

The background features a close-up, angled view of a microchip and its surrounding circuit board. The chip is a square, textured component with a grid of pins along its edges. The circuit board is dark blue with intricate white traces and numerous circular solder points. The overall composition is dynamic, with strong geometric lines and a sense of depth.

Análisis de sistemas microprogramables

10

Introducción

Para poder entender cómo surgió la necesidad de los sistemas electrónicos microprogramados, debemos conocer la forma en que se produjo el desarrollo y evolución de la técnica electrónica. Dicha evolución se efectuó en dos etapas: una de desarrollo cuantitativo y otra de desarrollo cualitativo.

Etapa de desarrollo cuantitativo

Fue la primera etapa de la evolución, caracterizada por una *filosofía de diseño* en la cual los componentes eran diseñados específicamente para una aplicación o un número muy reducido de ellas, no siendo fácil su inclusión en otras aplicaciones.

Su desarrollo, muy lento, duró desde los comienzos de la Electrónica hasta finales de la década de los cincuenta.

Etapa de desarrollo cualitativo

A principios de la década de los sesenta, la industria electrónica precisó un cambio en su filosofía de producción para conseguir abaratar los costes de fabricación. Surgió entonces

la idea de diseñar y producir componentes y circuitos de *elevada versatilidad* que poseyeran muchas aplicaciones y que pudieran ser fabricados en elevadísimas cantidades.

La aparición de la *Microelectrónica* con sus circuitos integrados a principios de la década de los setenta, permitió la obtención de circuitos altamente complejos que, integrados en reducidos encapsulados, servían para un elevado número de aplicaciones. La técnica microelectrónica tuvo un desarrollo vertiginoso que aún continúa en nuestros días y que permitió la idea de poder obtener el *circuito universal*, que debería tener las siguientes propiedades:

- Estar dotado de un método de programación que le permitiera trabajar de diferente forma, según la aplicación a la que fuera destinado.
- Ocupar un reducido espacio.
- Poder fabricarse mediante procesos de producción en serie para obtener cantidades enormes del mismo circuito y abaratar costes.

Los *sistemas microprogramables* son el producto más evolucionado de esta filosofía de producción y logran convertirse en el componente universal en los denominados *microcontroladores*.

10.1 Estructura de un sistema microprogramable

A. Definición de sistema microprogramable

Un sistema microprogramable es un sistema electrónico digital, formado por uno o varios chips integrados, capaz de interpretar y ejecutar secuencialmente las órdenes contenidas en un programa y a una velocidad muy elevada.

Sus aplicaciones son muchas, ya que con variar simplemente la secuencia de órdenes que contiene su programa de control, son capaces de:

- *Realizar cálculos matemáticos o aplicaciones informáticas.* Éste es el caso de los ordenadores personales que en realidad no son otra cosa que sistemas microprogramados.
- *Controlar procesos industriales de seguridad o producción,* como es el caso de los autómatas programables que controlan robots y cadenas de montaje.
- *Controlar el funcionamiento de aparatos domésticos,* como sucede con lavadoras, lavavajillas, hornos microondas, etc., que se encuentran controlados por microprocesadores o microcontroladores.

B. Hardware y software

Cuando se realiza el estudio de un sistema microprogramable, todos los conocimientos y conceptos relacionados con el sistema se agrupan bajo tres términos genéricos que son:

- **Hardware:** se denomina así a todos los circuitos y componentes electrónicos que constituyen el sistema. De esta forma, cuando hablemos de fabricantes o vendedores de circuitos microprogramables, los llamaremos fabricantes o vendedores de hardware.
- **Software:** este término define el *conjunto de instrucciones y rutinas* con que se programa el sistema, así como todo lo referente a los lenguajes empleados para obtener dichos programas.
- **Firmware:** es el conjunto de instrucciones y programas de un sistema microprogramable, que son inalterables por el usuario y se encuentran implícitos en la estructura electrónica de un circuito. Es, por tanto, un término que define elementos intermedios entre el hardware y el software.

C. Esquema de bloques de un sistema microprogramable

La estructura básica de cualquier sistema microprogramable es, independientemente de su capacidad de trabajo, la que muestra el esquema de bloques de la Figura 10.1. En el citado esquema se distinguen los siguientes bloques:

- **Reloj.** Es un generador de impulsos digitales en forma de onda cuadrada, cuya frecuencia define la velocidad de trabajo del sistema.
- **Unidad central de proceso, CPU.** Es el centro de interpretación y ejecución del sistema. Esta unidad, partiendo de las señales digitales producidas por el bloque de *reloj*, genera todas las señales digitales de gobierno del sistema en función de las órdenes enviadas por un programa. Es el verdadero cerebro del sistema microprogramable. Su estructura interna es compleja, realizándose actualmente de las siguientes formas:
 - Con complejos circuitos digitales cableados colocados en tarjetas de circuito impreso, agrupados en un mismo *chasis* o *armario*. Éste es el caso de la *CPU* de los *grandes ordenadores* o *mainframe*.

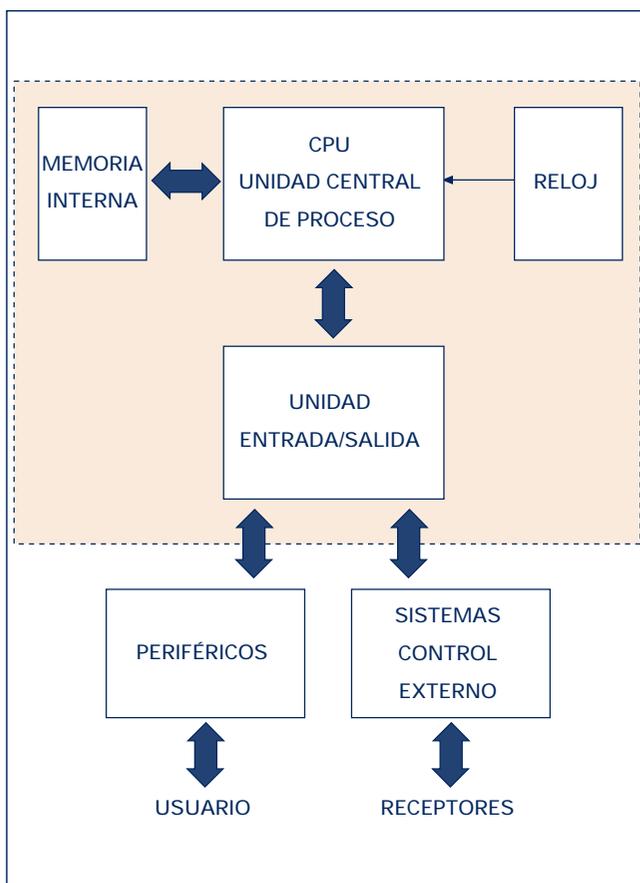


Fig. 10.1. Esquema de bloques de un sistema microprogramable

— Con *microprocesadores*, que es el nombre que recibe un tipo de *CPU* estandarizado y muy versátil integrado en un pequeño chip. Los microprocesadores son capaces de formar, en torno suyo, sistemas de mayor o menor capacidad de trabajo, tanto en el campo informático como en el de la automatización y control industrial o doméstico.

- **Memoria interna.** Es el dispositivo que se encarga del almacenamiento de datos y programas. No debemos confundirlo con las unidades de memoria masa, que poseen algunos sistemas de desarrollo, las cuales constituyen un tipo de periférico del sistema y se encuentran físicamente localizadas fuera de su estructura básica.
- **Unidad de entrada/salida (interface).** Es el circuito que controla la comunicación entre interior y exterior del sistema. Sirve para adaptar los códigos binarios de trabajo de la *CPU* con los de los periféricos o dispositivos de control externo, así como para adaptar las diferentes velocidades de trabajo entre el interior y el exterior.
- **Periféricos y sistemas de control externo.** Constituyen un amplio grupo de dispositivos, circuitos y máquinas que, controlados por el sistema, realizan un trabajo exterior. Comúnmente, se denominan *periféricos* a los aparatos que sirven para comunicar al sistema microprogramable con el usuario, dándose el nombre de *sistemas de control externo* a los circuitos empleados en automatización y que sirven tanto para controlar receptores o máquinas, como para enviar información del receptor o máquina al sistema microprogramable.

D. Sistemas microprogramables integrados en un solo chip

La evolución de la técnica microelectrónica ha permitido obtener sistemas microprogramables integrados en un solo chip; estos nuevos componentes-sistema son los siguientes:

- **El microcontrolador.** Es un chip que integra toda la estructura del sistema microprogramable, con la única excepción de los periféricos y sistemas de control externo. Aunque posee limitaciones en su capacidad de trabajo, es el tan deseado circuito universal.
- **Los dispositivos Lógicos Programables (PLD).** Son chips integrados que, con una estructura similar a una memoria, son capaces de realizar la función para la que fueron diseñados. Estos dispositivos no se ajustan internamente a la estructura típica de un sistema microprogramable; son, por tanto, dispositivos que externamente funcionan como un sistema microprogramado.

E. Clasificación de los sistemas microprogramables

Los sistemas microprogramables se clasifican, atendiendo a su capacidad de trabajo y tratamiento de datos, en los siguientes grupos:

- **Ordenadores personales:** son sistemas microprogramables que utilizan como CPU un chip microprocesador. Están destinados a los procesos informáticos, aunque, añadiéndoles los sistemas de control externo adecuados, son capaces de controlar procesos o maquinaria industrial.
- **Autómatas programables:** son sistemas microprogramables basados en un chip microcontrolador al que se le añaden sistemas de control de potencia, tales como contactores o relés. Están destinados al control de maquinaria industrial y automatización.
- **Pequeños automatismos:** son sistemas gobernados por microprocesadores, microcontroladores o dispositivos PLD, destinados a controlar electrodomésticos, rótulos luminosos, calculadoras programables, pequeños sistemas de alarmas, etcétera.

10.2 Hardware de un sistema microprogramable con microprocesadores

Actividades en el aula

1. Observación de la estructura de un sistema microprogramable real.

Abrir un ordenador y observar su placa base intentando realizar un esquema de bloques.

*Nota: se puede utilizar cualquier sistema que emplee microprocesadores.

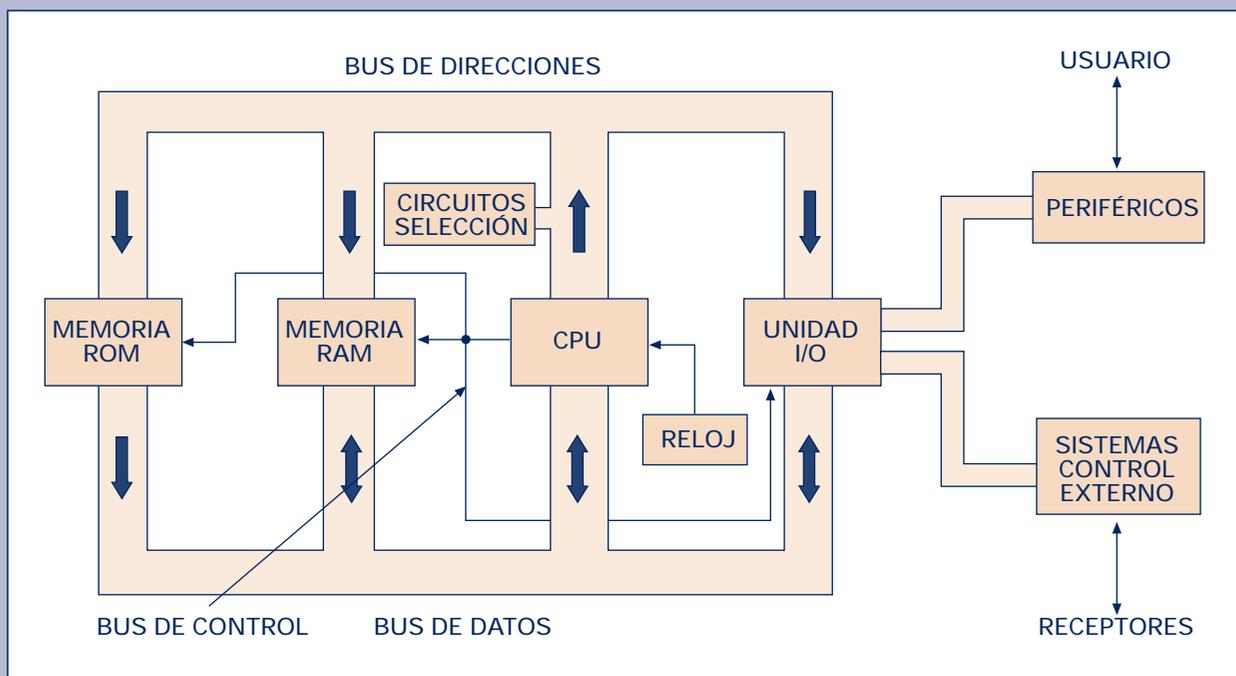


Fig. 10.2. Esquema de bloques de un sistema de desarrollo con microprocesador

A. Esquema de bloques de un sistema de desarrollo de microprocesadores

Un elevado número de sistemas microprogramables emplean un microprocesador en su estructura para realizar la función de CPU; ésta es la razón por la cual se suele denominar a estos sistemas *sistemas de desarrollo de microprocesadores*. Su estructura se adapta a la de cualquier sistema microprogramable, estando formada por una serie de bloques interconectados por grupos de cables que llamaremos *buses*. El esquema de bloques de la Figura 10.2 nos muestra esta estructura que, como mínimo, constituye el hardware de un sistema de desarrollo de microprocesadores.

Unidad central de proceso (CPU)

Formada por el chip integrado que constituye el microprocesador, es también denominada por algunos autores MPU. Su estructura es la de un circuito integrado LSI, que posee en su interior entre otros elementos:

- Una unidad de decodificación e interpretación.
- Una unidad de aritmética y lógica.
- Un contador de programa.
- Varios registros de almacenamiento de información.

La CPU controla, por medio de las instrucciones que decodifica e interpreta, a las memorias, a la unidad I/O y, a través de esta última, a los periféricos y sistemas de control externo.

Reloj

La CPU genera todos los impulsos de control de los restantes bloques, partiendo de ondas cuadradas de frecuencia constante generadas por el reloj. La frecuencia del reloj determina la *velocidad de operación o funcionamiento del microprocesador* y, por tanto, del sistema de desarrollo.

Circuitos de selección de chips

Es un conjunto de decodificadores y multiplexadores que tienen la finalidad de obtener, partiendo del bloque completo del bus de direcciones, las señales de autorización o *chip select* de cada bloque del sistema, evitando de esta forma que el grupo completo de líneas del bus de direcciones tenga que llegar a todos los bloques.

Memoria RAM

Su nombre corresponde a las iniciales del nombre en inglés *Random Access Memory*, que significa memoria de acceso

aleatorio. Se la suele denominar también *memoria de lectura/escritura*, ya que en ella se pueden realizar tanto operaciones de extracción de información o lectura (*Read*), como de almacenamiento o escritura (*Write*).

Su función en los sistemas microprogramables es contener los programas y datos que el usuario o la CPU puedan variar a voluntad.

En la Figura 10.3 se esquematiza la estructura interna de una memoria RAM. Su funcionamiento básico es el siguiente:

Cuando la CPU necesita leer o escribir una información en la memoria RAM, lo primero que hace es presentar, en el bus de direcciones, el código binario de la dirección de la memoria sobre la que necesita operar. Este código, al llegar al decodificador de direcciones de la RAM, es interpretado, activando la posición de memoria solicitada. Simultáneamente, el circuito de selección de chips enviará a la memoria RAM una señal de autorización ($\overline{CS} = \text{Chip Select o } \overline{E} = \text{Enable}$) generada a partir de la dirección solicitada por el microprocesador en el bus de direcciones.

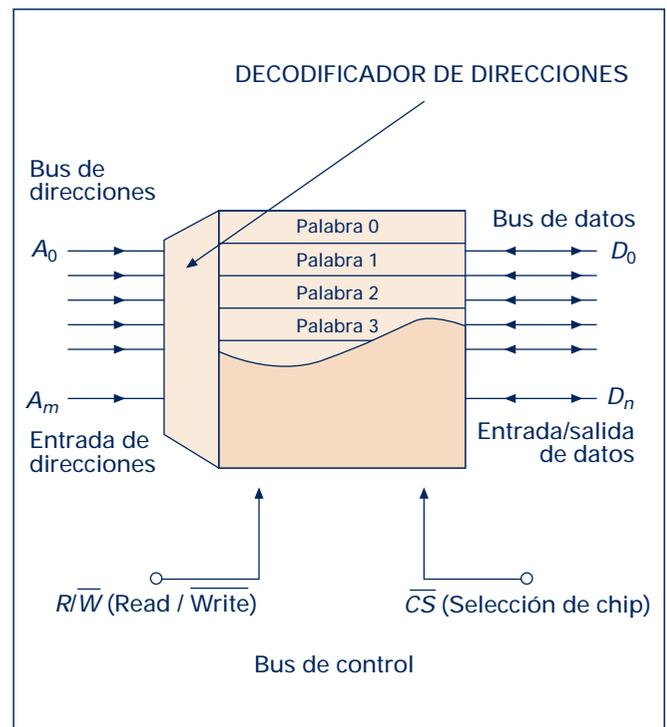


Fig. 10.3. Estructura básica de una memoria RAM

Dependiendo de la operación a efectuar (lectura o escritura), la CPU enviará simultáneamente por el bus de control la señal de *Read/Write*, cuyo valor indica si la operación a realizar es una lectura o una escritura de información (1 = lectura y 0 = escritura).

En función de la operación solicitada, el bus de datos de la memoria enviará o recibirá la información binaria solicitada.

Por último, cabe destacar que el bus de datos de una RAM es bidireccional, mientras que su bus de direcciones es unidireccional.

Memoria ROM

Su nombre corresponde a las iniciales de su denominación inglesa *Read Only Memory*, es decir, memoria de solo lectura. En esta memoria sólo se puede leer la información en ella almacenada. La función de este tipo de memoria es contener los datos y programas de arranque que precisa el sistema microprogramable para su activación, o el programa de funcionamiento en aquellos sistemas destinados a la automatización.

En la Figura 10.4 aparece en esquema la estructura de esta memoria.

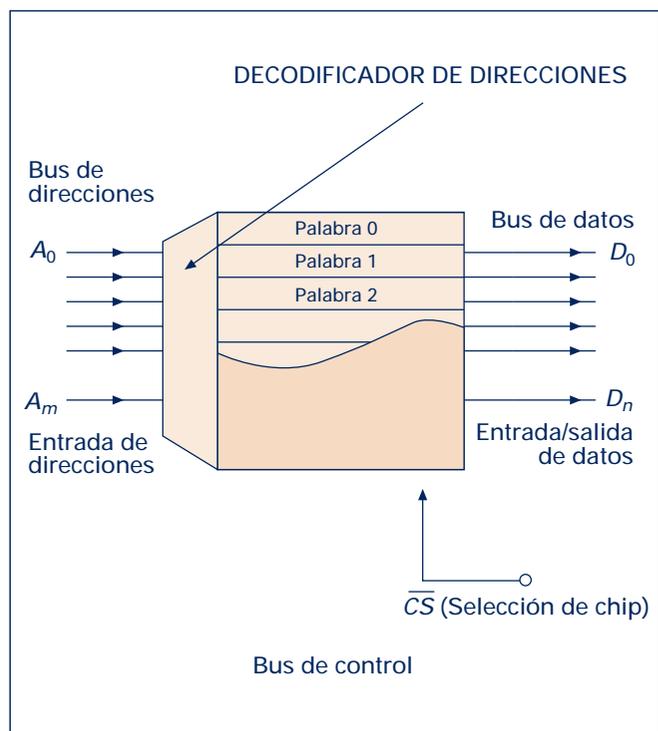


Fig. 10.4. Estructura básica de una memoria ROM

Su funcionamiento es similar al de la RAM, aunque más simple, al no ser necesaria la señal de R/\overline{W} . Por otra parte, al poder realizarse solamente operaciones de lectura, tanto el bus de direcciones como el bus de datos de las memorias ROM es unidireccional.

Actualmente, la denominación genérica de ROM comprende un conjunto de tipos de memoria entre las que se encuentran las memorias PROM, EPROM, EAROM, etc., que permiten, en determinadas circunstancias, la grabación o programación por parte del usuario y que veremos más ampliamente en el Capítulo 11.

Unidad de entrada/salida (Unidad I/O)

Todo sistema de desarrollo de microprocesadores necesita comunicarse con el exterior, bien sea para recibir información o para enviarla. Los dispositivos de entrada/salida constituyen la frontera entre los circuitos internos (CPU, memorias, etc.) y los externos. Este concepto de *frontera o interface* es el más claro para definir sus funciones.

La necesidad de emplear circuitos específicos para realizar las funciones de entrada/salida viene impuesta, básicamente, por las diferentes características eléctricas de las señales digitales empleadas en el interior y en el exterior del sistema. Las diferencias más importantes entre dichas señales son:

- *Diferente frecuencia de trabajo.* El microprocesador y las memorias trabajan a frecuencias superiores al megahercio, mientras que los periféricos y receptores rara vez pueden superar los kilohercios.
- *Diferencias en los códigos binarios de trabajo.*

Las interfaces o unidades de entrada/salida son circuitos cuya complejidad puede llegar a superar a la del microprocesador. Independientemente de su mayor o menor complejidad interna, poseen como mínimo estos elementos:

- **Registro de salida.** Es un registro de almacenamiento de información que posee un número de bits igual al de la palabra binaria utilizada, y cuya carga o escritura se realiza desde el interior del sistema.

El contenido de este registro es accesible, en cualquier momento, desde el exterior por medio de una serie de terminales O_0 a O_7 , que en lo sucesivo denominaremos *puerto de salida*. En el esquema de la Figura 10.5 vemos cómo el registro de salida está conectado al interior del sistema mediante las líneas D_0 a D_7 del bus de datos. Cuando una información procedente de la CPU o de cualquier otro bloque del interior ha de ser enviada al exterior, el registro de salida de la unidad recibe una *orden de carga* procedente de la CPU y, simultáneamente, la información a través de dichas líneas del bus de datos.

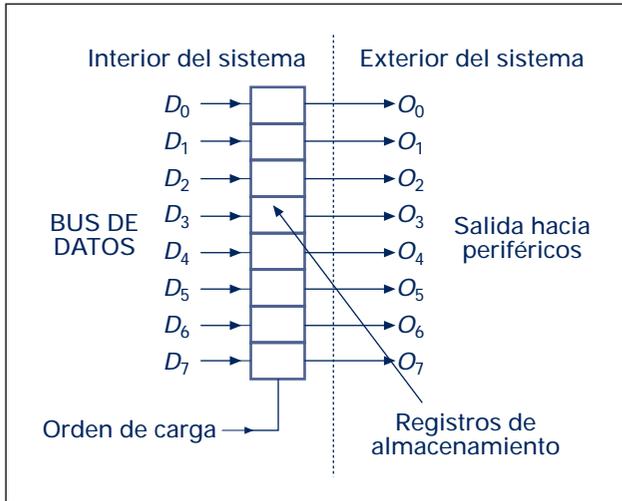


Fig. 10.5. Registro de salida de una unidad I/O

- **Registro de entrada.** Es un registro con un número de bits coincidente con el de la palabra binaria procesada y cuya carga está controlada por un conjunto de interruptores electrónicos, gobernados por la CPU (Figura 10.6).

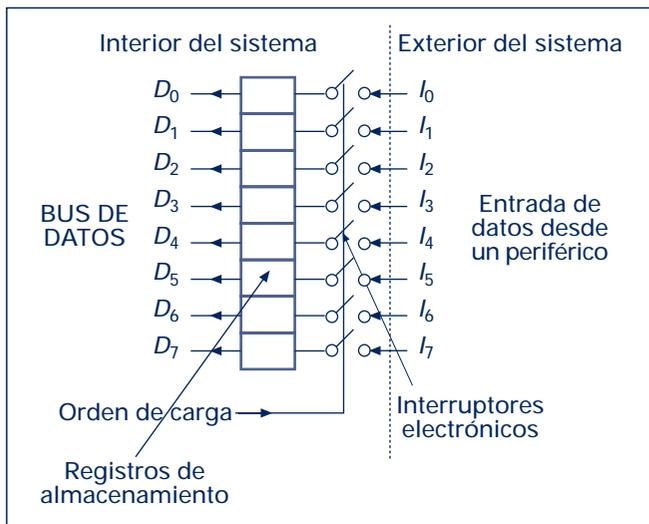


Fig. 10.6. Registro de entrada de una unidad I/O

De esta forma, la información sólo puede acceder desde los periféricos, a través de las líneas I_0 a I_7 , al interior del sistema, cuando la CPU lo autorice, quedando entonces almacenada dicha información en el registro de entrada. Desde este momento se podría leer dicha información en las líneas del bus de datos D_0 a D_7 . Al conjunto de líneas I_0 a I_7 las denominaremos en lo sucesivo *puerto de entrada*.

Buses

Se definen como el conjunto de grupos de hilos que transportan información del mismo tipo y que sirven para reali-

zar la comunicación entre los diferentes bloques del sistema microprogramable. Existen tres tipos de buses fundamentales, en dos de los cuales el número de hilos que poseen está en función del número de bits de la palabra binaria con que trabaja el sistema.

Los tres buses fundamentales de un sistema de desarrollo de microprocesadores son:

- **Bus de datos (Data Bus).** Es el destinado a transportar los datos entre los diferentes bloques. Su número de hilos es igual a la longitud de palabra binaria que procesa el microprocesador. Así, por ejemplo, un microprocesador de 8 bits posee un bus de datos formado por 8 hilos, cada uno de los cuales transporta un bit. Habitualmente, las líneas del bus de datos se denominan con la inicial *D*:

D_7 (bit de mayor peso) ... D_0 (bit de menor peso)

- **Bus de direcciones (Address Bus).** Su misión es transmitir, al bloque correspondiente, la dirección con la que va a trabajar la CPU. El número de líneas que lo forman depende del fabricante del microprocesador y determina la máxima cantidad de memoria que puede gobernar el sistema. El número de direcciones de memoria que es capaz de controlar un microprocesador, mediante su bus de direcciones, se obtiene mediante la siguiente fórmula:

$$N.^{\circ} \text{ direcciones de memoria} = 2^{N.^{\circ} \text{ líneas bus de direcciones}}$$

Así, por ejemplo, en los microprocesadores comerciales de 8 bits el bus de direcciones posee 16 líneas, lo que le permite gobernar hasta 65.536 posiciones de memoria (64K).

La denominación habitual de cada una de las líneas que componen este bus es la siguiente:

A_{15} (bit de mayor peso) A_{14} ... A_1 A_0 (bit de menor peso)

- **Bus de control (Control Bus).** Los microprocesadores emplean un cierto número de líneas para enviar o recibir órdenes que realizan fines diversos, tales como: línea de R/\overline{W} a las memorias, línea de \overline{RESET} o inicialización, líneas de interrupción, etc. A este conjunto de líneas, que depende y varía de unos microprocesadores a otros, se le denomina bus de control.

Periféricos

Estos dispositivos realizan dos tipos de funciones:

- Comunicación entre el sistema y el usuario.
- Almacenamiento masivo de información.

Existen infinidad de periféricos, pero podemos destacar entre los más comunes:

a) Periféricos de comunicación:

- **Teclado (*keyboard*)**. Es un *periférico de entrada* y sirve para que el usuario pueda introducir datos. Está formado por un conjunto de teclas, que pueden ser o no similares a las de una máquina de escribir y que están adaptadas al lenguaje de programación empleado por el sistema. Existen distintas clases:
 - Teclados hexadecimales, que sirven para programar en lenguaje máquina.
 - Teclados alfanuméricos, que sirven para programar en lenguajes de alto nivel.
 - Teclados especiales que, además de llevar la parte alfanumérica, contienen ciertas teclas con funciones específicas.
- **Pantallas y displays**. Son los *periféricos de salida* más usuales junto con las impresoras. En ellas aparecen tanto las informaciones que se introducen al sistema como las que éste indica al usuario. Existen diferentes tipos:
 - Monitores de televisión (TRC).
 - Pantallas de cristal líquido (LCD).
 - Visualizadores de tipo matricial, alfanuméricos o de 7 segmentos, empleados corrientemente en los sistemas de desarrollo de microprocesadores para aplicaciones de automatización.
- **Impresora**. Es un *periférico de salida* básico en cualquier sistema de desarrollo, está especializado en la producción de informes o listados de programas sobre papel. Existen muchos tipos en la actualidad, destacando como más importantes:
 - Impresoras de matriz de agujas.
 - Impresoras de inyección de tinta.
 - Impresoras láser.
- **Plotter**. Es un *periférico de salida* especializado en trasladar al papel dibujos y formas geométricas generados por el sistema.

- **Interfaces para comunicación de E/S**. Son dispositivos periféricos que no deben confundirse con las unidades I/O, y que sirven para intercomunicar sistemas microprogramables separados. Los tipos más corrientes de interface para comunicación son:

- Interface RS-232 para comunicación serie.
- Interface *Centronics* para comunicación paralelo.
- Interface I2C para interconexión de circuitos integrados.

- **Módem**. Es un interface que permite la comunicación serie a distancia, tanto de entrada como de salida de datos, entre dos sistemas microprogramables por medio del empleo de las redes telefónicas.

b) Periféricos de almacenamiento:

- **Unidades de cinta magnética**. Son dispositivos que emplean como soporte para el almacenamiento de datos las cintas magnéticas. Es el sistema más barato y uno de los primeros que se emplearon. Son los más lentos de todos los periféricos de almacenamiento. Existen dos tipos:
 - Los carretes de cinta para grandes ordenadores.
 - Las cintas tipo casete.
- **Unidades de disco magnético**. Esta clase de periférico emplea un soporte plano y circular recubierto de un material magnético donde se almacena la información. En la actualidad existen dos variedades fundamentales:
 - Disco duro (*hard disk*).
 - Disco flexible (*floppy disk*).
- **Unidades de disco óptico**. Este tipo de periférico, al igual que el anterior, emplea un soporte plano, ya sea circular o en forma de tarjeta, que utiliza un rayo láser para la lectura y grabación del soporte; su nombre habitual es el de CD-ROM o DVD y tiene una elevada capacidad de almacenamiento.

Sistemas de control externo

Constituyen un conjunto de circuitos que sirven para realizar la comunicación entre el sistema microprogramable y los procesos industriales o máquinas por él controlados. En su estructura intervienen los siguientes elementos:

- Transductores y sensores.
- Convertidores A/D, D/A, V/I, etcétera.
- Componentes de conmutación y potencia: relés, tiristores, contactores, etcétera.
- Circuitos de amplificación.

Actividades en el aula

2. Ensayo y experimentación de una puerta AND triestado.

Montar el circuito de la Figura 10.7 empleando puertas TTL y alimentar los transistores a +5 V. Realizar su tabla de verdad.

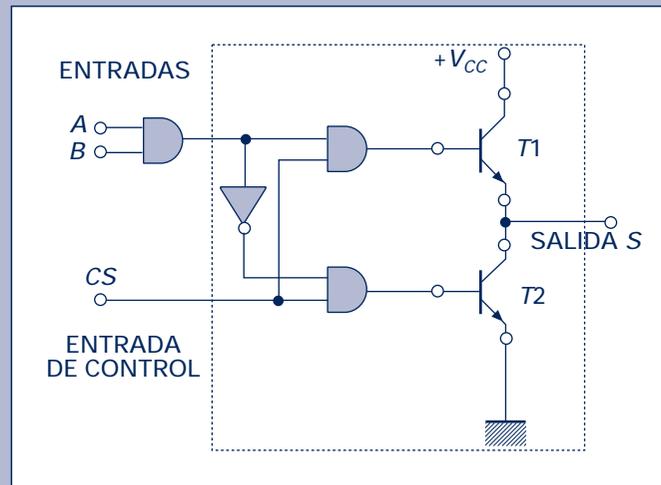


Fig. 10.7. Estructura de una puerta triestado

A. Funcionamiento del circuito

El circuito de la Figura 10.7 está formado por una puerta AND de dos entradas y un circuito adicional de control, que permite desconectar la salida de dicha puerta de circuitos posteriores mediante la entrada de control CS. La estructura del circuito adicional está formada por dos puertas multiplicadoras y una etapa de salida denominada *totem pole*, constituida por dos transistores que trabajan en conmutación.

El funcionamiento del circuito es el siguiente:

- Si $CS = 0$, las dos puertas multiplicadoras aplican un nivel bajo a las bases de los dos transistores, los cuales pasan al estado de corte y la salida S se desconecta de la masa y de la alimentación, quedando así en un estado inactivo o de alta impedancia, equivalente a una desconexión real de la salida de otros circuitos que se encuentren conectados a continuación.
- Si $CS = 1$, la puerta multiplicadora superior, al recibir el nivel alto de CS, transfiere a la base del transistor T_1 el nivel presente en la salida de la puerta controlada por el

bloque adicional de control, mientras que la puerta multiplicadora inferior transfiere a la base del transistor T_2 el nivel inverso al presente en la salida de dicha puerta controlada. Como consecuencia, uno de los dos transistores se saturará y conducirá, mientras que el otro se bloqueará al pasar al estado de corte.

Teniendo en cuenta lo dicho anteriormente, si el nivel presente en la salida de la puerta controlada es 1, el transistor T_1 se saturará y el T_2 se cortará, con lo cual, la salida S se pondrá al nivel de V_{CC} , es decir, a 1.

Asimismo, si el nivel en la salida de la puerta controlada es 0, el transistor T_2 se saturará y el T_1 se cortará, con lo cual, la salida S se pondrá al nivel de masa, esto es, a 0.

B. Lógica triestado

En los sistemas microprogramables basados en un microprocesador, los buses están permanentemente conectados a todos los bloques del sistema; esto supondría, teóricamente, que la información que es enviada por uno de los bloques entraría a todos ellos, lo cual produciría una mezcla de

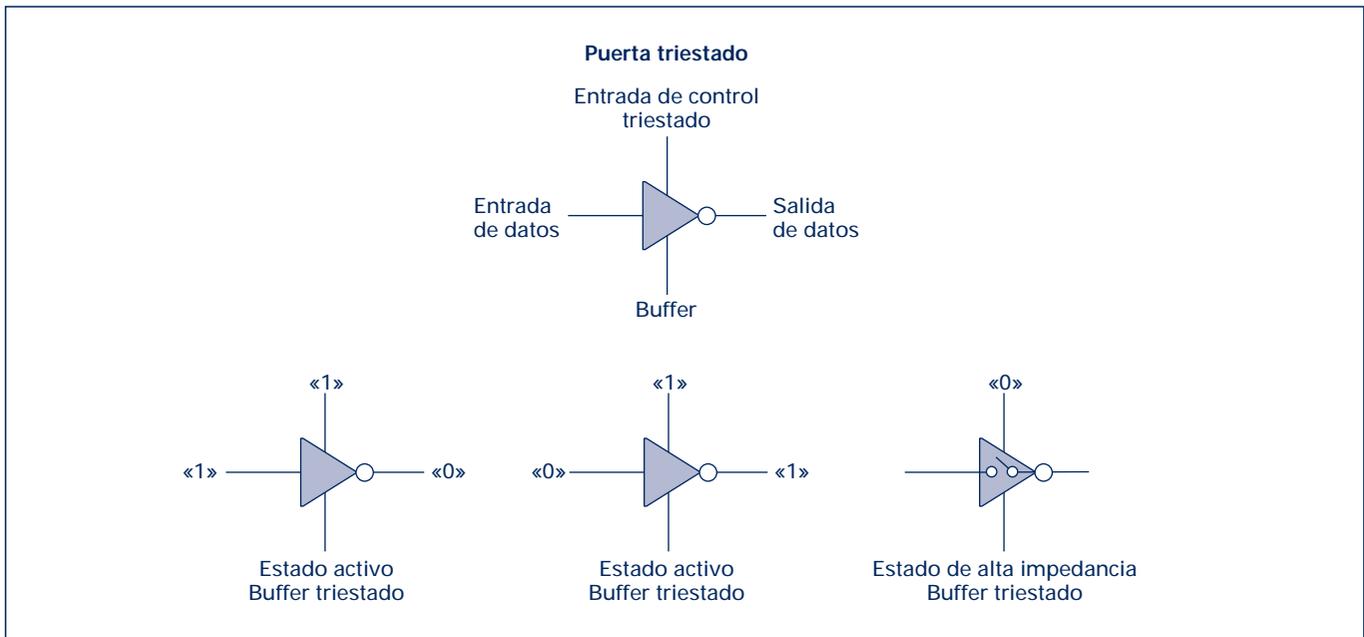


Fig. 10.8. Buffer inversor triestado

informaciones y errores de funcionamiento. La solución al problema anterior consiste en añadir dispositivos de lógica triestado al acceso de cada bus a un bloque. Este tipo de dispositivos trabajan con tres estados de salida posibles que son:

- **Estado 0 (L):** correspondiente a la información 0.
- **Estado 1 (H):** correspondiente a la información 1.
- **Estado inactivo:** correspondiente a la desconexión o alta impedancia.

Los tres estados de las entradas/salidas de cada bloque son controlados por la CPU, mediante señales de *enable* o *chip select*, enviadas a cada uno de ellos en el momento oportuno. De esta forma, el control de la información que circula por los buses es realizado enteramente por la CPU.

El circuito estudiado en la Figura 10.7 permite a cualquier puerta y circuito digital trabajar en lógica triestado, ya que, mediante el terminal CS, una CPU o cualquier otro circuito puede activar o desactivar la salida.

C. Dispositivos y chips triestado

En la práctica, cuando se trabaja con sistemas microprogramables, se utilizan dos tipos de circuitos integrados para lograr lógica triestado:

- **Buffers triestado:** sirven para conectarlos a la entrada y/o salida de un integrado digital normal, y, de este modo, permitir que éste pueda trabajar en lógica triestado. La Figura 10.9 muestra un circuito de este tipo.

Cuando $\overline{CS} = 1$, la transmisión está interrumpida en ambas direcciones (alta impedancia). Si $\overline{CS} = 0$, la transmisión se hace de izquierda a derecha, si la señal $\overline{DIEN} = 0$ y, al contrario, si $\overline{DIEN} = 1$.

- **Integrados digitales triestado:** llevan incluidos en su estructura registros triestado en cada una de sus entradas y salidas. Éste es el caso más habitual en los modelos comerciales de memorias y microprocesadores.

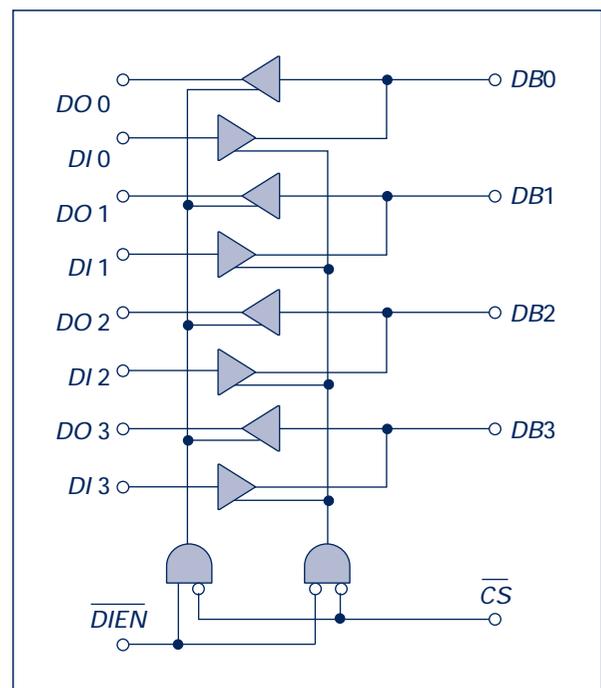


Fig. 10.9. Buffer 8216 Intel

Actividades en el aula

3. Comparación de los lenguajes máquina y ensamblador de diferentes microprocesadores.

Observar listados de programas en lenguaje máquina y lenguaje ensamblador para sistemas de desarrollo con microprocesadores.

A. Generalidades sobre el software de un sistema microprogramable

El software que reciben y procesan los sistemas microprogramables está formado por un conjunto de informaciones que se agrupan de la siguiente forma:

- **Instrucciones:** son las órdenes o tareas que debe realizar el sistema. La estructura habitual de una instrucción se compone de dos partes:
 - **Código de operación:** es la tarea a realizar por el sistema.
 - **Operando:** es el elemento sobre el que se ejecuta la instrucción.
- **Datos:** son grupos o tablas de datos que pueden ser empleados por las instrucciones en su ejecución.
- **Programa:** es el *conjunto ordenado de instrucciones y datos* que ha de recibir el sistema para realizar una determinada función.

Los programas, para ser introducidos en los sistemas con microprocesador, han de ser previamente codificados en binario, ya que tanto la CPU (microprocesador) como el resto del sistema, al ser circuitos electrónicos digitales, sólo son capaces de entender presencia de tensión (1) o ausencia de tensión (0).

La terminología empleada en los procesos de codificación y decodificación de las informaciones con las que trabaja el sistema es la siguiente:

- **Bit:** es la unidad básica de información que está formada por un dígito binario (0 o 1).

- **Palabra:** es el conjunto de bits con que se codifica una información para ser introducida a un sistema microprogramable. Este término es muy importante, ya que sirve para clasificar a los microprocesadores en función de la longitud de palabra con que son capaces de trabajar. Se dice, así, que un microprocesador de 8 bits tiene codificados sus programas con palabras binarias de 8 bits.
- **Repertorio de palabras:** como su nombre indica es el conjunto de palabras diferentes que es capaz de entender o procesar un sistema microprogramable. Su número máximo depende de la longitud de palabra con la que trabaja el sistema, pudiéndose calcular este valor con la siguiente fórmula:

$$\text{Repertorio de palabras máximo} = 2^{N.^{\circ} \text{ de bits de la palabra}}$$

Así, por ejemplo, un μP de 8 bits es capaz de trabajar con un repertorio de 256 palabras u órdenes diferentes, como máximo.

B. Sistema de numeración hexadecimal

Como ya hemos dicho en el apartado anterior, los sistemas microprogramables trabajan con datos e instrucciones codificadas en binario. Dichas codificaciones binarias resultan complicadas para la persona que trabaja con ellas, por lo que los sistemas microprogramables emplean, para facilitar la interpretación de los códigos máquina, la codificación hexadecimal.

El sistema de numeración hexadecimal, también llamado de base 16, tiene la propiedad de que cada dígito hexadeci-

mal se corresponde exactamente con cuatro dígitos binarios. En dicha propiedad está la razón de que el programador trabaje habitualmente con códigos en hexadecimal, en lugar de trabajar con códigos en binario, ya que pequeños circuitos digitales codificadores y decodificadores realizarán sin problemas el paso de un sistema de codificación a otro.

En la estructura del sistema de numeración hexadecimal o de base 16, los números básicos que se utilizan son los comprendidos entre el 0 y el 15 pero, dado que los dígitos entre el 10 y el 15 se forman con combinaciones de dos dígitos básicos, se sustituyen dichas combinaciones por las letras de la A a la F, tal y como aparece en la Tabla 10.1, en la cual se indican, asimismo, sus equivalentes en binario.

Los números en hexadecimal se representan bien indicando su base (16) bien seguidos de la letra H.

Tabla 10.1. Equivalencias entre sistema binario y sistema hexadecimal

Hexadecimal	Binario
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

Resolución de casos concretos

A. Codificar binario en hexadecimal.

Supongamos que deseamos codificar en hexadecimal el número binario $011100011101,00111100_2$.

Para realizar la conversión de binario a hexadecimal se sigue el proceso indicado seguidamente:

- Formaremos grupos de cuatro bits hacia ambos lados de la coma que separa la parte entera de la fraccionaria. Si faltan bits en la parte entera (izquierda), se completan con ceros hasta formar un grupo de cuatro bits; si faltan en la parte fraccionaria (derecha), no se codifica el último grupo incompleto.
- Codificaremos cada grupo de cuatro bits con sus equivalentes en hexadecimal.

0 1 1 1 0 0 0 1 1 1 0 1 , 0 0 1 1 1 1 0 0
 7 1 D , 3 C

El resultado es:

$$011100011101,00111100_2 = 71D,3C_{16}$$

B. Codificar hexadecimal en binario.

Supongamos que deseamos codificar en binario el número hexadecimal $AB1,F3B_{16}$.

Para realizar la conversión de binario a hexadecimal se sigue el proceso indicado seguidamente:

- Cada dígito hexadecimal se codifica por sus cuatro bits binarios equivalentes.

A B 1 , F 3 B
 1 0 1 0 1 0 1 1 0 0 0 1 , 1 1 1 1 0 0 1 1 1 0 1 1

El resultado es:

$$AB1,F3B_{16} = 101010110001,111100111011_2$$

C. Lenguajes de programación

Para codificar la información que se introduce a un sistema microprogramable, es necesario un método o lenguaje que sea fácilmente comprensible tanto para el sistema como para el usuario que realiza la programación. Desgraciadamente, no existe un lenguaje que sea igualmente fácil para el sistema y para el usuario, ya que el primero sólo entiende el binario, mientras que, para el segundo, su utilización es muy compleja.

Los lenguajes de programación se clasifican en función de su proximidad al lenguaje del sistema, en los siguientes tipos: lenguaje máquina o de bajo nivel, lenguaje simbólico o ensamblador y lenguaje de alto nivel.

Lenguaje máquina o de bajo nivel

Es el único lenguaje que entienden los sistemas de desarrollo de microprocesadores y es directamente interpretable y ejecutable por el microprocesador.

Está formado por palabras binarias cuya longitud depende del tipo de microprocesador empleado; así, uno de 8 bits empleará palabras de lenguaje máquina formadas por 8 dígitos binarios.

El problema del lenguaje máquina es que cada microprocesador posee sus propios códigos binarios, o *códigos máquina*, para codificar las instrucciones. Esto provoca que los programas en lenguaje máquina no sean intercambiables en sistemas de desarrollo que emplean diferente tipo de microprocesador.

Según lo explicado en el apartado 10.4.B, se pueden considerar como lenguaje máquina las codificaciones realizadas en hexadecimal.

En la práctica, el programador de sistemas microprogramables realiza los programas en otros lenguajes de más alto nivel y, luego, un *programa compilador, ensamblador o traductor* lo codifica en lenguaje máquina.

Lenguaje simbólico o ensamblador

El lenguaje máquina no es útil para el programador por su elevada complejidad. Por tanto, es preciso el empleo de un lenguaje más cercano al usuario.

El lenguaje simbólico o ensamblador emplea *nemónicos*, que son grupos de caracteres alfanuméricos que simbolizan

a las diferentes órdenes o tareas a realizar con cada instrucción. Los nemónicos se corresponden con las iniciales del nombre de la instrucción en inglés y sirven como regla nemotécnica que facilita su memorización.

Cada microprocesador posee su propio repertorio de instrucciones con sus correspondientes nemónicos. En la Tabla 10.2 aparece una serie de instrucciones y nemónicos, así como sus códigos máquina, para algunos microprocesadores de 8 bits.

Al observar en la Tabla 10.2 los nemónicos de diferentes microprocesadores, nos damos cuenta que son parecidos, pero, en cambio, sus códigos máquina son muy distintos.

Veamos un ejemplo de nemónicos: supongamos que queremos ordenar al microprocesador que sume el acumulador con el dato 5BH y le añada el *carry*. La instrucción será la siguiente en cada uno de los microprocesadores de la Tabla 10.2:

- Para el 6502: ADC #5BH] en código máquina: 69 5B.
- Para el 6800: ADCA #5BH] en código máquina: 89 5B.
- Para el 8085: ACI 5BH] en código máquina: CE 5B.

Ahora podemos comprobar lo indicado anteriormente, ya que, siendo los nemónicos parecidos (ADC, ADCA, ACI), sus códigos máquina son totalmente diferentes (69, 89, CE).

El lenguaje ensamblador es empleado en los sistemas microprogramables destinados al mundo de la automatización y la robótica. En el mundo de la informática el lenguaje ensamblador se utiliza normalmente a nivel de programador y no a nivel de usuario. Los programadores profesionales de microprocesadores emplean programas de ordenador que realizan la estructuración de los programas en lenguaje ensamblador y los codifican de forma automática a lenguaje máquina. Estos programas reciben el nombre de *programas editores-ensambladores*.

Lenguaje de alto nivel

Es el tipo de lenguaje más evolucionado y, por tanto, más cercano al del usuario. Está constituido por conjuntos de caracteres alfanuméricos y signos matemáticos y gramaticales, empleados en el lenguaje convencional y científico.

El nombre de sus instrucciones y sentencias se corresponde con el nombre en inglés de la tarea que realizan. El *set o repertorio de instrucciones* no depende del microprocesador que emplee el sistema de desarrollo con microprocesadores, sino del paquete de software con el que trabaja o de su sistema operativo. Es, por tanto, un tipo de lenguaje más universal.

Tabla 10.2. Tabla con nemónicos de diversos microprocesadores

Tipo μp	Nemónicos	Descripción de la operación	Direccionamiento inmediato (con dato). Código máquina	Direccionamiento directo (con dirección). Código máquina
6502 Rockwell	ADC	Suma de acumulador con el carry y el dato o contenido de la dirección.	69	6D
	LDA	Carga el acumulador con el dato o contenido de la dirección.	A9	AD
	STA	Lleva el contenido del acumulador a la dirección indicada.	—	8D
	JMP	Salta a la dirección indicada.	—	4C
	SEC	Pone a 1 el carry.	38	—
6800 Motorola	ADCA	Suma de acumulador con el carry y el dato o contenido de la dirección.	89	99
	LDAA	Carga el acumulador con el dato o contenido de la dirección.	86	96
	STAA	Lleva el contenido del acumulador a la dirección indicada.	—	97
	JMP	Salta a la dirección indicada.	—	7E
	SEC	Pone a 1 el carry.	OC	—
8085 Intel	ACI	Suma de acumulador con el carry y el dato.	CE	—
	ADC M	Suma de acumulador con el carry y el dato o contenido de la dirección.	—	8E
	MVI	Carga el acumulador con un dato.	3E	—
	LDA	Carga el acumulador con el contenido de la dirección.	—	3A
	STA	Lleva el contenido del acumulador a la dirección indicada.	—	32
	JMP	Salta a la dirección indicada.	—	C3
	STC	Pone a 1 el carry.	37	—

Los lenguajes de alto nivel más importantes son:

- **FORTRAN (1956):** *Formula Translation*. Es uno de los primeros lenguajes de alto nivel, diseñado para la realización de cálculos técnicos y científicos.
- **ALGOL (1958):** *Algorithmic Language*. Es un lenguaje ideado para cálculos numéricos.
- **COBOL:** *Common Business Oriented Language*. Es un tipo de lenguaje especializado en tareas de gestión.
- **BASIC (1965):** *Beginners All-Purpose Symbolic Instruction*. Es el lenguaje de aplicaciones generales más popular hoy en día, existiendo diversas versiones como, por ejemplo, el Q BASIC, el GW BASIC, el MSX BASIC, etc.
- **PASCAL (1969):** es un tipo de lenguaje estructurado que se emplea para generar programas cuya estructura sea fácilmente comprensible. Es parecido al ALGOL y al BASIC.
- **ADA (1975):** es un lenguaje inspirado en el Pascal y con posibilidades de convertirse en un estándar universal.
- **LOGO (1976):** orientado a la enseñanza asistida por ordenador.
- **C:** uno de los lenguajes más modernos, orientado a la automatización y la robótica.
- **LISP:** lenguaje para las aplicaciones de inteligencia artificial.
- **FORTH:** lenguaje moderno que permite crear al usuario sus propias instrucciones.

D. Proceso de programación

Como ya sabemos, con los lenguajes de programación se realizan los programas que el sistema debe procesar. Los programadores profesionales emplean el siguiente proceso en la elaboración de un programa:

- **Obtención del programa fuente:** partiendo de la tarea a realizar, el programador desarrolla un programa en lenguaje ensamblador o de alto nivel. Este programa, que se denomina programa fuente, se logra mediante el empleo de programas editores.
- **Ensamblado:** partiendo del programa fuente y mediante el empleo de *programas ensambladores* se obtiene,

después de eliminar los errores sintácticos cometidos y detectados por el programa, el llamado *programa objeto*.

Este programa objeto se encuentra limpio de errores gramaticales y, si se emplean lenguajes de alto nivel, totalmente codificado en código máquina y listo para ser ejecutado.

- **Linkado:** sólo en el caso de emplear lenguajes ensambladores es necesario que el programa objeto sea revisado por un *programa linkador*, que reajusta las direcciones y localizaciones internas del programa, generando un programa codificado totalmente en código máquina que puede ser directamente ejecutable.

10.5 Hardware de un sistema microprogramable con microcontrolador

A. Esquema de bloques de un microcontrolador

La evolución de los microprocesadores ha llevado a la aparición del **microcontrolador**, que consiste en la integración en un solo chip de todo el sistema de desarrollo de microprocesadores; esto es, integrar la CPU, la memoria RAM y la ROM, las unidades I/O, e incluso algunos periféricos o sistemas de control externo, como pueden ser los convertidores Analógico/Digital o Digital/Analógico. La Figura 10.10 nos muestra el esquema de bloques que suelen poseer estos dispositivos.

Actualmente existen en el mercado gran cantidad de microcontroladores, así como versiones de reducido tamaño que reciben el nombre de **microcontroladores PIC**.

B. Características del microcontrolador

Un microcontrolador presenta las siguientes características:

- Es capaz de procesar datos de 8 bits y, en algunos casos, de 16 bits.
- Tiene un **contador de programa (PC)** de 16 bits, empleado solamente para la búsqueda del *código* (nombre dado al programa codificado en código máquina) localizado en su memoria ROM-EPROM interna.
- Posee un **puntero de datos (DPTR)** de 16 bits, utilizado como contador de programa en la búsqueda de datos en la memoria externa que, cuando existe, es normalmente de tipo RAM.

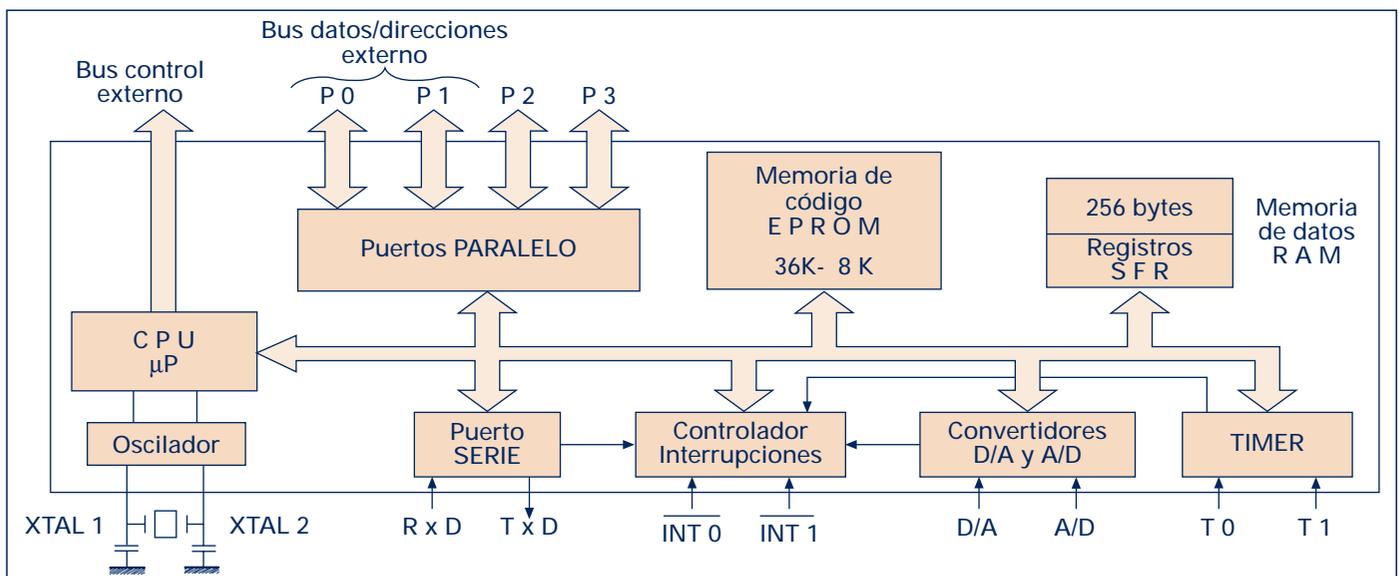


Fig. 10.10. Esquema de bloques de un microcontrolador

- Contiene un **puntero del pila** de 8 bits. La pila de memoria se localiza en su RAM interna.
 - Tiene una **unidad aritmético-lógica**, que es capaz de sumar, restar, multiplicar, dividir, realizar comparaciones y operaciones AND, OR, OR exclusiva y complementar.
 - Lleva en su interior los llamados **registros SFR** (*Special Function Registers*). Estos registros poseen múltiples aplicaciones y están localizados en direcciones de la memoria RAM interna. Entre ellos destaca el acumulador.
 - La memoria del microcontrolador se organiza en dos bloques:
 - **Memoria de código:** diseñada para contener los programas, se localiza sobre memoria de tipo ROM o EPROM.
 - **Memoria de datos:** diseñada para contener datos, se localiza sobre memoria RAM y se implementa de la siguiente forma: 256 bytes de memoria interior del microcontrolador, distribuidos en 128 bytes libres y 128 bytes empleados por los registros SFR.
- Los microcontroladores pueden contener las siguientes unidades I/O integradas:
 - **Puertos paralelos** de 8 o 16 líneas cada uno.
 - **Temporizadores/contadores programables** de 16 bits.
 - **Puertos serie**, que trabajan en modo *full duplex*.
 - **Controlador de interrupciones**, que admite varias posibles fuentes de interrupción.
 - **Convertidores A/D y D/A.**

Ejercicios propuestos

- Convertir a binario las siguientes cantidades hexadecimales:

a) $3F0_{(16)}$	b) $4A7_{(16)}$	c) $123_{(16)}$
d) $0,10F_{(16)}$	e) $62,1E_{(16)}$	f) $A30,22_{(16)}$
- Realizar el cambio de estas cantidades hexadecimales a binario:

a) $FA237CE,47D_{(16)}$
b) $FFF499B,CC1_{(16)}$
c) $AAFF471,CD5_{(16)}$
- Convertir a binario las siguientes cantidades hexadecimales:

a) $12002_{(16)}$	b) $10AF_{(16)}$
c) $C1AD_{(16)}$	d) $1111_{(16)}$
- Convertir a hexadecimal las siguientes cantidades binarias:

a) $10111101101_{(2)}$
b) $101010101111100_{(2)}$
c) $1010110011011111_{(2)}$
d) $101101010110_{(2)}$
e) $11110111010_{(2)}$
f) $111000000001010_{(2)}$
g) $100110101,01011101_{(2)}$
h) $1011000,1101010_{(2)}$
i) $1011111,0010001_{(2)}$
j) $111011010101_{(2)}$
k) $10101010101011_{(2)}$
l) $1010,1101101_{(2)}$
m) $0,1111001010_{(2)}$
n) $1011011101,11101111_{(2)}$
ñ) $100011,00110011_{(2)}$
o) $0,10101111_{(2)}$
- Un programa situado en la memoria RAM de un sistema microprogramable tiene su comienzo en la dirección 4F3BH y su final en la dirección 4F50H. Codificar dichas direcciones en binario y calcular el número de líneas de memoria que emplea.
- Indicar qué tres números siguen a cada uno de los siguientes números hexadecimales:

a) 10H
b) 2FEH
c) 042DH
d) A40FH
e) FF10H
f) C23EH
g) F27AH
h) FFFFH