría de una cierta ciudad.

Habitantes	*Categoría*
25.000	X_1
30.000	X_2
35.000	X_3
50.000	X_4
100.000	X_5
150.000	X_6

Supongamos que se desea almacenar esta tabla en la memoria, entonces se necesitarán seis posiciones de memoria para cada valor del subíndice (del 1 al 6), de modo que cada valor (25.000, 30.000,...) se almacenan en una posición distinta.

Por ejemplo, X_5 se refiere a la posición 5 de la tabla X, y contiene el valor 100.000.

El compilador se encarga de asignar las posiciones de memoria a las variables.

Para extraer los datos de la MC es necesario saber en qué posiciones de memoria se encuentran; para ello, cada posición de memoria tiene asignado un número que la identifica, llamado *dirección de memoria.*

Una posción de memoria puede contener un sólo carácter en las máquinas de *carácter direccionable,* y una palabra en las máquinas de *palabra direccionable.*

En las máquinas de palabra direccionable la dirección de la palabra es la que corresponde al primer octeto de más a la izquierda de la palabra.

El tiempo que transcurre desde que se establece una dirección de la memoria, mediante una orden de lectura, hasta que ésta se realiza, es decir, hasta que se obtiene la información deseada de la MC, se denomina *tiempo de acceso.*

El tiempo de acceso más el tiempo necesario en escribir la información leída se llama *ciclo de memoria.*

La velocidad del ciclo de memoria se mide, generalmente, en microsegundos (millonésimas de segundo) por carácter. Es decir, se puede tratar (leer o escribir) un millón de caracteres en un segundo si el tiempo de acceso es 1 microsegundo/carácter (1 μ seg/c).

Los ordenadores actuales disponen de memorias con un ciclo de memoria de 10 microsegundos aproximadamente.

En cuanto a la constitución física de la memoria central pueden distinguirse tres tipos de memoria, principalmente:

— Memoria de núcleos magnéticos
— Memoria de agujas magnéticas
— Memoria de hilos magnéticos

Memoria de núcleos magnéticos

Un núcleo de ferrita se obtiene a partir de una mezcla de óxido férrico, principalmente, y de otros compuestos. Además de su reducido tamaño (menos de un milímetro de diámetro), se caracteriza por la rapidez en magnetizarse (en millonésimas de segundo).

Para seleccionar un carácter o una palabra de la memoria, dos conductores perpendiculares atraviesan cada núcleo. La intensidad de la corriente de cada conductor corresponde a la mitad (I/2) de la corriente necesaria (I), para magnetizar un núcleo. Por lo tanto, sólo se magnetiza el núcleo que está en la intersección de los dos conductores.

El paso de la corriente por un conductor hace que el estado magnético del núcleo sea positivo o negativo (0 ó 1) según el sentido de la corriente. Este paso de corriente no es constante, sino que aparece en forma de impulsos eléctricos.

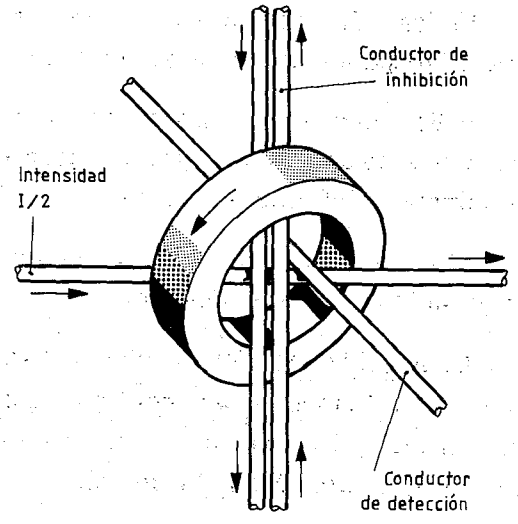


Fig. 1.25. Detalle de un núcleo magnético, con los conductores que lo atraviesan. El tamaño real de un núcleo es inferior a 1 mm de diámetro.

Cuando el estado magnético del núcleo se invierte por efecto de un impulso se induce una corriente en un tercer conductor (conductor de detección). La señal que pasa por este conductor puede detectarse para saber si el núcleo contenía un bit 1.

Al efectuar una operación de lectura de un bit 1 se devuelve al núcleo al estado 0. Los núcleos que contenían ceros permanecen en el estado 0.

Si se desean conservar los datos en la memoria después de un proceso de lectura, el ordenador debe reproducir los bits 1 al estado inicial; para ello, el ordenador debe poner unos en todas las posiciones leídas anteriormente y un impulso de inhibición suprime la escritura en los núcleos que contenían 0.

El impulso de inhibición se manda a través de un cuarto conductor para anular (inhibir) el efecto del impulso de escritura que pasa a través de uno de los dos conductores utilizados para la magnetización del núcleo.

Actualmente, existen otros tipos de memorias de núcleos que sólo emplean dos conductores para cada núcleo. De esta forma todos estos procesos de lectura y escritura se realizan más rápidamente y con mayor eficacia.

Aunque los elementos que más se utilizan como posiciones de memoria son los núcleos magnéticos, existen otros dispositivos como son los *hilos magnéticos*.

Memoria de agujas magnéticas

Una aguja magnética está constituida por una aleación de cobre y berilio, revestida por una capa delgada de material magnético (ferroníquel) de una milésima de milímetro de espesor.

La aguja se encuentra rodeada por dos hilos eléctricos aislados. Para escribir o registrar información se utilizan ambos conductores y para realizar una operación de lectura se emplea uno de estos dos conductores.

Las agujas magnéticas se agrupan en cuadrados o matrices. Cada línea y cada columna está formada por uno de los dos componentes de escritura que se arrollan en la aguja. En el cruce o intersección de un conductor de línea con otro de columna se encuentra una aguja magnética.

Durante una operación de escritura sólo la aguja situada en el cruce de los conductores de línea y de columna seleccionados recibe suficiente imantación para tomar el estado deseado.

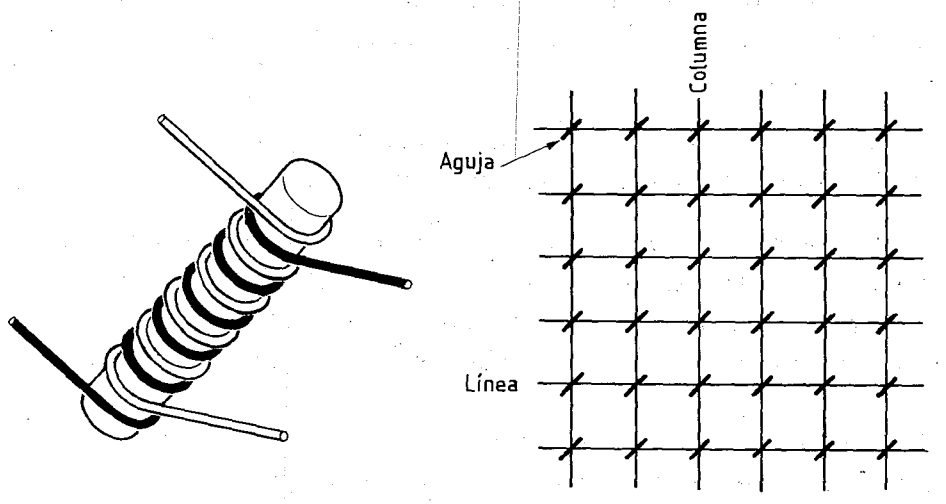


Fig. 1.26. Representación esquemática de una memoria de agujas magnéticas.

En la operación de lectura se escribe un 0 y se detectan los cambios del estado magnético 0 debe invertirse la señal eléctrica utilizada para escribir el 1.

En este tipo de memorias, un ciclo de memoria se reduce a medio microsegundo.

Memoria de hilos magnéticos

La memoria de hilos magnéticos es de funcionamiento similar a la de núcleos magnéticos. Esta memoria consta de varios alambres conductores de bits revestidos con una capa magnética en donde se graba la información. Estos conductores de bits se entrecruzan con conductores de palabras constituyendo en cada punto de cruce un bit que, según el sentido de la corriente que se les aplique, será 0 ó 1.

Los bits que componen una palabra están creados artificialmente en las intersecciones de los conductores de bit con un conductor de palabra.

El elemento magnético es un anillo; uno de los hilos, el conductor de bit, atraviesa el anillo; el otro hilo, el conductor de palabra, se arrolla alrededor del anillo. La señal leída se recoge sobre el conductor de bit.

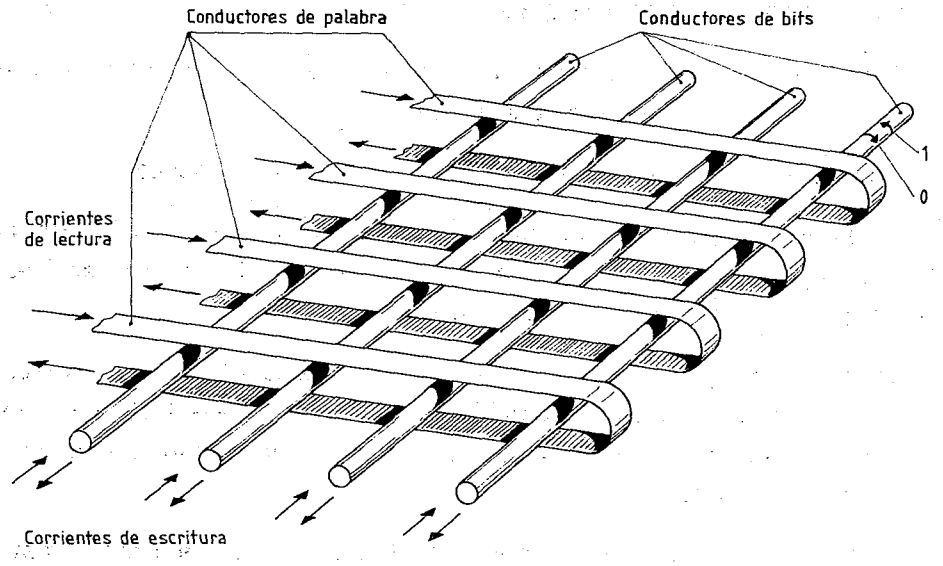


Fig. 1.27. Esquema de la disposición de una memoria de hilos magnéticos.

El conductor de palabra tiende a imantar el conductor de bit. La señal que recorre el conductor de bit produce el basculamiento, según su sentido, hacia la derecha o hacia la izquierda para escribir 1 ó 0.

En este tipo de memorias, la lectura no es destructiva. La señal que recorre el conductor de palabra produce el mismo efecto que la del conductor de bit. Cuando desaparece la señal de imantación vuelve a su sentido de origen, generando en el conductor de bit una señal distinta según que el sentido de la corriente de origen fuera 0 ó 1.

Los hilos magnéticos contienen una cantidad de material magnético muy inferior al de la aguja magnética. Un hilo magnético está constituido por cobre-helio y tiene 0,13 milímetros de diámetro, revestido de una película magnética de un espesor del orden de una diezmilésima de milímetro.

1.2.5. Memorias de gran capacidad

Las memorias de gran capacidad se utilizan para ampliar la memoria central (memoria de trabajo).

Las memorias de gran capacidad, o auxiliares, se dividen en dos grupos:

— Memorias de acceso aleatorio.
— Memorias de acceso secuencial.

Son de *acceso aleatorio (random access)* las memorias de tambor magnético y las memorias de discos magnéticos.

Son de *acceso secuencial* las memorias de cintas magnéticas.

En estos dispositivos (tambores, dicos y cintas magnéticas) la información se almacena sobre una superficie magnética móvil mediante unas técnicas de grabación muy similares en todos los casos.

En la cinta magnética, por ejemplo, la lectura de la información almacenada se realiza pasando la cinta magnética por un cabezal de lectural. Los puntos magnéticos producidos en la cinta por el cabezal de grabación son detectados por el cabezal de lectura.

El tiempo de acceso es más corto en los dispositivos de acceso aleatorio (tambor y disco) que en los de acceso secuencial (cinta). En la cinta magnética es necesario recorrer, por término medio, la mitad de la cinta hasta encontrar la información deseada. Mientras que en los tambores y discos magnéticos el tiempo de acceso es muy inferior al de la cinta magnética puesto que se accede directamente a la información requerida.

El sistema de grabación de la información en estos dispositivos magnéticos es el mismo en tambores, discos o cintas magnéticas. La superficie de grabación está formada por una capa de materal magnético. En el cabezal de lectura-escritura se dispone de un entrehierro que está en contacto con la superficie de la memoria. Cuando pasa esta superficie por el entrehierro se produce una variación de flujo que genera una corriente en las bobinas del cabezal. La parte magnética situada frente al cabezal queda magnetizado.

En una memoria de tambor magnético, el registro de dirección indica qué cabezal de lectura-escritura debe utilizarse y el sector en dónde se va a efectuar la operación de lectura o escritura.

1.2.6. Memorias flip-flop

Otro elemento de memoria básica que se utiliza para problemas especiales es el *flip-flop*, el cual es un elemento de memoria que puede adoptar dos estados eléctricos (0, 1) y puede pasar de un estado a otro mediante un impulso a la

entrada del circuito. Cada flip-flop puede tener una o varias entradas por las que se transfiere la información binaria al flip-flop.

Un elemento flip-flop es un circuito que posee dos estados estables y dos salidas complementarias. De forma que cuando una salida es 1 la otra es 0 y viceversa.

Este circuito posee dos entradas llamadas *puesta a uno* (*set*) y *puesta a cero* (*reset*). Este tipo de circuito denominado *flip-flop R-S* (*reset-set*) está representado en la figura 1.28.

Si se recibe un impulso de puesta a 1, se establece en una salida, B por ejemplo, un nivel 1. En ese instante, en la salida A (por ser complementaria a B) se establece un 0. En este caso se dice que el circuito está en 1 (o escrito).

Si, por el contrario (Fig. 1.29), el circuito recibe un impulso de puesta a cero, en la salida A se establece un 1 y en la B un 0. En cuyo caso se dice que el circuito está en 0 (o borrado).

Un circuito que esté escrito (en 1) y recibe un impulso de puesta a 1 permanece en estado 1; y si recibe un impulso de puesta a 0 y está en situación de borrado (en 0), permanece en 0.

En el caso de que se apliquen simultáneamente un impulso de puesta a 1 y otro de puesta a 0, la salida adoptará un nivel indeterminado.

La tabla de la verdad de este circuito es la siguiente:

Salidas actuales		*Entradas*		*Nuevas salidas*	
A	B	S	R	A	B
0	1	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	0	1	0	0	1
0	1	1	0	0	1
0	1	1	1	—	—

Existen otros tipos de flip-flop:

El *flip-flop J-K* es muy similar al tipo R-S, con la diferencia de que cuando se aplica un impulso a las dos entradas cambian las salidas: el 0 cambia por 1 y el 1 por 0.

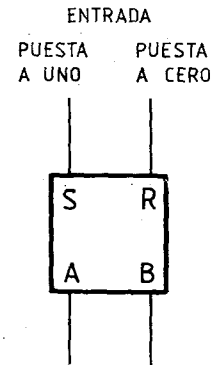


Fig. 1.28.

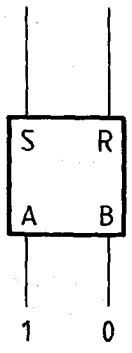


Fig. 1.29.

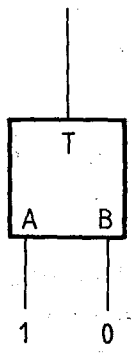


Fig. 1.30.

Otro tipo es el *flip-flop T* o *flip-flop basculante* (Fig. 1.30). Este circuito tiene una sola entrada y se produce el cambio de estado cada vez que se aplica un impulso en esta entrada.

La tabla de la verdad de este circuito es:

Salidas actuales		*Entrada*	*Nuevas salidas*	
A	B	T	A	B
1	0	0	1	0
1	0	1	0	1
0	1	1	1	0

Por último, existen otros tipos de flip-flop que contienen puertas lógicas incorporadas que deciden la aplicación del impulso a las entradas. El cambio de estado se produce cuando se aplica un impulso de reloj (impulso que se repite periódicamente en el tiempo).

El flip-flop consta de elementos electrónicos muy rápidos consiguiéndose operaciones de transferencia de información rapidísimas, por lo que las memorias formadas por flip-flop se les *llama memorias rápidas*. Se puede realizar una transferencia aproximadamente en 10 nanosegundos (1 nanosegundo es la milmillonésima parte de un segundo).

Las memorias flip-flop resultan muy caras y además tienen otras desventajas con respecto a las memorias de núcleos o hilos magnéticos. Suelen utilizarse en forma de pequeñas memorias llamadas *registros*.

1.2.7. Registros

Un *registro* es un conjunto de elementos binarios (flip-flop) en donde se puede almacenar un byte (en las máquinas carácter) o una palabra (en las máquinas palabra).

Mientras que en la MC se almacenan los datos e instrucciones para procesarse a continuación, en los registros se almacenan los datos y las instrucciones durante el desarrollo del proceso. Por ejemplo, se utilizan registros en la UAL para almacenar los operandos que intervienen en una operación de cálculo.

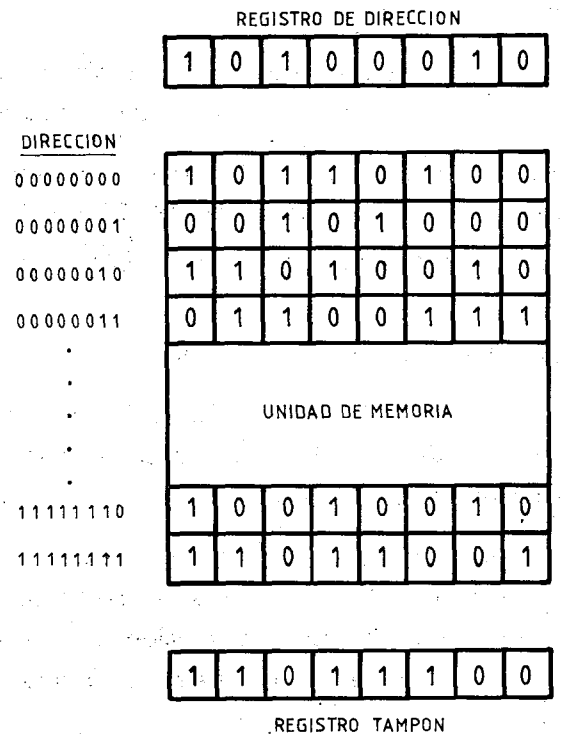


Fig. 1.31. Partes en que puede dividirse la MC: Unidad de memoria, Registro de dirección y Registro tampón.

Los registros pueden estar contenidos en la MC ocupando una o varias posiciones de memoria aunque, generalmente, un registro corresponde a una posición de memoria en donde se almacena una palabra.

La MC puede dividirse en tres partes: la unidad de memoria, el registro de direcciones y el registro tampón.

La *unidad de memoria,* propiamente dicha, está compuesta, generalmente, por núcleos magnéticos. En el ejemplo de la figura 1.31 están representados 256 caracteres (bytes) de 8 bits.

Cada una de estas posiciones de memoria tiene una dirección (del 0 al 225).

El *registro de dirección* contiene la dirección de la posición de memoria durante el acceso a esa posición, es decir, contiene la dirección del carácter (en el ejemplo) o de la palabra que se va a leer o escribir.

El registro de dirección consta de un número determinado de elementos biestables (flip-flop) en los que se escribe la dirección de la posición de memoria en donde se debe leer o escribir la información deseada.

El *registro tampón* tiene la función de aceptar el carácter o la palabra que se ha leído en la memoria y la almacena temporalmente hasta que se escribe en la memoria.

El registro también almacena la información hasta que se transfiera a la memoria o hasta que finaliza una determinada operación.

En una memoria de lectura destructiva de núcleos, dicho registro se utiliza para almacenar temporalmente la información que debe introducirse nuevamente una vez finalizado el proceso de lectura.

Circuito de mantenimiento

La acción de introducir información en la memoria es *registrar.* Un registro es un dispositivo que registra el paso de un impulso y lo «memoriza» hasta que sea necesario utilizarlo.

Cuando una combinación de impulsos ha suministrao una señal correspondiente al bit 1 o al bit 0, es necesario conservar el mismo estado (1 ó 0) después de que hayan desaparecido los impulsos necesarios. Esta función la realiza el *circuito de mantenimiento* o *circuito NOR complementado* (Fig. 1.32).

El circuito de mantenimiento está formado por dos operadores NOR (negación lógica de la función OR).

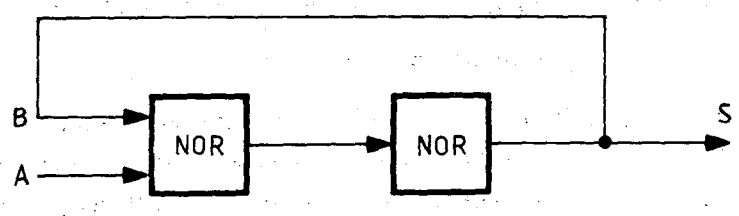


Fig. 1.32. Representación de un circuito de mantenimiento o circuito NOR complementado.

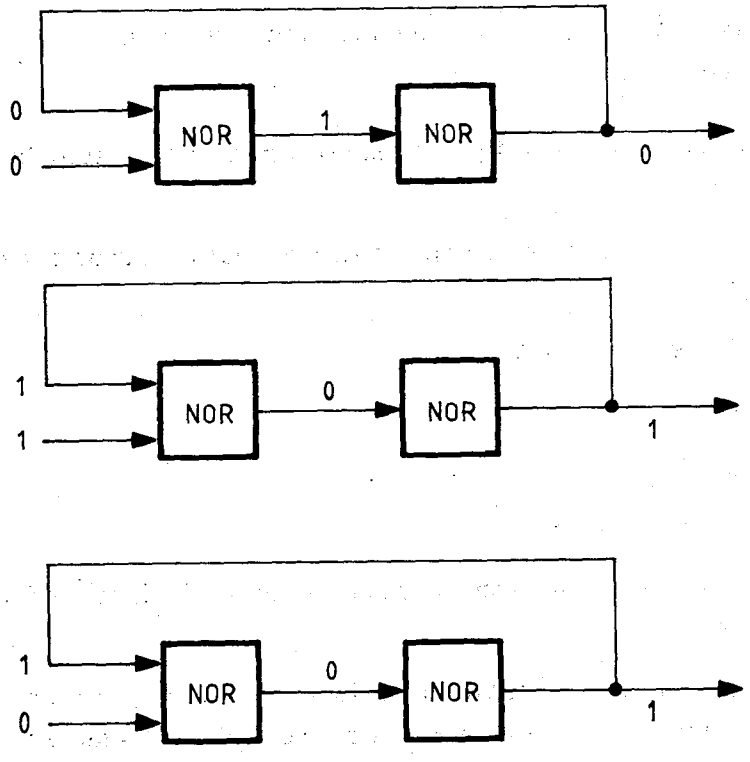


Fig. 1.33. Funcionamiento del circuito de mantenimiento según el valor de los impulsos de entrada.

La aplicación de una señal 1 a la entrada (A) del circuito NOR da a la salida el valor 0. Por lo tanto, el complemento de esta salida se lo da el valor 1 y este valor 1 de la salida del complemento tiene que ser aplicado a la entrada.

La salida coincide con la operación lógica OR.

En la figura 1.33 puede verse el funcionamiento de este circuito. En primer lugar, no existe impulso en la entrada (0) y la salida se encuentra en el estado 0. La operación OR es $0 + 0 = 0$. La salida del operador NOR es, por lo tanto, $\overline{0 + 0} = 1$.

Por último, la salida del segundo operador NOR es 0.

En segundo lugar, aparece un impulso de entrada (1) y se tiene:

$$A + B = 1 + 0 = 1$$

mediante el primer operador,

$$\overline{A + B} = 0$$

y con el segundo operador NOR, se complementa este valor,

$$S = \overline{0} = 1$$

es decir, la salida toma el valor 1 y éste valor se introduce de nuevo a la entrada B.

A continuación, se suprime el impulso en la entrada A y queda el valor 1, es decir,

$$A + B = 0 + 1 = 1$$

$$\overline{A + B} = 0$$

$$S = \overline{0} = 1$$

la salida S mantiene el valor 1.

Si después se aplicara un nuevo impulso a la entrada A, la salida S seguiría manteniendo el valor 1.

Para que este circuito pueda volver a utilizarse para otros procesos es necesario que en algún momento la salida esté a cero. Para ello se le aplica un impulso de entrada al segundo operador NOR de forma que se borre la información anterior.

Aunque se suprima este impulso al segundo operador NOR, seguirá conservándose el valor 0 para la salida S.

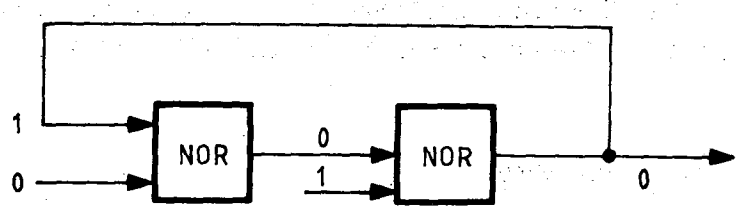


Fig. 1.34.

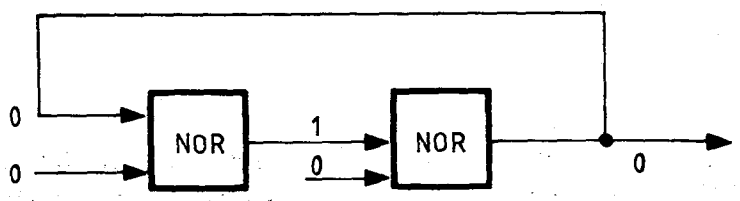


Fig. 1.35.

La figura 1.34 muestra cómo actúa el circuito al serle aplicado al segundo operador un impulso de entrada. Con este impulso se consigue el estado 0 en la salida.

La figura 1.35 muestra cómo el circuito permanece en estado 0 una vez suprimido el impulso adicional aplicado al segundo operador.

Así pues, puede definirse un registro como un conjunto de *células* de memoria (cada célula corresponde a un bit) provistas de entrada de impulsos, de salidas directas y de salidas complementadas.

Registros acumuladores

A los registros cuyo contenido depende solamente de la señal aplicada y no del contenido anterior del registro se les llama *registros independientes.*

Un registro que esté organizado de forma que su contenido sea una combinación del contenido inicial con las señales que se les apliquen se denomina *registro dependiente* o *acumulador.*

El contenido del acumulador depende de la señal que recibe y del contenido anterior.

Un acumulador sirve para contar y conservar disponible su contenido. Acumula la información recibida con la que contenía. El contenido se mantiene disponible hasta que se produzca una nueva acumulación o puesta a cero.

Registros de desplazamiento

Cuando se realiza un cálculo aritmético es necesario efectuar desplazamientos. Por ejemplo, si se registran los números 11010 y 101, la máquina los situará de la siguiente manera:

11010

101

por lo que tendrá que efectuarse un desplazamiento hacia la derecha hasta que se obtenga la posición correcta. En primer lugar se desplaza una sola posición.

11010 y por último, 11010

101 101

estos desplazamientos se pueden obtener por medio de un *registro de desplazamiento,* en el que se conectan las salidas de cada célula a las entradas de la siguiente, en el sentido del desplazamiento a realizar (derecha o izquierda).

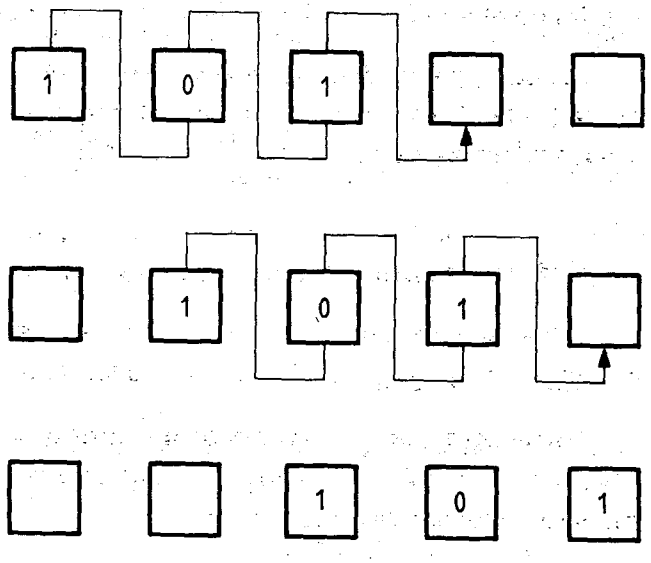


Fig. 1.36. Actuación de un registro de desplazamiento.

1.2.8. Clasificación de los soportes de almacenamiento

La gran velocidad de cálculo del ordenador permite el manejo de grandes cantidades de información. La memoria central no podría almacenar la totalidad de datos que debe manejar una gran empresa, por lo que se recurre a guardar los datos en otro lugar de almacenamiento como por ejemplo, discos o cintas magnéticas.

La MC se reserva sólo para lo indispensable, por ejemplo el almacenamiento del programa que se está ejecutando y de los datos que se van a procesar.

Los soportes físicos de almacenamiento de datos se clasifican según su uso en:

a) Por su uso posterior

Una clasificación de esos soportes o sistemas del almacenamiento se basa en el criterio de su utilización posterior para otros trabajos, según sean:

— Soportes reutilizables.
— Soportes no-reutilizables.

Los *reutilizables* permiten ser utilizados de nuevo para otro trabajo, borrando todo el contenido y volviendo a escribir los nuevos datos.

En los soportes *no-reutilizables* la situación es la contraria porque sólo se puede escribir una vez en ellos. Una vez efectuado el trabajo ya no pueden ser utilizados ninguna vez más, debiendo ser destruidos.

Un ejemplo de los primeros son las cintas magnéticas que pueden ser grabadas una y otra vez sin que se estropee la cinta.

La tarjeta perforada es un ejemplo del segundo tipo. Así cuando un trabajo ha terminado la tarjeta perforada se debe eliminar.

b) Por su capacidad

Otra clasificación tendría en cuenta el criterio de cantidad de almacenamiento y de coste. La clasificación sería:

— Memorias primarias
— Memorias secundarias
— Memorias terciarias.

Las memorias primarias son las memorias a las que la máquina tiene más rápido y fácil acceso. Son también las más caras. Su utilización es pues limitada a cálculos en el programa. La tecnología actual para la construcción de la memoria es la de los semiconductores.

Son las memorias que han sido llamadas memorias centrales (ROM, RAM) y forman parte integrante del ordenador.

Las memorias secundarias serían más lentas y baratas que las anteriores y ya estarían fuera físicamente del ordenador. Son los sistemas basados en discos magnéticos. Ultimamente se está investigando en la utilización de discos explorados por rayo laser en lugar de por una cabeza magnética. Estos discos proporcionan características de almacenamiento mejores, es decir, se podrán almacenar más bits por unidad de superficie. (Al número de bits grabados por unidad de superficie se le llama densidad de almacenamiento.) Este sistema de exploración por laser garantiza también una mayor protección frente a errores que se pudieran producir en la lectura.

Por último se consideran las memorias terciarias, que tienen aún menor costo por bit almacenado y que almacenan una mayor cantidad de datos. Un ejemplo de este tipo son las cintas magnéticas.

EJERCICIOS

1. Una puerta de autorización es:

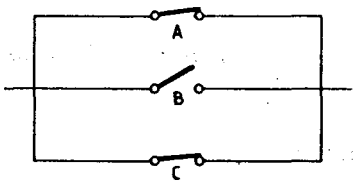
   a) Un circuito Y de una entrada
   b) Un circuito Y de dos entradas
   c) Un circuito O de dos entradas

2. Un decodificador es:

   a) Una combinación de selectores
   b) Varias puertas NO-O
   c) Una combinación de puertas NO-O y NO-Y

3. ¿Cuál sería la ecuación que representaría al siguiente circuito para que la salida fuese 0?

   a) $y = \overline{A} + \overline{B} + \overline{C}$
   b) $y = A + B + \overline{C}$
   c) $y = \overline{A} + B + \overline{C}$



4. El tiempo de acceso más el tiempo necesario en escribir la información leída se llama .....

5. El registro tampón se utiliza para:

   a) Realizar cálculos de precisión
   b) Almacenar información momentáneamente
   c) Contiene la dirección de memoria

6. Construir el diagrama por bloques a partir de la siguiente ecuación:

$$(\overline{A * B}) * (\overline{C + D}) * A * \overline{B} + (B * C) * D$$

7. ¿Quién se encarga de asignar las posiciones de memoria a las variables?

   a) El selector
   b) El carácter direccionable
   c) El compilador

8. El número – 14653.456 ¿de qué tipo es?

   a) Coma flotante.
   b) Alfanumérico.
   c) Coma fija.

9. La letra E ¿qué representa?

   a) Una coma.
   b) La base de exponentes 10.
   c) Un factor.

10. En el número – 2476 E–28, el – 2476 ¿qué representa?

    a) El exponente.
    b) Base de exponentes.
    c) La mantisa.

11. Las memorias de acceso aleatorio son las:

    a) ROM.
    b) PROM.
    c) RAM.

12. En una memoria ROM:

   a) Se puede grabar siempre.
   b) En ningún caso se puede grabar.
   c) Sólo puede grabarse en algunos casos.

13. El conocer el contenido de una posición de memoria significa

   a) Grabar.
   b) Borrar.
   c) Leer.

14. La cinta magnética es una memoria:

   a) Secundaria.
   b) Primaria.
   c) Terciaria.

15. Una memoria ROM es .....

2.

FUNCIONAMIENTO DEL ORDENADOR (II)

Unidad aritmética
Unidad de control
Unidades de E./S.

CUADRO RESUMEN DE LA LECCION

FUNCIONAMIENTO DEL ORDENADOR II

2.1. Funciones de la unidad A/L

- Unidad aritmética
  - Sumadores
    - En serie
    - En paralelo
  - Multiplicación
    - En serie
    - En paralelo
  - División
- Unidad lógica

2.2. Funciones de la unidad de control

- Reloj interno
- Registros de la unidad de control
- Ejecución de las instrucciones
  - Máquinas de una dirección
  - Máquinas de dos direcciones
  - Máquinas de tres direcciones

2.3. Funciones de la unidad de E/S

- E/S Off-line
- E/S On-line
- Métodos de E/S
- Flujo de información entre la UCP y las unidades de E/S

2.1. FUNCIONES DE LA UNIDAD ARIMETICA/LOGICA

En la unidad aritmética y lógica se efectúan las operaciones aritméticas (suma, resta, multiplicación y división) y las operaciones lógicas (comparaciones, transferencias, operaciones lógicas del álgebra booleana, bifurcaciones, etc.).

La U.A.L. realiza sus operaciones bajo el control de la Unidad de Control. La U.C. recoge las instrucciones de la memoria central y manda los operandos a la U.A.L., y, una vez efectuada la operación en la U.A.L., envía los resultados a la memoria en donde se almacenan.

Además de los dispositivos para realizar las operaciones lógicas y de comparación, la U.A.L. dispone, generalmente de tres registros para efectuar las operaciones aritméticas:

— Registro acumulador.
— Registro del primer operando
— Registro del segundo operando

El *registro acumulador* (AC) es el registro central de la U.A.L. En este registro se almacenan los operandos, los resultados parciales y los resultados finales de una operación.

Cuando comienza una operación aritmética el primer operando está contenido en el acumulador y cuando termina la operación el resultado se encuentra también en el acumulador. Mientras que el segundo operando se obtiene de la memoria.

Por ejemplo, supongamos la siguiente instrucción de una sola dirección:

Sumar	1000
Código de Operación	Dirección

Esta instrucción significa: sumar el contenido del acumulador al contenido de la posición de memoria 1000. El resultado de esta suma será el nuevo contenido del acumulador.

Si se quiere efectuar, por ejemplo, la suma:

$$5 + 3 = 8$$

en donde 5 es el primer operando, 3 el segundo y 8 el resultado (todos estos valores, en realidad, se encuentran en binario).

(AC) = 5 (valor inicial del acumulador)

contenido de la posición 1000 = (1000) = 3

(AC) = AC + (1000)

(AC) = 5 + 3

(AC) = 8 (resultado final)

Si la instrucción consta de dos direcciones entonces el primer y segundo operando están contenidos en sus respectivas posiciones de memoria.

Por ejemplo, sea la instrucción siguiente:

Sumar	1000	1010
Código de operación	Dirección del 1.^{er} operando	Dirección del 2.º operando

Esta instrucción significa: sumar al contenido de la posición de memoria 1000 a la posición 1110 y el resultado se sitúa en el acumulador.

Si en la posición 1000 se encuentran el valor 3, en la posición 1010 se encuentra el valor 12 y en el acumulador el valor 5.

Tendremos, por consiguiente:

(AC) = 5

(1000) = 3

(1010) = 12

la operación será:

(AC) = (1000) + (1010) = 3 + 12 = 15

por lo que el valor 5 del AC no ha influido para nada en la operación quedando en el acumulado el resultado de la operación, el valor 15.

Si la instrucción es de tres direcciones:

Sumar	1000	1010	1100
Código de operación	Dirección del 1.er operando	Dirección del 2.º operando	Dirección del resultado

entonces se suma el contenido de la posición 1000 al contenido de la posición 1010 y se sitúa el resultado en la posición de memoria 1100.

Si en la posición 1000 se encuentra el valor 3, en 1010 el valor 12 y en la posición 1100 el valor 7, tendremos:

(1000) = 3

(1010) = 12

(1100) = 7

de donde

(1100) = (1000) + (1010) = 3 + 12 = 15

por lo que el nuevo valor de la posición 1100 es 15.

En estas instrucciones de tres direcciones no se necesita el acumulador.

Para efectuar una suma o una resta se pueden utilizar los dispositivos sumadores, los cuales pueden realizar las operaciones *en serie*, bit a bit, o *en paralelo* donde se procesan todos los bits a la vez.

En los sumadores en serie los operandos se suministran en serie, un bit tras otro, obteniéndose el resultado de la suma del mismo modo, es decir, bit a bit.

En los sumadores en paralelo se suministran los operandos en paralelo, es decir, los bits se transmiten simultáneamente obteniéndose el resultado también en paralelo.

Las multiplicaciones y las divisiones se pueden hacer mediante sumas y restas consecutivas, por ejemplo, en una multiplicación se pueden obtener los productos parciales realizando sumas repetitivas y con la ayuda de un *registro multiplicador* y un *registro de operandos*.

Así pues, el ordenador dispone de una sección (U.A.L.) en donde se realizan operaciones aritméticas y operaciones lógicas. A su vez, esta sección puede considerarse como dos unidades por separado:

— Unidad aritmética
— Unidad lógica

2.1.1. Unidad aritmética

Las operaciones aritméticas se efectúan generalmente en una sola sección del ordenador (*unidad aritmética*). Estas operaciones son, básicamente: sumas con signo, almacenamiento temporal y desplazamientos de bits (o de palabras) entre los registros.

La unidad aritmética utiliza generalmente tres registros: dos registros de los operandos y un registro acumulador. Estos registros están controlados por señales de reloj procedentes de la unidad de control.

Los registros de la unidad aritmética pueden estar constituidos, según el tipo de sistema utilizado, por registros de desplazamiento, por registros de transferencia.

La unidad aritmética puede ser de tres tipos según el sistema de realización de las operaciones:

— Sistema serie
— Sistema paralelo
— Sistema serie-paralelo

Entre las máquinas que operan en sistema serie y las que operan en sistema paralelo no existen grandes diferencias en lo que respecta a los principios fundamentales, sólo se diferencian en el diseño de los circuitos.

En las unidades que trabajan en serie, para efectuar una operación aritmética se procesan las posiciones binarias de ambos operandos, bit a bit (Fig. 2.1).

En las unidades de cálculo en paralelo se procesan simultáneamente todas las posiciones binarias de los dos operandos (R1 y R2) y se obtiene en el resultado en otro registro de salida (RS) (Fig. 2.2).

En las unidades de cálculo que operan en el sistema serie-paralelo los bits de los operandos se procesan por grupos. Los dígitos binarios de cada grupo se procesan simultáneamente (en paralelo) y los grupos completos se procesan sucesivamente (en serie) (Fig. 2.3).

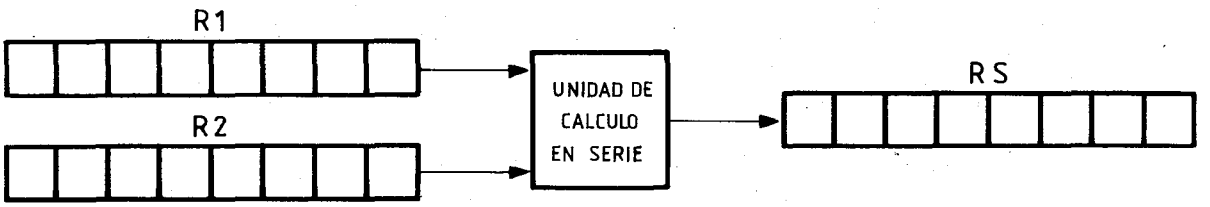


Fig. 2.1.

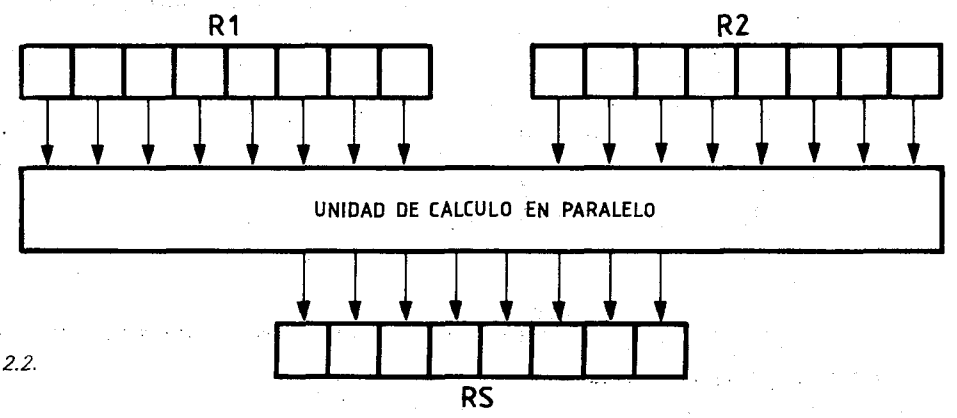


Fig. 2.2.

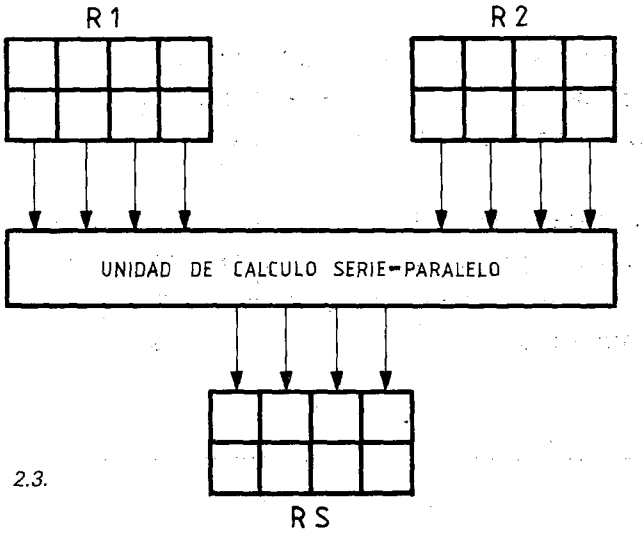


Fig. 2.3.

Sumadores

Los *sumadores* son unidades de cálculo, en donde se realizan operaciones de adición mediante circuitos apropiados. Un sumador puede efectuar también una resta complementando el sustraendo.

Un sumador está compuesto por tres puertas lógicas Y y una puerta O. Su funcionamiento puede verse en la siguiente tabla de la verdad, donde A y B son las entradas, S1 el resultado de la suma y S2 el bit de arrastre (*o de acarreo*).

A	*B*	*S1*	*S2*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Las operaciones de este circuito quedan definidas por las siguientes ecuaciones de álgebra de Boole:

$$S1 = \overline{A} * B + A * \overline{B}$$

$$S2 = A * B$$

De las dos ecuaciones anteriores, cuyo circuito conjunto se presenta en la figura 2.4, la ecuación S1 corresponde al operador lógico *O exclusivo* (EXOR):

$$A \oplus B = \overline{A} * B + A * \overline{B}$$

El diagrama puede simplificarse utilizando el símbolo del operador EXOR (Fig. 2.5). Este sumador tiene dos salidas: la de la suma y la del bit de acarreo.

El bit de acarreo debe volver a sumarse para dar el resultado exacto y obtener de esta forma una sola salida, S, del sumador.

Si se produce un bit de acarreo en la salida, se almacena en un elemento flip-flop, destinado para ello, y se suma mediante la entrada C (Fig. 2.6).

Estos sumadores pueden ser de dos tipos:

— Sumadores en serie.
— Sumadores en paralelo.

Sumador en serie

En un sumador en serie los operandos se introducen en serie, empezando por el bit de más a la izquierda.

Por la salida S se obtiene bit a bit (en serie) el resultado de la suma. Si se produce un bit de acarreo se almacena temporalmente en un elemento flip-flop y luego se agrega, mediante la entrada C, a la suma de la posición siguiente de orden superior.

La tabla de la verdad de este sumador es la siguiente:

A	*B*	*C*	*Suma*	*Acarreo*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

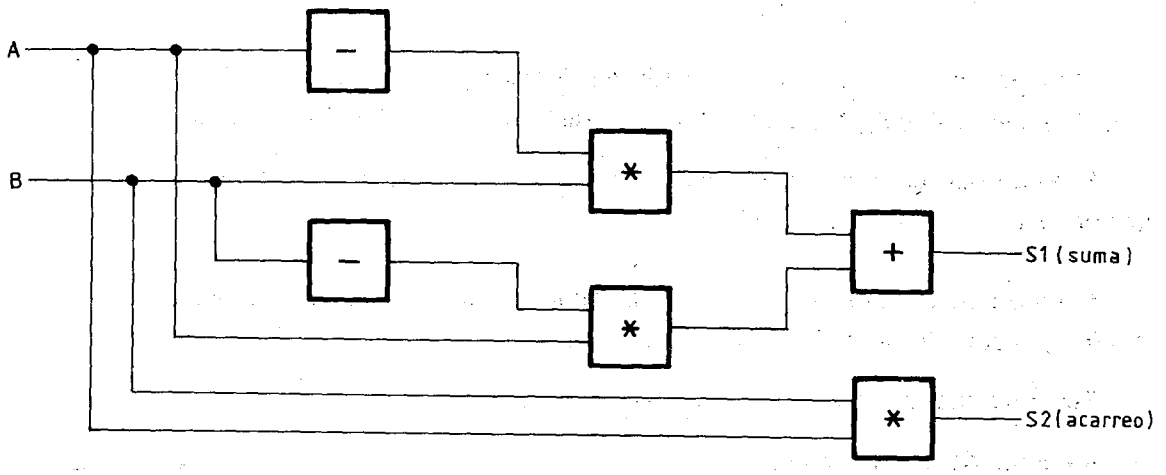


Fig. 2.4. Circuito correspondiente al conjunto de las ecuaciones:

$S1 = \bar{A} * B + A * \bar{B}$

$S2 = A * B$

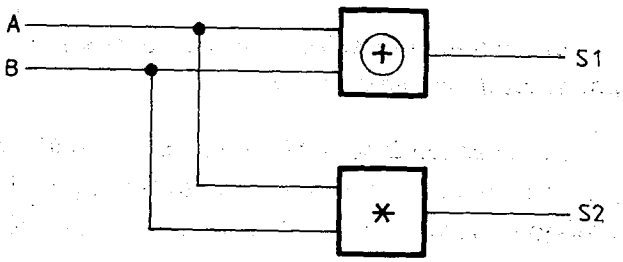


Fig. 2.5. El circuito anterior queda simplificado utilizando el símbolo del elemento EXOR.

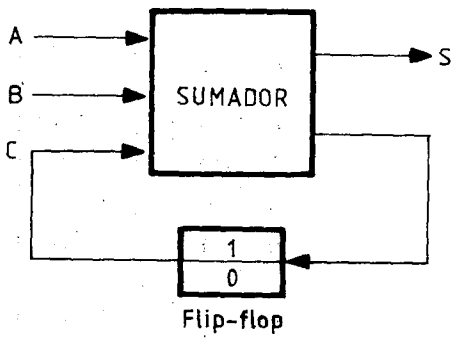


Fig. 2.6. El bit de arrastre se almacena en un elemento flip-flop y se introduce al sumador en el siguiente paso de la suma.

Veamos un ejemplo de una suma en serie:

	Paso 1	Paso 2	Paso 3	Paso 4	Paso 5
Operando 1	1 0 1 **1**	1 0 **1** 1	1 **0** 1 1	**1** 0 1 1	1 0 1 1
Operando 2	0 1 1 **0**	0 1 **1** 0	0 **1** 1 0	**0** 1 1 0	0 1 1 0
Acarreo		1	1 **1**	1 **1**	**1**
Suma	**1**	**0 1**	**0 0 1**	**0 0 0 1**	**1 0 0 0 1**

La suma se produce, por lo tanto, por resultados parciales almacenados sucesivamente en el registro. Esta operación está controlada por unos impulsos de sincronización.

Generalmente, el ordenador realiza la operación de adición de dos números con signo. El signo de cada número se identifica con el bit de signo.

Al efectuar una suma no se tienen en cuenta los bits de signo. Existe una célula especial llamda de *control de signo* encargada de detectar el bit de signo.

El proceso que se sigue en el ordenador para realizar una suma mediante un sumador en serie es el siguiente:

1) El primer operando (en este caso, primer sumando) se transfiere en serie desde la memoria al registro acumulador (AC) pasando por un registro tampón (RT).

2) El segundo sumando se transifere en serie (bit a bit) desde la memoria, pasando también por el RT, al sumador y, al mismo tiempo, se transfiere el primer sumando desde el acumulador al sumador, bit a bit. Por lo tanto, los bits que intervienen en la suma llegan al sumador al mismo tiempo desde el registro tampón y el acumulador.

El sumador realiza entonces la adición. El bit de más a la izquierda (bit de signo) es detectado por el *control de signo.* Si se detecta un 1 (signo negativo) el número pasa a través de un complementador antes de introducirse en el sumador; y si se detecta un 0 (signo positivo), el número llega directamente al sumador. El resultado de la suma queda, finalmente, en el acumulador.

El bit de signo es enviado en el momento adecuado por el elemento de *control de signo* quien ha efectuado las correspondientes comparaciones de los signos que han aparecido durante la operación.

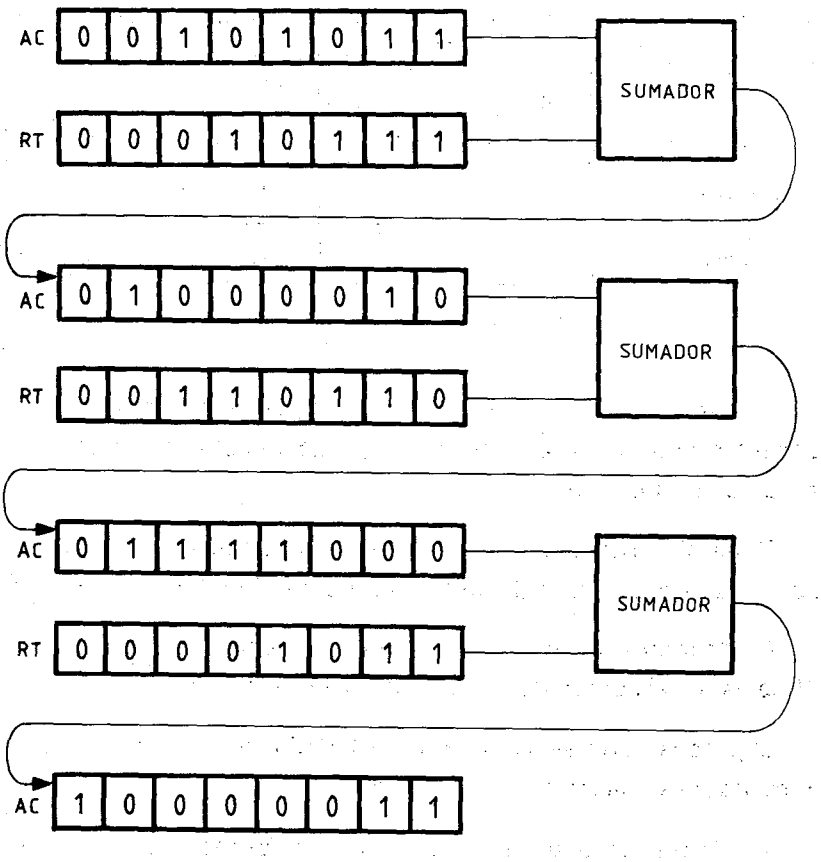


Fig. 2.7. Proceso seguido por el sumador para efectuar la suma de varios números.

Para realizar varias sumas, el proceso es el que muestra la figura 2.7.

La resta de dos números binarios se efectúa mediante un sumador en serie utilizando los bits de signo en una operación de adición.

Sumador en paralelo

Los sumadores en paralelo procesan todos los bits de los operandos simultáneamente. Los bits se transfieren en paralelo (simultáneamente) desde sus respectivos operandos hacia el sumador paralelo y el resultado de la suma se escribe en el registro acumulador.

El signo del resultado se obtiene siguiendo el mismo proceso que en un sumador en serie. En general, todo el proceso que se sigue en un sumador en paralelo es el mismo que el de un sumador en serie, salvo que la transferencia de los bits se efectúa en uno en paralelo y en el otro en serie.

Un sumador en paralelo está formado por varios sumadores en serie. Un ejemplo de sumador en paralelo es el de la figura 2.12, en donde los operandos son de 4 bits.

En este sumador pueden efectuarse sumas de grupos de 4 bits. Si se produce un bit de acarreo se suma a los bits contiguos. En el caso de que se pro-

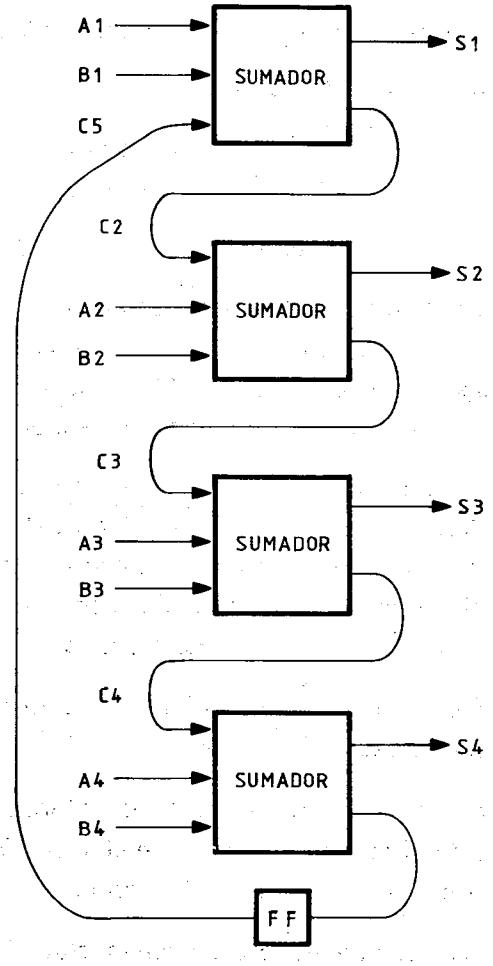


Fig. 2.8. Sumador en paralelo para operandos de 4 bits.

duzca un bit de acarreo en la última posición (C5), se almacena en un elemento flip-flop de transferencia y se suma al grupo de cuatro bits contiguos.

Multiplicación

La multiplicación puede considerarse como una extensión del proceso de adición excepto en la obtención del signo del resultado. En este caso el signo se obtiene a partir de los bits de signo de los dos números a multiplicar (multiplicando y multiplicador). La célula del control de signo detecta los dos bits de signo y a continuación determina el signo del producto.

En la multiplicación no se utilizan los complementadores ya que el resultado del producto (en valor absoluto) no se altera a causa de los signos de los dos números que se multiplican.

La multiplicación binaria puede realizarse de dos formas:

— En serie
— En paralelo

Multiplicación en serie

En el proceso de multiplicación en serie, el multiplicando se transfiere desde el registro tampón (RT) hacia un registro intermedio (MD) donde se almacena hasta que sea utilizado. El multiplicador se transfiere desde el RT a otro registro intermedio (MR) donde se almacena.

El siguiente paso consiste en desplazar el contenido del registro MR un bit a la derecha y se lee el bit de la izquierda del multiplicador. Si este bit es 1 se transfiere el multiplicando al sumador y se suma el contenido del acumulador (AC). La primera suma se efectúa con todos los bits del acumulador en 0, por lo que en el acumulador quedará almacenado el multiplicando.

A continuación se desplaza el acumulador un bit a la derecha y se realiza la misma operación de antes.

Si, por el contrario, el bit del multiplicador es un 0, se realiza un desplazamiento unitario a la derecha, en el acumulador y, al mismo tiempo, un desplazamiento similar en el multiplicador.

Después de estos pasos se efectúa el mismo proceso con el siguiente bit aparecido con el desplazamiento a la derecha del multiplicador y se repite el proceso anterior hasta que finaliza la multiplicación.

Ejemplo de multiplicación en serie

		111010
		× 101
1)	Sumar el multiplicando al acumulador	111010
2)	Desplazar un lugar a la derecha	111010
3)	Añadir 000000	000000
4)	Sumar 2) y 3)	111010
5)	Desplazar un lugar a la derecha	111010
6)	Sumar el multiplicando	111010
		100100010

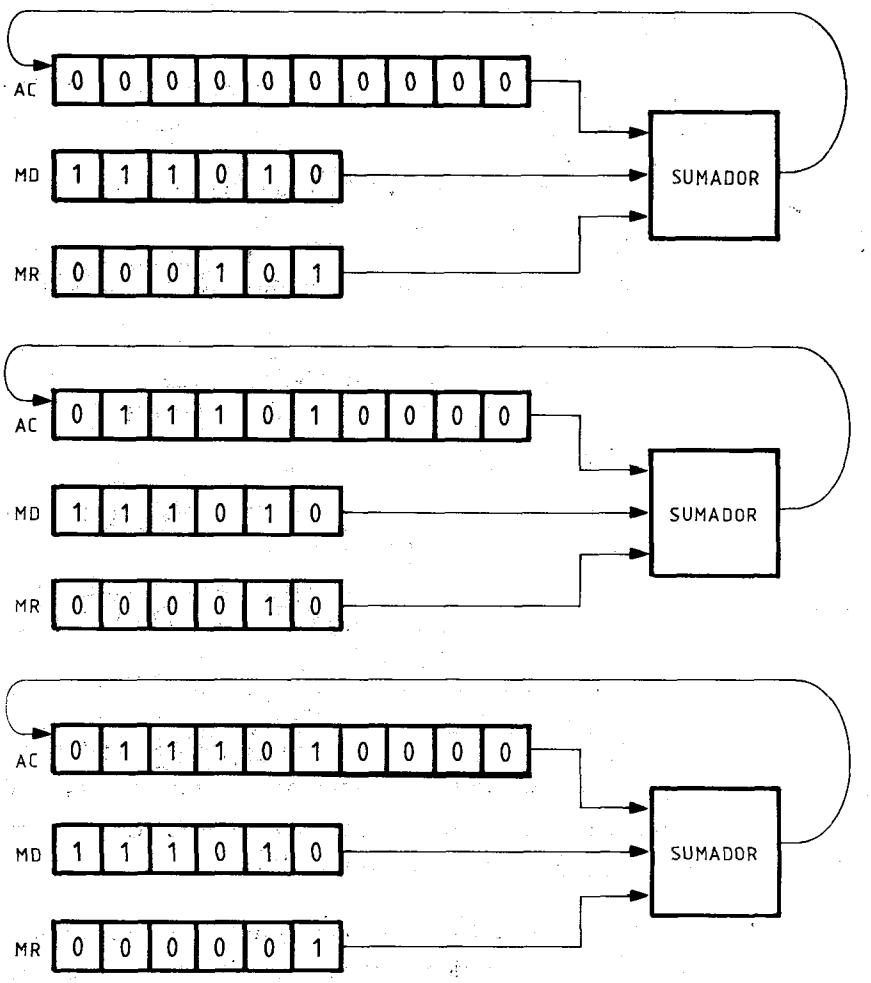


Fig. 2.9. Proceso seguido para la multiplicación en serie.

Multiplicación en paralelo

La multiplicación binaria en paralelo se realiza igual que la multiplicación en serie, excepto en el modo de transferir los datos que en este caso se realiza en paralelo.

Al igual que en la multiplicación en serie, cuando se efectúa una multiplicación en paralelo, el bit de más a la derecha del multiplicador determina si debe sumarse al acumulador el multiplicando (en el caso de que dicho bit sea 1) o una fila de ceros (en el caso de que el bit de la derecha del multiplicador sea 0).

Otra forma de realizar la multiplicación binaria consiste en utilizar un método serie-paralelo, consiguiéndose mayor rapidez.

División binaria

La división binaria puede efectuarse empleando dos métodos diferentes: mediante restas sucesivas o mediante desplazamientos.

Utilizando el método de los desplazamientos se necesita un circuito más complejo pero resulta mucho más rápido.

Por ejemplo, supongamos la siguiente división binaria:

```
101010   | 110
 110     ---
------    111
01001

 110
------
 00110

  110
------
  000
```

Para realizar esta división por desplazamientos deben efectuarse los siguientes pasos:

1) Comparar el divisor con el dividendo

```
------
101010
 110
------
```

El divisor es mayor que el dividendo. Entonces se coloca un 0 en el cociente.

```
101010  | 110
        |----
        | 0
```

A continuación, se desplaza el dividendo un lugar a la izquierda.

2) Comparar dividendo y divisor:

```
------
101010
 110
------
```

Ahora el divisor es menor que el dividendo. Se coloca un 1 en el cociente (a continuación del 0 que se había colocado anteriormente).

Se resta el divisor del dividendo:

```
 101010  | 110
-110     |----
------   | 01
 010010
```

y se desplaza el resto un lugar a la izquierda.

3) Se compara dividendo y divisor:

```
------
010010
 110
------
```

El divisor es menor que el dividendo: se coloca un 1 en el cociente y se resta el divisor del dividendo.

```
 101010  | 110
-110     |----
------   | 011
 010010
-110
------
 000110
```

El resto se desplaza un lugar a la izquierda.

4) Comparar dividendo y divisor:

```
000110
   110
```

Son iguales: se coloca un 1 en el cociente, se restan y queda 0. Por lo tanto el cociente es 0111.

```
 101010 | 110
-110    | 0111
 010010
-110
 000110
-110
 000000
```

En el proceso de una división binaria, se sitúa el dividendo en el acumulador. El contenido del acumulador se desplaza un lugar hacia la izquierda después de cada resta.

El divisor se coloca en un registro (D).

Al igual que en la multiplicación, la célula de control de signo determina el signo del resultado leyendo los bits de signo de los operandos. La división se efectúa considerando los números como positivos y en el resultado se coloca el signo correspondiente.

En primer lugar se envía el dividendo desde el acumulador, y el divisor, desde el registro D, al sumador. Para restar del acumulador el contenido del registro D mediante el sumador, se sitúa un complementador entre D y el sumador.

Si después de la resta queda un número positivo se coloca un 1 en el registro para el cociente (C). El resultado que queda en el acumulador, después de terminar la resta, se desplaza un bit a la izquierda para efectuar una segunda resta.

En el caso de que el resultado de la resta sea negativo, esto significa que el divisor es mayor que el dividendo, entonces se coloca un 0 en el registro C.

El proceso sigue del mismo modo sucesivamente hasta que finaliza la división.

Hasta aquí se ha visto la división binaria en serie.

La división binaria en paralelo se realiza siguiendo pasos semejantes a los de la división en serie, excepto en el modo de transferir los datos. En este caso se efectúan las transferencias en paralelo.

2.1.2. Unidad lógica

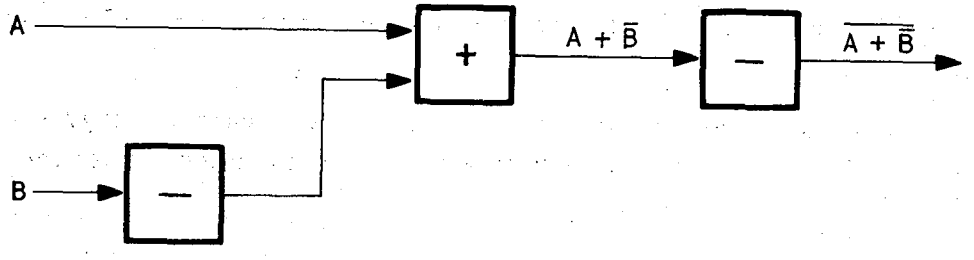
Se denomina *unidad lógica* al conjunto de dispositivos que realizan básicamente comparaciones.

Para determinar si un número es mayor, igual o menor que otro se utiliza un circuito lógico llamado *comparador.*

En el resultado de una comparación de dos elementos binarios sólo hay dos valores posibles que se traducen en 0 y 1.

Así, por ejemplo, si se cumple la relación A < B, sólo existe una posibilidad: A = 0 y B = 1.

Un circuito NO-O con la entrada B complementada puede detectar que A < B, es decir que A es 0 y B es 1.



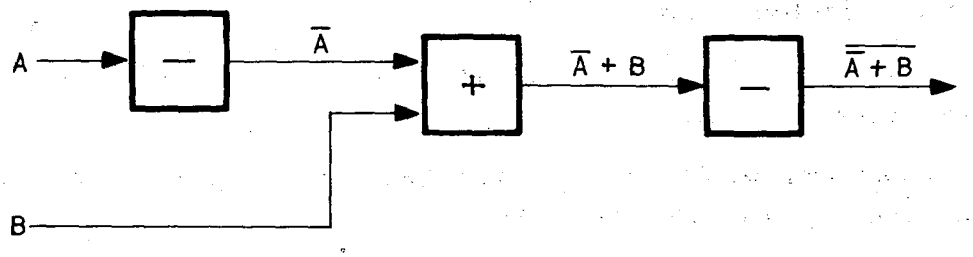
La tabla de la verdad de este circuito es:

A	*B*	$\overline{B}$	*S*	*Relación*
0	0	1	0	A = B
0	1	0	1	A < B
1	0	1	0	A > B
1	1	0	0	A = B

Por lo tanto, este circuito sólo emite una señal (1) cuando A es menor que B (es decir, A < B).

Si en este circuito NO-O se complementa la entrada A en lugar de la entrada B se obtiene un circuito que detecta la relación A > B. En este caso, la salida será:

$$S = \overline{\overline{A} + B}$$



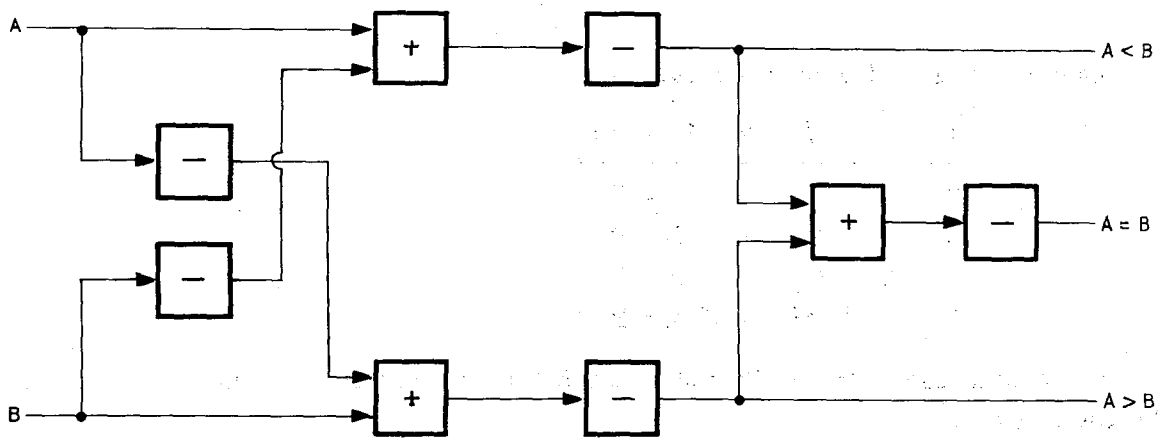
La tabla de la verdad de este circuito es:

A	*B*	$\overline{A}$	*S*	*Relación*
0	0	1	0	A = B
0	1	1	0	A < B
1	0	0	1	A > B
1	1	0	0	A = B

= 0, es decir, cuando A > B.

Este circuito sólo da señal (1) de salida en el caso en que A = 1 y B = 0, es decir, cuando A > B.

El circuito comparador puede detectar cada uno de los casos A > B y A < B y además detecta A = B si no se detecta señal en las dos relaciones anteriores.



La tabla de la verdad del anterior circuito comparador es la siguiente:

A	B	$\overline{A}$	$\overline{B}$	$A < B$	$A > B$	$A = B$
0	0	1	1	0	0	1
0	1	1	0	1	0	0
1	0	0	1	0	1	0
1	1	0	0	0	0	1

Cada una de las relaciones entre A y B puede representarse en binario según el valor de las entradas del circuito comparador. Así por ejemplo, si las entradas son A = 1 y B = 0, es decir, A > B, las salidas del circuito son 100. Si las entradas son A = 0 y B = 0, es decir A = B, las salidas son 010. Y, por último, si las entradas son A = 0 y B = 1, es decir, A < B, las salidas son 001.

Así pues, puede construirse la siguiente tabla:

Estado de las salidas			*Significado*
$A < B$	$A = B$	$A > B$	
0	0	1	A > B
0	1	0	A = B
1	0	0	A < B
0	1	1	A ⩾ B
1	1	0	A ⩽ B
1	0	1	A ≠ B
1	1	1	No tiene validez en una comparación
0	0	0	

Cualquiera que sea el resultado de una comparación puede representarse en binario.

Las salidas del circuito 011 son una combinación de las relaciones A = B y A > B:

$A = B \rightarrow 010$

$A > B \rightarrow \underline{001}$

$A \geqslant B \rightarrow 011$

Lo mismo sucede con las salidas 110,

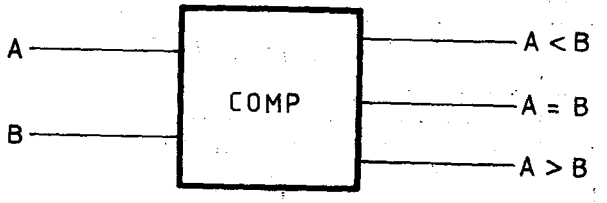
$A < B \rightarrow 100$

$A = B \rightarrow \underline{010}$

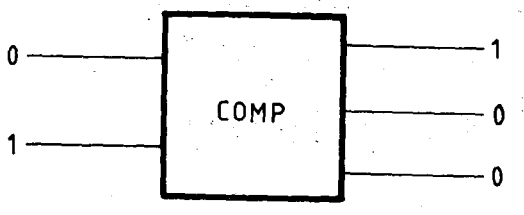
$A \leqslant B \rightarrow 110$

Las salidas 101 son una combinación de 100, es decir, A < B, y de 001, es decir, A > B. Así pues, 101 indica que tanto puede ser A < B como A > B, o lo que es lo mismo, A ≠ B.

Para simplificar el esquema del circuito comparador, puede utilizarse la representación siguiente:



Por ejemplo, si las entradas son A = 0 y B = 1 puede representarse como se muestra a continuación:



2.2. FUNCIONES DE LA UNIDAD DE CONTROL

El funcionamiento de la unidad de control depende esencialmente de las instrucciones que componen los programas.

La programación consiste en dirigir el ordenador a través de un determinado proceso.

El conjunto de instrucciones del programa se almacena en la memoria central antes de procesarse. Después, el ordenador interpreta cada una de las instrucciones y las ejecuta.

La unidad de control coordina y controla las operaciones en el orden adecuado; decodifica las instrucciones y determina las señales de tiempo necesarias para todo el ordenador.

En general, la unidad de control realiza sus funciones empleando elementos tales como:

1) Un reloj interno (para sistemas síncronos).

2) Señales de tiempo que indican:

   — El comienzo o el final del tiempo destinado a una palabra o a grupos de palabras de las instrucciones.

   — Un impulso para indicar el comienzo de las operaciones principales de cálculo.

   — Impulsos distintos para cada intervalo dentro del tiempo de duración de una palabra.

2.2.1. Reloj interno

Las operaciones que realiza la unidad de control le son ordenadas por las instrucciones del programa almacenado en la memoria central.

La U.C. se encarga de controlar el desarrollo de ejecución de las instrucciones, de controlar las operaciones de cálculo, de entrada y salida, etc., así como del control del tiempo que emplea cada operación.

Para el control del tiempo, la U.C. dispone de un registro especial que varía aumentando o disminuyendo una cantidad determinada en ciertos períodos de tiempo. Este registro se denomina *reloj interno* y está formado por un circuito eléctrico que es capaz de generar señales o impulsos eléctricos.

El *reloj* es un elemento básico de control que suministra una sucesión continua de impulsos a la frecuencia necesaria.

El reloj establece el ritmo de funcionamiento del ordenador.

Los circuitos eléctricos de la máquina se adaptan sincrónicamente a este reloj. Este modo de trabajo de la U.C. se llama *funcionamiento síncrono,* es decir, cuando la U.C. controla temporalmente el ordenador por medio del reloj que emite impulsos en períodos o intervalos fijos.

Durante la ejecución de una instrucción se producen varios impulsos de reloj y varias *conexiones lógicas* que no tienen que emplear necesariamente un período completo de impulsos, de forma que se produce una *interrupción* en la ejecución de la instrucción desde el momento que termina una conexión lógica hasta que aparece un nuevo impulso de reloj.

Las *conexiones lógicas* son representaciones de funciones de conmutación que utilizan operadores lógicos (AND, OR, NOT, etc.).

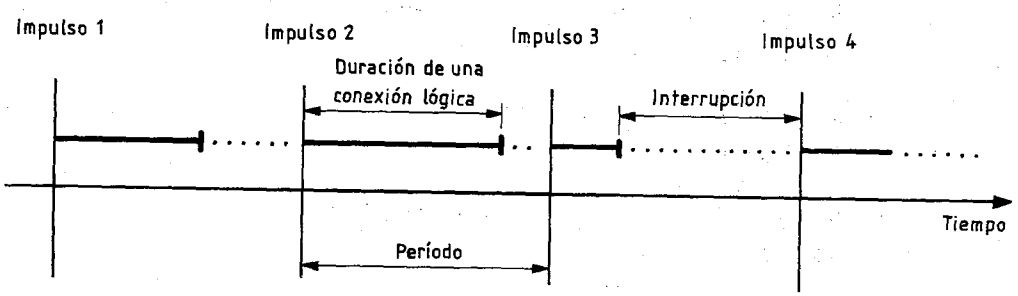


Fig. 2.10. Funcionamiento síncrono de la unidad de control.

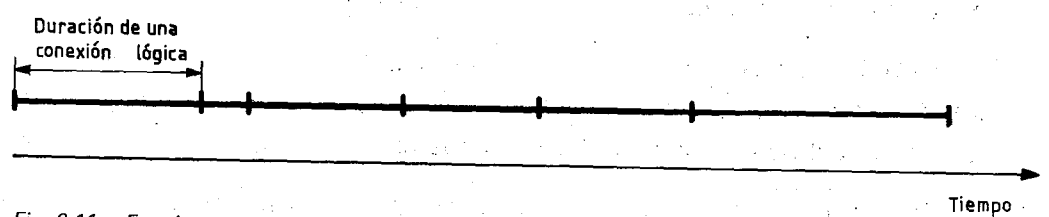


Fig. 2.11. Funcionamiento asíncrono de la unidad de control.

Otro tipo de funcionamiento de la unidad de control es el *funcionamiento asíncrono,* en el que no se utiliza el reloj interno. Cuando finaliza una operación se manda una señal a la U.C. quien da comienzo a la siguiente instrucción.

Con el funcionamiento asíncrono desaparecen las interrupciones en la ejecución de una instrucción con lo que se consigue disminuir el tiempo de ejecución.

2.2.2. Registros de la U.C.

La unidad de control se encarga de controlar la secuencia de la ejecución de las instrucciones de un programa; interpreta el significado de las instrucciones que recibe de la memoria central, modifica las instrucciones necesarias y manda las señales que producen la ejecución de las instrucciones.

Para poder interpretar una instrucción, la parte direccional contiene la dirección del operando, y para interpretar la parte operativa la U.C. dispone de un dispositivo traductor que mediante mecanismos electrónicos transmite señales eléctricas a las distintas unidades del ordenador, traduciendo los caracteres de la parte operativa.

Para poder realizar estas funciones, la U.C. dispone de los siguientes registros:

— Registro contador de instrucciones

— Registro de instrucción

— Registro índice

El *registro contador de instrucciones* (RCI), llamado también *registro de dirección de la instrucción,* contiene la dirección de la siguiente instrucción a ejecutar.

El RCI controla la secuencia de la ejecución de las instrucciones mediante cambios de la dirección que contiene dicho registro.

La dirección del registro contador aumenta en una unidad cada vez que se ejecuta una instrucción. Por lo que el contenido (la dirección) del registro contador pasa por un dispositivo de sumar, se le suma 1 y queda registrado con el nuevo valor, es decir: RCI $\rightarrow$ RCI + 1.

Por ejemplo, si el RCI contiene la dirección A entonces se ejecuta la instrucción que hay en la posición A de la memoria. Una vez ejecutada dicha instrucción el contenido del RCI aumenta en una unidad pasando a A + 1 de forma que

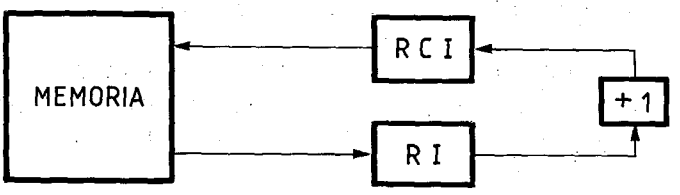


Fig. 2.12.

la siguiente instrucción a ejecutar es la que esté contenida en la posición A + 1 de la memoria (Fig. 2.12), excepto cuando la instrucción contenida en A sea de bifurcación, en tal caso, se cambia el contenido del RCI por la dirección que contiene la instrucción de bifurcar.

El *registro de instrucción* (RI) llamado también *registro decodificador* es el que se está ejecutando en un momento determinado.

Cuando se extrae una instrucción de la memoria, se transfiere al registro de instrucciones.

La instrucción se carga en el RI mediante la dirección de la posición de memoria que contiene el registro contador de instrucciones.

El *registro índice* se utiliza para modificar las direcciones contenidas en las instrucciones, para controlar las instrucciones de bifurcación y para efectuar operaciones de cálculo en direcciones.

El registro índice permite al ordenador efectuar el *direccionamiento indexado* mediante el cual se suma el contenido del registro índice a la dirección de una instrucción para calcular la dirección real de esa instrucción. Con el direccionamiento indexado, por lo tanto, no se obtiene directamente la dirección de la posición de memoria que interesa sino que se obtiene mediante el cálculo especificado.

Así pues, el registro índice se emplea para modificar una instrucción. Por ejemplo, en lugar de transferir el contenido, puede sumarse el contenido del registro índice al RCI (el contenido de ambos registros pasa a un sumador). El resultado se escribe de nuevo en el RCI.

Los registros índice se utilizan también como contadores.

2.2.3. Ejecución de las instrucciones

Las instrucciones o el programa que contiene estas instrucciones se almacena en la memoria central por medio de una operación de lectura en la unidad de entrada. Esta operación la efectúa la consola del ordenador.

En la memoria se destinan ciertas posiciones para las instrucciones, otras para los datos que se van a procesar y, por último, se destinarán otras posiciones de memoria para situar los resultados parciales y finales de las operaciones.

La Unidad de Control (UC) de la unidad central de proceso coordina las operaciones de la UCP y de los dispositivos periféricos.

La UC selecciona, en primer lugar, las instrucciones del programa almacenado en la memoria central y luego las ejecuta. El procedimiento que sigue para ejecutar una instrucción es el siguiente:

1) Localizar y tomar la instrucción de la memoria.
2) Transferir la instrucción de la memoria a la propia unidad de control.
3) Situar la dirección de la siguiente instrucción en la posición de memoria correspondiente.
4) Ejecutar la instrucción.

Estos pasos se repiten hasta que se ejecutan todas las instrucciones del programa. En realidad estos pasos pueden resumirse en dos fases en el desarrollo del programa:

— Fase de toma de la decisión.

— Fase de ejecución de la instrucción.

Las instrucciones que van a ser ejecutadas se encuentran almacenadas en posiciones de memoria. Para ejecutar una instrucción, por tanto, la UC toma dicha instrucción de la memoria y la sitúa en el registro de instrucción (RI).

La dirección de la primera instrucción a tomar de la memoria se encuentra en el registro contador de instrucciones (RCI) de la UC. El RCI contiene la instrucción que ha de ser interpretada por la UC. La interpretación del código de operación se efectúa, generalmente, por un traductor o decodificador, ya que la operación a realizar que indica el código de operación suele estar codificada. Por ejemplo si en el código de operación aparece la palabra «SUMAR», ésta se ha de traducir (decodificar) a los ceros y unos correspondientes de forma que la máquina entienda que se trata de una operación de sumar.

La interpretación de la dirección es más sencilla ya que está representada generalmente por ceros y unos y no necesita decodificarse.

Una vez interpretado el código de operación de la instrucción, la UC activa los circuitos necesarios para la ejecución de la instrucción, emite señales a la unidad aritmética/lógica, toma los datos necesarios de la memoria y al finalizar la operación de cálculo guarda el resultado en los registros o posiciones de memoria designados para almacenar los resultados parciales o finales. En el caso de una instrucción de salida las señales de la UC van dirigidas a la memoria y a la unidad de salida.

Para ejecutar la siguiente instrucción, el registro contador de instrucciones contiene la dirección, de forma que el contenido de la posición de memoria con dicha dirección pasa al registro de instrucción. Después, la dirección del RCI aumenta, mediante un dispositivo de suma, en una unidad siempre que la instrucción no sea de bifurcación sino que siga un desarrollo secuencial (o *lineal*). En caso de bifurcación el RCI no aumenta en uno sino que se sustituye la dirección anterior del RCI por la dirección del registro de instrucción.

La velocidad y el modo de operación de un ordenador dependen de la estructura de la instrucción que se utilice. Por ejemplo, si se emplea una instrucción de tres direcciones se obtiene mayor velocidad de operación que empleando una instrucción de una dirección. Además con instrucciones de tres direcciones se codifican menos instrucciones que utilizando instrucciones de una o dos direcciones.

Así pues los ordenadores pueden clasificarse en tres grupos según el tipo de instrucciones utilizadas:

— Máquinas de una dirección

— Máquinas de dos direcciones

— Máquinas de tres direcciones

Máquinas de una dirección

Las instrucciones se transfieren desde la memoria al decodificador de instrucciones (RI) pasando por el registro tampón.

El decodificador de instrucciones se divide, a su vez en dos elementos que codifican por separado cada una de las partes de una instrucción: uno de estos elementos decodifica la parte operativa de la instrucción y el otro elemento decodifica la parte direccional.

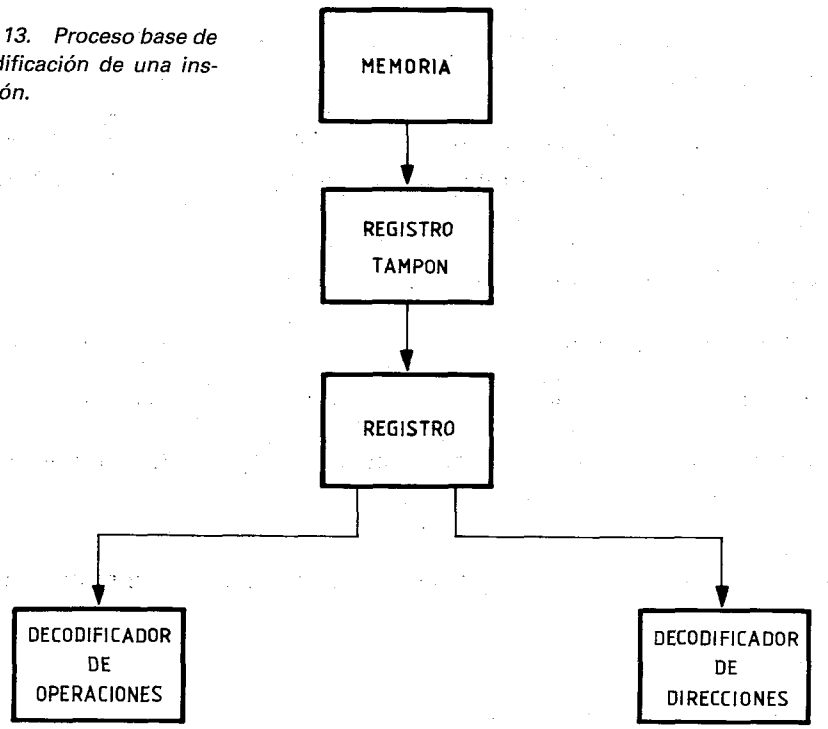


Fig. 2.13. Proceso base de decodificación de una instrucción.

La ejecución de una instrucción en un ordenador de una dirección consta de dos fases:

— Tiempo de programa

— Tiempo de operación

En el *tiempo de programa* se realizan las siguientes etapas:

1) Buscar la dirección de la instrucción. Esta operación se realiza generalmente mediante el registro contador de instrucciones.

2) Transferir la instrución desde el registro tampón al decodificador de operación y decodificador de dirección.

3) Decodificar la instrucción.

El *tiempo de operación* es el tiempo necesario para la ejecución de la instrucción.

La secuencia de operaciones en la ejecución de una instrucción de una dirección es en general:

1) Introducir la dirección de la primera instrucción en el registro contador de instrucciones (RCI).

2) Transferir la primera instrucción (indicada por el RCI) desde el registro tampón al registro decodificador (es decir, al registro de instrucción, RI).

3) Decodificar la primera instrucción.

4) Sumar 1 al RCI.

5) Introducir la dirección de la segunda instrucción en el RCI.

6) Transferir la segunda instrucción desde el registro tampón al RI.

7) Etc.

Y así sucesivamente hasta que se han procesado todas las instrucciones.

Máquinas de dos direcciones

En los sistemas de dos direcciones para cada instrucción se indican dos posiciones de memoria (que corresponden a una palabra o a un carácter).

SUMAR	15	28

Por ejemplo, se va a efectuar la suma de los números 486 y 162. Estos números están almacenados en la memoria central. Supongamos que la dirección del primer operando (486) es 15 y la dirección del segundo operando (162) es 28.

En primer lugar, se transfieren las posiciones de memoria situadas a la derecha de ambos operandos a los registros de los operandos, R01 y R02. Después se efectúa la suma de ambas posiciones en el sumador. El resultado de esta suma se transfiere al registro acumulador. En caso de que se produzca acarreo, éste se transfiere al registro de acarrreo. La posición inferior del resultado se lleva a la posición de memoria de dirección 28. A continuación se repite el mismo proceso con el resto de las posiciones de ambos operandos y así sucesivamente hasta que los operandos se han sumado completamente.

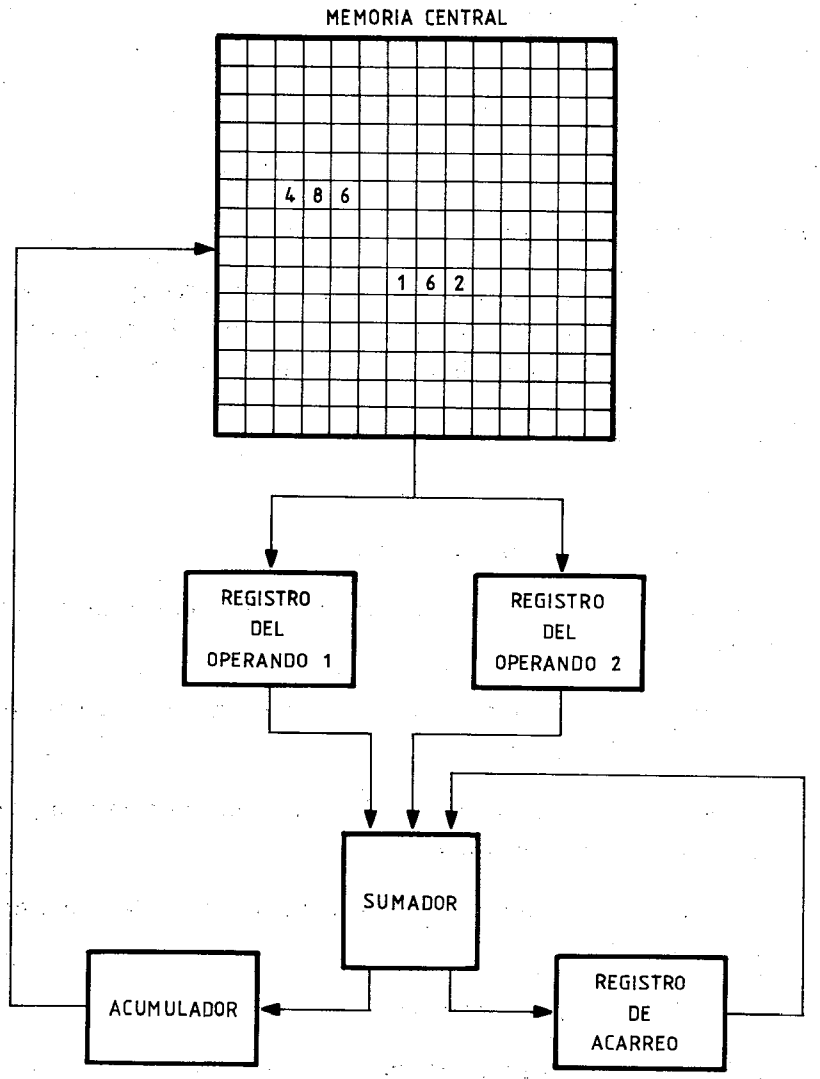


Fig. 2.14. Suma de dos sumandos en una máquina de dos direcciones.

El resultado de la suma (648) queda almacenado en el AC. En primer lugar se ha almacenado el 8, después el 4 y por último el 6 (como puede verse en la tabla).

R01	R02	AC	RA
6	2	0	0
8	6	8	0
4	1	4	1
0	0	6	0

Máquinas de tres direcciones

En un sistema de tres direcciones se decodifica la parte operativa al mismo tiempo que las direcciones 1 y 2. La decodificación de la dirección 3 se efectúa después en uno de los decodificadores de las direcciones 1 y 2.

Las tres direcciones mencionadas son:

— Dirección en la memoria de la palabra A

— Dirección en la memoria de la palabra B

— Dirección de la posición de memoria en donde debe introducirse el resultado.

Para realizar una operación aritmética con una instrucción de tres direcciones se siguen los siguientes pasos:

1) Transferir la instrucción a los decodificadores de operación y de dirección y decodificar la instrucción.

2) Transferir al mismo tiempo las palabras A y B desde la memoria a la unidad arimética.

3) Efectuar la operación aritmética.

4) Escribir los resultados en la memoria.

2.3. FUNCIONES DE LA UNIDAD DE ENTRADA/SALIDA

La unidad o sección de entrada/salida del ordenador tiene la función de enlazar o acoplar (*interface*) entre las unidades periféricas y la unidad central de proceso.

Mediante la sección de entrada/salida se establece una comunicación con la máquina. Se proporcionan los impulsos que representan informaciones a tratar (datos o instrucciones) y una vez finalizado el tratamiento se extraen los resultados (o informaciones tratadas).

Para establecer una correcta comunicación con la máquina hay que tener en cuenta una serie de limitaciones, tales como: su alfabeto, el formato de las palabras, la adaptación de velocidad, etc.

Puesto que el ordenador funciona mediante impulsos eléctricos, su alfabeto se limita a dos símbolos. Es decir, su alfabeto es binario, y deben representarse las informaciones por medio de combinaciones de dos elementos (presencia o ausencia de impulsos).

Las palabras empleadas en el ordenador tienen sus limitaciones (en cuanto a longitud). Se utilizan palabras de formato fijo (longitud fija) y de longitud variable. En cualquier caso, si se sobrepasan ciertos límites para representar una determinada palabra, ésta debe fraccionarse de forma que quede representada en palabras tratables por el ordenador.

Respecto a la adaptación de velocidades, es conveniente tener en consideración el tiempo con el que opera la máquina y el tiempo que puede emplear el operador en introducir las informaciones. Para obtener el máximo rendimiento del ordenador pueden utilizarse memorias tampones que, a la salida, se cargan sucesivamente al ritmo del ordenador y se descargan a la velocidad adecuada para el operador. Y a la entrada, permiten suministrar información a la UCP a la velocidad adecuada para que trabaje de forma continua.

En un sistema de proceso de datos existe un conjunto de terminales (impresoras, teclados, pantallas, etc.) conectados a unas memorias tampones que están en conexión directa *(on-line)* con el ordenador.

Los elementos básicos para las memorias tampón están constituidos por soportes magnéticos tales como tambores magnéticos, discos magnéticos o cintas magnéticas.

Las memorias tampón son capaces de adaptarse a la velocidad de operación de la unidad central de proceso y al ritmo lento de los periféricos de entrada/salida.

Así pues, en general, las funciones de la unidad de E/S son la operación de transferencia de la información entre la UCP y los periféricos y la preparación de la información en el código adecuado para hacer posible la comunicación con otros dispositivos de la máquina.

El acoplamiento de los periféricos con la UCP está supervisada por la unidad de control. La información puede suministrarse por diversos canales y debe establecerse el orden de prioridad de forma que cuando la información ocupe un canal, los otros estén interrumpidos.

El funcionamiento de la sección de E/S viene determinada principalmente por los tipos de periféricos utilizados. Algunos de estos periféricos están destinados a recibir y transmitir información a los usuarios. Otros periféricos tienen distinas funciones (por ejemplo, los convertidores analógico-digital y digital-analógico).

La entrada y salida de la información puede efectuarse por medio de soportes de datos (*off-line*) o directamente (*on-line*).

2.3.1. E/S off-line

La entrada/salida *off-line* (fuera de línea) utiliza medios indirectos con el fin de establecer mayor rapidez de comunicación con la memoria central del ordenador.

Estos medios indirectos (soportes de información) se preparan por medio de máquinas auxiliares que no forman parte del ordenador propiamente dicho.

La *entrada off-line* consiste en la entrada de datos por medio de soportes tales como:

— Tarjetas perforadas

— Cintas perforadas

— Documentos escritos.

La entrada de los datos contenidos en las tarjetas perforadas se efectúa por medio de las lectoras de tarjetas en donde se suministran las tarjetas de forma que van llegando secuencialmente al dispositivo de lectura y una vez efectuada la lectura las tarjetas son enviadas al dispositivo de salida de la lectora.

Las tarjetas leídas se almacenan en un *registro tampón* desde donde puede transportarse la información mediante el canal de entrada al ordenador. Cuando el registro tampón se llena la lectora envía una señal al ordenador para que se vacíe antes de que tenga que interrumpir la operación de lectura.

Para la entrada de datos por medio de cintas perforadas se emplean unidades lectoras de cintas perforadas cuyo funcionamiento es semejante a la lectora

de tarjetas, aunque la información es leída carácter a carácter, mientras que en la lectora de tarjetas se puede leer por filas o por columnas de la tarjeta.

Existen también unidades lectoras capaces de leer documentos escritos que estan diseñados de forma que puedan ser leídos por el hombre y por el ordenador.

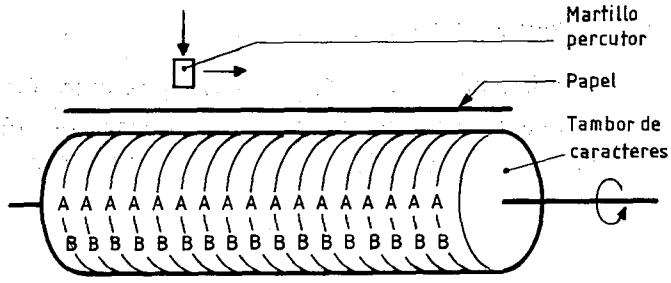
La *salida off-line* consiste en obtener los resultados por medio de soportes de datos tales como:

— Tarjetas perforadas

— Cintas perforadas

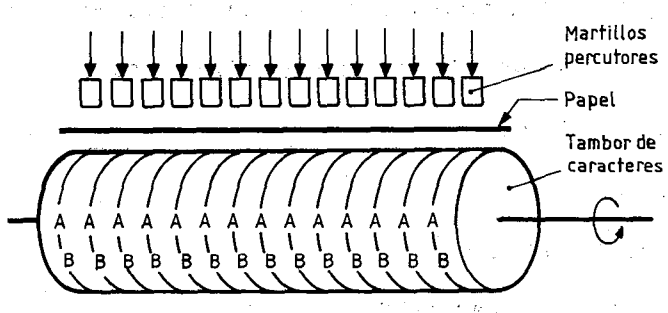
— Documentos escritos

— Listados



Impresora en serie

Fig. 2.15.



Impresora en paralelo

Fig. 2.16.

El soporte más utilizado para la salida de datos es el papel continuo (listado) que se obtiene de la impresora.

Las impresoras pueden ser de dos tipos según el sistema de impresión:

— Impresoras de caracteres

— Impresoras de líneas

Las impresoras de caracteres o impresoras en serie imprimen en serie, carácter a carácter. El carro de la impresora se va desplazando longitudinalmente de forma semejante al funcionamiento de una máquina de escribir. La velocidad de impresión puede ser hasta de 50 caracteres/segundo.

Las impresoras de líneas o impresoras en paralelo imprimen una línea completa de una sola vez, consiguiendo velocidades de hasta 30 líneas por segundo. Cada línea tiene de 100 a 160 caracteres.

En las impresoras en serie se utiliza un tambor con todos los caracteres, que puede girar en ambos sentidos. Un martillo percutor se desplaza a lo largo de la línea de forma que cuando recibe una señal eléctrica se dispara y presiona el papel contra el tambor de caracteres. El número de caracteres repetidos en una fila del tambor es, generalmente, de 64 caracteres.

En las impresoras en paralelo se dispone de un martillo percutor fijo para cada posición del tambor de caracteres, de manera que en cada rotación del tambor se puede imprimir una línea completa.

2.3.2. E/S on-line

El proceso de interpretación de los datos pasando a la memoria central, así como el proceso de escritura sobre un soporte de información desde la memoria central puede considerarse como operaciones de entrada/salida *on-line*.

Los dispositivos de entrada/salida on-line más utilizados son:

— Teclado y máquina de escribir (entrada y salida)

— Impresora (salida).

— Pantalla de visualización (salida)

— Teclado y pantallas (entrada y salida).

La *entrada on line*, es decir, la entrada directa de datos al ordenador se puede hacer bien mediante unidades especiales acopladas al ordenador, donde

se transmiten los datos por medio de líneas, o bien a través de un soporte de datos intermedio.

La *consola* es una unidad de entrada y salida on line. El operador encargado de la máquina puede manejar el sistema dando instrucciones por medio del teclado de la consola. Estas instrucciones pueden consistir en instrucciones de control, o iniciar o parar un proceso, etc. Para la salida de información, la consola dispone de unos dispositivos con pilotos luminosos.

Otras unidades de entrada y salida directa son las máquinas de escribir on line, las pantallas, etc.

Por ejemplo, en una compañía aérea se pueden efectuar consultas al ordenador acerca de las plazas disponibles en los vuelos mediante el empleo de terminales con teclados. Estas terminales están conectadas directamente (on line) con el ordenador, de forma que se pueden hacer las consultas y obtenerse las respuestas en unos segundos.

Los dispositivos de salida directa (on line) son los que transmiten información directamente desde el ordenador a una terminal.

Con los dispositivos de entrada y salida off-line (fuera de línea) los tiempos de consulta son relativamente grandes, sobre todo si se comparan con los tiempos que se obtienen en un sistema on line (en línea), lo cual significa que se puede pasar de horas e incluso días en preparar los datos de entrada hasta obtener los resultados en un sistema off-line, mientras que en un sistema on line esas operaciones se realizan en segundos.

Los sitemas on line resultan mucho más caros que los sistemas off-line por lo que estos últimos se utilizan con más frecuencia.

2.3.3. Métodos de E/S

Los métodos de E/S varían de uno a otro sistema. Existen diversos métodos. Tres de ellos son:

— Entrada/salida descentralizada

— Entrada/salida centralizada

— Entrada/salida compensada

La *E/S descentralizada* es el más básico.

Los periféricos pueden encontrarse a gran distancia del ordenador. Las co-

municaciones se realizan solamente con métodos digitales, transmitiéndose las palabras tanto en serie como en paralelo.

Cada periférico dispone de su propio decodificador de instrucciones. Las instrucciones deben contener la información siguiente:

— Operación a ejecutar

— Dirección del periférico

— Dirección de la memoria donde va destinada la información

— Líneas de transmisión

En este caso, la sección de entrada/salida está acoplada directamente con el ordenador. Además, cada periférico puede ser direccionado independientemente.

En el caso de *E/S centralizada,* los periféricos están montados en la sección de E/S. Para el control de E/S se dispone de un decodificador de instrucciones que sirve para todos los dispositivos periféricos que se utilicen.

En el método de *entrada/salida compensada,* los periféricos están conectados con canales intermedios.

Los sistemas de E/S compensados se diseñan de forma que las interrupciones procedentes de los periféricos puedan detener la secuencia del programa y solicitar la atención del ordenador. Esto puede hacerse de forma que se atienda primero a los programas que tengan más prioridad.

2.3.4. Flujo de información entre la UCP y las unidades de E/S

Entre las unidades de E/S y la UCP existe una unidad de control de la sección de E/S encargada de diversas funciones, tales como:

— Interpretación de las instrucciones enviadas desde la UCP.

— Generar señales de control para producir determinadas funciones en las unidades de E/S (por ejemplo, avance del papel de la impresora).

— Interpretación de señales enviadas desde la UCP o desde las unidades de E/S (por ejemplo, una lectora de tarjetas comunica el llenado del registro tampón y solicita su vaciado para no interrumpir el proceso de lectura).

— Emitir señales adecuadas para mantener las unidades de E/S a plena capacidad.

— Coordinar las diferentes velocidades de trabajo del ordenador y de las unidades de E/S.

Para la conexión de unidades de control de los equipos de E/S se emplean generalmente canales *multiplex* (o *multiplexor*). (El multiplexor se utiliza también para la E/S on-line.)

El multiplexor es un interruptor que consulta continuamente a los canales. Cuando detecta un carácter lo transporta a un registro tampón de la UCP. Al mismo tiempo transporta la dirección del canal a la entrada de un sumador donde se calcula la dirección de la instrucción del canal. Este cálculo se efectúa sumando a la dirección del canal la dirección de la posición de memoria en donde se encuentran las instrucciones correspondientes a todos los canales del multiplex. La dirección hallada se envía al registro de direcciones (RCI) de la memoria central. A continuación se lee la instrucción y se transporta a través del registro tampón al registro de instrucciones.

EJERCICIOS

1. El registro acumulador se utiliza para

   a) Mover información desde la memoria a la U.A.L.

   b) Almacenar resultados parciales y finales

   c) Efectuar operaciones lógicas

2. En la suma el bit de acarreo

   a) No se almacena

   b) Se suma automáticamente

   c) Se almacena en un elemento de flip-flop y después se suma

3. ¿Qué circuito detecta que A < B?

   a) NO-O

   b) NO-Y

   c) COIN

4. ¿Cuál es la diferencia entre el funcionamiento síncrono y asíncrono?

   a) El primero no utiliza reloj y es más rápido

   b) El segundo utiliza reloj y es más rápido

   c) El primero utiliza reloj y es más rápido

5. La velocidad y el modo de operación depende:

   a) De la estructura de la instrucción que utilice

   b) De los periféricos de entrada

   c) De la U.A.L.

6. ¿Cuáles son los elementos básicos de las memorias tampones?

   a) Registros de instrucciones

   b) Soportes magnéticos

   c) Periféricos de baja velocidad

7. Un sumador, está compuesto por:

   a) Tres puertas O y una Y

   b) Tres puertas Y y una puerta O

   c) Tres puertas Y y EXOR

8. ¿Qué otro nombre recibe el registro contador de instrucciones?

   a) Registro decodificador

   b) Registro tampón

   c) Registro de dirección de instrucción

3.

FUNCIONAMIENTO DEL ORDENADOR (III)

Los sistemas operativos

CUADRO RESUMEN DE LA LECCION

FUNCIONAMIENTO DEL ORDENADOR III

3.1. Software del ordenador

3.2. Sistemas operativos

- Programas de control
  - Supervisor
  - Cargador inicial
  - Sistemas de control de E/S
  - Controlador de trabajos
  - Monitor
- Programas de proceso
  - Traductores de lenguajes
  - Programas de servicio
    - Cargador del sistema
    - Editor
    - Bibliotecario
    - Clasificación
    - Intercalación
    - De utilidad

3.1. SOFTWARE

Un sistema de proceso de datos es una combinación de los distintos componentes mecánicos, eléctricos y lógicos, es decir, elementos físicos (*hardware*) y lógicos (*software*).

Para la resolución de un problema son necesarios una serie de recursos físicos, como por ejemplo, la memoria de núcleos, los distintos dispositivos periféricos, los canales entre los periféricos y la memoria, etc. A estos recursos o componentes físicos se les llama *hardware* o *sistemas físicos.*

Además de estos componentes físicos, la ejecución de un problema necesita de recursos o componentes lógicos como pueden ser los programas especializados que permiten simplificar la programación del ordenador. El conjunto de estos programas se denomina *software.* Y a los diversos recursos de software utilizados para la resolución de un problema en un sistema de proceso de datos se les llama *sistemas de funcionamiento* o *sistemas operativos.*

El software realiza la mayoría de las operaciones de control del sistema ya que, actualmente, el hardware no puede realizarlas en su mayor parte. Sin embargo, la tendencia actual es que los dispositivos periféricos estén cada vez más capacitados para realizar operaciones de control, detección automática de errores, etc.

Los programas de software se realizan, generalmente, por el fabricante del ordenador y se suministran con el ordenador. En otros casos, los realizan empresas de servicios que los venden o alquilan. También en otras ocasiones, el propio programador de la empresa interesada es quien desarrolla los programas de software.

Los programas del software pueden ser de dos tipos:

— Programas de aplicación
— Programas de sistema

Los *programas de aplicación*, o *programas del usuario,* son los que ejecutan una aplicación determinada como por ejemplo: programas de facturación, de nómina, de contabilidad, etc.

Los *programas de aplicación* pueden ser de dos tipos: a) Programas escritos por el usuario y b) *PacKages de aplicación.*

Los PacKages de aplicación son programas que solucionan problemas en distintos sistemas efectuando algunas modificaciones. Son programas standard confeccionados por los constructores del sistema.

Los *programas de sistema,* llamados también *módulos del sistema,* están considerados como parte del sistema al igual que está considerada parte del sistema la memoria de núcleos, con la diferencia de que unos son sistemas operativos y los otros sistemas físicos.

La función de los programas de sistema es coordinar, supervisar y controlar las diversas unidades del sistema de proceso de datos. Estos programas ayudan a la tarea de confeccionar un programa de aplicación para la resolución de un problema y a controlar su ejecución.

Un sistema operativo (OS *operating system*) está formado por una serie de programas denominados *programas del sistema* que se encargan de controlar y supervisar el desarrollo de los programas de aplicación (programas del usuario).

En la ejecución de un programa, el sistema ha de resolver diversos problemas mediante la utilización de los programas del sistema: programas de organización, programas traductores de lenguajes, programas generadores de programas, programas de mantenimiento, etc.

Estos programas se clasifican en dos grupos: *programas de control* y *programas de proceso.*

3.2. SISTEMAS OPERATIVOS

Un sistema operativo es un conjunto de *programas de proceso* y *programas de control* creados para conseguir el máximo rendimiento del ordenador.

Mediante unas fichas de control preparadas por el ~~ordenador~~ operador (usuario) se le indica al sistema operativo que pase automáticamente de un trabajo a otro lo más rápido posible y con la mínima intervención del operador. En este caso, se establece la comunicación con el ordenador por medio del sistema operativo.

El sistema operativo permite al operador apilar trabajos para conseguir un proceso continuo de modo que se consiga mayor rapidez en la preparación de trabajos.

Un *trabajo* (JOB) puede considerarse como un conjunto de programas relacionados entre sí de forma lógica y que han de ejecutarse en una secuencia determinada.

El operador prepara unas fichas de control de trabajos que contienen la información necesaria para cada trabajo. El programa de control lee estas fichas que le indican en qué secuencia debe realizarlas. Estas fichas de control también contienen información sobre los dispositivos de E/S que se utilicen.

Los sistemas operativos pueden, además, compilar programas escritos en lenguajes de alto nivel, organizar ficheros y controlar o supervisar todo el sistema.

El sistema operativo se encarga, entre otras, de las siguientes funciones:

— Cargar los programas en la memoria.

— Iniciar la ejecución de los programas.

— Supervisar la ejecución de los programas.

— Asignar las posiciones de memoria.

— Asignar las unidades de E/S.

— Iniciar y controlar las unidades de E/S.

— Controlar los errores y las reanudaciones.

— Controlar las comunicaciones entre el operador y el sistema.

— Efectuar procesos de traducción.

— Tratar correctamente las prioridades.

— Cambiar de programa cuando se produce una interrupción.

3.2.1. Programas de control

Los *programas de control* supervisan la ejecución de los programas de proceso. Controlan el almacenamiento y la recuperación de la información, y planifican los trabajos para conseguir un proceso continuado.

Supervisor

El supervisor o programas de supervisión son los encargados de la gestión del sistema, de la gestión de los trabajos y de la gestión de los datos.

El objetivo del supervisor es la coordinación de la UCP con la gestión de los periféricos consiguiendo el máximo aprovechamiento del sistema.

En general, las funciones del supervisor son las siguientes:

— Tratamiento de errores de la memoria.

— Tratamiento de errores del programa.

— Control de interrupciones.

— Acoplamiento entre el proceso y las operaciones de E/S.

— Enlazar con los demás programas del sistema operativo.

En el caso de que se ejecuten varios programas al mismo tiempo (multiprogramación) pueden citarse además otras funciones, tales como:

— Control de los programas, de los periféricos y de los terminales.

— Control de los programas que se encuentran en la memoria.

— Asignación de memoria a los programas que se estén procesando o vaya a procesarse.

— Repartición del tiempo de proceso para los diferentes programas.

— Asignación de prioridades en la ejecución de trabajos.

El programa supervisor controla las operaciones de entrada/salida y las interrupciones que se producen en la ejecución de un programa. Estas interrupciones pueden estar motivadas por errores del programa, entradas incorrectas, errores del hardware, etc. Una vez corregidos los errores, el supervisor reanuda la ejecución del programa.

Las interrupciones pueden producirse, además, voluntariamente desde el exterior del sistema por medio de unas teclas especiales de que disponen las unidades del sistema. Pueden producirse interrupciones, también, cuando se inicia o finaliza una operación de entrada/salida.

El supervisor comienza la operación de E/S y devuelve el control al programa que se está ejecutando (programa problema). Cuando finaliza la operación,

el supervisor comprueba si existen condiciones de error del dispositivo de E/S y se encarga de ellas.

El supervisor dispone de unas rutinas de punto de control/reanudación que proporcionan un medio de registrar el estado del programa en puntos determinados para que más tarde continúe a partir de ese punto.

El programa problema continúa su proceso a partir de cada punto de control (*checkpoint*).

Algunas rutinas del supervisor se cargan en la memoria central cuando se inicia la operación del sistema. Otras rutinas del supervisor se transfieren a la memoria central desde memorias auxiliares siempre que sea necesario. De este modo, el supervisor puede realizar más funciones utilizando menos memoria.

Programa cargador inicial (IPL)

El programa *cargador inicial,* IPL (Initial Program Loader), se carga en la memoria y éste a su vez carga el programa supervisor cuando se inician las operaciones del sistema.

La carga del IPL puede efectuarse por medio de la consola del sistema accionando una tecla especial de carga.

Este programa carga el supervisor en la memoria cuando comienza la operación del sistema. El cargador marca su dirección en los conmutadores de carga que se encuentran en la consola del sistema, y pulsando la tecla de carga («load»).

Sistemas de control de entrada/salida (IOCS)

Los sistemas de control de entrada/salida, IOCS (Input/Output Control System) son programas que controlan las entradas y las salidas de los distintos dispositivos periféricos.

Estos programas se encargan de escribir en lenguaje máquina las operaciones necesarias para la lectura o escritura de los datos del programa.

El IOCS proporciona al programador la posibilidad de la programación de entrada/salida.

El IOCS se encarga de las siguientes funciones:

— Inicia o solicita las operaciones de E/S necesarias.

— Suministra registros de entrada al programa problema y recibe registros de salida del programa.

— Agrupa los registro en bloques.

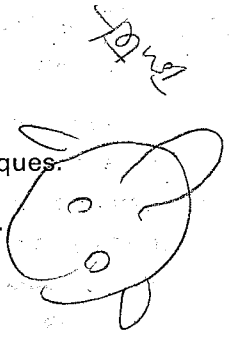
— Proporciona el tiempo necesario para el proceso de lectura y escritura de los registros.

— Trata las condiciones de fin de fichero.

— Controla los canales y las unidades de Entrada/Salida.

— Controla la transmisión de los registros agrupados en bloques.

— Controla la manipulación de los registros dentro del bloque.



Programa controlador de trabajos

El programa controlador de trabajos (Job Control) suministra varios trabajos y controla su ejecución en serie, uno tras otro. Este sistema de procesamiento se le denomina proceso *Batch.* De este modo, pueden haber varios programas en la memoria central para su procesamiento (*multiprogramación*).

El programa controlador de trabajos (Job Control) prepara al sistema para la ejecución del resto de los programas. El supervisor lo carga en memoria cuando se necesita.

Existe otro programa llamado *controlador de tareas* (Task Control) que suministra diferentes programas a la vez en los sistemas de proceso de tiempo compartido (*Time sharing*).

En los sistemas de proceso *batch* (proceso por lotes) los programas comparten la memoria y se ejecutan uno tras otro, mientras que en los sistemas de tiempo compartido los programas se reparten el tiempo total de ejecución de manera que en un momento determinado se está ejecutando un solo programa. Cuando pasa el tiempo destinado a su ejecución se comienza otro programa y así sucesivamente hasta que vuelve el control al primero.

Monitor

El *monitor* es el programa o conjunto de programas encargado de interpretar, iniciar y finalizar los diferentes trabajos. Además, encadena los programas de acuerdo con las instrucciones o fichàs de control.

El monitor llama al programa cargador inicial o al *editor* (*linkage editor*) que es un programa con las mismas funciones que el cargador, es decir, almacenar los programas en la memoria.

Mediante el monitor, el operador puede establecer una comunicación con el ordenador.

En multiprogramación, el monitor se encarga de la programación de los turnos de espera, debiendo elegir el trabajo a iniciar según el orden de prioridad, y de almacenar los resultados en memorias auxiliares.

3.2.2. Programas de proceso

Además de los programas de control, los sistemas operativos disponen de un conjunto de programas que facilitan el proceso de los trabajos. Estos programas, llamados *de proceso,* se clasifican en:

— Programas traductores de lenguajes

— Programas de servicio

Los programas de aplicación (escritos por el usuario) pueden considerarse también como programas de proceso, aunque se hayan tratado anteriormente como no incluidos en el conjunto de programas del sistema operativo.

Traductores de lenguajes

Los programas traductores son los que aceptan instrucciones codificadas en un lenguaje de programación y producen instrucciones equivalentes en lenguaje máquina.

Uno de estos programas es el *compilador,* el cual traduce el programa fuente (escrito en un lenguaje simbólico de alto nivel) a *programa objeto* (escrito en lenguaje máquina).

Programas de servicio

Los programas de servicio del sistema operativo realizan determinadas funciones que pueden utilizarse para efectuar un proceso sin tener que programarlas cada vez que se necesiten.

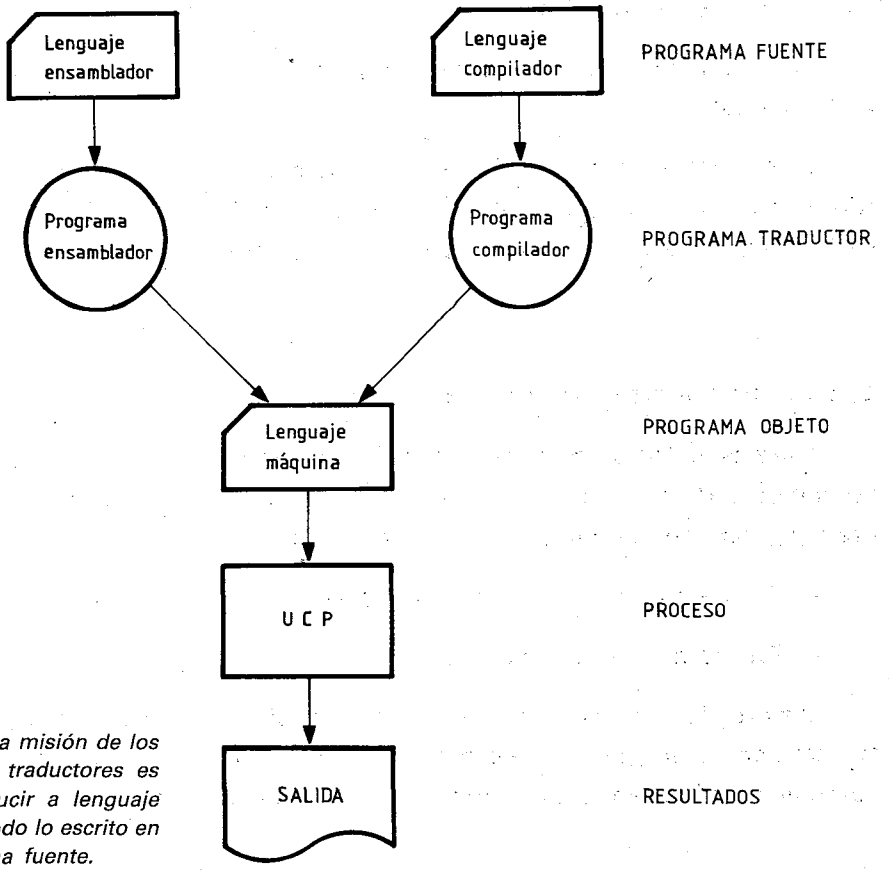


Fig. 3.1. La misión de los programas traductores es la de traducir a lenguaje máquina todo lo escrito en el programa fuente.

Los programas de servicio se clasifican en dos grupos:

— De funcionamiento del sistema

— De tratamiento de fichero

En el grupo de programas de funcionamiento del sistema están incluidos programas tales como:

— Cargador del sistema

— Editor (o Linkage Editor)

— Montador

— Bibliotecario (programa de manejo de bibliotecas)

En el grupo de programas de tratamiento de ficheros se incluyen programas de:

— Clasificación

— Intercalación

— Utilización de ficheros (programas de utilidad)

Programa cargador del sistema

Este programa carga en la memoria central los programas de sistema que son necesarios para la ejecución del problema y que están almacenados en las bibliotecas del sistema operativo. Estos programas que se cargan desde la biblioteca a la memoria central han de estar escritos en lenguaje máquina, y si no lo están se han de traducir o compilar antes de ser cargados en memoria.

Linkage editor

Muchos sistemas operativos disponen de programas traductores que en el momento de la traducción no obtienen programas directamente cargables, sino programas que deben someterse a una operación de *encuadernación* antes de que puedan funcionar. La encuadernación la efectúa el programa *editor* (*linkage editor*) y consiste en reunir en un programa partes de programas desarrollados independientemente. De esta forma se pueden hacer funcionar, como un sólo programa, partes de programa traducidas en diferentes momentos.

Montador

El *programa montador* convierte la salida de los traductores en módulos que puedan cargarse y ejecutarse. Este programa enlaza partes de programas traducidos por separado e incluye módulos desde la biblioteca de programas.

Programas bibliotecarios

Para las bibliotecas del sistema operativo existen unos *programas bibliotecarios* que se encargan de controlarlas. Estos programas dan información al usuario sobre la dirección de los programas o módulos de la biblioteca, pueden introducir nuevos programas en la biblioteca, etc.

Programas de clasificación (SORT)

Estos programas se encargan de ordenar y clasificar los datos de un fichero que no esté ordenado previamente y que se vaya a utilizar en la ejecución de un problema.

Programas de intercalación (MERGE)

Estos programas se utilizan para conseguir un fichero ordenado secuencialmente a partir de otros dos ficheros de entrada que estén también ordenados. Esta operación consiste en intercalar los datos de ambos ficheros de entrada siguiendo un criterio de clasificación o de orden.

Programas de utilidad

Los *programas de utilidad* se encargan de la conversión de un conjunto de datos de un soporte de información a otro. Por ejemplo, de ficha a cinta, de cinta a disco, etc.

Otros programas de utilidad sirven para ayudar la operación del sistema. Por ejemplo, los programas de comparación de registros.

EJERCICIOS

1. Un sistema operativo está formado por:
   a) Programas del sistema
   b) Programas de aplicación
   c) Programas del usuario

2. ¿Quién supervisa la ejecución de los programas de proceso?
   a) Los programas del usuario
   b) Los programas de control
   c) Los programas de entrada/salida

3. ¿Quién se encarga de la coordinación entre la UCP y los periféricos?
   a) Programa de control
   b) Supervisor
   c) El compilador

4. ¿Qué programa traduce el programa fuente a programa objeto?
   a) El supervisor
   b) Editor
   c) El compilador

5. ¿Quién efectúa la operación de encuadernación?

   a) El programa cargador del sistema

   b) El linkage editor

   c) Merge

6. ¿Qué significa SORT?

   a) Programas de utilidad

   b) Programas de intercalación

   c) Programas de clasificación

7. El control de las interrupciones en la ejecución de un programa está a cargo del programa:

   a) IPL

   b) Supervisor

   c) IOCS

8. El programa IOCS se encarga de:

   a) Cargar el supervisor en la memoria

   b) Iniciar las operaciones de E/S

   c) Tratar los errores de la memoria

4.

FUNCIONAMIENTO DEL ORDENADOR (IV)

Métodos de proceso Teletratamiento

CUADRO RESUMEN DE LA LECCION

FUNCIONAMIENTO DEL ORDENADOR IV

4.1. Métodos de funcionamiento

- Proceso por lotes
  - Proceso por lotes sin prioridades
  - Proceso por lotes con prioridades
  - Proceso por lotes remoto
- Tiempo compartido
  - Reparto del tiempo de proceso
  - Partición de la memoria
- Memoria virtual
- Multiprogramación
- Tiempo real
- Multiproceso

FUNCIONAMIENTO DEL ORDENADOR IV

4.2. Sistemas de teletratamiento

- — Ordenador central
- — Terminales
  - — Líneas de comunicación
  - — Controlador de líneas
  - — Concentradores
  - — Multiplexores
  - — Modems
- — Vías de transmisión
- — Tipos de transmisión
  - — Símplex
  - — Semidúplex
  - — Dúplex

4.1. METODOS DE FUNCIONAMIENTO

Con el fin de obtener el máximo aprovechamiento del ordenador son necesarias determinadas funciones de mando y control del sistema de proceso de datos. Estas funciones son realizadas por el sistema operativo del ordenador.

Las funciones del sistema operativo vienen condicionadas frecuentemente por el método de funcionamiento del sistema de proceso de datos. Algunos de estos métodos de tratamiento son: Proceso por lotes (Batch-Processing), tiempo compartido (Time Sharing), tiempo real (Real Time), Multiprogramación, etc.

4.1.1. Proceso por lotes

El *proceso por lotes* (*Batch-Processing*), o proceso de datos convencional, consiste en tratar sucesivamente, por etapas, varios programas.

En el proceso por lotes puede ocurrir que se trate un solo programa compuesto de varios trabajos independientes. En este caso, se carga el programa en la memoria y estos trabajos se procesan uno tras otro sin tener que volver a cargar el programa. (A estos trabajos se les denomina, frecuentemente, *jobs*.)

Cuando uno de los trabajos ha sido procesado totalmente, la unidad central de proceso y las unidades periféricas quedan disponibles para procesar el siguiente trabajo y así sucesivamente hasta que se han procesado todos los trabajos que componen el programa.

Por ejemplo, supongamos un programa de nómina que trata el sueldo de cada empleado individualmente. En primer lugar, se recopilan los datos necesarios para efectuar el trabajo.

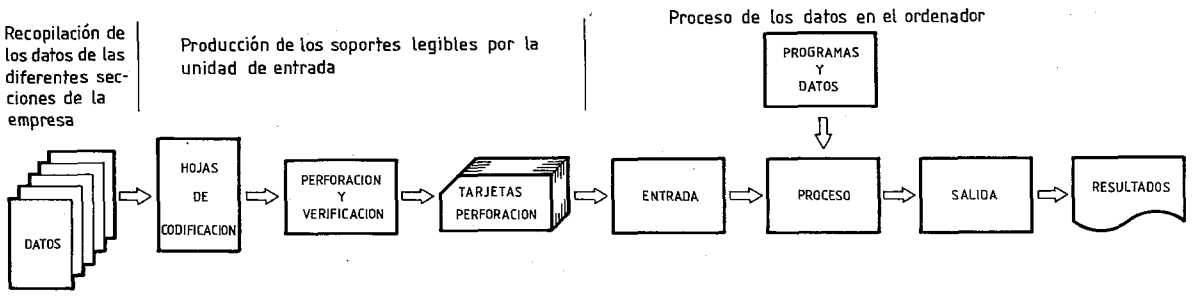


Fig. 4.1. Proceso seguido por un trabajo (Job) para su tratamiento por ordenador.

Una vez recopilados los datos, éstos se transcriben en soportes de información legibles por la unidad de entrada del ordenador (por ejemplo, mediante tarjetas perforadas).

Cuando se introducen los datos en el ordenador no se procesan inmediatamente sino que se necesita cierto tiempo para procesar, uno tras otro, la totalidad de los trabajos. Hay que tener en cuenta que un trabajo debe estar completamente procesado antes de comenzar con el trabajo siguiente. Es decir, el proceso por lotes es un proceso que se realiza por etapas.

En la salida de resultados también se produce una espera. Los resultados pueden obtenerse, por ejemplo, mediante la impresora o a través de memorias de gran capacidad, como son los discos magnéticos. Si los resultados van a obtenerse por impresora, puede ocurrir que ésta no se encuentre disponible, por lo que se producirá un tiempo de espera hasta que pueda utilizarse.

En el proceso por lotes pueden distinguirse diferentes tipos de funcionamiento:

— Proceso por lotes sin prioridades

— Proceso por lotes con prioridades

— Proceso por lotes remoto

Proceso por lotes sin prioridades

Si en el método de funcionamiento no se ha establecido ningún orden de preferencia o de prioridad, los programas o jobs se procesan en el orden que se introducen en la unidad central. Las salidas se obtienen en el mismo orden del de llegada.

Supongamos, por ejemplo, un programa almacenado en la memoria central dividido en tres trabajos (jobs) independientes J1, J2 y J3.

En primer lugar se introduce, desde la unidad de entrada a la unidad central, una parte del job J1 y se procesa en la UCP. Después se introduce otra parte del job J1, se procesa y se obtiene una parte de los resultados por la

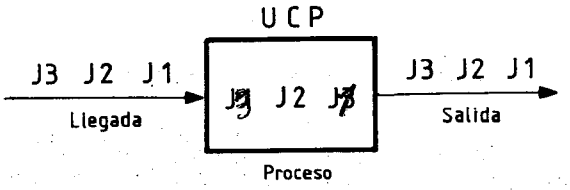


Fig. 4.2. En un proceso por lotes sin prioridades los trabajos se ejecutan por riguroso orden de llegada.

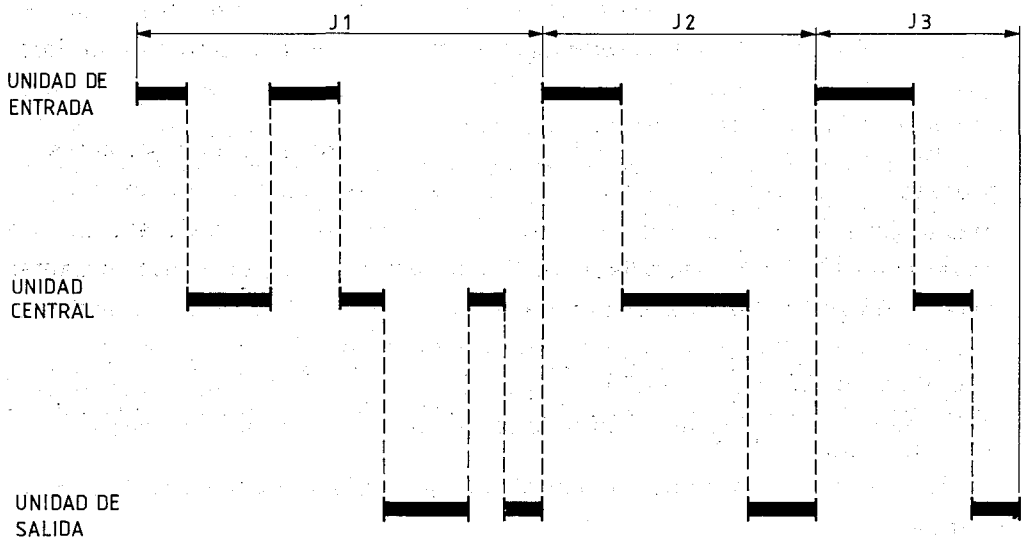


Fig. 4.3. Secuencia de tiempos útiles y tiempos de espera de cada una de las unidades, en un proceso por lotes sin prioridades.

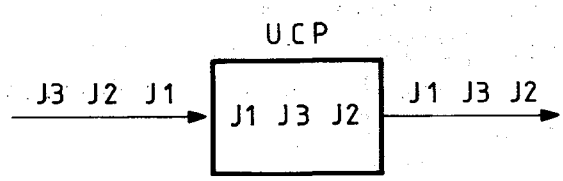


Fig. 4.4. Cuando concurren varios trabajos a la vez, con orden de prioridad marcado, el procesamiento y salida se efectúa cumplimentando dichas prioridades.

unidad de salida. A continuación, se procesa el resto del job J1 y se obtiene el resultado. A partir de ese momento se introduce el job J2, se procesa y una vez obtenido el resultado se repite la misma operación con el job J3.

Proceso por lotes con prioridades

Generalmente, se aplican prioridades en el proceso de programas cuando se procesan varios programas al mismo tiempo, utilizando el tiempo de espera de la UCP, empleado en operaciones de entrada/salida de un programa que está ejecutándose, para ejecutar parte de otro programa hasta que el primer programa finalice sus operaciones de E/S (este método se denomina *multiprogramación*).

Además de utilizarse en multiprogramación, existen también sistemas de proceso por lotes en los que se tienen en cuenta las prioridades de los programas a ejecutar. En este caso, los programas no se procesan según el orden de llegada, sino según el orden de prioridades. De forma que se procesa en primer lugar el programa con mayor prioridad.

Por ejemplo, si el orden de prioridad de los jobs es: J2>J3>J1. Cualquiera que sea el orden de llegada, la salida será: primero el J2, después el J3 y por último el J1.

Proceso por lotes remoto

En el *proceso por lotes remoto,* la entrada y la salida de los programas y datos se efectúa a través de terminales, situados a distancia, por medio de líneas de transmisión de datos.

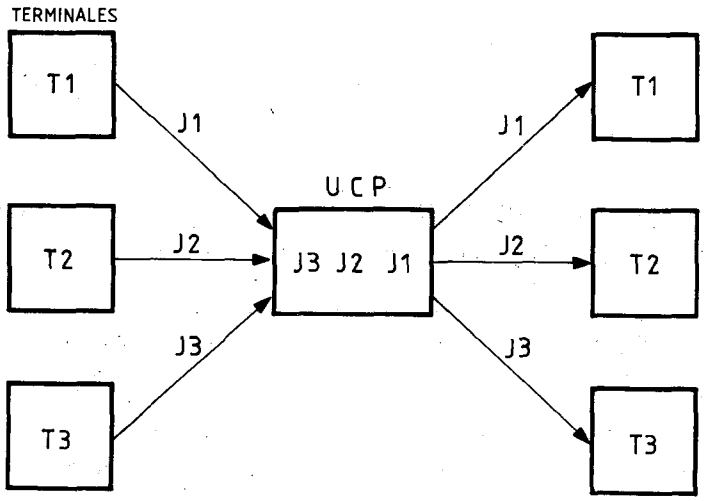


Fig. 4.5. Utilización de la UC en un proceso por lotes remoto.

Con este sistema, muchos usuarios pueden utilizar el mismo ordenador desde sus terminales situados, generalmente, lejos del ordenador central.

A cada uno de los terminales se le puede asignar una cierta prioridad. De esta forma, algunos de los usuarios puede ser que tengan que esperar un tiempo considerable hasta obtener los resultados.

4.1.2. Tiempo compartido

El método de *tiempo compartido* (Time Sharing) consiste en utilizar simultáneamente el ordenador por varios usuarios independientes. Cada usuario puede resolver un determinado problema usando sus propios programas y mediante unidades de entrada/salida independientes.

Así pues, el sistema de tiempo compartido consiste en la utilización de un ordenador desde varios terminales.

Se entiende por terminal el dispositivo por el que los datos pueden introducirse o extraerse del sistema. Generalmente, la palabra «terminal» se refiere a

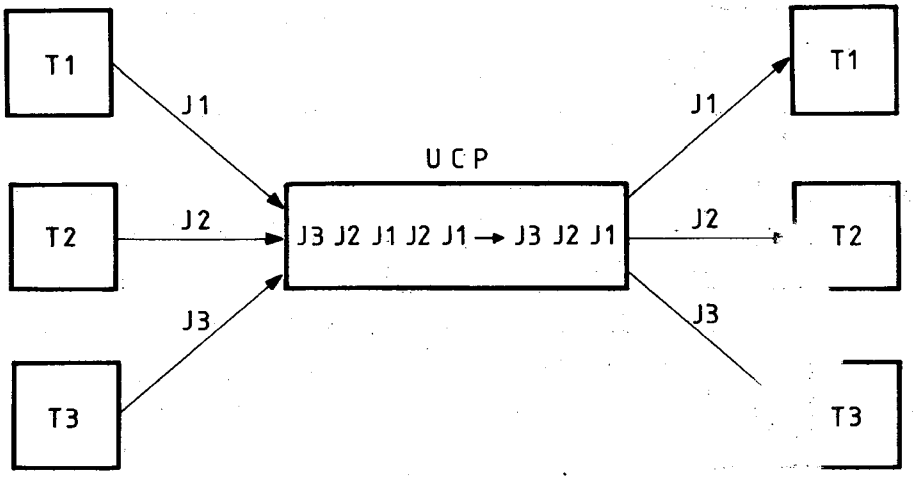


Fig. 4.6. Utilización de la UC en un proceso con tiempo compartido.

un dispositivo por el que se introducen o reciben datos en un extremo de una línea de transmisión de datos.

Por lo tanto, un ordenador que trabaje en tiempo compartido puede ser utilizado por varios usuarios simultáneamente. Para ello, cada usuario dispone de un terminal a distancia.

Los componentes principales de un sistema de tiempo compartido son los siguientes:

— Una unidad central para cálculos de alta velocidad.

— Un fichero de acceso directo para el almacenamiento de información.

— Un supervisor de E/S para controlar las comunicaciones.

— Líneas de transmisión de datos (líneas telefónicas para largas distancias y cables coaxiales para distancias cortas).

— Terminales con diversos recursos de E/S.

Los sistemas de tiempo compartido permiten el completo aprovechamiento de un cierto número de terminales remotos conectados directamente a la

unidad central de proceso. Para los usuarios supone la posibilidad de obtener la respuesta inmediata a sus peticiones de trabajo.

Al igual que en el proceso por lotes remoto, en un sistema de tiempo compartido se acoplan varios terminales a un ordenador central. En realidad, cada terminal dispone periódicamente del ordenador durante un cierto intervalo de tiempo. La elevada velocidad de trabajo del ordenador permite que el usuario se considere en contacto continuo con el sistema de proceso de datos, dándole la impresión de que sólo trabaja para él. Así pues, el método de funcionamiento de tiempo compartido requiere un ordenador que disponga de una elevada velocidad interna de trabajo.

Los sistemas de tiempo compartido necesitan dispositivos para la entrada y salida directa de la información. Para ello se utilizan canales multiplexores que permiten la entrada simultánea de información sobre un conjunto de terminales.

Algunos de los terminales más utilizados son el teletipo y los terminales de pantalla y teclado.

El tiempo compartido tiene la finalidad de reducir el tiempo que transcurre entre la entrada de un trabajo hasta su terminación o salida de los resultados, además del máximo aprovechamiento del ordenador.

En resumen, un sistema de tiempo compartido debe realizar principalmente dos tareas:

— Controlar las comunicaciones con varios usuarios.

— Ejecutar los programas de los diversos usuarios.

Para ofrecer un servicio simultáneo a varios usuarios se utilizan memorias auxiliares. Sin embargo, en un cierto instante, sólo se está ejecutando el programa de un usuario.

El usuario debe tener conectado su terminal con el *controlador de comunicaciones.* Cuando se encuentra disponible una entrada, se inicia la utilización simultánea.

El número de terminales conectadas a un sistema de tiempo compartido queda limitado por el número que pueda atender el controlador de comunicaciones o por la velocidad de la UCP.

Cuando varios usuarios solicitan el proceso en la unidad central al mismo tiempo, el monitor sitúa los programas en cola y el proceso se efectúa según el orden de prioridades.

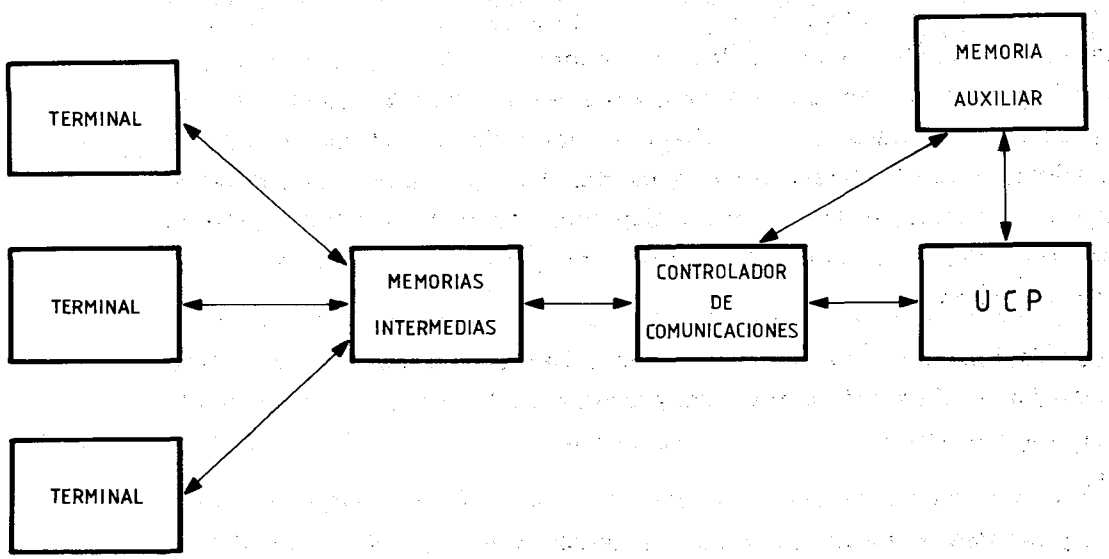


Fig. 4.7. Proceso con tiempo compartido.

Reparto del tiempo de proceso

La UCP procesa los diferentes programas de los usuarios durante ciertos intervalos de tiempo. El paso de un programa a otro se efectúa por medio de interrupciones o por reloj. De forma que cuando se ha dedicado un cierto tiempo a un programa se pasa al siguiente.

En tiempo compartido los programas no pueden estar todos a la vez en la memoria. Cuando se está procesando un programa y se va a procesar otro en un determinado instante, se sustituye el primero sin finalizar por el nuevo que procede de la memoria auxiliar.

En muchos sistemas de tiempo compartido se utilizan dos ordenadores acoplados, de forma que uno de ellos supervisa las comunicaciones y asigna el tiempo a cada terminal resolviendo las prioridades, mientras que el otro se encarga de la ejecución de los programas.

En otros sistemas superiores de tiempo compartido las funciones anteriores las realiza un solo ordenador, aunque utilizando varias unidades de proceso.

Para que el tiempo de proceso por usuario y para que cada uno de los usuarios dispongan de las mismas prioridades, se efectúa una repartición del tiempo total de proceso, proporcionando a un usuario el mismo intervalo de tiempo que al resto de los usuarios.

En la mayoría de los casos los usuarios no utilizan completamente su intervalo de tiempo asignado, ello representa disponer de más tiempo para poder realizar el proceso por lotes. Así, un usuario que tenga que realizar un trabajo de gran volumen generalmente puede disponer de un tiempo mayor que el que le han asignado.

La diferencia entre este sistema y el sistema de multiprogramación consiste en que el sistema operativo no puede determinar, en el caso del tiempo compartido, cuántos programas van a procesarse y cuál es el tiempo asignado a cada uno de ellos.

El número de programas que pueden estar procesándose simultáneamente es mucho mayor en tiempo compartido que en multiprogramación.

Partición de la memoria

Otro de los problemas que se presentan es el de la repartición de la memoria central entre los diversos usuarios.

Uno de los métodos de partición de la memoria consiste en que, durante la ejecución de un determinado programa, se produce la carga en memoria de los programas que, a continuación, poseen mayor prioridad. Así, mientras un programa se está ejecutando, existen otros disponibles en la memoria. Cuando el primer programa termine con su tiempo de proceso pasará el control al programa con mayor prioridad almacenado en la memoria, en ese momento otro programa ocupará la zona de memoria que ha quedado libre. Este proceso se repite hasta que se ejecuten todos los programas.

Para la carga y descarga de los programas en la memoria central se exige una gran velocidad de transferencia desde la memoria central a las memorias auxiliares y viceversa.

Otro procedimiento de partición de la memoria consiste en dividir la memoria disponible en zonas parciales, denominadas *páginas.*

Las páginas son de dimensión fija y constan de 1 K palabra a 4 K palabras, siendo cada palabra de 4 octetos, generalmente.

Las direcciones de las instrucciones, en este método, deben contener el número de página y el número de la palabra dentro de cada página.

Los programas de los diferentes usuarios se encuentran en memorias auxiliares de gran capacidad dispuestos en páginas. Y en la MC se reserva espacio para el mismo número de páginas de cada usuario.

Cuando se está procesando un programa, sólo una parte se encuentra ocupando la memoria central, produciéndose frecuentemente transferencias de control a páginas todavía no cargadas en la MC.

Utilizando este sistema de partición de la memoria se requieren memorias auxiliares (de gran capacidad) muy rápidas y con tiempos de acceso muy pequeños.

Un ejemplo de la utilización de la partición de la memoria en páginas es el sistema de *memoria virtual,* que sirve para la ampliación de la memoria central del ordenador.

4.1.3. Memoria virtual

El sistema de memoria virtual se utiliza cuando la memoria central de un ordenador no es suficiente para almacenar completamente la información del programa o programas que están procesándose.

Así pues, con la memoria virtual se evita el tener que aumentar la capacidad de la memoria central. El método consiste en utilizar *segmentaciones* (que son realizadas por los compiladores), es decir, los programas se dividen en trozos (*segmentos* o *páginas*) de un determinado tamaño y la Memoria Central también se estructura en trozos (llamados *áreas de solapamiento*) sobre los que se van cargando los segmentos del programa que vayan a procesarse en un momento determinado.

El Sistema Operativo va pasando el programa, a trozos (por *páginas*), por la memoria central.

De esta forma, pueden estar ejecutándose varios programas o varios segmentos de programas al mismo tiempo.

El programa monitor carga en la memoria central los segmentos de los programas que en un determinado instante se necesiten, mientras que el resto está disponible en una memoria auxiliar (de acceso directo).

Supongamos un ordenador con una memoria central de 16 K palabras. Dispone además de un tambor magnético de 64 K palabras. La memoria y el tambor se dividen en *páginas,* siendo cada página igual a 4 K palabras. Cada página de la MC se numera desde la 1 a la 4, y cada página del tambor se numera de 1 a 16. Cada palabra de una página se numera de 1 a 4096.

De esta forma cada palabra se identifica por el número de página a que pertenece y por su número de palabra dentro de la página. Así por ejemplo, 10001 hará referencia a la primera palabra de la página 1, 10002 será la segunda palabra de la página 1, 40003 será la tercera palabra de la página 4, etc.

Para un programa que necesite 64 K palabras, el ordenador (con una capacidad de memoria de 16 K palabras) puede hacer uso de este tambor de 64 K palabras (o de más capacidad).

En primer lugar, el programa se almacena en el tambor, ocupando las 64 K palabras (es decir, las 16 páginas). En el momento de la ejecución del programa, el ordenador lee las 3 primeras páginas del tambor en las 3 primeras páginas de la MC (no es conveniente ocupar completamente la memoria central, ya que es mejor reservar una página para operaciones). Las páginas se introducen por lectura desde el tambor a la memoria central. Asimismo, las páginas de la memoria también se devuelven al tambor mediante lectura.

Por lo tanto, un ordenador que dispone solamente de una memoria central de 16 K palabras (de las cuales utiliza 12, es decir, de 4 páginas utiliza sólo 3)

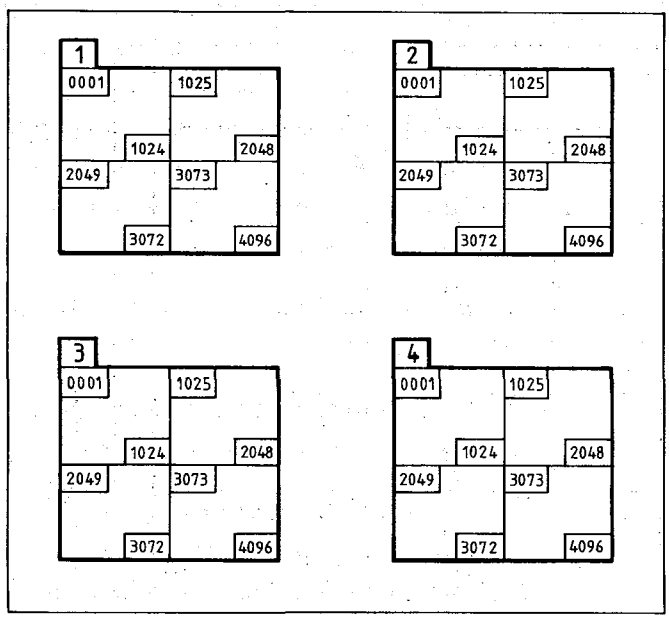
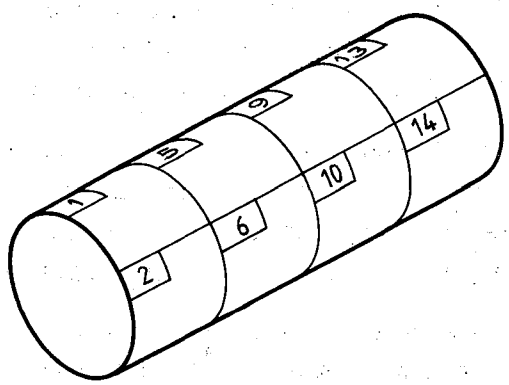
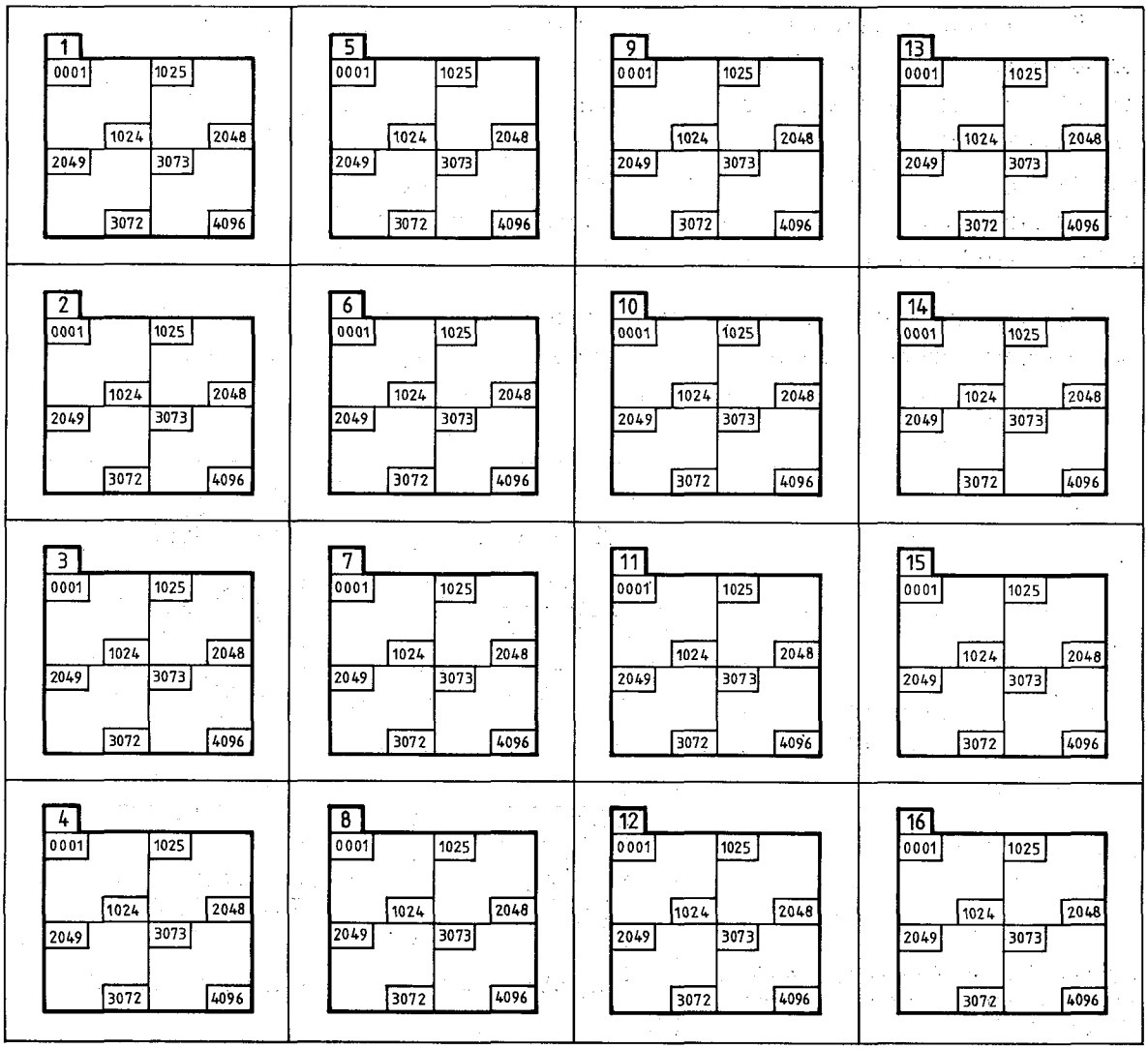


Fig. 4.8. Capacidad de la memoria central en el ejemplo que se está estudiando.

Fig. 4.9. Capacidad del tambor magnético, o menoria auxiliar, correspondiente al mismo ejemplo. ▶ ▼





puede procesar un programa que necesita 64 K palabras utilizando la técnica de la memoria virtual (mediante un tambor magnético de 64 K palabras). Es decir, el programador dispone de un espacio de 64 K palabras para ejecutar su programa.

4.1.4. Multiprogramación

La *multiprogramación* es un método de funcionamiento del ordenador en el que el sistema operativo controla el desarrollo simultáneo de varios programas.

En este método de funcionamiento, varios trabajos (jobs) comparten la utilización de la Unidad Central de Proceso. La memoria central debe esperar a que finalicen las operaciones de E/S para poder tratar la información introducida. Así pues, se dispone de tiempo libre para el tratamiento interno de otro trabajo.

La multiprogramación tiene por objeto el de obtener el máximo rendimiento del ordenador. En la mayoría de los procesos, la UCP se encuentra en espera de operaciones de E/S durante ciertos intervalos de tiempo. Para ocupar este tiempo disponible, o tiempo de espera, se utilizan los *canales* con el fin de conseguir una sincronización entre las operaciones del proceso y las de entrada/salida. De esta forma, mientras se realiza una operación de E/S bajo el control de una canal, la UCP pueden continuar trabajando. Así pues, cuando la UCP ha terminado con una determinada tarea, no tiene que esperar (o tiene que esperar mucho menos) a que se efectúe la operación de E/S.

Con la multiprogramación se consigue aumentar en gran proporción las velocidades de operación. Se consigue ocupar la máxima capacidad de la Unidad Central de proceso y pueden ejecutarse varios programas a la vez, pasando el control de un programa a otro sin necesidad de tener a la UCP en espera.

Cuando se produce una interrupción en el proceso, a causa de una petición de E/S de uno de los programas almacenados en la memoria, se transfiere el control a otro programa y éste se procesa hasta que se produzca una nueva interrupción.

Supongamos que se van a procesar tres trabajos (jobs), J1, J2 y J3, con el siguiente orden de prioridades: J1 > J2 > J3. El orden de prioridades de los jobs se refiere al orden de proceso en la unidad central.

Un job dejará libre la UCP cuando pida una unidad de E/S, o bien cuando un job de mayor prioridad solicita la UCP.

Como J1 tiene mayor prioridad que J2, la unidad central estará disponible para J2 hasta que J1 haya finalizado la operación de salida y solicite de nuevo la UCP.

El ordenador ejecuta, en primer lugar, cierto número de instrucciones del primer programa (J1) de los tres que se encuentran en la memoria central. Luego pasa el segundo programa (J2) para ejecutar una serie de instrucciones. Después vuelve al primer programa J1 para ejecutar algunas instrucciones más, y así sucesivamente hasta completar los tres programas o jobs.

La finalidad de la multiprogramación reside, entre otras cosas, en la posibilidad de que muchos usuarios compartan el mismo ordenador, manteniendo en activo todo el equipo físico y eliminando al máximo el tiempo de espera. La multiprogramación permite, además, que el ordenador responda en un tiempo

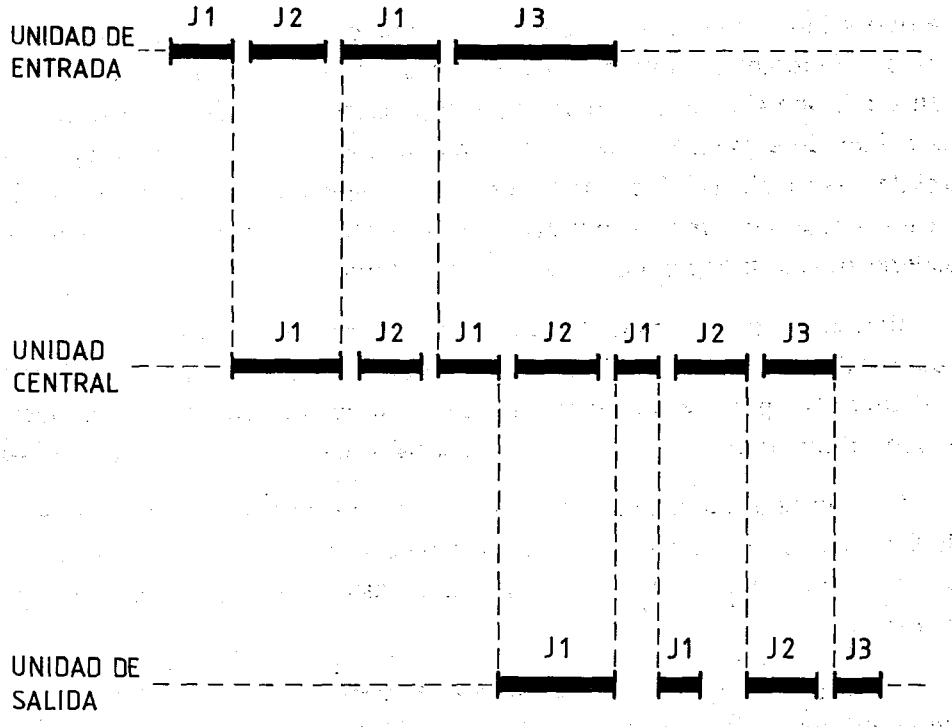


Fig. 4.10. Secuencia de trabajo del ordenador en un proceso de multiprogramación, según su estado de prioridad.

límite (*tiempo real*) cuanto se le pregunte. El tiempo de respuesta es del orden de unos pocos milisegundos.

El ordenador pasa de un trabajo a otro operando con intervalos de tiempo fijos, planificados y siguiendo un orden.

4.1.5. Tiempo real

Se dice que un sistemas trabaja en *tiempo real* cuando procesa una determinada información y proporciona la respuesta en un tiempo mínimo (tiempo de respuesta).

Los sistemas en tiempo real efectúan su respuesta en segundos o milisegundos.

Un sistema en tiempo real controla un proceso en marcha y proporciona sus salidas (o entradas de control) en el instante en que se necesiten para el control del sistema.

El tiempo real se refiere a la duración real del tiempo en que transcurre un acontecimiento.

Un sistema de tiempo real es el sistema de control de entrada/salida (IOCS). Este sistema efectúa la respuesta a una interrupción siempre que un periférico finaliza alguna tarea. El sistema IOCS, por lo tanto, controla un proceso en marcha (proceso de E/S) y debe proporcionar las entradas antes que continúe el proceso.

Si el tiempo de proceso no puede sobrepasar algunos segundos entonces debe usarse un sistema en tiempo real.

El tiempo de respuesta (*response time*) del sistema de proceso de datos debe ser muy pequeño, por lo que los programas que se traten en tiempo real poseen la máxima prioridad, pudiendo interrumpir en cualquier momento otro programa que se esté ejecutando en el ordenador.

El tiempo real (*real time processing*) se utiliza frecuentemente en combinación con sistemas de tiempo compartido y multiprogramación.

El tiempo real se utiliza en sistemas de interrogación, en los cuales se consultan las informaciones almacenadas en un fichero de información según determinados criterios de búsqueda. Después de la consulta, el fichero permanece con la misma información, efectuándose la actualización periódicamente utilizando un sistema batch (proceso por lotes).

En otros casos, en los que se utilizan los sistemas en tiempo real, se producen alteraciones en los ficheros al efectuar una consulta. Por ejemplo, en un sistema de reserva de plazas de transporte aéreo. En las oficinas de despacho de billetes se encuentran instalados los terminales correspondientes.

Si se pide información desde un terminal sobre un determinado vuelo, la respuesta se recibe en unos segundos. En esos momentos sí había solamente una plaza disponible y se efectúa la reserva correspondiente, se actualiza el fichero y en la siguiente consulta, la información recibida sobre ese vuelo indicará que ese avión está completo.

Este sistema trabaja en tiempo real, puesto que la información enviada desde el terminal la procesa y proporciona el resultado inmediatamente.

En general, en un sistema de tiempo real, la información debe ser aceptada por el ordenador en el momento en que se le suministra dicha información.

4.1.6. Multiproceso

El *multiproceso* consiste en la utilización de varias unidades de proceso funcionando simultáneamente y compartiendo memorias centrales y auxiliares.

Los trabajos a tratar se comparten entre las distintas unidades de proceso, pasándose de unas a otras cuando sea preciso.

En un sistema de multiproceso varios procesos poseen un acceso simultáneo a la memoria central del ordenador.

Los diferentes procesos trabajan independientes entre sí en varios programas (multiprogramación).

La elevada capacidad de proceso de un sistema de multiproceso se debe a la implantación múltiple de ciertos componentes. Por ejemplo, varias unidades aritméticas y unidades de control están comunicadas con una memoria central a través de unas líneas de transmisión de datos. De esta forma, esta memoria central se encuentra a disposición de varios sistemas de proceso de datos al mismo tiempo.

Para obtener mayor seguridad y mejor funcionamiento en un sistema de multiproceso, es necesario que las unidades funcionales individuales tengan un funcionamiento independiente entre sí. De esta forma, cualquier unidad en la que se produzca una avería puede ser reemplazada por otra que realice las mismas funciones.

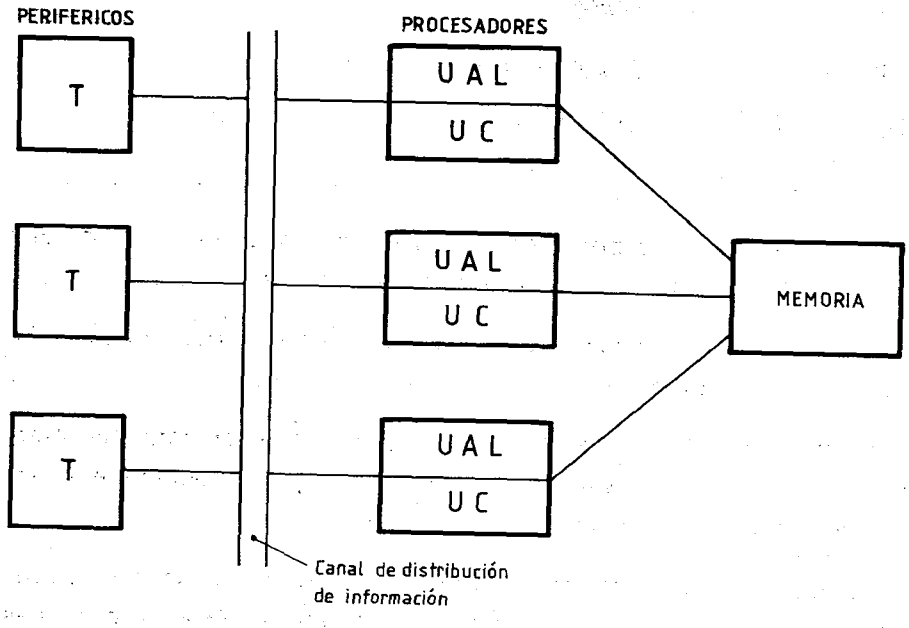


Fig. 4.11. Sistema de multiproceso con instalación múltiple de algunos componentes.

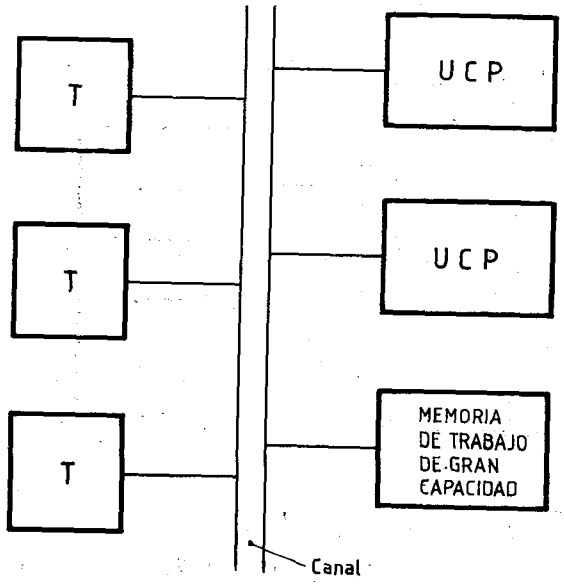


Fig. 4.12. Sistema de multiproceso en el que cada uno de los módulos puede funcionar individualmente.

Así pues, en multiproceso es necesario que cada módulo del sistema pueda funcionar individualmente.

Existen tres tipos de módulos:

— Módulo de proceso (procesador)

— Módulo de memoria (M.C.)

— Módulo de E/S

El *módulo de proceso* o *procesador* (*processor*) está compuesto por una unidad de control y por una o varias unidades aritméticas/lógicas.

El procesador es un dispositivo controlado por el ordenador y que no dispone de memoria central ni de unidades de entrada/salida.

El *módulo de memoria* o *memoria central* posee un funcionamiento independiente. Si existen varios módulos de memoria, puede accederse independientemente a cada uno de ellos.

El *módulo de entrada/salida* está compuesto por todos los dispositivos para unir las distintas unidades de control de las unidades de E/S y las memorias

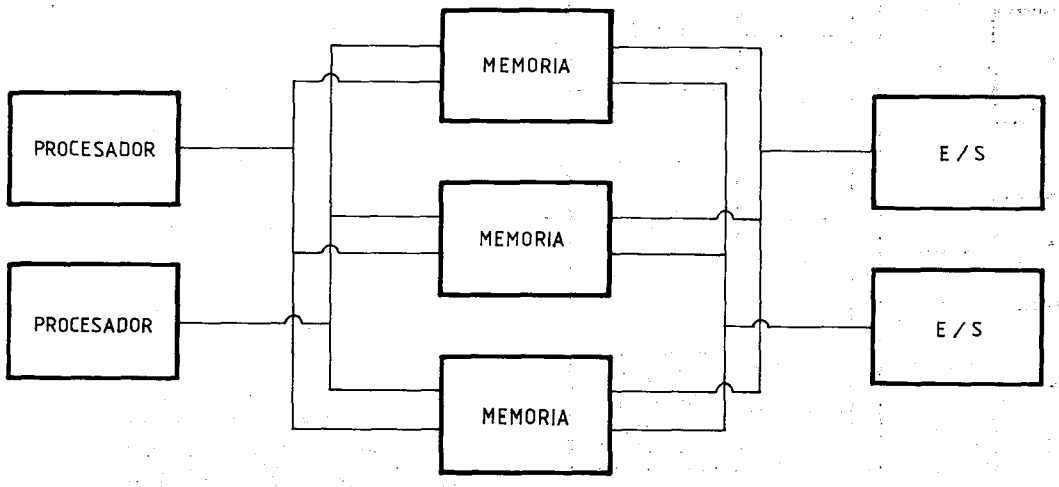


Fig. 4.13. Conexión múltiple entre los módulos componentes de un sistema de multiproceso.

de gran capacidad del sistema de proceso de datos. Algunos de estos dispositivos son los canales selectores, los canales multiplexores, etc.

Los procesadores y los módulos de entrada/salida tienen acceso a cada uno de los módulos de memoria, es decir, en un cierto momento pueden acceder a cualquier módulo de memoria.

En algunos sistemas de multiproceso, están conectados entre sí una serie de ordenadores completos.

Estos ordenadores pueden estar conectados mediante sistemas de transmisión de datos a distancia, constituyendo lo que se llama una *red de multiproceso* o red de ordenadores.

En el caso de que los ordenadores interconectados se encuentren físicamente próximos entre sí, se dice que constituyen un sistema *multiplex.*

Las redes de ordenadores se caracterizan porque las instalaciones individuales están en lugares muy separados y, por lo tanto, se conectan mediante unidades de transmisión de datos a distancia.

4.2. SISTEMAS DE TELETRATAMIENTO

En un sistema de *teletratamiento,* es decir, que en un sistema de tratamiento de datos a distancia, se utilizan para la transmisión de los datos los medios actuales de comunicación, como son las transmisiones telegráficas, telefónicas, etc. Este tipo de tratamiento transmite los datos desde donde se generan hasta el sistema de proceso de datos que los trata. Una vez procesados, el ordenador transmite los resultados al punto de origen. El tiempo que tarda el ordenador en devolver los resultados se llama *tiempo de respuesta.*

En un sistema de Tiempo Real (*Real Time*), el tiempo de respuesta ha de ser el mínimo que se necesite para obtener el resultado después del proceso efectuado al realizar la pregunta correspondiente.

En un sistema de Tiempo Compartido (*Time Sharing*), diferentes usuarios comparten el mismo ordenador en tiempo real.

Un sistema de teletratamiento está compuesto por los siguientes elementos:

— Ordenador
— Terminales
— Vías de transmisión de datos

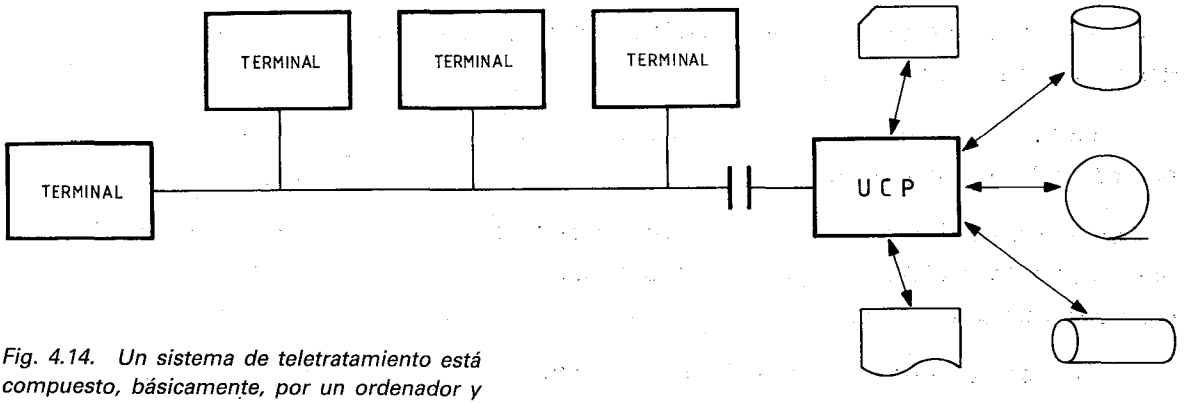


Fig. 4.14. Un sistema de teletratamiento está compuesto, básicamente, por un ordenador y unos terminales, situados a distancia, unidos al ordenador por unas vías de comunicación.

4.2.1. El ordenador central

En un sistema de teletratamiento, el ordenador controla el funcionamiento de los demás elementos del sistema. Es decir, controla la comunicación o intercambio de información dentro del sistema de teletratamiento.

En el ordenador utilizado en un sistema de teletratamiento deben tenerse en cuenta los siguientes elementos: dispositivos de interrupción, memorias intermedias, reloj, etc.

Para procesar la información que llega al ordenador deben crearse una serie de *colas* en donde estarán determinadas informaciones en espera de su tratamiento. Para ello se utilizan memorias intermedias (*buffer*).

4.2.2. Terminales

Los *terminales* son dispositivos transmisores y/o receptores que permiten al operador mantener un diálogo con el ordenador o con otros terminales.

Los terminales suelen ser dispositivos periféricos sin capacidad de cálculo que pueden ser manejados a distancia, manual o automáticamente, mediante los dispositivos de control adecuados.

Los terminales transmiten la información que debe ser procesada a un ordenador mediante fichas perforadas, cintas perforadas, cintas magnéticas o por medio de un teclado.

Otra de las misiones de los terminales consiste en transformar las informaciones emitidas desde el ordenador en información impresa, perforada en fichas o cintas o por medio de pantallas (representación visual).

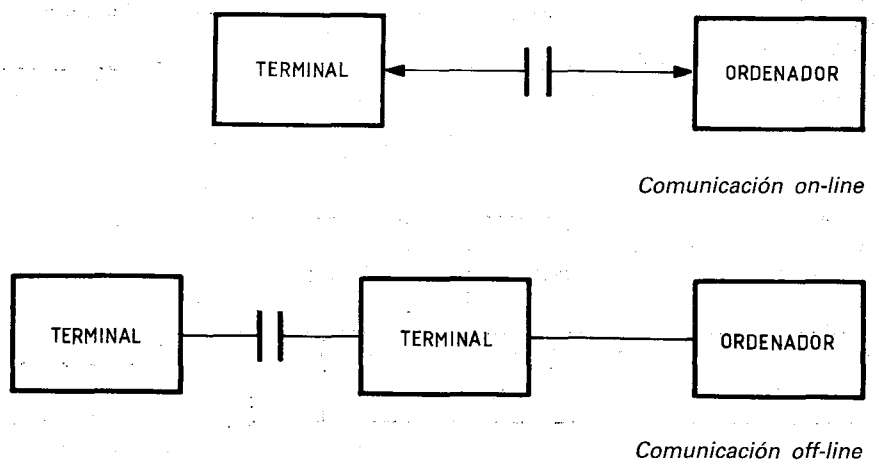
La forma de comunicación entre el ordenador y un terminal puede ser de dos tipos según se utilice un sistema *on-line* (en línea) o un sistema *off-line* (fuera de línea).

En un sistema *on-line,* los datos llegan directamente al ordenador y, del mismo modo, el ordenador devuelve los resultados al usuario.

En los sistemas *off-line,* los datos los recibe un *terminal* (por ejemplo, una perforadora de fichas, una perforadora de cintas, etc.). Después se envían los datos al ordenador a través de este terminal.

Mediante un sistema on-line se obtiene mayor rapidez en el proceso y mayor capacidad de control que utilizando un sistema off-line.

Fig. 4.15. La comunicación entre ordenador y terminal puede ser directa (on-line), o a través de otro terminal (off-line).



4.2.3. Vías de transmisión

Las *vías de transmisión* son los medios de comunicación entre el ordenador y los terminales, es decir, constituyen el medio físico por el cual se transporta la información desde los terminales al ordenador o desde el ordenador a los terminales.

Las vías de transmisión de datos son los componentes que más encarecen el sistema de teletratamiento.

Según el modo en que se encuentran conectados los terminales con el ordenador, las vías de transmisión pueden ser de diferentes tipos:

— Punto a punto

— Multipunto

— A través de una red conmutada

En un sistema de transmisión *punto a punto* cada terminal está conectado al ordenador por una sola línea de transmisión.

En un sistema *multipunto* varios terminales comparten la misma línea de transmisión.

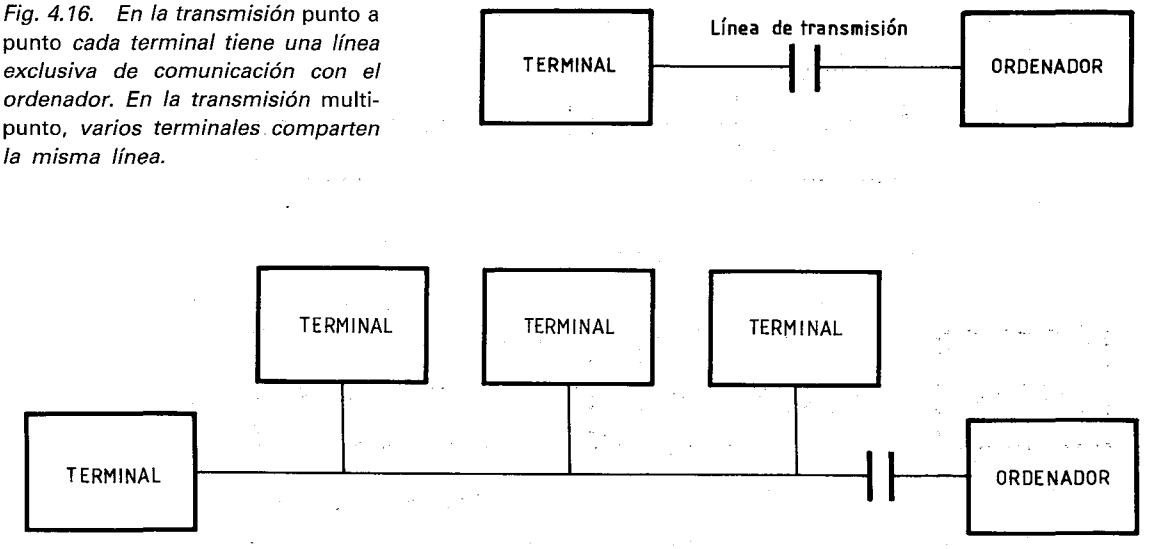


Fig. 4.16. En la transmisión punto a punto *cada terminal tiene una línea exclusiva de comunicación con el ordenador. En la transmisión* multipunto, *varios terminales comparten la misma línea.*

En la transmisión a través de una *red conmutada,* los terminales se conectan a través de una red pública de conmutación. Cuando la red establece conexión entre el ordenador y el terminal, se efectúa una transmisión punto a punto.

Una vía de transmisión puede contener elementos como:

— Líneas de comunicación

— Controlador de líneas

— Concentradores

— Multiplexores

— Modems.

Líneas de comunicación

Los medios utilizados como líneas de comunicación suelen ser líneas telefónicas y líneas telegráficas. En algunos casos especiales, se utilizan otros medios como, por ejemplo, enlaces de radio por cables coaxiales (un cable coaxial contiene un conductor dentro de otro, separados ambos por material aislante).

Una de las características principales de las líneas es la velocidad de transmisión, la cual se mide en *baudios,* siendo un baudio la unidad elemental de transmisión por segundo. Generalmente, la unidad de transmisión es un bit.

Por lo tanto, 1 baudio = 1 bit/seg.

La velocidad de transmisión de las líneas va desde 50 baudios (líneas a baja velocidad) hasta 40800 baudios (líneas de alta velocidad).

Controlador de líneas

El controlador de líneas es un dispositivo que controla la transmisión, actuando de *interface* (*punto de conexión*) entre la línea de comunicación y el ordenador. Se encuentra situado próximo al ordenador (Fig. 4.17).

Algunas de las funciones del controlador consisten en: convertir en paralelo la información transmitida en serie, controlar los posibles errores de transmisión, etc.

Cuando un terminal va a emitir una información al ordenador, envía una señal de control para preguntar al ordenador si se encuentra disponible para

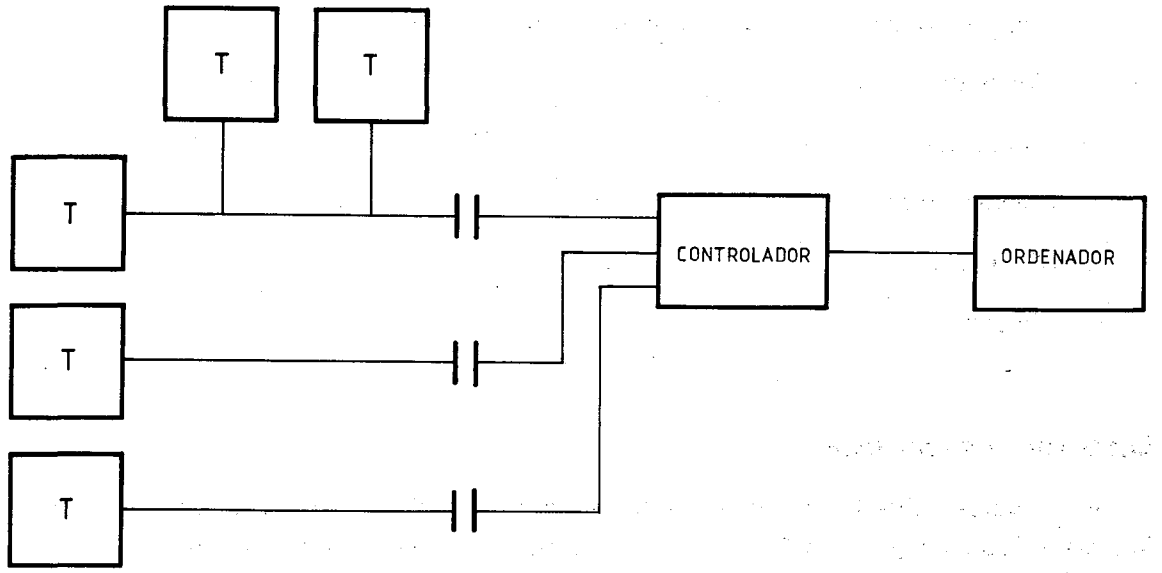


Fig. 4.17. La principal misión del controlador es la de regular los contactos entre los terminales y el ordenador.

recibir información. El ordenador efectúa la respuesta y si es negativa no se emite la información hasta que el ordenador envíe una señal indicando que está disponible para recibir información desde ese terminal.

Existen dos métodos para el control de líneas:

— Método *polling*

— Método de *contención.*

En el método *polling,* el ordenador emite por la línea a cada uno de los terminales, preguntándoles si tienen información para enviar. Primero le pregunta a un terminal y si éste le responde negativamente, entonces el ordenador consulta al siguiente terminal en orden de prioridad. En caso contrario el terminal envía la información.

En el método de *contención* no existe prioridad en los terminales. El terminal envía información cuando el ordenador está disponible para recibirla.

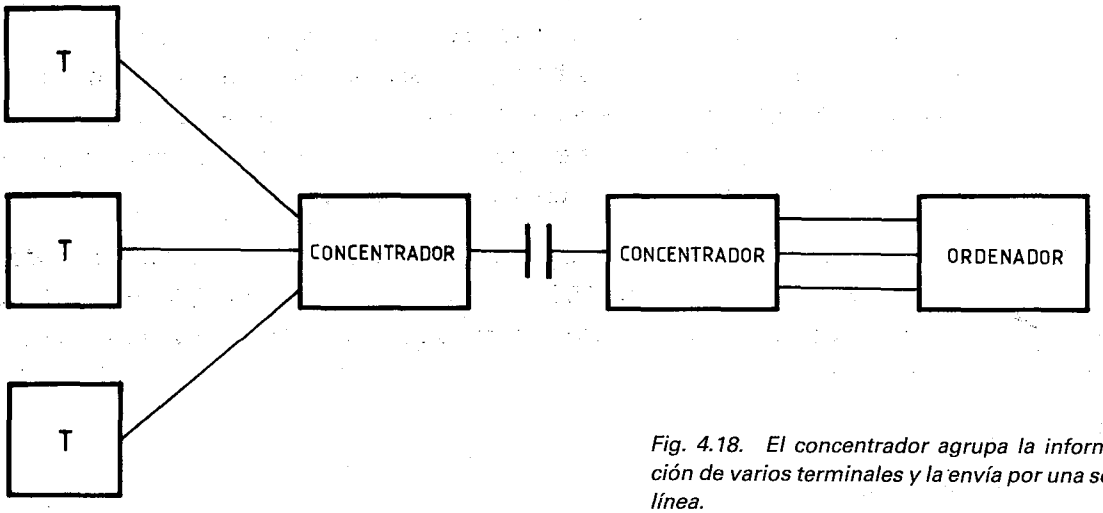


Fig. 4.18. El concentrador agrupa la información de varios terminales y la envía por una sola línea.

Concentradores

Los *concentradores,* situados cerca de los terminales, son dispositivos que agrupan o concentran la información de varios terminales enviándola al ordenador por una sola línea de comunicación y viceversa, recibe información del ordenador y la envía a cada terminal.

Los concentradores reciben la información a baja velocidad y la envían a los ordenadores a alta velocidad.

Los concentradores exploran periódicamente las líneas y va almacenando la información recogida en *buffers* (dispositivos que almacenan temporalmente la información, se denominan también *memorias de amortiguación*). En cada una de las líneas hay un buffer y, cuando se llena, el concentrador envía la información al ordenador.

Multiplexores

Los *multiplexores,* situados próximos a los terminales, son dispositivos que, al igual que los concentradores, agrupan la información de varios terminales y la envían por una sola línea, y la información que reciben del ordenador la envían a los terminales.

La operación de multiplexar consiste en transmitir simultáneamente dos o más mensajes por una sola línea.

Cuando se conectan varios terminales a un ordenador, puesto que una línea de alta velocidad es más barata que varias líneas de baja velocidad, es conveniente agrupar las líneas de baja velocidad en un punto de conexión, es decir, en el multiplexor, desde el que se puede efectuar la transmisión hacia el ordenador mediante una sola línea de alta velocidad.

Por ejemplo, si están conectados 5 terminales con el ordenador a través de líneas cuya velocidad es de 200 baudios, mediante el multiplexor se obtiene una línea de 1200 baudios. Esta conversión suele efectuarse carácter a carácter o en bloques.

Modems

Las señales que se envían desde el ordenador hacia los terminales o desde estos al ordenador son digitales (es decir, impulsos o ausencia de impulsos).

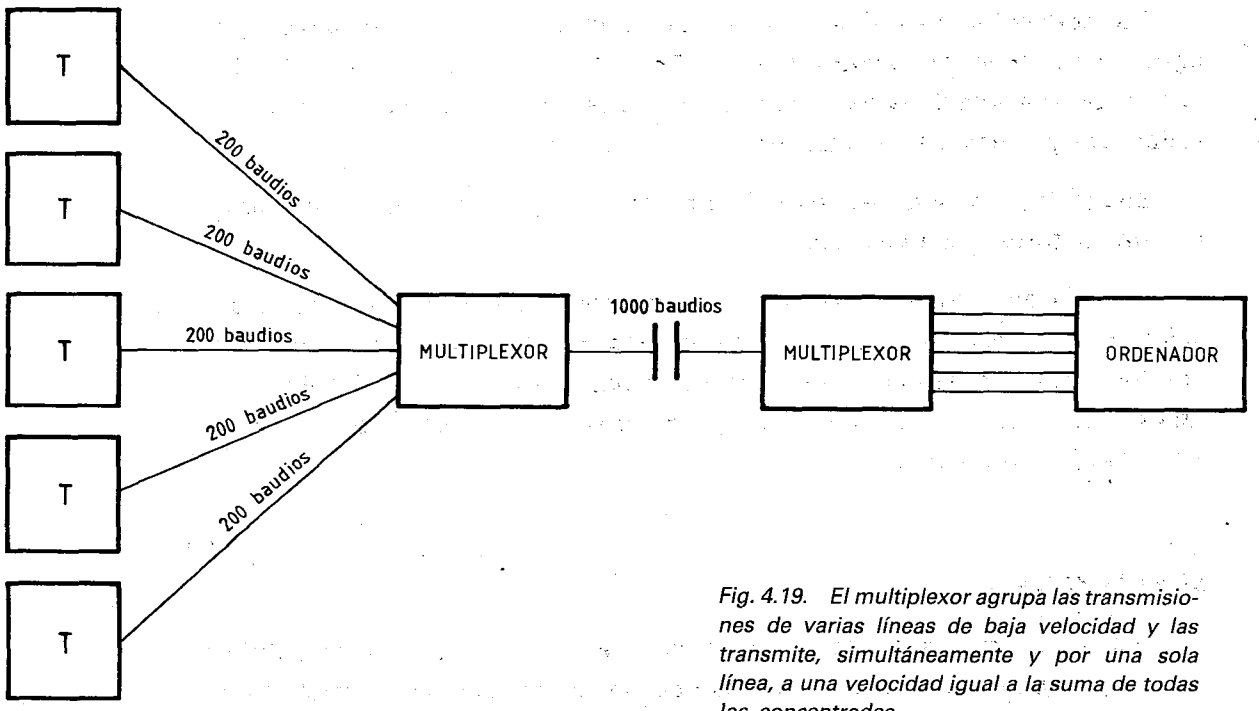


Fig. 4.19. El multiplexor agrupa las transmisiones de varias líneas de baja velocidad y las transmite, simultáneamente y por una sola línea, a una velocidad igual a la suma de todas las concentradas.

Por su parte, en las líneas de comunicación (por ejemplo, en las líneas telefónicas) sólo puede realizarse la transmisión en corriente alterna, mediante otro tipo de señales de oscilación constantes llamadas señales sinusoidales. Por lo tanto, para poder transmitir sobre estas líneas información digital, los ceros y unos deben traducirse a este tipo de señales. Para ello se hacen variar ciertas magnitudes de estas señales, como son la *frecuencia,* la *amplitud* y la *fase.*

La transformación de estas magnitudes a partir de las señales digitales se denomina *modulación;* y la obtención de la información digital a partir de las oscilaciones se llama *demodulación.* El dispositivo utilizado para efectuar estas transformaciones se llama *Modem* (MOdulador-DEModulador).

La modulación de frecuencia aumenta la frecuencia de la señal cuando se transmite un bit 1. Y continúa con la frecuencia normal cuando se transmite un bit 0. Con la modulación de amplitud, los bits 1 tienen diferente amplitud que los bit 0. Mediante la modulación de fase, cuando se transmite un bit 1 y el anterior bit era un cero, se cambia la fase, y viceversa.

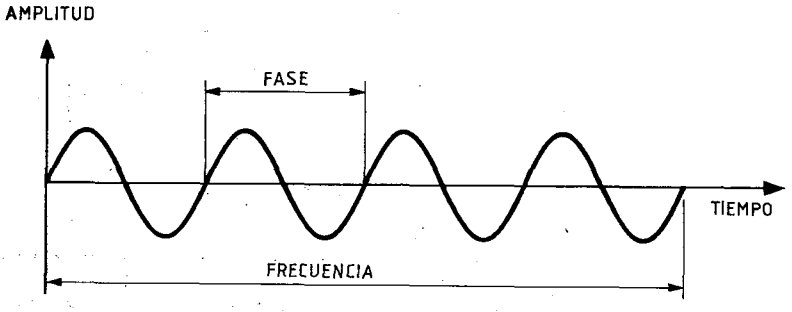


Fig. 4.20. Magnitudes en una señal sinusoidal.

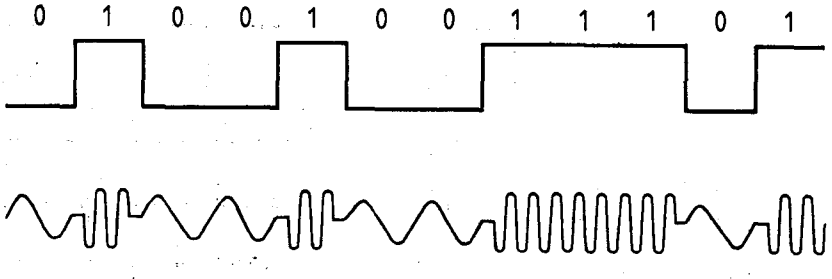


Fig. 4.21. El Mo-dem produce alteraciones en la frecuencia, amplitud y fase de una señal sinusoidal.

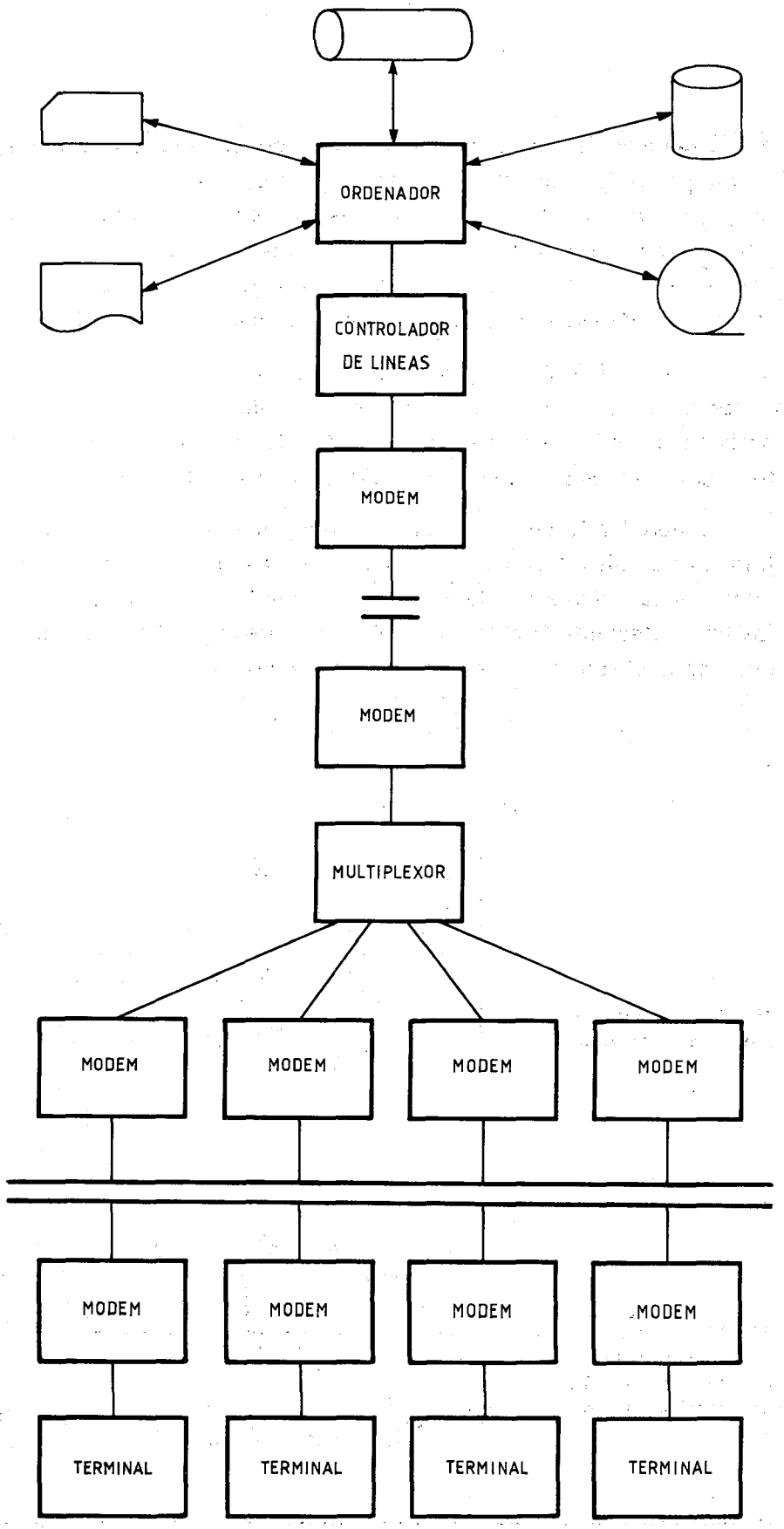


Fig. 4.22. Situación de los Mo-dems en las líneas de comunicación, dentro de un sistema de teletratamiento.

4.2.4. Tipos de transmisión

La transmisión de datos entre un terminal y el ordenador central al que está conectado puede ser de diversos tipos según el sentido de la información en la línea de comunicación se clasifica que:

— Símplex

— Semidúplex

— Dúplex

En la transmisión *símplex,* la información se transmite siempre en un mismo sentido. Es decir, sólo se puede enviar información al ordenador, sin que se obtengan respuestas por el terminal (Fig. 4.23), o bien, dispositivos que sólo reciben información desde el ordenador sin poder transmitir (Fig. 4.24).

En la transmisión *semidúplex* (*Half-Duplex*) puede transmitirse información en ambos sentidos, pero no al mismo tiempo, es decir, el terminal (o el ordenador) debe esperar a que termine completamente la recepción de información para poder enviar más información al ordenador (o al terminal) (Figura 4.25).



Fig. 4.23.

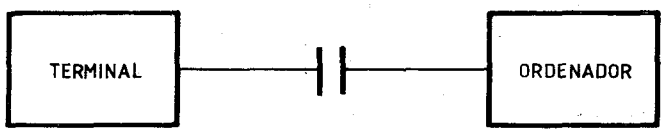


Fig. 4.24.

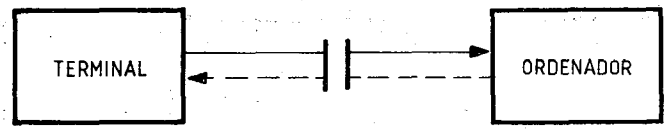


Fig. 4.25.

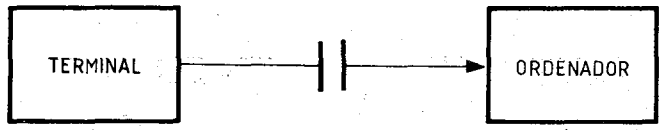


Fig. 4.26.

En la transmisión *dúplex (Full-Duplex)* la información se transmite en ambos sentidos simultáneamente (Fig. 4.26).

Otra clasificación según el modo de transmitir la información es:

— Transmisión asíncrona

— Transmisión síncrona

Generalmente, la transmisión de los bits se efectúa en serie, es decir, cada uno de los bits de un carácter se transmite sucesivamente. Para que se pueda realizar esta transmisión, el emisor y el receptor, es decir, el terminal y el ordenador, deben estar *sincronizados.*

El *sincronismo* (es decir, la igualdad cronológica de los procesos) entre el emisor y el receptor puede obtenerse mediante información adicional en el principio y final del carácter a transmitir. Este método se denomina transmisión *asíncrona,* en la que sólo existe sincronismo entre el emisor y el receptor durante el tiempo de emisión de un carácter.

El método de transmisión asíncrona se llama también START-STOP (arranque-parada) (Fig. 4.27).

En la transmisión *síncrona* se transmiten la información en bloques. En este caso, al principio de cada bloque se transmiten algunos caracteres que controlan el sincronismo. Después de emitir estos caracteres de control, ya se encuentran el emisor y el receptor totalmente sincronizados y se transmite, a continuación, el bloque correspondiente (Fig. 4.28).

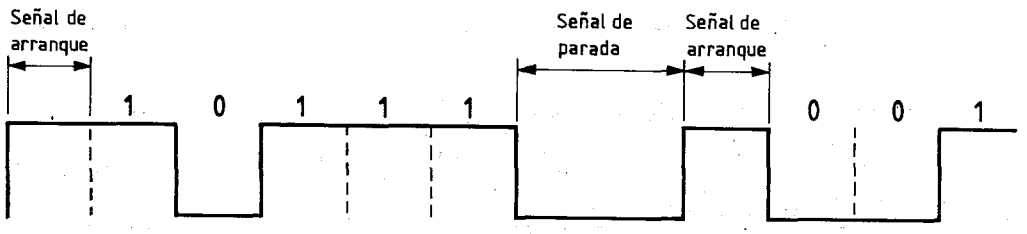


Fig. 4.27.

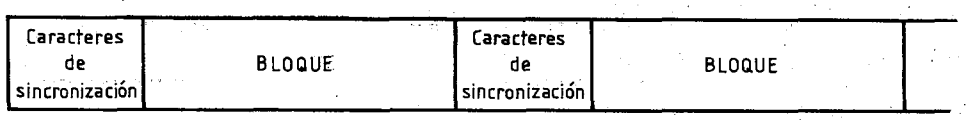


Fig. 4.28.

EJERCICIOS

1. ¿Cuál es la diferencia entre un sistema de tiempo compartido y un sistema de multiprogramación?

   a) El sistema operativo no puede determinar cuántos programas van a procesarse y cuál es el tiempo asignado

   b) El tiempo compartido es más lento que la multiprogramación

   c) El tiempo compartido no puede utilizarse a distancia mientras que la multiprogramación sí

2. ¿A qué se llama *paginación*?

   a) A las hojas del listado

   b) A la división de la memoria en zonas parciales

   c) A las hojas de codificación

3. ¿Qué programas tienen prioridad sobre los otros?

   a) Los que trabajan en tiempo compartido

   b) Los que trabajan en tiempo real

   c) Los que trabajan en multiprogramación

4. Cuando varios usuarios solicitan el proceso al mismo tiempo, ¿quién los sitúa en cola y según el orden de prioridades?

   a) El compilador

   b) El editor

   c) El monitor

5. La memoria virtual se utiliza:

   a) Para ampliar la memoria central

   b) Para ampliar las memorias auxiliares

   c) Para contener información por si tiene que utilizarse

6. ¿Quién efectúa las representaciones?

   a) Los propios programas

   b) La memoria central

   c) El compilador

7. Cuando los datos se envían directamente al ordenador sin utilizar periféricos adicionales se dice que es un sistema:

   a) Off-line

   b) On-line

   c) De teleproceso

8. En un sistema de transmisión en el que cada terminal está conectado al ordenador por una sola línea de transmisión se llama:

   a) Punto a punto

   b) Multipunto

   c) Polling

9. La velocidad de transmisión se mide en baudios, siendo un baudio igual a:

   a) 1 bit/seg

   b) 1 carácter/seg

   c) 1 bloque/seg

10. Los multiplexores:

   a) Transmiten simultáneamente mensajes por una sola línea

   b) Reciben la información a baja velocidad y la transmiten a alta velocidad

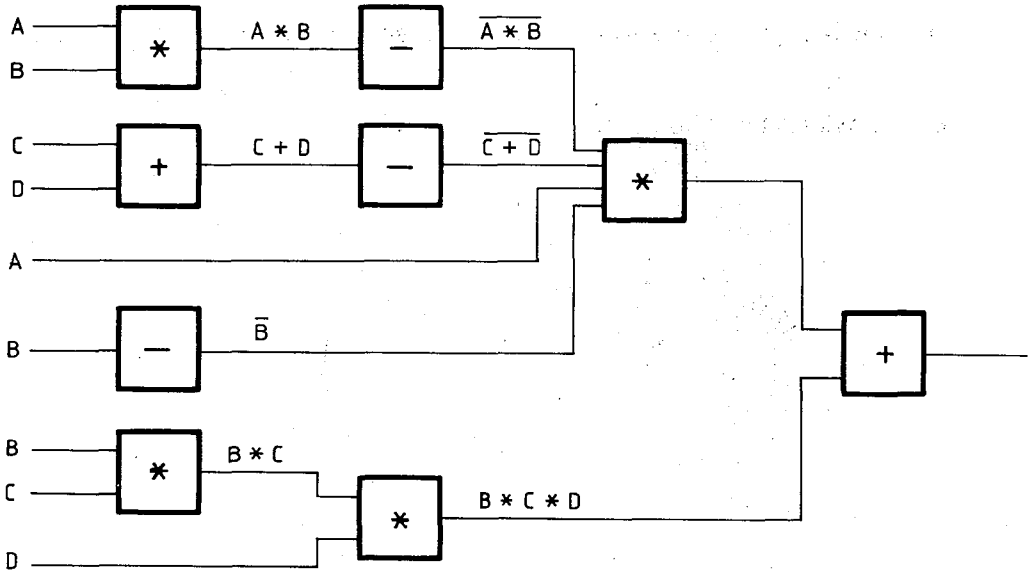
   c) Convierten la información de serie a paralelo.

SOLUCIONES DE LOS EJERCICIOS

1. Funcionamiento del ordenador (I)

1. *b);* 2. *a);* 3. *c);* 4. Ciclo de memoria; 5. *b);* 6. (Ver figura); 8. *c);* 9. *b);* 10. *c);* 11. *c);* 12. *b);* 13. *c);* 14. *c);* 15. Una memoria de solo lectura (Read Only Memory).

6.



2. Funcionamiento del ordenador (II)

1. *b*);2. *c*); 3. *a*); 4. *c*); 5. *a*t); 6. *b*); 7. *b*); 8. *c*).

3. Funcionamiento del ordenador (III)

1. *a);* 2. *b);* 3. *b);* 4. *c);* 5. *b);a);* 6. *c);* 7. *b);* 8. *b).*

4. Funcionamiento del ordenador (IV)

1. *a*); 2. *b*); 3. *b*); 4. *c*); 5. *a*); 6. *c*); 7. *b*); 8. *a*); 9. *a*); 10. *a*).

INDICE GENERAL

Contenido 7

Introducción 9

1. FUNCIONAMIENTO DE ORDENADOR (I).
Transmisión y almacenamiento de la información

Cuadro resumen 12
Funcionamiento del ordenador 14
1.1. Transmisión de la información 16
1.1.1. Transmisión en serie 18
1.1.2. Transmisión en paralelo 19
1.1.3. Transmisión mixta 20
1.1.4. Circuitos de conmutación 20
— Circuito Y 21
— Circuito O 23
— Combinación de circuitos 24
1.1.5. Puertas lógicas 25
— Puertas Y 25
— Puertas O 26
— Inversor 26
— Puertas NO-O 26
— Puertas NO-Y 27
— Combinaciones de puertas 27
— Puerta de autorización 28
— Puerta de prohibición 30
— Selector de dirección 30
— Decodificador 31
1.2. Almacenamiento de la información 33

1.2.1. Unidades de información 34
1.2.2. Representación de los datos en la MC 43
— Datos numéricos 43
— Coma fija 45
— Coma flotante 47
— Datos alfanuméricos 48
1.2.3. Acceso a la MC 49
— Memoria de acceso aleatorio 51
— Memoria de sólo lectura 51
— Memoria asociativa 52
1.2.4. Constitución de la memoria central 52
— Memoria de núcleos magnéticos 55
— Memoria de agujas magnéticas 56
— Memoria de hilos magnéticos 57
1.2.5. Memorias de gran capacidad 58
1.2.6. Memorias flip-flop 59
1.2.7. Registros 62
— Circuito de mantenimiento 63
— Registros acumuladores 66
— Registros de desplazamiento 67
1.2.8. Clasificación de los soportes de almacenamiento 68

Ejercicios 70

2. FUNCIONAMIENTO DEL ORDENADOR (II). Unidad aritmética. Unidad de control. Unidades de E/S

Cuadro resumen 75
2.1. Funciones de la unidad aritmética/lógica 76
2.1.1. Unidad aritmética 79
2.1.2. Unidad lógica 92
2.2. Funciones de la unidad de control 96
2.2.1. Reloj interno 96
2.2.2. Registros de la UC 98
2.2.3. Ejecución de las instrucciones 100
2.3. Funciones de la unidad de entrada/salida 105
2.3.1. E/S off-line 107
2.3.2. E/S on-line 109
2.3.4. Flujo de información entre la UCP y las unidades de E/S 111

Ejercicios 113

3. FUNCIONAMIENTO DEL ORDENADOR (III). Los sistemas operativos

Cuadro resumen 117
3.1. Software 118
3.2. Sistemas operativos 119
3.2.1. Programas de control 120
— Supervisor 121
3.2.2. Programas de proceso 124

Ejercicios 128

4. FUNCIONAMIENTO DEL ORDENADOR (IV). Métodos de proceso. Teletratamiento

Cuadro resumen 132
4.1. Métodos de funcionamiento 134
4.1.1. Proceso por lotes 134
— Proceso por lotes sin prioridades 136
— Proceso por lotes con prioridades 137
— Proceso por lotes remoto 137
4.1.2. Tiempo compartido 138
— Reparto del tiempo de proceso 141
— Partición de la memoria 142
4.1.3 Memoria virtual 143
4.1.4. Multiprogramación 146
4.1.5. Tiempo real 148
4.1.6. Muliproceso 149
4.2. Sistemas de teletratamiento 152
4.2.1. El ordenador central 153
4.2.2. Terminales 153
4.2.3. Vías de transmisión 155
— Líneas de comunicación 156
— Controlador de líneas 156
— Concentradores 158
— Multiplexores 158
— Modems 159
4.2.4. Tipos de transmisión 162

Ejercicios 164

Soluciones a los ejercicios 167

En este volumen se describen los fundamentos principales para la introducción de pleno en el campo de la Programación. Los conceptos expuestos aparecen de una forma progresiva que desemboca en la preparación del lector a un más profundo estudio de los campos específicos de la operativa informática.

El proceso de datos, instrumentos, partes, componentes, y relaciones entre todos ellos, vienen descritos de manera que su asimilación y entendimiento sea perfectamente asequible a toda persona que acompañe a la exposición con un mínimo de estudio, constancia e interés.

Al final de los capítulos se expone una serie de ejercicios que sirven para apreciar si la materia estudiada está perfectamente comprendida y el mismo lector puede juzgarlo mediante el solucionario que subsiguientemente se adjunta.

Una característica importante del libro es la subdivisión con que se presentan los temas más importantes que conduce a una exposición explícita y clara, siempre de gran ayuda para el lector.

El dibujo de la cubierta es cortesía de GENERAL DE COMPUTADORES, S.A.