# Resumen de Electrónica Digital

## Códigos

Un código es un sistema de reglas para convertir información representada de una forma, en otra que posea propiedades deseadas para su transmisión, almacenamiento y procesamiento.

### Partes del código

* Conjunto fuente: es el conjunto de información original, si este es finito, se le llama alfabeto.
* Conjunto objetivo: es el conjunto transformado, está constituido por elementos llamados “palabras códigos”.
* Tamaño del código: es el número de elementos utilizado por el código.

### Propiedades del Código

* Longitud: es la cantidad de elementos que componen una palabra código. Un código puede ser de longitud fija, cuando el número de elementos de salida es fijo para un número fijo de entradas, o variable, cuando el número de elementos de salida es variable para un número fijo de entradas.
* Distancia: es una función matemática que mide la similitud entre dos palabras código, si bien todas deben cumplir ciertas propiedades(d(ti ,tj ) = d(tj ,ti ); d(ti ,ti ) = 0; d(ti ,tj ) + d(ti ,tm ) ≥ d(ti ,tm ))existen diferentes formas de definirla, por ejemplo, la distancia de haming es definida como el número de posiciones en las que los símbolos correspondientes son diferentes.
* Prefijo: es cuando una palabra de código es un segmento inicial de cualquier otra.

### Tipos de códigos

* Continuos: son códigos en los que la distancia de haming entre sus palabras código, es uno.
* Cíclicos: son códigos en los que la última palabra código es continua con la primera.
* Auto complementarios: es un código en el cual sus palabras código se complementan entre sí, es decir palabras código del conjunto objetivo se obtienen a partir de la negación de otras p del conjunto.
* completos: son códigos que utilizan todas las palabras código disponibles.
* Instantáneos: son códigos en los que se puede decodificar cada palabra de Código, en cualquier cadena, tan pronto como se recibe. Este tipo de Código es expresable mediante “arboles de código”.
* Óptimos: son códigos unívocamente decodificables y cuya longitud media es mínima
* Ponderado: son códigos en los cuales hay una conexión aritmética entre el código, donde cada posición representa un peso específico.

#### Decodificación

Es el proceso inverso a la codificación, en este los códigos se convierten a su forma original para ser utilizado o comprendido por el receptor.

#### Usos básicos de la codificación

* Compresión de datos: eliminan la redundancia de los datos de una fuente para trasmitirla de manera más eficiente.
* Control de errores: Agrega redundancia a un mensaje, de modo que los receptores pueden usarla para verificar la consistencia del mensaje entregado y recuperar los datos perdidos.
* Codificación criptográfica: se trata de construir y analizar protocolos que eviten que terceros accedan a la información privada.
* Codificación de línea: se utiliza para representar la señal digital a transportar por una señal analógica que cambia alguna de sus propiedades en el tiempo de manera óptima para ser trasmitida por el canal físico.

### Códigos detectores-correctores

Estos son códigos que incluyen información redundante para poder detectar y corregir errores en los mensajes. Generalmente estos códigos se realizan en esquemas sistemáticos, en estos, el emisor agrega información al mensaje siguiendo algún algoritmo, el cual será utilizado por el receptor para verificar el estado de la información.

#### Solicitud de repetición automática

Si las características de los errores no se pueden determinar, o son muy variables, se combina la detección de errores con un “sistema de retransmisión de datos”, y se lo conoce como “Solicitud de repetición automática”, el cual se utiliza principalmente en internet.

#### Adelanto de información

Es un proceso de agregar datos redundantes para recuperar la información aun cuando hay varios errores, ya sea durante la transición o el almacenamiento.

#### Códigos de corrección

* Códigos de bloque: son códigos que procesan la información bloque a bloque de longitud fija. En particular los códigos de bloque lineales son códigos que también tienen la propiedad de linealidad (la suma de dos palabras código es otra palabra código).
* Códigos convolucionales: son códigos en los que cada símbolo de palabra de código es una suma ponderada de los símbolos de mensaje de entrada, de modo que la salida del codificador es la convolución de la entrada contra los estados almacenados en registros.

En el algoritmo utilizado por los códigos convolucionales se logra un mejor rendimiento en la corrección que en los de bloque lineales.

#### Tipos de códigos de bloque lineales

* Códigos de repetición: son códigos que repiten el mensaje varias veces con la probabilidad de que el canal corrompa una minoría de las repeticiones, de forma que el receptor detecte el error de transmisión.
* Códigos cíclicos: son códigos donde rotaciones de la palabra código dan otra palabra código.
* códigos polinomiales: son códigos cuyo conjunto de palabras de código validas consiste en aquellos polinomios que son divisibles por un polinomio fijo dado, de longitud más corta llamado “polinomio generador”.
* Códigos BCH: son una clase de códigos que se construyen utilizando polinomios sobre un conjunto de información.

### Detección de errores

La detección de errores se realiza utilizando una “función de hash” adecuada que agrega una “etiqueta de longitud fija” al mensaje, lo que permite a los receptores verificar el mensaje entregado recalculando la etiqueta y comparándola con la proporcionada.

### Funciones de Hash

* Bit de paridad: es un bit que se agrega al mensaje para garantizar que el número de bits con valor 1 sea par o impar. El problema es que un número par de bits invertidos en el mensaje no pueden ser detectados.
* Checksum: es una suma aritmética modular de palabras de código de mensaje de una longitud de palabra fija.
* Cheque de redundancia y cíclica (CRC): esta es una función de hash diseñada para detectar cambios accidentales en datos digitales. No es adecuado para cambios introducidos maliciosamente.

### Código de Hamming

Los códigos de haming son una familia de códigos de bloque lineales correctores, capaces de detectar errores de hasta dos bits o corregir errores de un bit. Se dice que los códigos de haming son perfectos, ya que logran la tasa más alta posible para códigos con su longitud de bloque. Solo pueden detectar y corregir errores cuando la tasa de error es baja. Este es el caso de la memoria de la computadora (memoria ECC), donde los códigos de Hamming son ampliamente utilizados.

Los pasos que implementan el algoritmo para el diseño de un código Hamming son:

1. Numere los bits a partir de 1 (1, 2, 3, 4, 5, 6, 7, etc.)

2. Escriba los números de bits en binario (1, 10, 11, 100, 101, 110, 111, etc.)

3. Todas las posiciones de bits que son potencias de dos son bits de paridad (1, 2, 4, 8, etc.)

4. Todas las demás posiciones de bits, con dos o más bits en la forma binaria de su posición, son bits de datos

5. Cada bit de datos se incluye en un conjunto único de bits de paridad, según lo determinado por la forma binaria de su posición de bit.

*Códigos de línea*

Estos son códigos elegidos ser usados dentro de un sistema de comunicaciones o almacenamiento, con fines de transmitir o almacenar información. estos son patrones de voltaje, corriente, etc. utilizado para representar datos digitales a ser transmitidos y/o almacenados.

### Algebra de Boole

Es una estructura matemática que sistematiza operaciones lógicas, utilizando técnicas algebraicas para tratar expresiones de la lógica proposicional.

* Ver que tiene que ocurrir para que sea un Algebra de Boole (axiomas) y que si se cumple entonces se deducen los teoremas fundamentales
* Lógica Binaria: es un sistema lógico cuyas variables adoptan solo dos valores (0,1), es un algebra de Boole porque cumple con los axiomas y también se deducen los teoremas

#### Función lógica

Una función lógica es una función matemática y lógica, cuyos argumentos, y la función misma, asumen uno de los dos valores posibles del conjunto que define las variables.

#### Representaciones de una función lógica

* Expresión algebraica: es una representación matemática de las expresiones lógicas utilizando las tres operaciones básicas.
* Tabla de verdad: es una tabla matemática utilizada en lógica que establece los valores de las expresiones lógicas para cada una de las combinaciones de los argumentos de la función.
* Gráfica: es la que se utiliza en el desarrollo de circuitos electrónicos. Las funciones algebraicas se representan gráficamente con símbolos normalizados para cada operación básica.

### Formas canónicas

La estandarización de los términos de las funciones lógicas hace que su evaluación, análisis, simplificación e implementación sea sistemática y sencilla. Un término canónico de una función es todo producto o suma en la cual aparecen TODAS las variables en su forma directa o inversa, estos se conocen como:

* Minitérminos: es una expresión lógica de n variables donde solo aparece el operador and y la negación. VER MÁS DESARROLLADO EN CARPETA
* Maxitérminos: es una expresión lógica de n variables donde solo aparece el operador or y la negación. VER MÁS DESARROLLADO EN CARPETA

Toda expresión booleana se puede transformar en cualquiera de las siguientes formas estándar:

* Suma de productos: una función lógica puede representarse como una suma de minitérminos. VER MÁS DESARROLLADO EN CARPETA
* Producto de sumas: una función lógica puede expresarse como el producto de maxitérminos. VER MÁS DESARROLLADO EN CARPETA

## Circuito Combinacional / Lógica combinacional

Se le dice circuito combinacional a todo sistema digital cuyas salidas son función exclusiva del valor de sus entradas, sin que intervengan en ningún caso estados anteriores de las entradas o salidas, es decir carecen de memoria y realimentación. En electrónica digital un circuito combinacional esta formado por ecuaciones simples a partir de las operaciones básicas del algebra de Boole, y se pueden representar mediante una tabla de verdad. Los circuitos están compuestos SOLO por puertas lógicas interconectadas, sin ninguna celda de memoria.

Las compuertas lógicas son los componentes básicos de los circuitos combinacionales.

Entre los circuitos combinacionales clásicos tenemos:

• Lógicos:

- Generador / Detector de paridad;

- Conversor de código;

- Codificadores;

- Generadores de códigos detectores de error.

• Gestión de datos:

- Multiplexores y demultiplexores;

- Codificadores / Decodificadores de para transmisión de datos (códigos detectores/correctores, línea, criptografía).

• Aritméticos:

- Comparador;

- Sumador / Restador;

- Operaciones matemáticas (Multiplicación, división, etc.).

• Aritméticos y lógicos:

- Unidad aritmético lógica (ALU).

## Codificador

Un codificador es un circuito combinacional que convierte símbolos de un alfabeto en otro, es decir implementa un proceso de codificación. Uno de sus usos más comunes es el de comprimir información para hacer más sencillo su procesamiento, almacenamiento y transmisión.

Codificadores con prioridad

Son codificadores en los cuales, si se activan dos o más entradas simultáneamente, el codificador generará el código correspondiente a la entrada de mayor prioridad.

Conversores

Otra de las aplicaciones de la codificación es cambiar la representación de la información (conversión de código) para facilitar su procesamiento, presentación o transmisión, entre otras. Algunos de ellos son: BCD-7 segmentos, Binario-Gray, Decimal-BCD.

## Decodificador

Un decodificador es un circuito combinacional que realiza el proceso inverso a la codificación. Convierte símbolos de un alfabeto en otro. Uno de sus usos más comunes es para descomprimir información. También son utilizados para las memorias, ya que permiten acceder a cada celda a través de su dirección

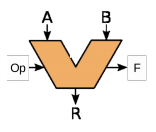
Los decodificadores incluyen una entrada – En – que permite habilitar o deshabilitar el proceso de decodificación en el circuito. Cuando esta entrada está activa, las salidas son 0 independiente del código presente en las entradas

## MULTIPLEXOR Y DEMULTIPLEXOR EN CARPETA

## Unidad Aritmética Lógica

Una ALU es un circuito que se especializa en realizar operaciones aritméticas y operaciones lógicas entre uno o dos argumentos. La ALU consta de:

* A y B son los operandos
* R es el resultado de la operación
* F es el estado de la operación
* Op es el código de operación que indica la operación realizar.



Las ALU realizan las siguientes operaciones:

* Operaciones aritméticas de números enteros
* Operaciones lógicas bit a bit
* Operaciones de desplazamiento y rotación de bits.

# Parte 2

## Circuitos secuenciales

### Definición

Un circuito secuencial es un tipo de circuito lógico cuya salida depende del valor presente de sus entradas y de la secuencia de entradas o salidas pasadas.

### Propiedades

* El estado de salida de un circuito secuencial es una función de la entrada actual, la entrada pasada y/o de la salida pasada.
* Se dice que los circuitos secuenciales tienen “memoria” ya que recuerdan las condiciones de las entradas o salidas pasadas y permanecen fijos en su estado actual hasta el siguiente cambio.
* Además de los dos estados lógicos 0 y 1, hay un tercer elemento que separa los circuitos lógicos secuenciales de las lógicas combinacionales: el tiempo.
* Los circuitos secuenciales también poseen una entrada de reset la cual, según la preferencia del usuario, puede poner el circuito en un estado inicial de interés.

### Categorías

* Circuitos controlados por eventos: Son circuitos secuenciales asincrónicos cuyos estados cambian con sus entradas.
* Controlados por reloj: Son circuitos secuenciales síncronos cuyos cambios se habilitan cuando el reloj lo habilita.
* Controlado por pulsos: Es una combinación de los otros dos tipos de circuitos secuenciales, respondiendo a impulsos de activación asíncronos.

#### Flip Flops

Estos son circuitos secuenciales básicos, capaces de almacenar hasta un bit. El término flip-flop se refiere a la operación real del dispositivo, ya que puede flip a un estado de configuración o flop al estado de reinicio opuesto. Existen diferentes tipos:

* Flip Flop SR sincrónico:

Este dispositivo consta de tres entradas:

Una entrada Clk que refiere al reloj.

Una que establece la salida (Q=1), llamada Set (S).

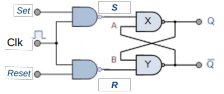
Una que la reestablece (Q=0), llamada Reset ®.

y dos salidas:

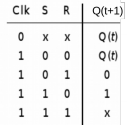
La salida principal Q.

Su complemento Q.

Se implementa con dos compuertas NAND y dos and de la siguiente forma:



Su tabla de verdad es:



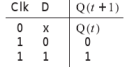
Como se puede observar, el problema de esta implementación es que, si S y R están en 1, entonces el dispositivo se rompe, entra a un estado indeterminado.

* Flip Flop D sincrónico

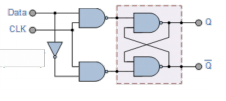
tiene solo una entrada de datos (D) con la entrada Clk

Transfiere los datos entre su entrada y su salida en cada impulso de reloj, es decir que Qn=D

Su tabla de verdad es:



Su implementación es la siguiente:



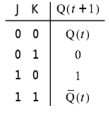
Los flip-flops D se construyen a partir de un flip-flop SR conectando un inversor entre las entradas S y R. Esto implica resolver el problema de la condición prohibida a partir de eliminar esa condición (R = S).

* Flip Flop JK sincrónico:

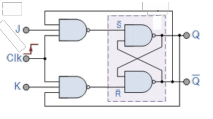
Tiene dos entradas (J y K) con J actuando como S y K actuando como R.

Las cuatro configuraciones de entrada son válidas: sin cambio, establecer, restablecer y alternar.

Su tabla de verdad es:



Su implementación es la siguiente:



En esta implementación se resuelve el problema de la condición prohibida al establecer el comportamiento del flip flop para ese caso.

*Parámetros de tiempo*

Tiempo de propagación:

Es el tiempo que tarda en producirse un cambio en la salida después de un flanco del reloj.

Tiempo de establecimiento:

Es el mínimo tiempo necesario que la señal de entrada debe mantenerse estacionaria antes del flanco ascendente del reloj, de manera que el dato sea leído de manera confiable.

Tiempo de retención:

Es el mínimo tiempo que la señal de entrada debe permanecer sin cambio necesario después del flanco ascendente del reloj de manera que el dato sea leído de manera confiable.

Apertura:

Es la suma de los tiempos de retención y establecimiento. Las entradas deben permanecer sin cambios durante este periodo.

Tiempo de recuperación:

Es la cantidad mínima de tiempo que la entrada asincrónica de seteo o reseteo debe estar activa antes del flanco de ascendente del reloj, de modo que el reloj lea de manera confiable los datos. El tiempo de recuperación para la entrada asincrónica es, por lo tanto, similar al tiempo de establecimiento para la entrada de dato.

Tiempo de eliminación:

Es la cantidad mínima de tiempo que la entrada asincrónica establecida o restablecida debe estar inactiva después del flanco ascendente del reloj, de modo que el reloj lea de manera confiable los datos. El tiempo de eliminación para la entrada asincrónica establecida o restablecida es, por lo tanto, similar al tiempo de retención para la entrada de datos.

## Metaestabilidad:

Los flip-flops están sujetos a un problema de Metaestabilidad, que ocurre cuando dos entradas cambian aproximadamente al mismo tiempo, lo cual ocasiona que la salida pueda ser impredecible o que oscile varias veces antes de establecerse. Los problemas de Metaestabilidad se pueden evitar asegurando que las entradas de control se mantengan constantes durante el periodo de apertura, pero no siempre es posible cumplir con ese periodo.

Pero se soluciona conectando dos o más flip Flops en cadena, de modo que la salida de cada uno alimente la entrada de datos de la siguiente y así todos los dispositivos compartan un reloj común.

## Registro

*¿Qué es?*

Un registro es un grupo de n flip-flops, donde cada uno de ellos almacena un bit de información, tiene dos funciones básicas: almacenamiento y movimiento de datos, generalmente se implementan de forma síncrona con el reloj.

¿Para qué se utilizan dentro de los sistemas digitales?

Almacenamiento temporal de bits sobre los que se está realizando una labor de procesamiento

Se emplean como líneas de retardo digitales y en tareas de sincronización.

Convertir los datos de formatos serie a paralelo y paralelo a serie.

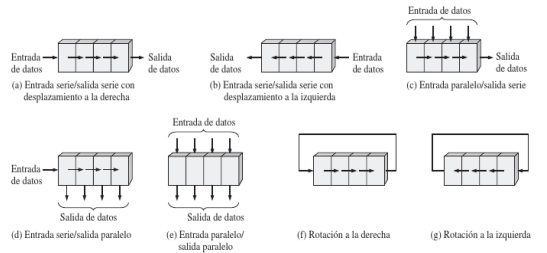
¿Qué tipos de registros existen?

Paralelo-Paralelo – las entradas y las salidas de todos los flip-flops son accesibles (figura e).

Serie-Serie – la entrada del primer (o último) flip-flop y la salida del último (o primero) son accesibles (figura a,b,f y g).

Paralelo-Serie – las entradas de todos los flip-flops y la salida del último son accesibles (figura c).

Serie-Paralelo – la entrada del primer flip-flop y las salidas de todos son accesibles(figura d).



*Figura 1: entradas y salidas de registros según su tipo*

Registros de desplazamiento

A estos se les llama de desplazamiento ya que los datos enviados se desplazan un bit a la derecha por cada movimiento del reloj, existen diferentes tipos:

* Serie-serie unidireccional: Los bits de datos son cargados en el registro usando la entrada del registro (Din). El contenido de cada flip-flop es transferido al siguiente en cada flanco ascendente del reloj (Clock).
* Serie-serie bidireccional: Los registros de desplazamiento serie bidireccional es aquél en el que los datos se pueden desplazar a izquierda o a derecha. Se implementan utilizando multiplexores a la entrada de los flip flops que permitan controlar la transferencia de los datos de una etapa a la siguiente o a la anterior. Poseen una entrada extra llamada D/I la cual controla la dirección de desplazamiento de los datos.
* Serie-paralelo: Los registros de desplazamiento con entrada serie y salida paralelo serie aceptan datos en serie, un bit cada vez por una única línea, y salen todos juntos en paralelo. Los bits de datos son cargados en el registro usando la entrada del registro (Din). El contenido de cada flip-flop es transferido al siguiente flip-flop en cada flanco ascendente del reloj (Clock). La información almacenada es entregada a la salida también en forma paralela de manera simultánea.
* Registro paralelo a serie: es un registro con entradas en paralelo y salida serie, se pueden implementar de dos maneras, una utiliza flip flops RS y otra flip flop D, en la práctica no se utilizan.
* Registro universal: Un registro de desplazamiento universal tiene capacidad de entrada y salida serie y paralelo. El 74L194 es un ejemplo de este tipo de registros.

## Memorias

¿Qué son?

Las memorias son dispositivos capaces de almacenar información digital en forma estructurada. Las memorias pueden clasificarse según diferentes criterios:

#### Persistencia: la capacidad de retener la información

* Volátil : la integridad de la información está garantiza cuando está alimentada;
* No volátiles : preservan la información aún en ausencia de energía. Estas se utilizan en la microprogramación de sistemas, existen diferentes tipos según si pueden reprogramarse y como se lleva a cabo este proceso,

#### Acceder a los datos: como se leen o escriben los datos

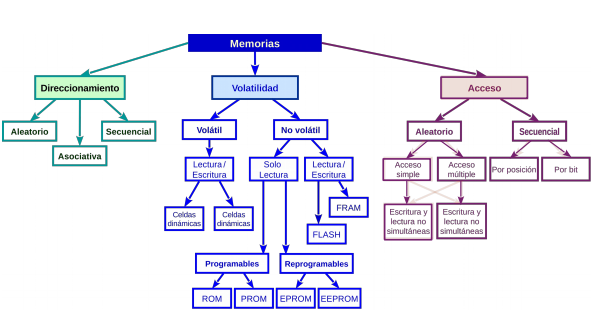
* Paralelo – los bits de los datos se acceden todos al mismo tiempo;
* Serie – los bits de los datos se acceden de a uno a la vez.

#### Operaciones que se pueden realizar – sólo leer o leer y escribir

* Lectura-escritura – son memorias en la que los datos se pueden leer y escribir;
* Solo lectura – son memorias en la que los datos solo pueden leerse.

#### Mecanismo de direccionamiento – como se accede a un dato particular

* Aleatoria – los datos pueden ser accedidos en cualquier orden;
* Secuencial – para acceder un dato, debe accederse a todos los datos que lo anteceden.



Términos de importancia para las memorias

Palabra de memoria: es un conjunto de n bits adyacentes.

Dirección de memoria: es un conjunto de m bits que muestra la posición de la palabra dentro de la memoria, utilizando un decodificador se pueden direccionar 2^m posiciones.

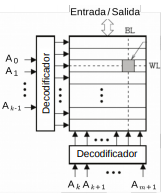
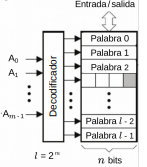
Capacidad de la memoria: es la cantidad de palabras que se pueden almacenar en la memoria, se puede expresar como el número de posiciones, es decir 2^m palabras o n\*2^m bits.

Bus de datos: este es el cable que lleva a los datos desde las memorias a los destinos.

Bus de direcciones: este es el cable que lleva la información de cuál es la palabra a la que deseamos acceder en la memoria.

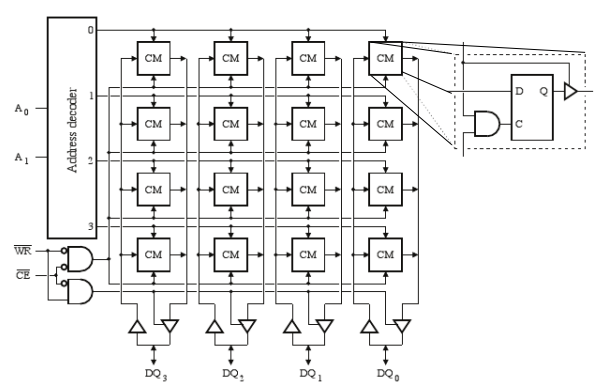
### Organización de las memorias de acceso aleatorio (RAM)

una memoria se puede organizar de dos formas, como una columna con un solo decodificador para acceder a cada palabra (ya que habrá una por fila), o como una matriz, se utiliza cuando el número de bits en la memoria se vuelve muy alto, para acceder a una palabra se utilizan dos decodificadores, uno para las filas y el otro para las columnas.



*Figura 3: Configuración de columna y matriz en una memoria*

### Implementación de una memoria columna de 16 bits



*Figura 4: Diagrama de circuito de la memoria hecha con flip flops tipo d.*

Para cada combinación de bits de dirección, el decodificador activa una línea horizontal lo que permite escribir o leer una palabra de 4 bits a través de los pines de entrada/salida.

### Memorias RAM multipuerto

Existe otra forma de implementar una memoria de acceso aleatorio, es utilizando más de un puerto de lectura y escritura. El objetivo de una memoria con múltiples puertos es proporcionar varias rutas de comunicación simultaneas a una matriz de datos. Cada puerto proporciona una ruta de acceso independiente para leer o escribir datos en la matriz de memoria. Se puede acceder a la matriz de manera aleatoria a través de cada puerto. Cada uno de dichos accesos puede leer o escribir cualquier posición de memoria, independientemente de a qué otra posición se acceda en otros momentos o a través de otros puertos. Una memoria multipuerto de p-puerto actúa “casi” como p-memorias independientes, con idéntico contenido. La memoria de p-puerto admite el acceso paralelo de p-vías a la matriz de datos, permitiendo leer hasta p veces más rápido.

Las memorias multipuertos se pueden clasificar de acuerdo a su modo de operación y su organización interna:

* Memoria multipuerto asíncrona
* Memoria multipuerto síncrona
* Memoria multipuerto conmutables por banco
* Memoria multipuerto de acceso secuencial y aleatorio

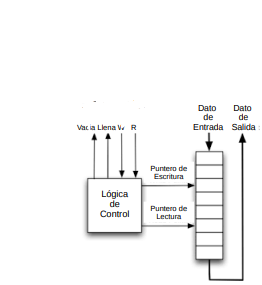
### Memorias de acceso secuencial FIFO

Una FIFO(First In, First Out) es una memoria de acceso secuencial utilizada para el registro de datos en la cual la información que entra en primer lugar va a ser la primera en salir.

Generalmente se utilizan entre sistemas que operan a diferentes velocidades diferentes y tienen que comunicarse. Los datos pueden entrar a la FIFO con una velocidad y salir con otra. Básicamente.

### Implementación

Una FIFO se implementa utilizando registros de desplazamiento con circuitos de control. Los datos ingresan en el Buffer(puntero) de Entrada, controlados por el Control de Desplazamiento. Cuando un dato es almacenado en la FIFO, el Buffer de Entrada está listo para recibir nuevos datos lo que es indicado por la señal Buffer Vacío. Los datos son leídos en el Buffer de Salida. Cada vez que se lee un dato, el Control de Desplazamiento actualiza la FIF



###### Figura 5: Funcionamiento de memoria FIFO

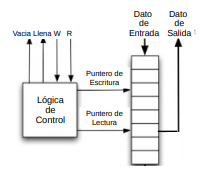
#### Memorias de acceso secuencial LIFO

Una LIFO(Last In, First Out) es una memoria de acceso secuencial utilizada para el registro de datos en la cual la información que entra en último lugar va a ser la primera en salir.

Una memoria LIFO a menudo se llama pila. La mayoría de los microprocesadores usan una pila para guardar bits de indicador de estado y el contenido de ciertos registros, en caso de interrupciones. Un uso común de las pilas es la asignación y acceso a la memoria.

### Implementación de memorias LIFO

La estructura de una memoria LIFO es similar a la de una memoria FIFO y puede basarse en registros de desplazamiento o memoria RAM. En principio, los datos almacenados en una memoria LIFO basada en registros se pueden mover de una posición a otra en cada pulso de reloj. Sin embargo, para una memoria LIFO basada en RAM, no son los datos sino las posiciones de acceso las que se mueven, controladas por contadores que se llaman punteros de escritura o lectura.



###### Figura 6: Funcionamiento de memorias LIFO

### Memoria Asociativas

Una memoria asociativa, también conocida como memoria direccionable por contenido (CAM), compara los datos de entrada con los que están almacenados para generar un indicador de correspondencia y luego devolver la dirección de los datos coincidentes.

Se utiliza en aplicaciones que requieren las operaciones relacionadas con la detección de coincidencia de patrones en los datos, tales como controladores de caché, redes de comunicación, de codificación de imágenes y compresión de datos.

## Máquinas de Estado Finito

Una máquina de estado finito (FSM) o simplemente una máquina de estado, es un modelo matemático de computación, ya que tiene la capacidad de representar secuencias que evolucionan en el tiempo, es decir algoritmos.

### Definición

Se define como una lista de estados, la cual tiene un estado inicial y las entradas que desencadenan cada transición. Existen dos tipos:

* Determinísticas: son maquinas donde las transiciones están definidas.
* No determinísticas o probabilísticas: las transiciones son realizadas según probabilidades de moverse de un estado al otro.

Las máquinas de estado finito NO son capaces de representar TODOS los algoritmos.

### Estado

Un estado es una descripción de las variables internas de un sistema, situación, que está esperando ejecutar una transición. En las máquinas de estado finito, existe una cantidad de estados finita, la cual es su principal limitación.

### Transición

Una transición es un conjunto de acciones que se ejecutan cuando se cumple una condición o cuando se recibe un evento.

#### Componentes de las máquinas de estados

En una máquina de estados finitos, el séxtuple (Σ, Γ, S, s0, T, G) representa los seis componentes que definen la máquina., donde:

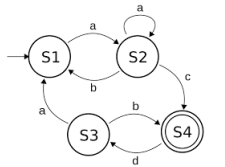
* Σ es el alfabeto de entrada, un conjunto finito y no vacío de símbolos.
* Γ es el alfabeto de salida, conjunto finito y no vacío de símbolos.
* S es un conjunto finito y no vacío de estados, que incluye el estado inicial s0 ∈ S y al conjunto de estados finales F , el cual puede ser vacío.
* T : S×Σ → S es la función de transición de estado. En una FSM no determinista estaría definida por T : S×Σ →P(S), es decir devolverá un conjunto de estados.
* G es la función de salida.

### Diagrama de estado

Un diagrama de estado describe el comportamiento de los sistemas por una serie de eventos que pueden ocurrir en uno o más estados posibles.

Una forma clásica de diagrama de estado es un grafo orientado con elementos que representan los componentes de las FSM:

* Vértices Q – representan los estados si ∈ S a través de círculos y etiquetados con símbolos o palabras únicos dentro de ellos. Estas etiquetas representan el estado representado. Los estados finales se dibujan con un doble circulo. El estado inicial s0 no suele representarse con un vértice, se lo representa mediante una flecha sin origen que apunta al siguiente estado.
* Arcos t(qi , σj ) ∈ T – representan las transiciones entre estados causadas por la entrada σj , identificada en los bordes. Un arco se dibuja como una flecha dirigida desde el estado actual al siguiente, describiendo que la transición que ocurrirá si el símbolo de entrada aparece. Este símbolo y el valor correspondiente aparecen junto al arco que representa la transición.



###### Figura 7: ejemplo de diagrama de estado, con 4 estados.

### Tabla de transición de estados

Una tabla de transición de estados es una tabla que muestra a qué estado (o estados en el caso de un autómata finito no determinista) se moverá una FSM, en función del estado actual y las entradas.

* VER DIFERENCIA ENTRE MEALY Y MOORE

## Microprogramación

### Sistema micro programado

Un sistema micro programado consiste en un conjunto de elementos que permiten almacenar y ejecutar una secuencia de microinstrucciones (instrucciones que determinan las operaciones elementales que puede ejecutar el sistema).

### Lenguaje de maquina

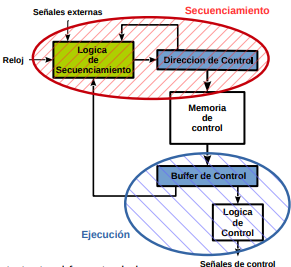
Un lenguaje de máquina es un sistema de códigos interpretable por el controlador de un autómata, es un tipo de lenguaje de programación muy básico que permite controlar una máquina.

### Unidad de control micro programada

La unidad de control microprogramada incluye las lógicas para secuenciar a través de microoperaciones, ejecutar las microoperaciones y tomar decisiones basadas en entradas externas. Su arquitectura está claramente definida, con muy pocas variantes. Esta unidad 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.

### Partes de un sistema microprogramado



###### Figura 8: unidad de control micro programada.

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

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

Lógica de secuenciación – 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. La lógica de secuenciación puede ser generada con3 arquitecturas diferentes Campo con dos direcciones, Campo de dirección única y de Formato variable.

## Maquinas Abstractas

Una máquina abstracta es un modelo de un sistema informático, considerado como hardware y/o software, construido para realizar un análisis de cómo funciona el sistema. El modelo consiste en entradas, salidas y las operaciones que se pueden realizar.

La abstracción de los procesos informáticos se utiliza en:

* Informática para analizar la complejidad de algoritmos;
* Ingeniería de la computación para calcular la cantidad de recursos (tiempo, memoria, etc.) necesarios para implementar un algoritmo particular;
* Ingeniería de la computación para implementar los sistemas al definir las posibles formas en que se conectan los recursos físicos (arquitectura).