TEMA VII: DISEÑO SECUENCIAL PROGRAMABLE

De igual forma que podíamos disponer de dispositivos combinacionales programables para poder implementar funciones combinacionales en un solo integrado, en el dominio secuencial también podemos encontrar dispositivos similares.

Existen diferentes grados de complejidad de estos sistemas programables, los cuales fueron apareciendo a medida que han ido aumentando las capacidades tecnológicas.

1. Introducción.

Los dispositivos secuenciales programables siguen el esquema básico de cualquier sistema secuencial, es decir, constará de un bloque combinacional, que generará las diferentes funciones lógicas, y un conjunto de biestables, que almacenarán el estado en el que se encuentre el sistema, como podemos ver en la figura 7.1. Según las misiones de las dos partes fundamentales, se observa que la programabilidad sólo es necesaria en el bloque combinacional, ya que la misión de los biestables es la misma independientemente del comportamiento del sistema.

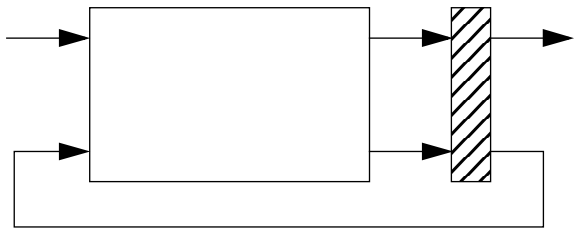

Figura 7.1.- Esquema básico de un sistema secuencial.

A diferencia de los dispositivos programables combinacionales, sus contrapartidas secuenciales han sufrido una mayor evolución motivada por dos causas fundamentales:

- La mayor utilización de los dispositivos secuenciales.
- La necesidad de aumentar la velocidad de sistemas grandes, que desembarcó en la técnica de *pipeline*.

Algunas de las características, y por tanto divisiones, de los dispositivos combinacionales se encuentran en los secuenciales. Así podemos encontrar dispositivos reconfigurables, en

los que podemos devolver los fusibles a su estado inicial y así configurarlo más de una vez, y dispositivos configurables, en los que la configuración es única.

A continuación vamos a tratar los principales sistemas programables respetando su evolución natural en la mayor medida de lo posible.

2. Sistemas basados en dispositivos combinacionales programables.

Siguiendo estrictamente el esquema general de un sistema secuencial, estos sistemas no están integrados en un solo chip, sino que disponemos de un integrado para el dispositivo combinacional y otro para los elementos de memoria.

Así, la misma clasificación que teníamos para los dispositivos combinacionales puede ser aplicada a estos sistemas, es decir, podemos encontrar:

- Sistemas completos, los cuales son capaces de implementar cualquier función con un número de entradas dado. Estos sistemas estarán formados por una memoria ROM o PROM y biestables, como se puede ver en la figura 7.2.

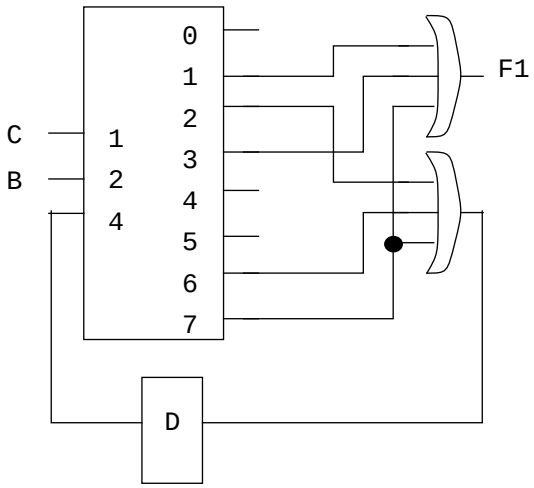

Figura 7.2.- Sistema secuencial programable basado en memorias ROM o PROM.

- Sistemas incompletos, los cuales sólo son capaces de implementar las funciones con un número determinado de entradas que cumplan una serie de restricciones como pueden ser el número de términos productos. Estos sistemas estarán formados por dispositivos PAL o PLA y biestables, como se puede ver en la figura 7.3.

El principal problema de esta solución consiste en el número de entradas y salidas. El factor básico de limitación de los dispositivos programables combinacionales es el número de entradas y salidas. Y debido a la necesidad de extraer fuera del dispositivo la dependencia temporal, a través de las señales de estado, estamos reduciendo aun más este factor limitante. Así si tenemos un dispositivo combinacional de N entradas y M salidas, y el sistema que vamos a diseñar tiene Q señales de estado, el dispositivos equivalente secuencial pasará a tener N-Q entradas y M-Q salidas (en el caso más favorable en el que los biestables sólo tengan una señal de entrada, tipo D o T) o M-2Q (en el caso más desfavorable en el que los biestables tengan dos

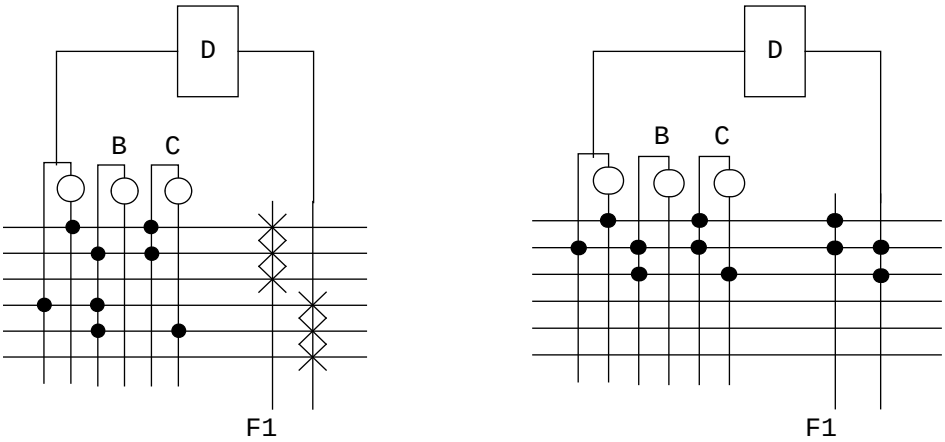

Figura 7.3.- Sistema secuencial programable basado en dispositivos PAL y PLA combinacionales.

señales de entrada, tipo RS o JK). Es decir, necesitamos Q señales de entrada para introducir las señales de estado presente, y Q (o 2Q) señales de salida para obtener las señales de próximo estado. En el ejemplo mostrado en las figuras anteriores pasamos de tener 3 entradas y 2 salidas a tener 2 entradas y 1 salida.

3. Sistemas secuenciales programables monochip.

El siguiente paso consiste en integrar todo el sistema secuencial en un solo integrado, es decir, la lógica combinacional y los elementos de memoria. Por lo tanto, las principales características de estos elementos serán el número de entradas, el número de términos producto, el número de biestables y el número de salidas, así como el tipo de biestables que utilizan (del cual van a depender las funciones de próximo estado).

Dentro de los primeros sistemas secuenciales programables podemos encontrar el **secuenciador lógico programable** o **PLS** (Programmable Logic Sequenciator). Este sistema está basado en una arquitectura PLA (matrices AND y OR programables) a la que se le ha añadido un conjunto de biestables (tipo D por lo general) con un reloj común, como podemos apreciar en la figura 7.4.

En la figura anterior podemos apreciar la manera en la cual podemos tener conectada las salidas de los biestables:

- Sus salidas se realimentan internamente a la matriz AND, sin ser accesibles desde el exterior.
- Sus salidas se realimentan internamente a la matriz AND, y además son accesibles desde el exterior.
- Sus salidas sólo se conectan al exterior, sin contar con ninguna realimentación interna.

Estos dispositivos son especializados para el diseño síncrono ya que todas sus salidas están conectados a biestables y todos ellos son disparados por la misma señal de reloj. Por lo tanto, los PLS no se pueden utilizar en el diseño de sistemas asíncronos, ya sean del modo fun-

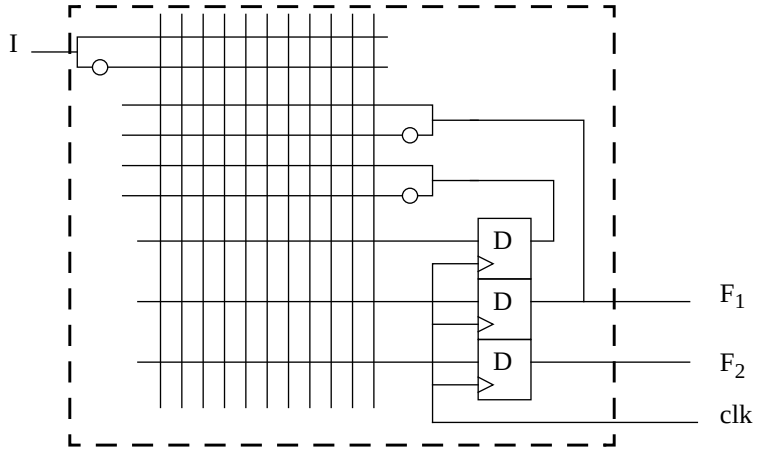

Figura 7.4.- Esquema básico de un PLS.

damental o autotemporizados.

Análogos a los PLS podemos encontrar sistemas basados en arquitecturas PAL (matriz AND programable y OR fija). La principal diferencia con los PLS radica en que estos dispositivos suelen ser más generales. La mayor generalidad consiste en el sentido de que disponen una serie de salidas combinacionales y otra serie de salidas a través de biestables, como se puede ver en la figura 7.5. Debido a esta generalidad, este tipo de dispositivos reciben el nombre genérico de **PAL**.

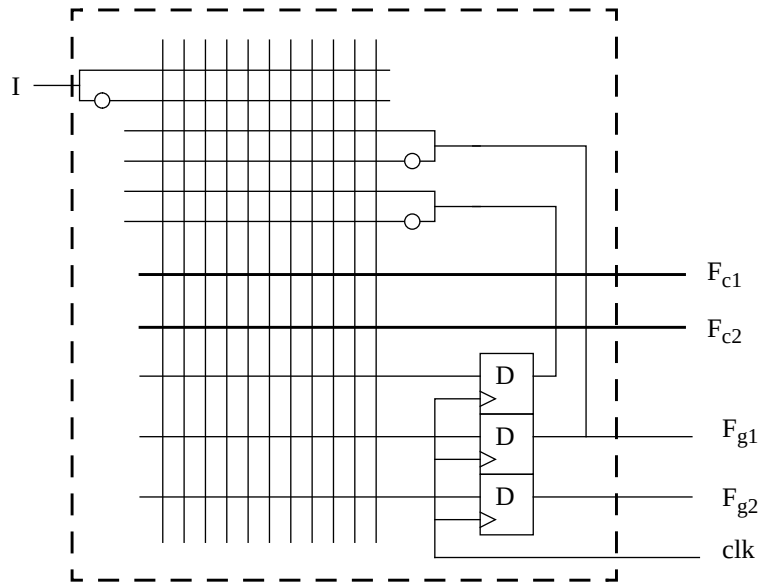

Figura 7.5.- Esquema básico de un PAL.

Este nuevo dispositivo tiene el problema de la rigidez del tipo de salidas, siendo ésta una nueva limitación. Esta rigidez viene determinada porque no podemos cambiar salidas combinacionales por registradas (colocadas después de un biestable) y viceversa. Luego, estos dispositivos tendrán algunas salidas sin aprovechar.

Para tratar de paliar este problema, surgió un nuevo dispositivo programable denominado **GAL** (Generic Array Logic). Estos dispositivos están basados en un arquitectura PAL, cuyas salidas atacan a un bloque o macrocelda de salida. Esta macrocelda es programable de tal forma que podemos utilizar las salidas de forma directa (como si fuesen combinacionales o asíncronas) o a través de un biestable (como si fuesen síncronas), además de ser realimentadas hacia el interior sin necesidad de utilizar ningún terminal de entrada. Un posible esquema de estas macroceldas puede ser el mostrado en la figura 7.6. Cada macrocelda tiene dos señales de control para configuración: la señal S_0 saca al exterior la salida complementada o sin complementar, según valga '1' ó '0'; mientras que la señal S_1 selecciona la salida combinacional o registrada., según valga '1' ó '0'.

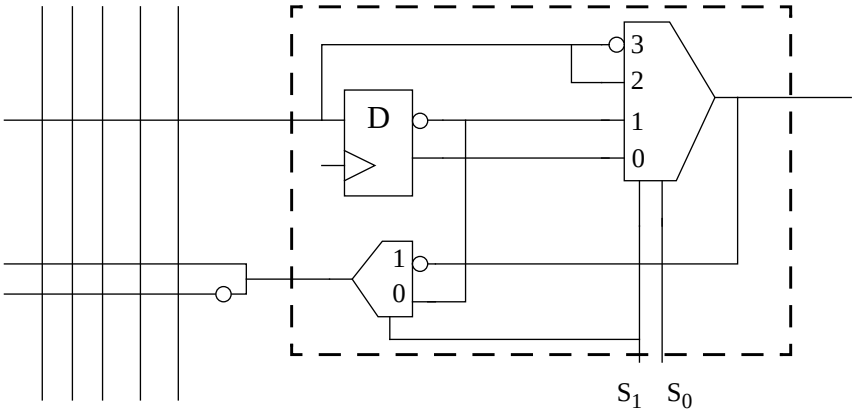

Figura 7.6.- Esquema básico de un macrocelda de salida correspondiente a una GAL.

4. Dispositivos programables avanzados.

Con los GAL llegó el despegue de la Informática, y por tanto, los diseños electrónicos se empezaron a realizar mediante software. No obstante, también se ha seguido avanzando en el diseño electrónico hardware.

Cuando adquirió importancia la técnica de *pipeline*, y por lo tanto, el particionado de sistemas en bloques más pequeños, los esquemas anteriores tuvieron la desventaja de desaprovechar salidas para generar las etapas de *pipeline*. Esto es debido a que necesitamos biestables para cada etapa de pipeline pero no necesitamos tener accesibles ninguna de las salidas de las etapas intermedias (salvo en el caso de prototipos experimentales y para cuestiones de testado). Debido a esta nueva necesidad, se requiere la presencia de un mayor número de biestables aunque no sean todos accesibles desde el exterior. Luego, se originó una nueva generación de dispositivos programables que podemos denominar **CPLD** (Complex Programmable Logic Device). Su nomenclatura no está bien definida, y cada familia da sus propios nombres. De entre los fabricantes que se han introducido en este campo, destacan las empresas XILINX (que denomina a sus dispositivos FPGA) y ALTERA (que los denomina EPLD).

Estos dispositivos tienen la misma arquitectura básica. Como se muestra en la figura 7.7, existen tres partes fundamentales:

- Bloques de procesado (CLB para XILINX y LAB para ALTERA).

- Bloques de entrada/salida (IOB para XILINX).
- Matriz de conexiones (PIA para ALTERA).

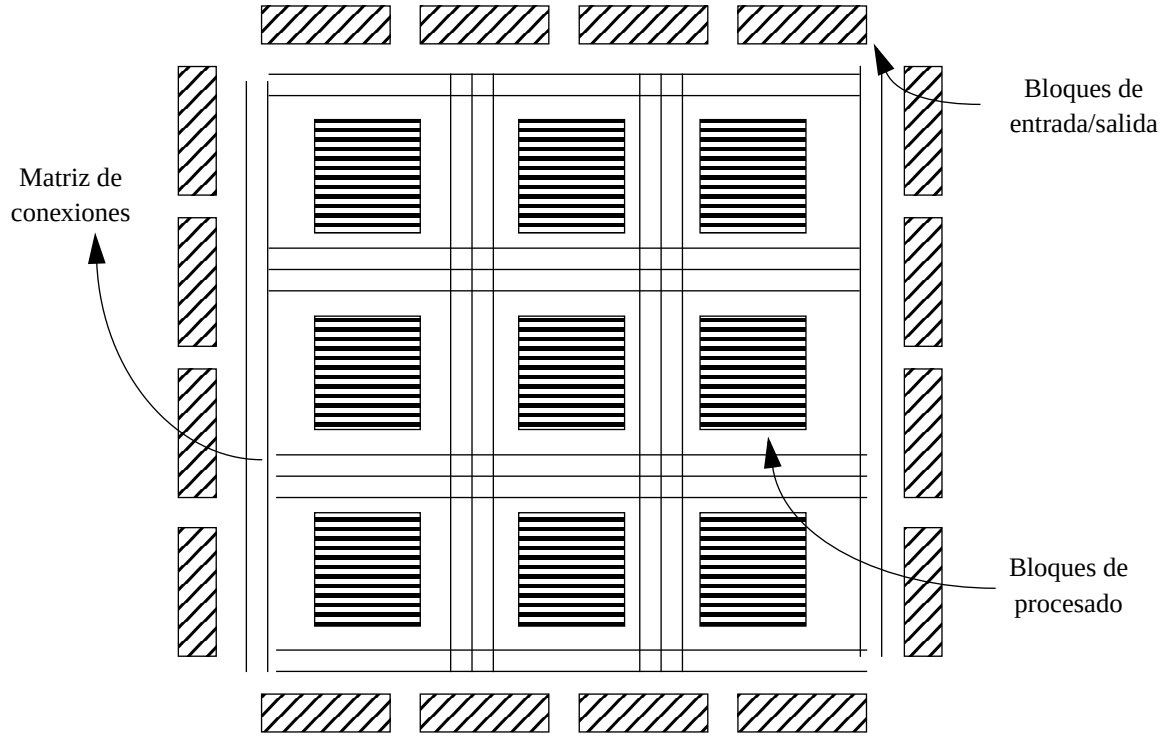

Figura 7.7.- Arquitectura básica de un CPLD.

Los bloques de procesado pueden ser vistos como pequeñas GAL, en la que las etapas de salida son algo más complejas. Un esquema simplificado para un bloque de procesado se muestra en la figura 7.8. Los bloques que generan las funciones combinacionales G, F y H son bloques completos, por lo que pueden generar cualquier función lógica de cuatro (F y G) y dos entradas (H). En cuanto a las entradas, disponemos de una serie de entradas de control (que no se muestran en la figura por motivos de claridad) cuya misión básica es seleccionar las salidas del bloque, nueve entradas de datos (D, F's y G's) y una señal de reloj. En cuanto a las salidas, disponemos de dos salidas registradas, que pueden ser F, G, H o D, y dos salidas combinacionales, que pueden ser F, G o H. Como podemos ver se mantienen todas las mejoras introducidas en las GAL.

Los bloques de entrada/salida consiguen eliminar la restricción rígida de limitar el número de entradas y el de salidas. En estos dispositivos, la limitación se encuentra en el número de puertos, ya que todos los bloques de entrada/salida pueden ser configurados como entradas o como salidas de forma independiente. Un esquema simplificado de un bloque de entrada/salida se muestra en la figura 7.9. En estos bloques podemos acceder a las señales directamente o pasando previamente por un biestable, a partir de una señal de selección que no ha sido incorporada al esquema. Los biestables de entrada/salida han sido incluidos para permitirnos sincronizar las señales que pasan a través de este bloque. Además de esta señal, también tenemos una señal de control, T, que deshabilitará el camino de salida a través de un buffer triestado. Adicionalmente tendremos dos señales de datos, *In* y *Out*, y dos relojes para

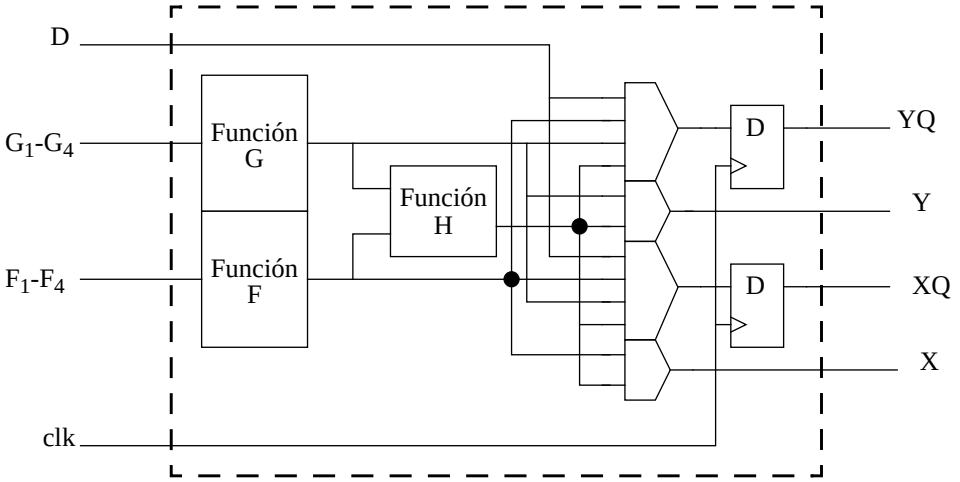

Figura 7.8.- Esquema básico de un bloque de procesado.

cada biestable, clk_i y clk_o. La programación de entrada o salida se realiza mediante la utilización del terminal correspondiente, *In* o *Out*.

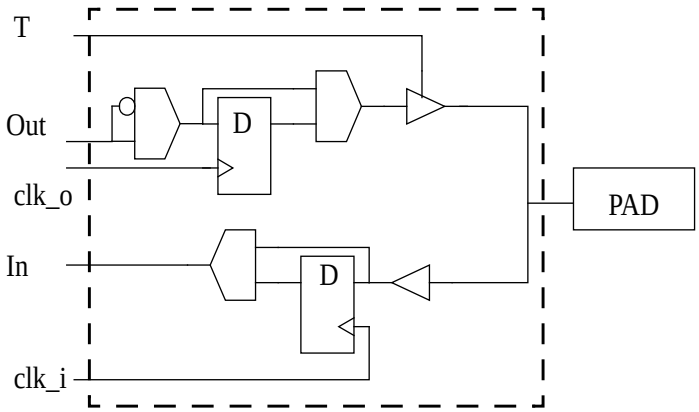

Figura 7.9.- Esquema básico de un bloque de entrada/salida.

Por último, la matriz de conexiones suele ser el elemento más limitante en la utilización de este tipo de dispositivos. Este bloque es el que evita que tales dispositivos no sean completos, ya que los bloques de procesado lo son. A medida de que se van sacando nuevos dispositivos, casi siempre va mejorando esta matriz, ya sea en mayor número de conexiones y/o en mejores prestaciones (como puede ser la velocidad).

5. Microcontroladores.

Estos dispositivos realizan la programación mediante instrucciones software, es decir, introducimos el comportamiento del sistema a través de una sucesión de instrucciones a modo de programa.

Los microcontroladores surgen cuando se pretende llevar la generalidad al máximo, es decir, implementa una gran cantidad de instrucciones, que son más comprensibles para el usuario. Luego, estos dispositivos serán utilizados cuando el comportamiento del sistema es excesivamente complejo, y utilizamos suficientes recursos del microcontrolador como para que sea rentable. Así el microcontrolador realizará la misma operación tantas veces como queramos, pero **sólo puede ejecutar un programa**. En adición a los microcontroladores, también podríamos utilizar microprocesadores; no obstante, estos últimos entienden un número muy elevado de instrucciones como para limitar su operación a un solo programa.

La arquitectura básica de un microcontrolador, que es la misma que la de un microprocesador, se muestra en la figura 7.10. Esta basado en una unidad central de proceso (CPU), un sistema de memoria y un sistema de entrada/salida.

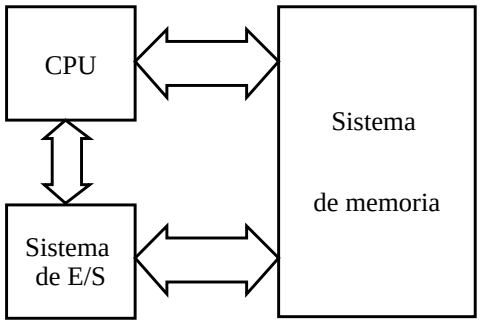

Figura 7.10.- Arquitectura básica de un sistema microcontrolador.

La unidad central de proceso es la parte encargada de realizar las operaciones requeridas. Estas operaciones son introducidas a través de instrucciones de programa, que debe ser almacenado en el sistema de memoria. Esta unidad se divide en:

- Unidad de control, la cual está encargada de interpretar las instrucciones y realizar las conexiones necesarias para poder realizar las diferentes operaciones.
- Unidad aritmético-lógica, la cual está encargada de realizar las operaciones que le indique la unidad de control.

El sistema de memoria es la parte encargada de almacenar los datos necesarios para la correcta realización del desarrollo del programa. Por lo general, este sistema se divide en dos partes:

- Memoria de programa, la cual está encargada de almacenar el programa que ejecutará el microcontrolador. Al ser este programa único, solamente será necesario almacenar una sola vez, y no cada vez que se conecte el microcontrolador; luego este memoria será del tipo no volátil. Los tipos más utilizados son memorias EPROM-FLASH
- Memoria de datos, la cual está encargada de almacenar los datos con los que el programa realizará las diferentes operaciones. Como los datos sólo serán útiles cuando el programa esté ejecutándose, no es necesario mantenerlos cuando el microcontrolador esté desconectado; luego utilizará memorias no volátiles. Los tipos más utilizados son memorias RAM dinámicas debido a la mayor rapidez y menor coste.

El sistema de entrada/salida es la parte utilizada para la comunicación con el usuario. A través de este sistema, el usuario puede obtener los datos de salida, introducir los datos de entrada, así como realizar cambios en el contenido del programa a ejecutar.

5.1. Ejecución microprogramada

Hasta este momento, el tipo de operación que se debe realizar en cada momento no tiene una relación directa con los datos de entrada, sino que éstos me determinan los valores con los que deberá trabajar el sistema. A este tipo de operación se denomina **operación cableada**. Por lo tanto, la siguiente operación a realizar dependerá directamente del estado del sistema y no de los valores de las señales de entrada.

En contraposición a este tipo de operación, tenemos la **operación microprogramada**. En este tipo, el sistema será de carácter general, y los datos de entrada tendrán que indicar explícitamente la operación que hay que realizar en cada momento.

En la figura 7.11 mostramos parte del esquema de una CPU basada en un acumulador. En ella podemos distinguir varias partes:

- La unidad aritmética-lógica (ALU), que realiza las diferentes operaciones en función de unas determinadas señales de control.
- El acumulador, que es el registro donde se almacenará el resultado de la operación de la ALU.
- El registro de instrucciones, donde se almacenará la instrucción que se deba realizar en ese momento. Este registro de instrucciones tiene (en este caso particular) dos partes bien diferenciadas: una parte de código de operación, que me indica la operación que vamos a realizar; y otra parte de datos, donde se encuentra el dato con el que se realizarán las operaciones.

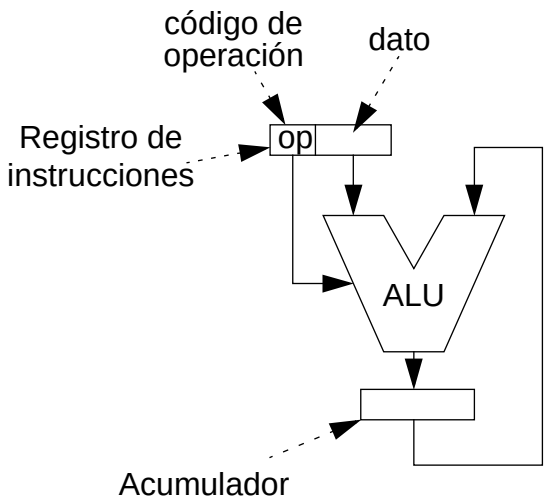

Figura 7.11.- Parte del esquema de una CPU basada en acumulador.

Veamos cómo se realizaría la suma de dos datos, D1 y D2. Para ello, la ALU debe poder realizar dos operaciones posibles:

- Cargar en el acumulador, identificada como LOAD, y con un código de operación 01.
- Sumar con el acumulador, identificada como SUM, y con un código de operación 10.

El esquema temporal de estas operaciones se muestra en la figura 7.12. La secuencia de operaciones para realizar una suma microprogramada es la siguiente:

- En primer lugar se deberá realizar una carga en el acumulador del dato D1, por lo que el código de operación será “01”, y en el campo de dato estará almacenado D1. De esta forma, la ALU dejará pasar directamente este dato al acumulador que será almacenado en el siguiente ciclo de operación.
- Seguidamente, en el registro de instrucciones se almacenará la siguiente instrucción, es decir, suma del dato D2 con el contenido del acumulador. Luego, el código de operación será “10” y en el campo de dato estará almacenado D2. Así, la ALU realizará la suma del contenido del acumulador con el contenido del campo de dato. Este resultado será almacenado en el acumulador en el siguiente ciclo de operación.

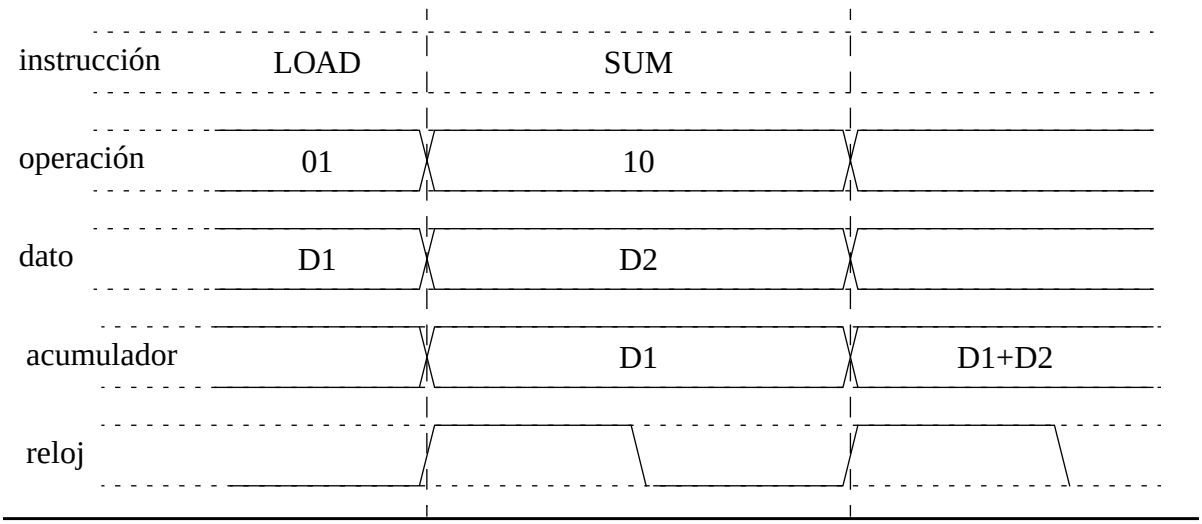

Figura 7.12.- Ejemplo de una suma microprogramada.

Por lo tanto, con estas dos instrucciones realizamos la suma de dos datos.

La ventaja de esta técnica es que únicamente cambiando las instrucciones que deseamos realizar, cambiaremos la operación del sistema sin necesidad de alterarlo de forma física. Por lo tanto, debido a que el sistema no se altera, estamos ante un sistema programable, pero en lugar de físicamente, este sistema es programable a través de un programa.