

Electrónica Digital

Ingeniería Informática – FICH, UNL
Leonardo Giovanini



Máquina algorítmica de estados

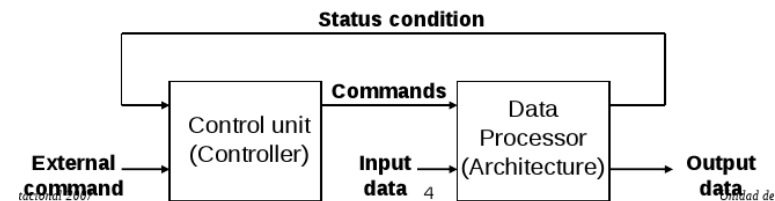
En esta se estudiarán los siguientes temas:

- Sistemas digitales;
- Maquinas algritmicas de estados:
 - Introducción;
 - Definición;
 - Diseño;
 - Operaciones RTL
 - Temporización;
 - El Procesador de datos;
 - El Controlador.

Los sistemas digitales más grandes necesitan técnicas de diseño más abstractas y sistemáticas.

Los métodos de diseño sistemático tiene las siguientes características:

- **Enfoque Top – down** – se enfatiza la vista macroscópica, comenzando por el problema original y refinándolo gradualmente hacia la solución. Los pasos para un procedimiento de diseño top–down son:
 1. Especifique el problema claramente (a nivel global / superior sin detalles innecesarios);
 2. Divide el problema en subproblemas más pequeños;
 3. Repita el proceso hasta que los subproblemas sean lo suficientemente pequeños como para ser resueltos directamente (implementable).
- **Separar el control de las operaciones** – los sistemas digitales procesan información. Lo hacen utilizando circuitos combinacionales. Un sistema digital consta de dos componentes:
 - Un algoritmo de control proporciona comandos que dirigen las operaciones de procesamiento de datos para realizar las tareas deseadas; y
 - Una arquitectura que procesa los datos de acuerdo con los requisitos;
- **Desarrollar una arquitectura general** – a nivel de bloque, antes de continuar con los detalles del hardware.



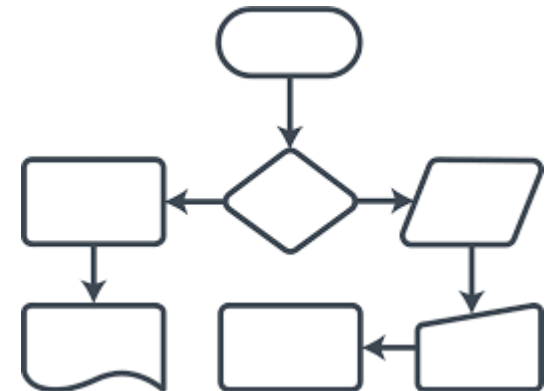
Un **diagrama de flujo** es la **representación gráfica de un algoritmo** o proceso.

Los diagramas **usan símbolos**, con significados definidos, **para representar los pasos** del algoritmo, y representan el **flujo de ejecución mediante flechas** que conectan los puntos de inicio y de fin del proceso.

Un diagrama de flujo es **una variación del diagrama de estados**, donde los "estados" representan operaciones, y las transiciones representan las actividades que ocurren cuando la operación termina.

Los pasos a seguir para construir el diagrama de flujo son:

- Establecer el alcance del proceso a describir;
- Identificar y listar las principales actividades y su orden cronológico;
 - Si el nivel de detalle incluye actividades menores, listarlas;
- Identificar y listar los puntos de decisión;
- Construir el diagrama respetando la secuencia cronológica y asignando los correspondientes símbolos.



Las ventajas de los diagramas de flujo son

- Favorecen la comprensión del proceso; y
- Permiten identificar los problemas y las oportunidades de mejorar del proceso. Se identifican los pasos, los flujos de los reprocesos, los conflictos, los cuellos de botella, y los puntos de decisión.

La máquina de estado algorítmico (ASM) es un método para diseñar máquinas de estado finito a partir de describir la secuencia de operaciones de un sistema digital.

Es como un diagrama de estado, pero más estructurado y por lo tanto más fácil de entender.

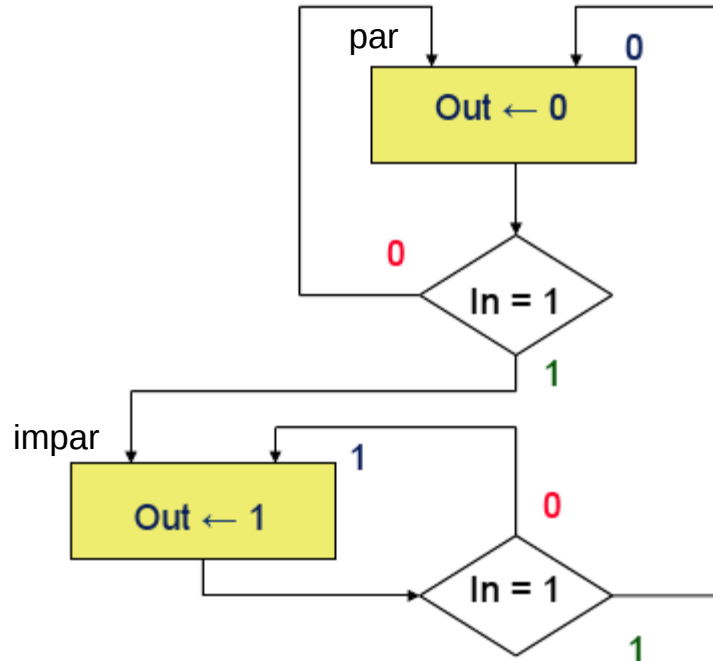


Diagrama ASM

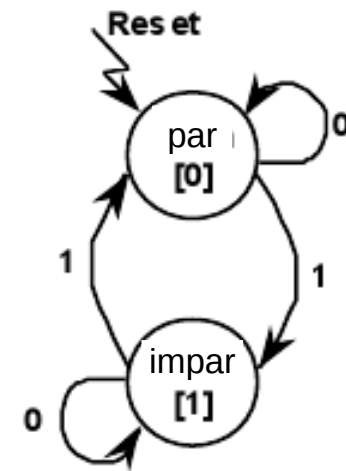
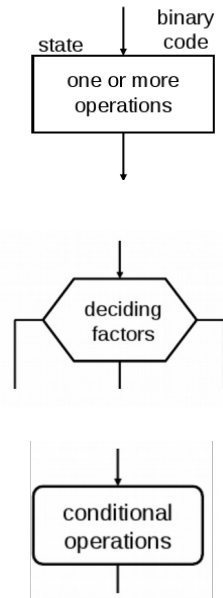


Diagrama de estados

Se diferencia de los diagramas de flujo en:

- Transición está gobernada por reloj y ocurre entre bloques ASM;
- Utiliza tres tipos de bloques que contienen **información de sincronización** exacta:
 - **Bloque de estado** – tiene forma rectangular. Es similar a la caja de operación y tiene como máximo un punto de entrada y un punto de salida y se utiliza para especificar una o más operaciones que podrían completarse simultáneamente en un ciclo de reloj;
 - **Bloque de decisión** – tiene forma de diamante. Tiene un punto de entrada y múltiples puntos de salida. Se utiliza para especificar una serie de rutas alternativas que se pueden seguir; y
 - **Bloque condicional** – está representado por un rectángulo con esquinas redondeadas. Sigue a un bloque de decisión y contiene operaciones condicionales que se invocan cuando el bloque de decisión selecciona la ruta que lo contiene.
- Los bloques de control y procesamiento se obtiene a partir del diagrama
 - El controlador se determina a partir de los bloques de decisión y las transiciones de estado; y
 - El procesador de datos se obtiene de las operaciones especificadas en los bloques de estados y condicionales.



El procedimiento de diseño de una ASM se compone de los siguientes pasos:

1. Crear un algoritmo, utilizando pseudocodigo para describir el comportamiento deseado;
2. Convertir el pseudocodigo en un diagrama ASM;
3. Diseñar el procesador de datos (Datapath) basado en el diagrama ASM;
4. Crear diagrama ASM detallado basado en el procesador de datos a partir de sustituir la notación del RTL por las señales definidas en el procesador de datos ;
5. Diseñar la lógica de control (Unidad de Control) basada en el diagrama ASM detallado.

Una vez que la operación del un circuito ha sido descrito utilizando operaciones de transferencia de registro (RTL), los componentes del procesador de datos se pueden derivar.

Cada variable única que se le asigna un valor al programa RTL puede ser implementado como un registro.

Dependiendo de la operación a realizar cuando se asigna un valor a una variable, el registro para esta variable se puede implementar como:

- Un **registro sencillo**;
- Un **registro asociado con un bloque combinatorio** — que puede llevar a cabo un sumador, restador, multiplexor, o algún otro tipo de función lógica combinatoria.

La **transferencia a nivel de registro** (RTL) es una **abstracción de diseño** que modela un circuito digital síncrono en términos del flujo de señales digitales entre registros de hardware y las operaciones lógicas realizadas en esas señales.

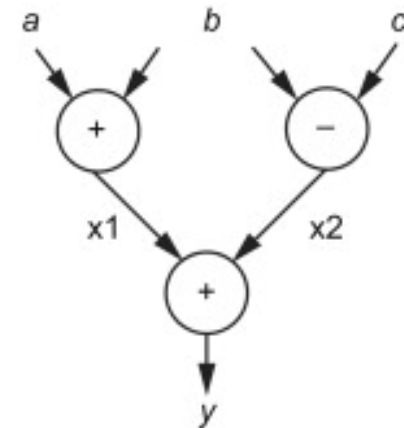
RTL se **utiliza en lenguajes de descripción de hardware** como Verilog y VHDL para crear representaciones de alto nivel de un circuito, a partir de las cuales se pueden derivar representaciones de nivel inferior y el cableado real.

Los sistemas digitales a nivel RTL esta especificado por tres componentes

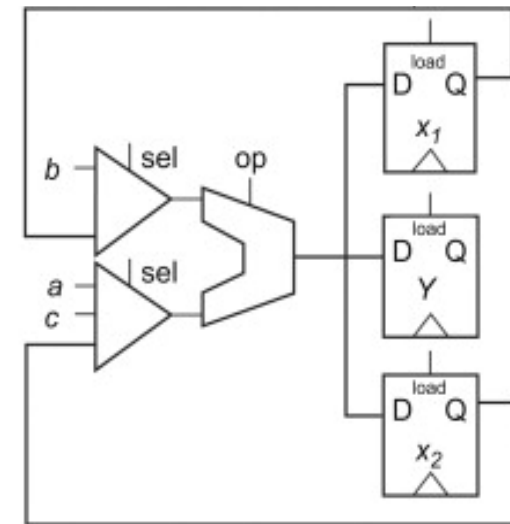
- El conjunto de registros;
- Las operaciones a realizar en los registros; y
- Las señales de control que se aplican a los registros y la sequencia de operaciones.

```
x1 <= a + b;  
x2 <= b - c;  
y <= x1 + x2;
```

Codigo



Flujo de datos



Implementacion

Los registros se usan para almacenar y procesar datos en el procesador de datos.

Los flip-flops (registros de 1 bit) and las memorias (conjunto de n registros) tambien son considerados registros.

Las operaciones en registros se pueden especificar en los bloques de estados y/o bloque condicionales usando la siguiente sintaxis

registro destino \leftarrow **funcion**(registros, parametros)

Ejemplos de operaciones en registro:

- | | |
|------------------------|---|
| $A \leftarrow B$ | Transfiere el contenido del registro B al A; |
| $A \leftarrow 0$ | Resetea el registro A; |
| $A \leftarrow A - 1$ | Decrementa el registro A en 1; |
| $A \leftarrow D - C$ | Sustraer el registro C a D y lo guarda en A; |
| $A \leftarrow f(A, p)$ | Aplica la funcion logica f al registro A. Por ejemplo si f es un barrel shifter esta sentencia rota el dato en A p posiciones y lo guarda en A; |

La señal de reloj generalmente no se incluye en la declaración de nivel de transferencia de registro, el comportamiento secuencial está implícito

Maquinas Algoritmica de Estados – Temporización

La temporización entre bloques está implícitamente presente en los gráficos ASM.

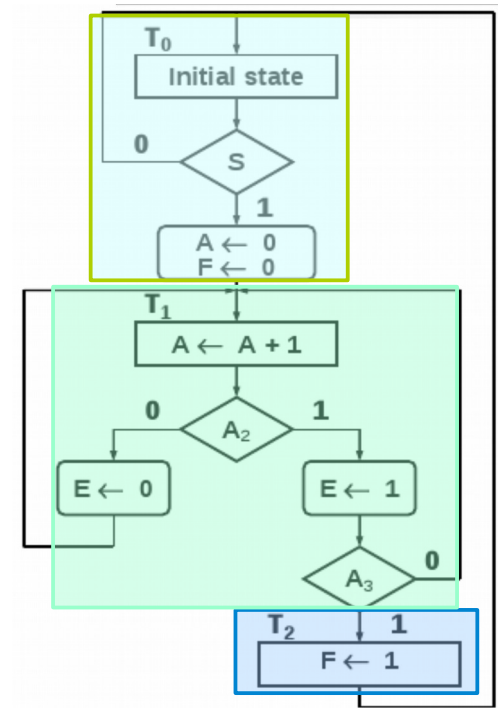
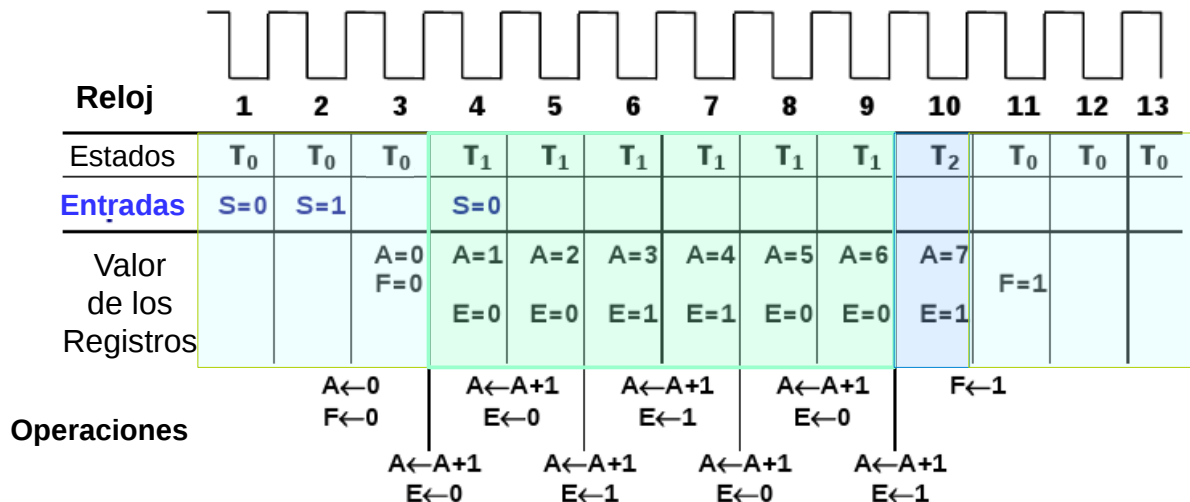
Cada bloque de estado, su decisión y bloques condicionales, se ejecuta en un ciclo de reloj.

Un **grupo de bloques que se ejecutan en un solo ciclo** de reloj se llama **bloque ASM**.

Las operaciones de ASM se pueden ilustrar a través de un diagrama de tiempos.

Dos factores que deben considerarse son

- Las operaciones en un bloque ASM ocurren simultáneamente en un ciclo de reloj;
- Los bloques de decisión dependen del estado del ciclo de reloj anterior (no dependen de las operaciones del bloque actual).



La arquitectura del procesador de datos (Datapath) se obtiene del diagrama ASM.

En particular, las operaciones realizadas en el diagrama determinan:

- Los registros que se utilizaran;
- Como deben estar conectados;
- Que operaciones se deben implementar;
- En que orden y que señales activan cada operacion

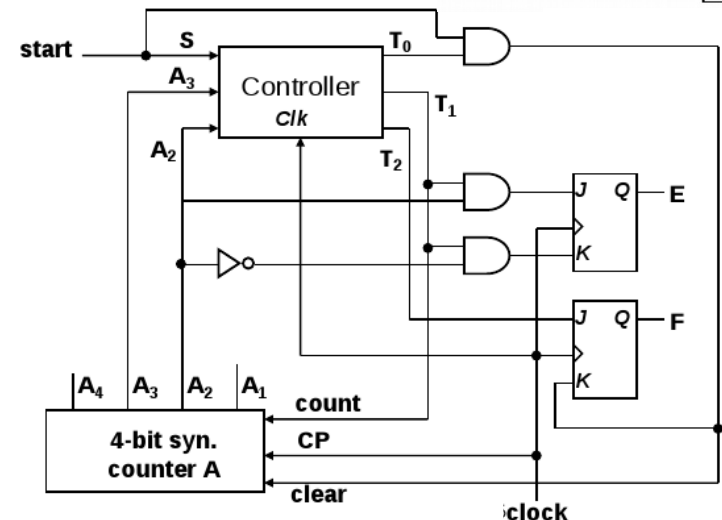
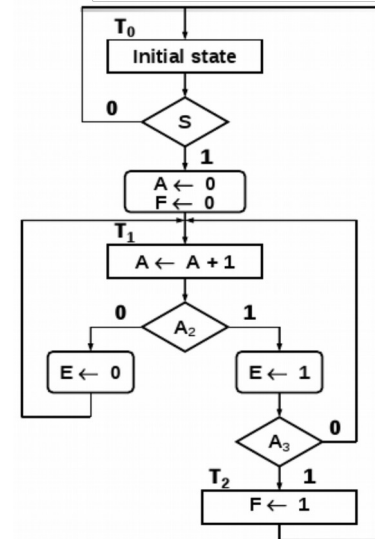
Como criterio general la arquitectura del procesador de datos debe ser lo más simple posible a partir del uso de unidades de alto nivel (contadores, registros de desplazamientos, etc.)

En el ejemplo podemos ver que

- Hay un incremento del registro A ($A \leftarrow A + 1$) en el estado T1.
- Se resetea el contador ($A \leftarrow 0$) en el estado T0 y $S = 1$.
- Se setea E ($E \leftarrow 1$) en el estado T1 y $A_2 = 1$.
- Se resetea E ($E \leftarrow 0$) en el estado T1 y $A_2 = 0$.
- Se setea F ($F \leftarrow 1$) en el estado T2.

Podemos ver que el procesador de datos necesita:

- Un contador sincrono ascendente de 4 bits con reset; y
- Dos flip-flops para los registros E y F.

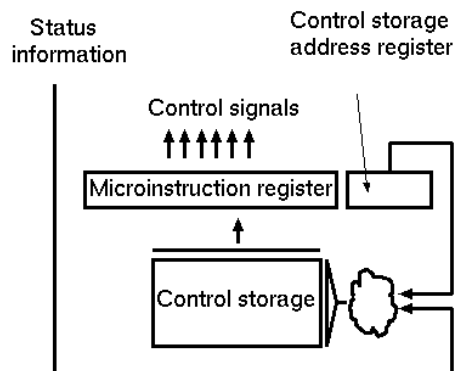
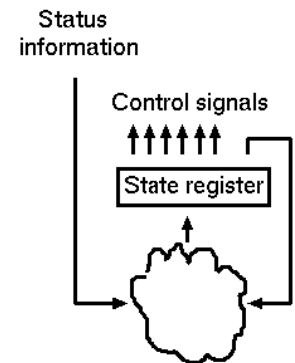


El controlador inicia comandos secuenciando correctamente el procesador de datos, utiliza condiciones de estado emitidas por procesador para determinar la secuencia correcta de comandos a ejecutar.

El **controlador requiere memoria** por lo tanto es un **sistema secuencial**.

Hay dos formas de implementar el controlador, utilizando

- **Máquinas de estados finitos** – son sistemas digitales que se diseñan para implementar un algoritmo específico. Una vez diseñado e implementado no se puede cambiar. El procedimiento de diseño ya se explicó en clases anteriores.



- **Microprogramación** – cualquier algoritmo ejecutado por un sistema digital se puede representar como una secuencia de operaciones elementales de procesamiento de información. Estas **operaciones elementales** se denominan **microoperaciones**.

Un **conjunto de microoperaciones** ejecutadas durante un **ciclo de operación** de un sistema digital se denomina **microinstrucción**. Se utilizan **condiciones lógicas** (señales de estado o indicadores) para **controlar el orden de ejecución** de las microoperaciones.

Atributos	Sistema cableado	Sistema microprogramado
Velocidad	Rápido	Lento
Costo	Creciente con el tamaño	Constante
Tamaño	Pequeño	Grande
Flexibilidad	Nula	Muy Alta
Manejo de la complejidad	Baja	Alta
Complejidad de Diseño	Creciente	Constante