

Electrónica Digital

Ingeniería Informática – FICH, UNL
Leonardo Giovanini



Microprogramación

En esta se estudiarán los siguientes temas:

- Sistemas microprogramados;
- Estructura;
- Logica de control;
- Lógica de secuenciación.

Un **algoritmo** es una **secuencia finita de instrucciones**, o reglas, **no ambiguas, ordenadas, finitas e implementables**, que se usa para resolver problemas, realizar un cálculos y razonamientos automatizados, procesar datos y llevar a cabo cualquier tarea.

Los algoritmos pueden ser **expresados de muchas maneras**, incluyendo al pseudocódigo, diagramas de flujo, automatas y lenguajes de programación.

Los **automatas** son tan precisos como un **lenguaje máquina** para describir un algoritmo.

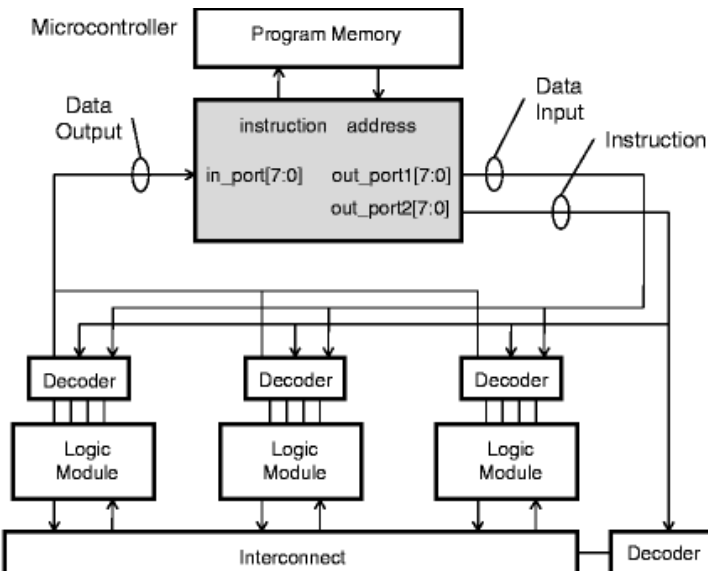


Un lenguaje de máquina es un **sistema de códigos interpretable** por el controlador de un autómata.

Está compuesto por un conjunto de instrucciones que determinan las operaciones que puede ejecutar el automata.

Estas **operaciones elementales** de procesamiento de información se denominan **microinstrucción**.

Un **sistema microprogramado** consiste en un conjunto de elementos que permiten **ejecutar una secuencia de microinstrucciones**.



La unidad de control microprogramada incluye las lógicas para secuenciar a través de microoperaciones, ejecutar las microoperaciones y tomar decisiones basadas en entradas externas.

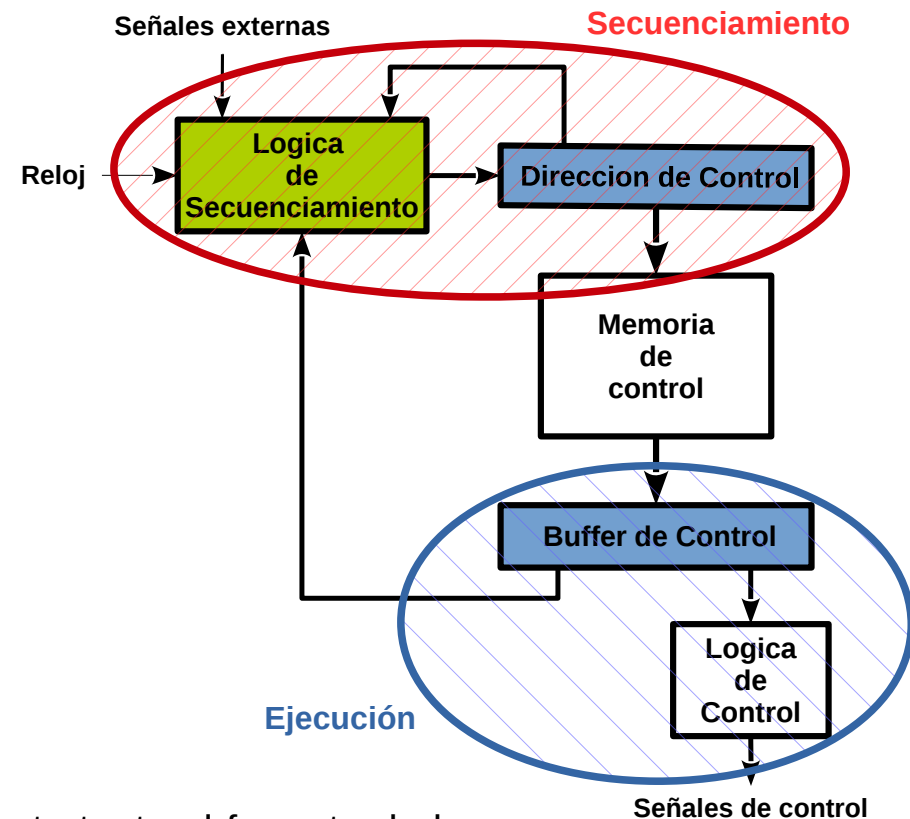
Entonces su **arquitectura está claramente definida, con muy pocas variantes**.

La unidad de control realiza lecturas secuenciales de la memoria de control para generar las salidas de control correspondientes.

La unidad de control microprogramada realiza **dos tareas básicas**:

- **Secuenciamiento** – la unidad de control busca y obtiene la próxima microinstrucción a ejecutar de la memoria de control;
- **Ejecución** – una vez disponible, la unidad de control genera las señales de control necesarias para ejecutar la microinstrucción y generar las señales de salida .

El diseño de la unidad de control debe tener en cuenta tanto el formato de la microinstrucción como la sincronización de la unidad de control.



Un sistema microprogramado consiste esencialmente en un conjunto de elementos que permiten almacenar y ejecutar la secuencia de microinstrucciones.

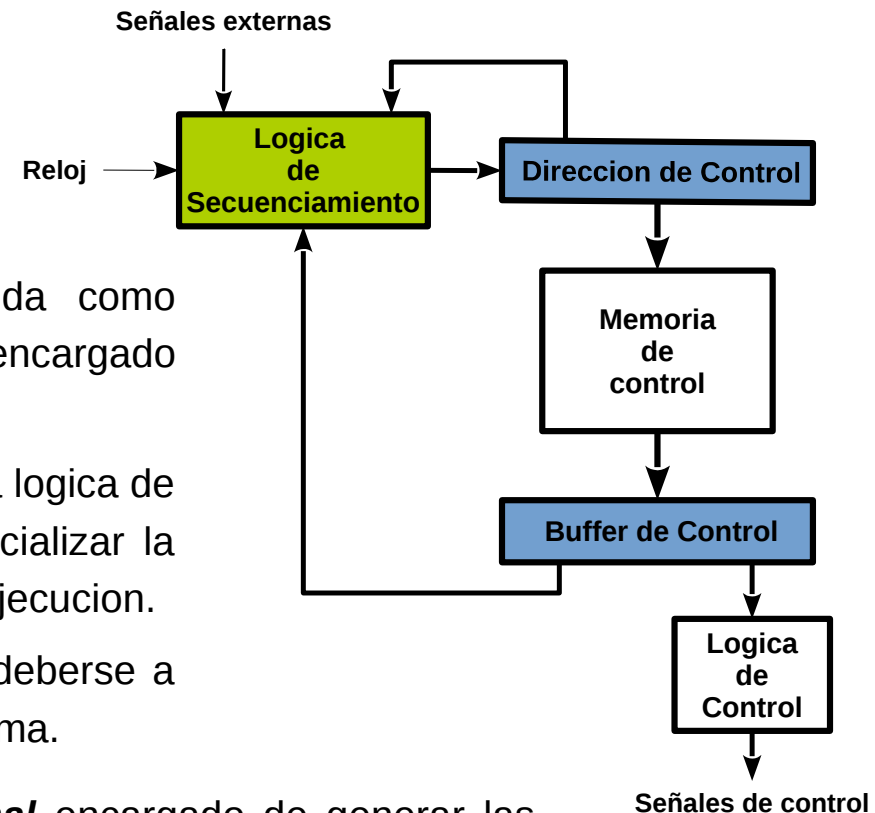
Memoria de programa – es una memoria de sólo lectura que almacena las operaciones a realizar codificadas como una secuencia de microinstrucciones (microprograma);

Lógica de secuenciamiento – también conocida como *microsecuenciador*, es un **circuito secuencial** encargado de **elegir la próxima microinstrucción a ejecutar**.

Básicamente es un contador al que se le agrega una lógica de inicialización y una lógica de salto que permite inicializar la ejecución del microprograma y modificar el flujo su ejecución.

Lo cambios en la secuencia de ejecución pueden deberse a cambios programados y/o entradas externas al sistema.

Lógica de control – es un **circuito combinacional** encargado de generar las señales de control a partir de información almacenada en las microinstrucciones y señales externas.



Microprogramacion – Estructura del sistema

Lógica de control

El ciclo de microinstrucción es el evento básico en un sistema microprogramado, que se compone de dos partes: i) búsqueda y ii) ejecución de las microinstrucciones.

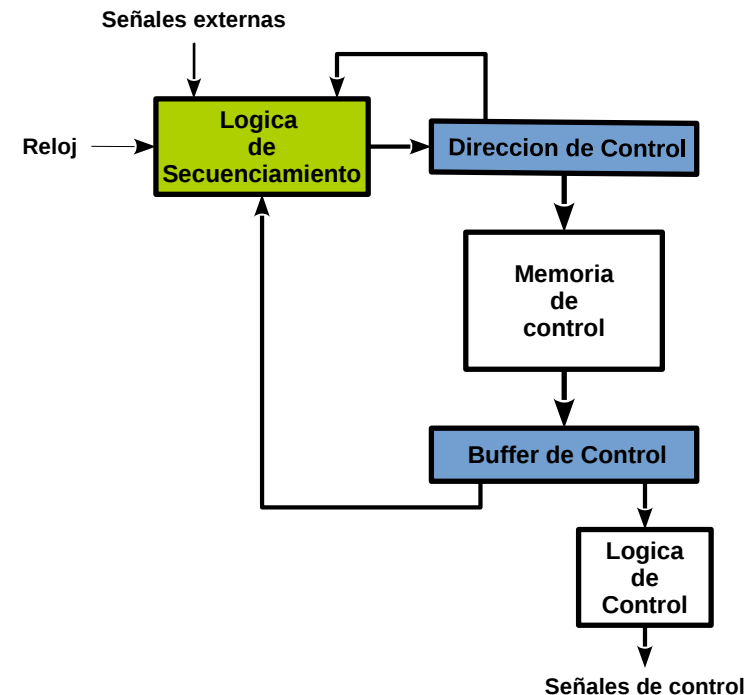
Para realizar la **busqueda de microinstrucciones**, la unidad de control puede utilizar un microprograma común.

Este microprograma se **ejecuta al comienzo de la ejecución de cada instrucción**, con la instrucción almacenada en el microsecuenciador, se calcula la dirección del microprograma correspondiente.

Después de esto, el microsecuenciador controla la ejecución del microprograma que genera la secuencia señales de control apropiadas.

Durante la ejecución de un microprograma, el **microsecuenciador se incrementa cada vez que se extrae una microinstrucción** de la memoria, excepto en las siguientes situaciones:

- Cuando se **encuentra** una instrucción que indica el **fin del microprograma**;
- Cuando se **carga una nueva instrucción**, el microsecuenciador se carga con la dirección inicial del microprograma correspondiente a esa instrucción;
- Cuando se encuentra una **microinstrucción de salto** y se cumple la condición para ejecutar el salto.

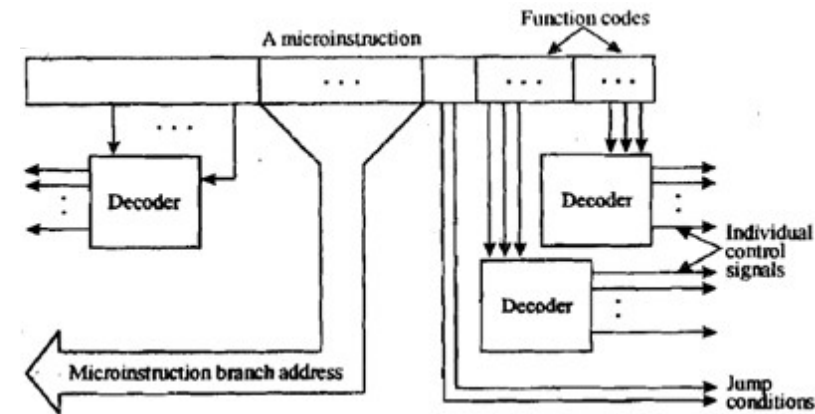
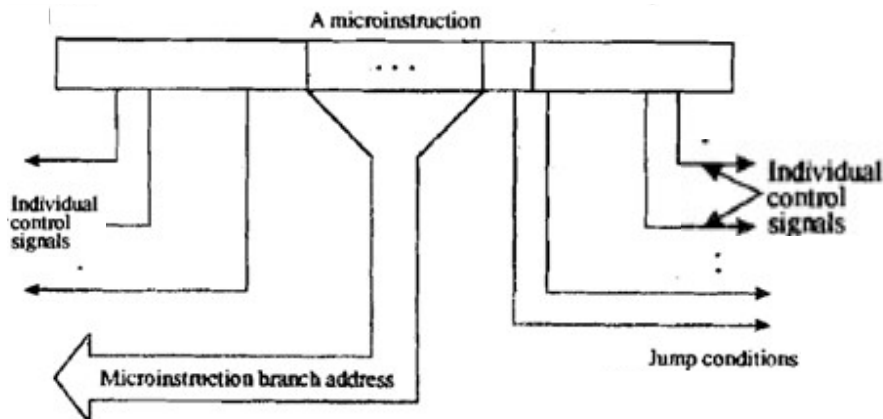


El diseño de un sistema microprogramado implica resolver dos problemas: i) la información contenida en la microinstrucción y ii) la generación de la dirección.

Hay dos formas en las que la información se puede organizar en una microinstrucción:

Microprogramación vertical – cada microinstrucción especifica una sola microoperación a realizar a través del uso de codificación de información.

- El ancho de la palabra de memoria es estrecho ya que se especifica un código para las señales;
- Capacidad limitada para expresar paralelismo;
- Requiere un decodificador para identificar la línea de control exacta que se está manipulando.



Microprogramación horizontal – cada microinstrucción especifica muchas microoperaciones diferentes que se realizan en paralelo a partir de generar todas las señales que se usan.

- Palabra de memoria amplia;
- Alto grado de operaciones paralelas son posibles.

Lógica de secuenciación – también conocida como **microsecuenciador**, es un **circuito secuencial** encargado de **elegir la próxima microinstrucción a ejecutar**.

Básicamente es **un contador con una lógica de carga paralela** que permite **cargar una dirección arbitraria** en el contador para inicializar la ejecución del microprograma y modificar el flujo de ejecución, o solo incrementar la dirección en ejecución de instrucciones secuenciales.

Cada microinstrucción **proporciona las señales de control** requeridas en ese paso de tiempo y el microsecuenciador **asegura que esas señales** se envíen al sistema en la **secuencia correcta**.

La ejecución de algunas instrucciones depende del estado de códigos de condición, indicadores de estado y/o entradas externas.

Para manejar este tipo de instrucciones, la unidad de control implementa la ejecución de saltos condicionales, los cuales requieren especificar la dirección de la memoria a la que se debe dirigir el control (dirección de salto) y cuáles de los códigos de condición, bits del registro de instrucciones y/o entradas deben verificarse como condición para que se produzca el salto.

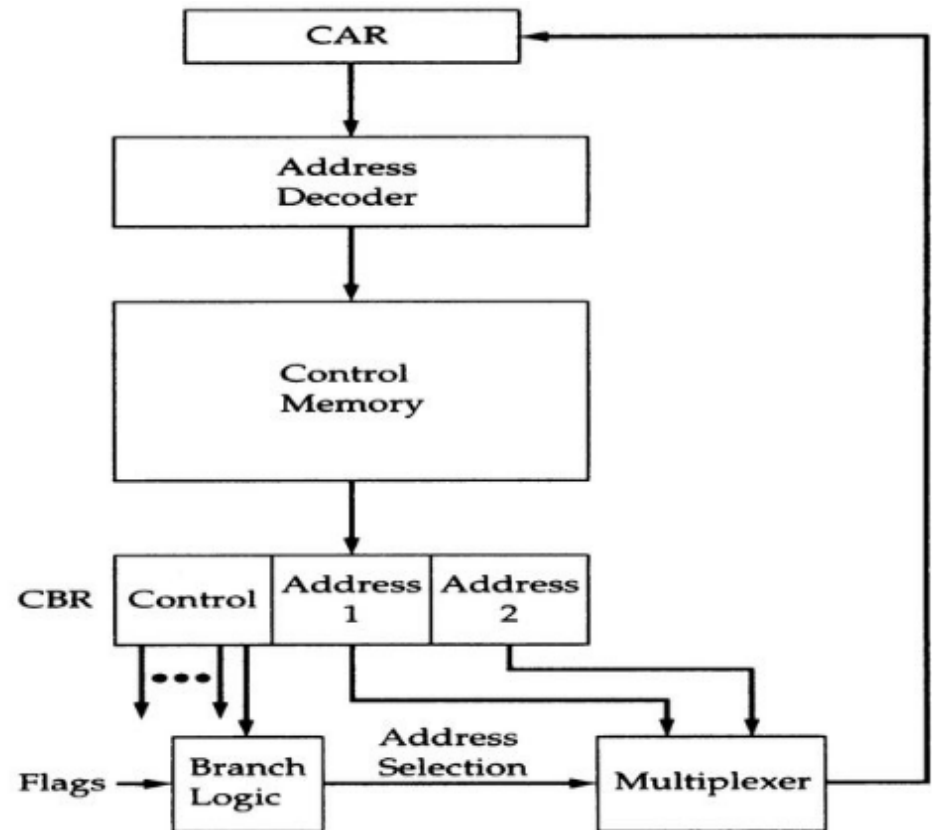
Esto se logra combinando circuitos combinacionales, que verifiquen el cumplimiento (o no) de las condiciones lógicas requeridas, con multiplexores que permitan seleccionar la fuentes de datos correspondiente con la condición lógica que se verifica.

Hay **tres arquitecturas básicas** para generar la lógica de secuenciación

Campo con dos direcciones – el enfoque más simple es proporcionar dos campos de direcciones en cada microinstrucción, una por cada rama de ejecución.

Las señales de salidas son generadas por un módulo de lógica de salida.

Las señales de control de flujo de ejecución son proporcionadas por un módulo de lógica de salto cuya entrada consiste en señales externas (banderas) más bits de la partición de control de la microinstrucción.

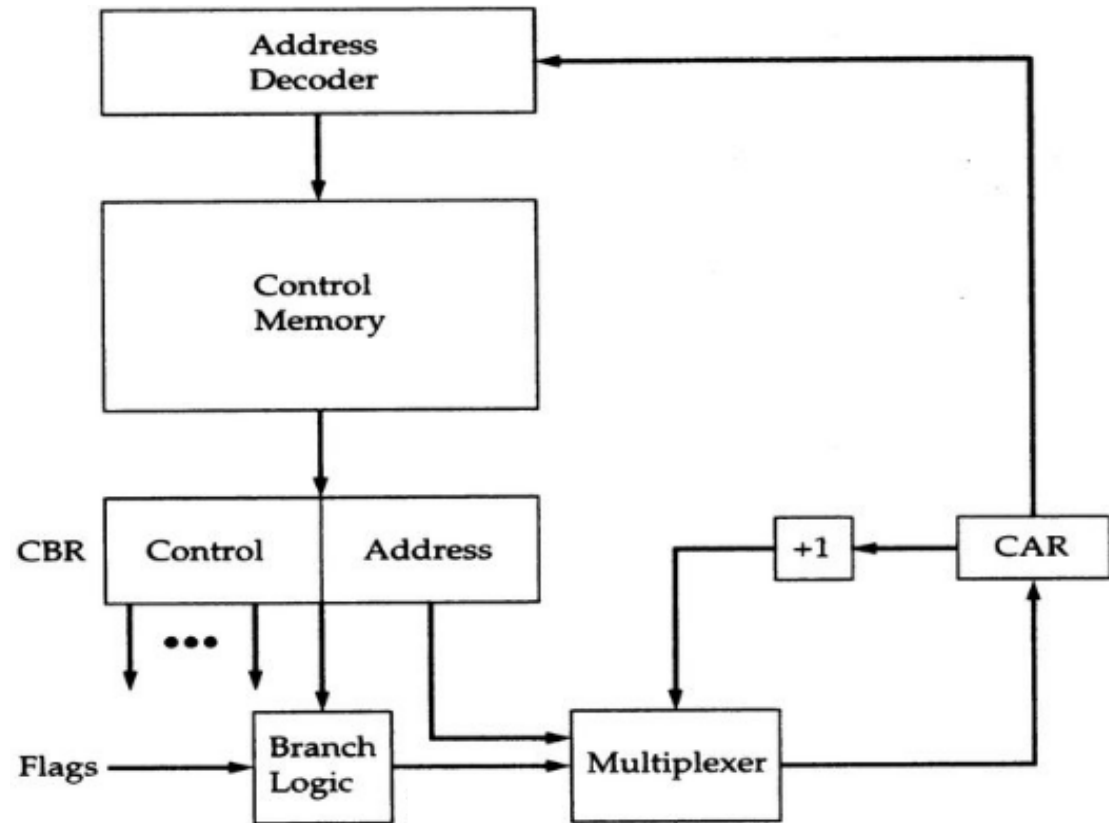


Campo de dirección única – las señales de selección de dirección determinan qué opción se selecciona. La dirección de ejecución secuencial se calcula automáticamente, la de salto es almacenada en la instrucción.

Este enfoque reduce el número de campo de dirección a uno.

En la mayoría de los casos, cuando se utiliza la ejecución secuencial, el campo de dirección no se utiliza.

Por lo tanto, esta codificación de microinstrucciones no utiliza de manera eficiente toda el ancho de la microinstrucción.



Formato variable – hay dos formatos de microinstrucción diferentes, que se selecciona con un bit que determina el formato utilizado en cada instrucción.

El primer formato utiliza los bits que no son dirección para activar las señales de control. En este formato, la próxima dirección es la siguiente a la actual (ejecución secuencial).

El segundo formato utiliza los bits que no son dirección para controlar la lógica de salto. En este formato se especifica un salto condicional o incondicional y la próxima dirección está codificada en la microinstrucción.

