

Unidad 4 / Escenario 7

Lectura Fundamental

Máquinas de Estados Finitos

Contenido

- 1 Introducción a las Máquinas de Estados ¿Qué son?
- 2 Máquinas de Moore
- 3 Máquinas de Mealy
- 4 Algunos ejemplos

Referencias

Palabras clave: Máquinas de estados, Moore, Mealy, Diagramas de estados, Tablas de estados

1. Introducción a las Máquinas de Estados ¿Qué son?

Las máquinas de estados finitos (FSM, del inglés, *Finite-State Machine*), son una herramienta teórica que permite modelar sistemas en los que los valores de las salidas para un momento dado dependen tanto de las entradas como de la historia del sistema. Esto es, el valor de las salidas depende del valor presente y los valores pasados de las entradas.

Para simplificar el modelado de este tipo de sistemas, se ha tomado el concepto de ESTADO que surge de otras disciplinas, particularmente del control automático. Un estado puede definirse como:

El menor conjunto de variables que, conocidas para un momento del tiempo, junto con la entradas del sistema, permite determinar completamente el comportamiento del sistema en cualquier momento posterior. (Ogata, 2001)

De esta manera, una máquina de estados se puede decir que consta de **estados** y de **transiciones** entre dichos estados. Aunque la definición pueda parecer un poco confusa, su aplicación en los sistemas digitales secuenciales no es tan compleja, teniendo en cuenta que las variables del sistema serán de tipo discreto. De manera general, una máquina de estados, basada en lógica secuencial, está formada “por una etapa de lógica combinacional y una sección de memoria (flip-flops)” (figura 1 (Floyd, 2010, p.499). La etapa de memoria se utiliza para almacenar los estados, mientras que la lógica combinacional permite codificar y decodificar dichos estados para que el sistema pueda actuar adecuadamente.

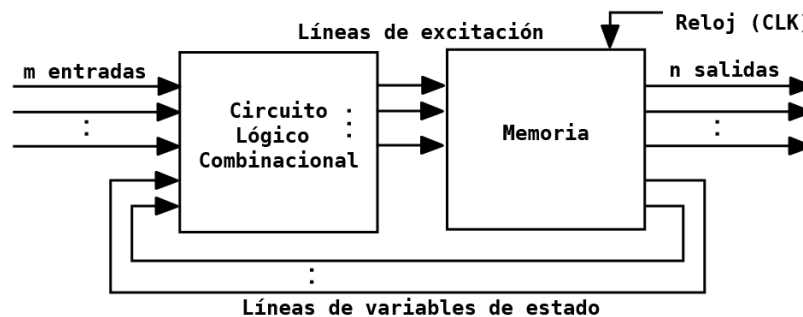


Figura 1: Máquina de estados con circuitos digitales. Basado en (Floyd, 2010)

Existen dos elementos sobre los que se apoya el diseño e implementación de las máquinas de estado, uno gráfico y uno tabular: los **diagramas de estados** y las **tablas de estados**.FSM-CircuitoSecuencial

1.1. Diagramas de estados

El diagrama de estados es en esencia un grafo, compuesto por nodos y arcos, que muestra de manera gráfica el comportamiento dinámico del sistema modelado mediante la máquina de estados (figura 2).

Cada círculo (nodo) en el diagrama representa un estado. El texto que aparece dentro del círculo hace referencia

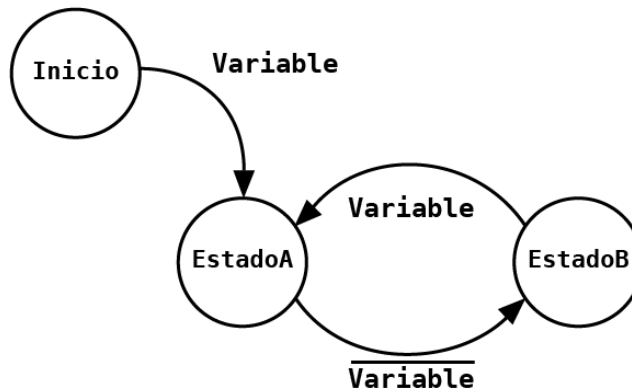


Figura 2: Diagrama de estados simple. *Elaboración propia.*

a un nombre arbitrario que posteriormente se codifica de manera adecuada. El nombre del estado típicamente describe el comportamiento del sistema en ese estado en particular y responde a preguntas como: ¿Qué ocurre en el sistema? ¿Cómo está el sistema en este momento? ¿Qué está “haciendo” el sistema?. El estado inicial permite indicar el primer estado en el que se encuentra la máquina una vez se enciende.

Por otro lado, cada flecha (arco) indica la condición necesaria para pasar de un estado a otro (transición). Esta condición se refiere específicamente a los valores de las entradas requeridos para cambiar del estado de donde sale el arco hacia el estado al que llega el arco. Es posible que para que las transiciones sucedan se requiera del pulso de reloj (máquina síncrona) o que se realicen monitoreando únicamente las variables de entrada (máquina asíncrona). Las variables que se evalúan también se conocen como **variables de estados** y son por lo tanto las encargadas de controlar en qué estado se encuentra la máquina.

1.2. Tabla de estados

La tabla de estados, también llamada tablas de transición (Mano, 2014, 181) relaciona toda la información contenida en el diagrama de manera extensiva. Está formada por cuatro secciones:

- **Estado presente:** muestra el estado actual del sistema. En términos de circuitos secuenciales, en esta sección se indica el estado actual de los flip-flops que conforman la máquina de estados.
- **Entrada:** muestra todos los posibles valores de la variable (o variables) de entrada para cada estado presente.
- **Estado siguiente:** muestra el estado al que pasará la máquina, dada la combinación entre estado presente y la variable de entrada. En el circuito, esto hace referencia al estado en el que quedarán los flip-flops, si la entrada se mantiene y hay un pulso de reloj (para un circuito síncrono).
- **Salida:** muestra la variable de salida, para el estado actual y la entrada presente. Este valor puede hacer referencia a las salidas de los flip-flops de un circuito secuencial, dependiendo de si el circuito cuenta con lógica combinatorial a la salida o no.

Para el diagrama de la figura 2 la tabla de estados asociada podría tener la siguiente forma:

Estado presente	Entrada	Estado siguiente	Salida
Inicio Inicio	$\overline{Variable}$ $Variable$	Inicio Estado A	Inactiva Activa
Estado A Estado A	$\overline{Variable}$ $Variable$	Estado A Estado B	Activa Inactiva
Estado B Estado B	$\overline{Variable}$ $Variable$	Estado B Estado A	Inactiva Activa

Tabla 1: Ejemplo de tabla de estados. *Elaboración propia.*

Nótese que la tabla de estados incluye también la salida del sistema, que hasta el momento no había aparecido en el diagrama de estados de manera explícita. La razón para esto, es que existen dos modelos de máquinas de estados y la ubicación del valor de las salidas en el diagrama depende del tipo de máquina que se esté empleando para modelar el sistema.

Los dos modelos de máquina de estados que se mencionan son: **Máquinas de Moore** y **Máquinas de Mealy**. Cada uno de estos tipos de máquinas tiene sus características particulares, así como unas ventajas y desventajas asociados. Por esta razón, es adecuado conocer los dos tipos y evaluar cual resulta más pertinente de acuerdo con el comportamiento deseado del sistema que se modela y las restricciones en su implementación.

2. Máquinas de Moore

En una máquina de estados de Moore, el valor de las salidas depende **únicamente** del estado en el que la máquina se encuentra. Este “aislamiento” entre el valor instantáneo de las entradas y las salidas del sistema presenta una ventaja importante: la lógica que calcula el valor de las salidas del sistema resulta comparativamente más simple que el de una máquina equivalente de Mealy, (por cuanto tienen un menos número de variables). Como contrapartida, esto generalmente implica un mayor número de estados para conseguir la misma funcionalidad de una Máquina de Mealy equivalente.

En la figura 3 se encuentra el diagrama de estados para una máquina de Moore. Dependiendo de la variable de entrada los estados cambian en un sentido (ABCD) o en el sentido contrario (ADCBA). Esta configuración podría ser usada por ejemplo para crear un contador ascendente/descendente. En este caso, cada estado se encuentra asociado a una salida y su valor (o valores) se escribe debajo del nombre del estado. En la tabla 2 se muestra la información detallada de los estados para la misma máquina.

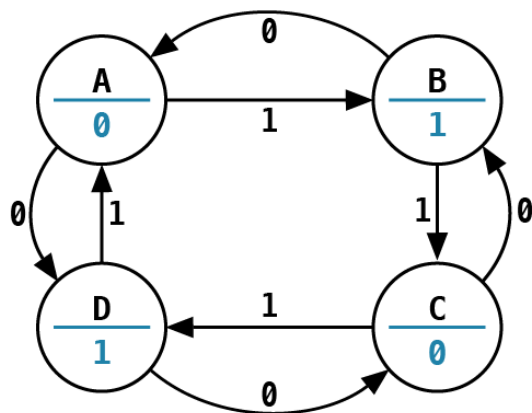


Figura 3: Diagrama de estados de una máquina de Moore. *Elaboración propia.*

Estado presente	Entrada	Estado siguiente	Salida
A	0	D	0
A	1	B	0
B	0	A	1
B	1	C	1
C	0	B	0
C	1	D	0
D	0	C	1
D	1	A	1

Tabla 2: Tabla de estados para una máquina de Moore. *Elaboración propia.*

3. Máquinas de Mealy

En una máquina de estados de Mealy, el valor de las salidas del sistema depende del estado en el que se encuentre y del valor de la entrada. Una ventaja importante de este tipo de máquinas es que permite tener cambios instantáneos en la salida al cambiar la entrada, sin que estos cambios estén limitados por la velocidad del reloj general del sistema. Esto es especialmente útil en aplicaciones en las que la velocidad de respuesta del sistema es de gran importancia.

En la figura 5 se presenta el diagrama de estados para una máquina de Moore, siguiendo el mismo ejemplo anterior. Fíjese que los cambios en las salidas se representan en los arcos, debajo del nombre de las condiciones de entrada. En la tabla 3 se muestra la tabla de estados.

Una forma alterna de ver la tabla de estados para el diagrama se presenta en la tabla 4. Difiere un poco de lo que se ha estudiado, pues para un mismo estado presente, las salidas, dadas por la variable “x” en este caso, van a cambiar.

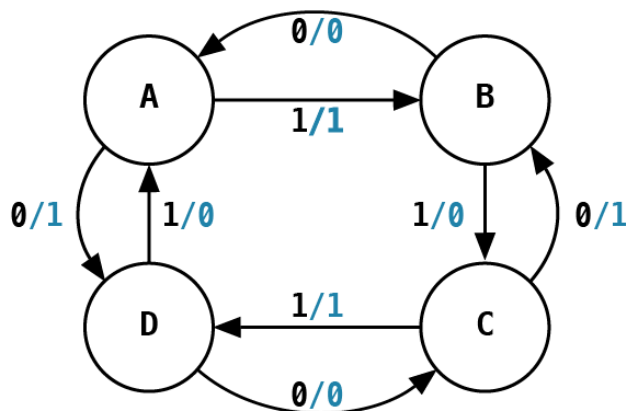


Figura 4: Diagrama de estados de una máquina de Mealy *Elaboración propia*.

Estado presente	Entrada	Estado siguiente	Salida
A	0	D	1
A	1	B	1
B	0	A	0
B	1	C	0
C	0	B	1
C	1	D	1
D	0	C	0
D	1	A	0

Tabla 3: Tabla de estados para una máquina de Moore. *Elaboración propia*.

Con el fin de conocer mejor las características, ventajas y desventajas de las máquinas de Moore y de Mealy, lo invito a regresar al escenario, donde encontrará un recurso con esta información.

4. Algunos ejemplos

Las máquinas de estados pueden ser utilizadas en la solución de problemas de diferentes áreas. Se pueden encontrar casos como el control para un semáforo o controles para electrodomésticos en los que se presenten estados diferenciados, como hornos, lavadoras, etc. En un ámbito industrial también es usual modelar procesos que desean controlarse utilizando FSM. La implementación puede diferir dependiendo del problema y el escenario en el que se aplica. Las tecnologías de circuitos integrados MSI y LSI son las que usualmente se emplean en los primeros casos, mientras que en los ámbitos industriales se prefiere el uso de controladores Lógicos Programables (PLC) que permiten interactuar con diversos tipos de sensores y actuadores teniendo además la posibilidad de modificar el modelado de los sistemas/procesos mediante programación.

Estado presente	Estado siguiente / Salida	
	x=0	x=1
A	D / 1	B / 1
B	A / 0	C / 0
C	B / 1	D / 1
D	C / 0	A / 0

Tabla 4: Tabla de estados alterna para una máquina de Mealy. *Elaboración propia.*

4.1. Un ejemplo de aplicación

Para ilustrar el uso de la teoría de máquinas de estados, a continuación se estudiará como podría abordarse la resolución de un problema de diseño de un sistema de control para la conmutación de un LED basada en un pulsador.

El comportamiento deseado es el siguiente:

1. Al iniciar el sistema el LED se encuentra apagado.
2. Al oprimir el pulsador (P) el LED (L) se enciende y permanece encendido mientras que el pulsador (P) no vuelva a ser activado.
3. Si el LED se encuentra encendido y el pulsador (P) es activado, el LED (L) se apaga y permanece apagado mientras que el pulsador (P) no sea activado.
4. Si el LED se encuentra apagado y el pulsador (P) es activado, el LED (L) se enciende y permanece encendido mientras que el pulsador (P) no sea activado.
5. El comportamiento se repite de manera cíclica.

4.1.1. Diagrama de bloques

Un primer paso para el diseño consiste en hacerse una idea de los elementos generales sobre los que actúa el sistema, es decir: sensores y actuadores. Claramente, en este ejemplo existen dos elementos identificados:

- El pulsador P envía al sistema el valor 1 cuando es activado (pulsado) y 0 cuando es liberado.
- El LED L se enciende al recibir un nivel ALTO y se apaga al recibir un nivel BAJO.

4.1.2. Identificación de entradas y salidas

El siguiente paso consiste en reconocer, de acuerdo a nuestro sistema, cuáles son aquellas entradas que permitirán el cambio entre estados y cuáles salidas reflejan dicho cambio de estados. En el ejemplo tenemos:

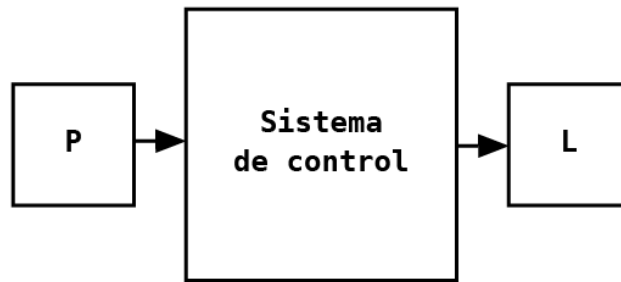


Figura 5: Diagrama de bloques del sistema. *Elaboración propia.*

- **Entradas:** Pulsador P, entrada de 1 bit.
- **Salidas:** LED L, salida de 1 bit.

4.1.3. Solución empleando máquinas de Moore

Al tratarse de una máquina de Moore, cada estado presenta un valor específico de salida que se incluye en el grafo (figura 6 especificando el valor de la salida bajo el nombre del Estado. Esto se ve reflejado también en la tabla de estados (tabla 5).

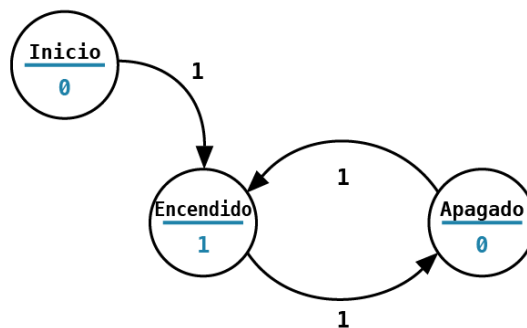


Figura 6: Diagrama de estados, para la máquina de Moore del ejemplo. *Elaboración propia.*

Estado presente	Entrada	Estado siguiente	Salida
Inicio	0 1	Inicio Encendido	0
Encendido	0 1	Encendido Apagado	1
Apagado	0 1	Apagado Encendido	0

Tabla 5: Tabla de estados, para la máquina de Moore del ejemplo. *Elaboración propia.*

4.1.4. Solución empleando máquinas de Mealy

En este caso, tratándose de una máquina de Mealy el valor de la salida (L) depende tanto del estado en el que se encuentra el sistema como del valor de la entrada (P) (figura 7). Esto se ve reflejado también en la tabla de estados (6).

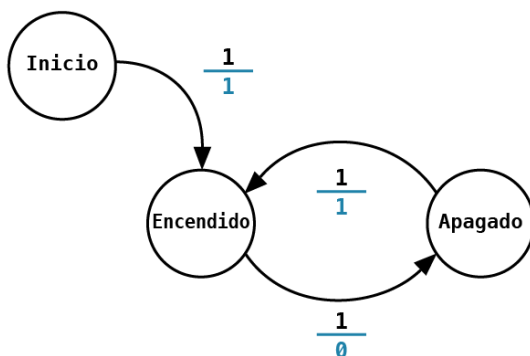


Figura 7: Diagrama de estados, para la máquina de Mealy del ejemplo. *Elaboración propia.*

Estado presente	Estado siguiente / Salida	
	x=0	x=1
Inicio	Inicio / 0	Encendido / 1
Encendido	Encendido / 1	Apagado / 0
Apagado	Apagado / 0	Encendido / 1

Tabla 6: Tabla de estados, para la máquina de Mealy del ejemplo. *Elaboración propia.*

Hasta el momento se ha hecho solamente el diseño a nivel de diagrama de estados y de tablas. En el siguiente escenario se aborda el diseño desde los circuitos secuenciales, aplicando lo visto en la anterior unidad.

Ahora lo invito a regresar al escenario, donde encontrará otro ejemplo detallado sobre el diseño de máquinas de estados, sus diagramas y las tablas de estados asociadas.

Índice de figuras

1	Máquina de estados con circuitos digitales. <i>Basado en (Floyd, 2010)</i>
2	Diagrama de estados simple. <i>Elaboración propia.</i>
3	Diagrama de estados de una máquina de Moore. <i>Elaboración propia.</i>
4	Diagrama de estados de una máquina de Mealy <i>Elaboración propia.</i>
5	Diagrama de bloques del sistema. <i>Elaboración propia.</i>
6	Diagrama de estados, para la máquina de Moore del ejemplo. <i>Elaboración propia.</i>
7	Diagrama de estados, para la máquina de Mealy del ejemplo. <i>Elaboración propia.</i>

Índice de tablas

1	Ejemplo de tabla de estados. <i>Elaboración propia.</i>
2	Tabla de estados para una máquina de Moore. <i>Elaboración propia.</i>
3	Tabla de estados para una máquina de Moore. <i>Elaboración propia.</i>
4	Tabla de estados alterna para una máquina de Mealy. <i>Elaboración propia.</i>
5	Tabla de estados, para la máquina de Moore del ejemplo. <i>Elaboración propia.</i>
6	Tabla de estados, para la máquina de Mealy del ejemplo. <i>Elaboración propia.</i>

Referencias

- Floyd, T. L. (2010). *Fundamentos de sistemas digitales*. Pearson Prentice Hall. (OCLC: 893578510)
- Mano, M. M. (2014). *Diseño digital: con una introducción a Verilog HDL*. Naucalpan de Juárez, México: Pearson Educación. (OCLC: 881629816)
- Ogata, K. (2001). *Modern control engineering* (4th ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.

INFORMACIÓN TÉCNICA



Módulo: Sistemas digitales y ensambladores

Unidad 4: Máquinas de Estados Finitos (FSM)

Escenario 7: Máquinas de Estados Finitos

Autor: Gabriel Eduardo Ávila Buitrago

Asesor Pedagógico: Jeimmy Lorena Romero Perilla

Diseñador Gráfico: Leonardo Stiglitch Campos

Asistente: Jhon Edwar Vargas Villa

*Este material pertenece al Politécnico Grancolombiano.
Por ende, es de uso exclusivo de las Instituciones
adscritas a la Red Ilumino. Prohibida su reproducción
total o parcial.*