

# RESOLUCIÓN DE PROBLEMAS DE MÁQUINAS DE ESTADO FINITO

Bryan V. Alvarado <sup>1\*</sup>, Evelin E. Hidalgo <sup>1</sup>, Jorge A. Cruz <sup>1</sup>

*Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador*

bvalvarado@espe.edu.ec

**Resumen** -Las máquinas de estado finito son una herramienta muy útil para especificar aspectos relacionados con tiempo real, dominios reactivos o autónomos, computación reactiva, protocolos, circuitos, arquitecturas de software, etc. El modelo de FSM (Finite State Machine) es un modelo que posee sintaxis y semántica formales y que sirve para representar aspectos dinámicos que no se expresan en otros diagramas.

**Índice de Términos** -Autónomos-Estado- Semántica-Sintaxis-software.

## I. INTRODUCCIÓN

Para la resolución de problemas de máquinas de estado finito es indispensable tener claro su concepto.

Las máquinas de estado finito FSM, representan una técnica especial de modelado de circuitos lógicos secuenciales; esta concepción es de extrema utilidad en el diseño de circuitos donde todas sus funciones pueden ser exactamente listadas, conteniendo todos los posibles estados y sus condiciones para que estos evolucionen de uno a otro estado, con el desarrollo de los sistemas embebidos y su cada vez mayor velocidad, las máquinas de estado finito se han convertido en una poderosa herramienta para el desarrollo de controladores digitales.

Existen dos representaciones fundamentales de las FSM una relacionada con sus especificaciones llamada diagrama de transición de estados y otra relacionada al hardware conocida como de lógica combinacional/secuencial. Durante esta entrada se brindará los fundamentos de estas técnicas de modelado, la forma como codificar las FSM con lenguaje VHDL y su implementación en los FPGA.

## II. ESTADO DEL ARTE

En 2002, Mikael Kerttu, Per Lindgren, Mitch Thomson y Rolf Drechsler en su estudio sobre la Estimación de la actividad de conmutación de máquinas de estados finitos para síntesis de baja potencia, en su investigación se describe una técnica para calcular la actividad de conmutación de implementaciones de máquinas de estados finitos síncronas (FSM) que incluye la influencia de la correlación temporal entre las siguientes señales de estado. El enfoque se basa en el cálculo de que una FSM se encuentra en un estado dado que, a su vez, se usa para calcular la probabilidad condicional de que el siguiente bit de estado cambie dado su valor de estado actual. Todos los cálculos se realizan utilizando estructuras de datos de diagrama de decisión (DD). Como una aplicación de este método, la siguiente información de actividad de estado se utiliza para la optimización de baja potencia en la síntesis de circuitos mapeados con diagramas de decisión binaria (BDD). (Kerttu, M., Lindgren, P., Thornton, M. y Drechsler, R., 2002, p.1) [1].

Miroshnyk, M., Shkil, A., Kulak, E., Rakhlis, D., Filippenko, I. y Miroshnyk, A. miembros de la asociación de profesores KnNURE Kharkiv, Ucrania en 2020 implementaron un sistema para la Verificación de los sistemas de control FPGA mediante el análisis de la corrección de los diagramas de estado. El trabajo está dedicado

a la verificación de los sistemas de control lógico automático mediante el análisis de la corrección de los diagramas de estado de las máquinas de control de estados finitos que se representan en forma de código en el lenguaje de descripción del hardware. Como método para el análisis de diagramas de estados, se propone utilizar el concepto de ortogonalidad, como un sistema de eventos incompatibles. El análisis de la corrección se lleva a cabo mediante el análisis de los resultados del modelado de comportamiento y la síntesis lógica utilizando herramientas CAD. (Miroshnyk M., Shkil A., Kulak, 2020, p.1) [2].

En 2002, Ming-Der Shieh, Wann-Shyang Ju y Ming-Hwa Sheu miembros del Departamento de Ingeniería Electrónica del Instituto Nacional de Tecnología de Yunlin, Taiwán presentaron un artículo sobre la Asignación de estado de bajo consumo para máquinas de estado finito asíncronas, en este artículo se presenta un modelo preciso para calcular las probabilidades de estado estable y las probabilidades de transición de estado de las máquinas de estado finito asíncronas (AFSM). Sobre la base de este modelo y el diagrama de adyacencia ponderado derivado, se propone una técnica de asignación de estado sin carrera unívoca eficiente para minimizar la actividad de conmutación promedio de las variables de estado, y así reducir potencialmente la disipación de potencia promedio en AFSM. (Ming-Der Shieh, Wann-Shyang Ju, 2002, p.1) [3].

En 2018, Daniel Rivas, Pragna Das, Joaquín Saiz-Alcaine y Lluís Ribas-Xirgo miembros del Departamento de Microelectrónica y Sistemas electrónicos de la Universidad Autònoma de Barcelona, España realizaron un artículo sobre la Síntesis de controladores a partir de diagramas de máquina de pila de estados finitos. En este artículo, combinan máquinas de estados finitos extendidos (EFSM) con pilas de estados. Si bien EFSM coincide con la mayoría de las plantillas de diseño, las máquinas de pila de estado finito extendidas (EFS2M) se adaptan a arquitecturas de agentes de procedimiento para módulos en niveles de abstracción más altos de las redes de los controladores. (Daniel Rivas, Pragna Das, 2018, p.1) [4].

Para el presente producto de unidad, se utilizó Creately que es un programa para crear diagramas de flujo aplicando la temática y conceptos básicos de las máquinas de estado. (Miroshnyk M., Shkil A., Kulak, 2020, p.1), el análisis y el diseño de una máquina de estado finito utiliza módulos en niveles de abstracción (Daniel Rivas, Pragna Das, 2018, p.1) .

## III. MARCO TEÓRICO

Una máquina de estado finito, MEF, es una abstracción matemática que nos permite modelar la computación. En esta lección vamos a usar una MEF para controlar a una unidad aritmética en el cálculo del máximo común divisor, de dos números naturales a, b, denotado por MCD(a,b).

Representación por diagrama de transición de estados

Esta representación se basa en burbujas que representan el estado e incluyen la salida y flechas curvas que especifican la función de evolución de estados estimulada por las entradas, este diagrama debe ser de naturaleza cerrada y debe considerar una condición para

que la maquina alcance su estado inicial, tipicamente este condición debe ser la señal de reinicio Reset[5].

Los tres diagramas representan la misma maquina, que tiene tres estados. A,B y C, tres entradas: rst"que representa la señal de RESET, "x"que estimula la evolución de los estados y clk", la cual no se ve representado en el diagrama pero es implícito ya que estas son unidades secuenciales, por lo que dependen de una señal de reloj, por ultimo se tiene una sola salida llamada z", es posible observar que el estado inicial de la FSM es A, ya que es el único estimulado por la señal de RESET, la evolución de la maquina depende de la entrada y de la progresión de la señal de reloj, por ultimo son los valores de "x"los que determinan el cambio o no de estado, si bien para el ejemplo de basan en una función constante, estas pueden ser de múltiples tipos segun la necesidad o complejidad del diseño del controlador[5].

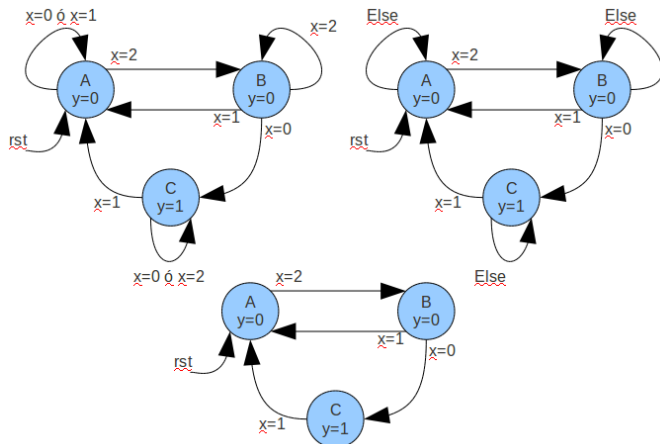


Fig 1. Diagrama de Transición

#### Diagramas de Hardware de lógica combinacional/secuencial

La representación de las FSM con esta perspectiva se basa en los dos bloques perfectamente representados: las sección superior conocida como bloque de lógica combinacional y la sección inferior que es el bloque de lógica secuencial, como se observa esta representación ilustra mejor el proceso controlado por el reloj pero no representa claramente el proceso de los estados.

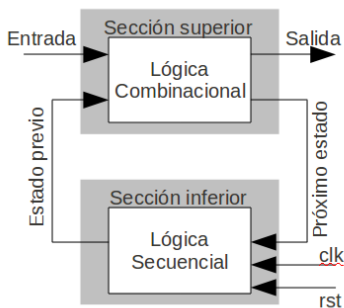


Fig 2. Representación FSM

Con esta representación se puede hacer una mejor extrapolación de la maquina completa a VHDL, separando los dos bloque es dos procesos perfectamente codificables para la sección superior de la lógica combinacional y la parte inferior para la lógica secuencial[6].

Una de las principales limitaciones de la anterior arquitectura, es que la salida generalmente producida por el circuito combinacional puede ser susceptible a fallas transitorias (glitches) que para algunas aplicaciones pueden afectar su rendimiento, si ese fuera el caso es posible corregir este problema agregando un bloque adicional que permita contener el transiente, como lo muestra la siguiente figura:

adicional solamente contiene un flip-flop D que garantiza que tiempo suficiente para que el transiente se supere, es necesario notar

que esta adición hace que la salida final se retrase un ciclo respecto a la obtenida por el modelo de controlador[6].

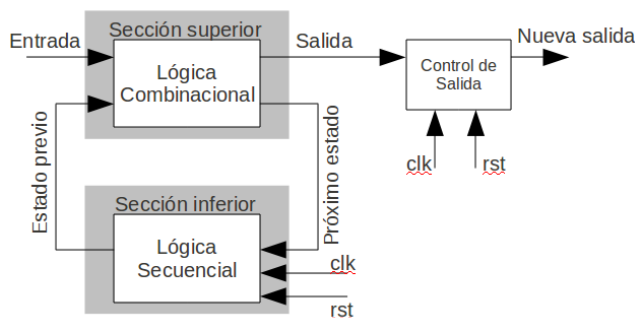


Fig 3. Representación FSM

## IV. PROPUESTA

1. Dibuje el diagrama de estados para la máquina de estado finito cuya tabla de estados es la siguiente. Partiendo del estado s0, calcula la salida para la cadena de entrada 1000110.

Estados	Transición	Salida
	Entrada	Entrada
S0	S0	S4
S1	S0	S3
S2	S0	S2
S3	S1	S1
S4	S1	S0

Fig 4. Tabla de Estados

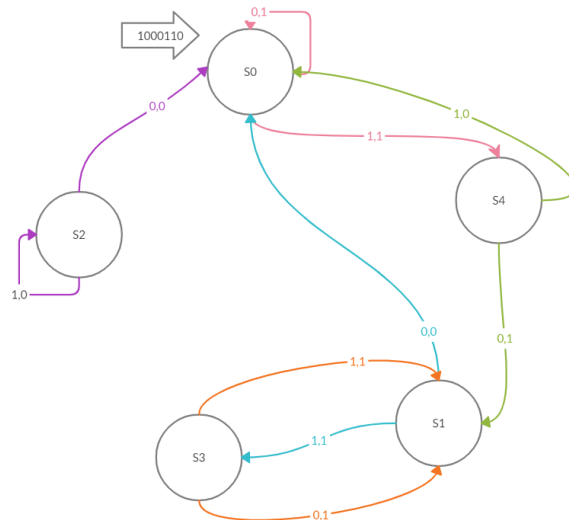


Fig 5. Diagrama de Estados

ESTADO	S0	S4	S1	S0	S0	S4	S0
ENTRADA	1	0	0	0	1	1	0
SALIDA	1	1	0	1	1	0	1

Fig 6. Tabla de Resultados

2. Dibuje el diagrama de estados para la máquina de estado finito cuya tabla de estados es la siguiente. Partiendo del estado inicial s0, calcula la salida para la cadena de entrada abbcc.

Estados	Transición	Salida
	Entrada	Entrada
S0	S0	S3
S1	S1	S1
S2	S1	S1
S3	S2	S3

Fig 7.Tabla de Estados

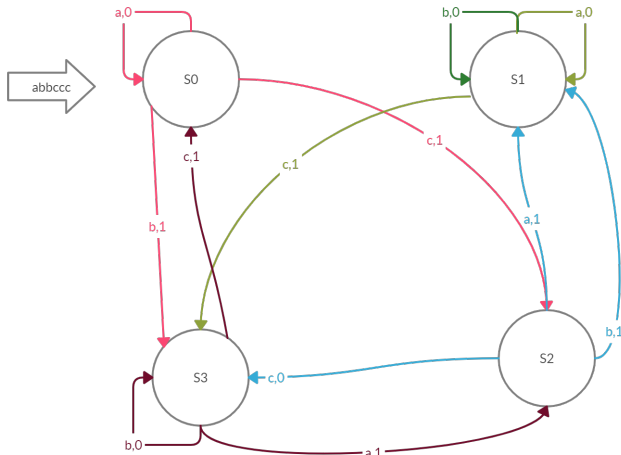


Fig 8.Diagrama de Estados

ESTADO	S0	S4	S3	S3	S0	S2
ENTRADA	a	b	b	c	c	c
SALIDA	0	1	0	1	1	0

Fig 9.Tabla de Resultados

3. Halle la tabla de estados para la máquina de estado finito cuyo diagrama de estados es:

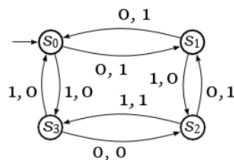


Fig 10.Diagrama de Estados

Entrada	Transición		Salidas	
	Entradas		Entradas	
	0	1	0	1
S0	S1	S3	1	0
S1	S0	S2	1	0
S2	S1	S3	1	1
S3	S2	S0	0	0

Fig 11.Tabla de Estados

4. Construya una máquina de estado finito que modele una máquina expendedora de bebidas que acepta monedas de 5, 10 y 20 centavos. La máquina acepta cualquier moneda hasta que se introducen 25 centavos y devuelve cualquier cantidad que supere los 25 céntimos. Entonces, el cliente puede pulsar los botones y elegir una bebida de cola (C), cerveza (Z) o agua (A).

ESTADO ACTUAL	ESTADO SIGUIENTE									
	ENTRADAS									
	0 CTVS	5 CTVS	10 CTVS	15 CTVS	20 CTVS	25 CTVS	25 CTVS o más	C (coca)	Z (cerveza)	A (agua)
E0 (0 ctvs)	E0	E1	E2	E3	E4	E5	E6	E0	E0	E0
E1 (5 ctvs)	E1	E2	E3	E4	E5	E6	E6	E1	E1	E1
E2 (10 ctvs)	E2	E3	E4	E5	E6	E6	E6	E2	E2	E2
E3 (15 ctvs)	E3	E4	E5	E6	E6	E6	E6	E3	E3	E3
E4 (20 ctvs)	E4	E5	E6	E6	E6	E6	E6	E4	E4	E4
E5 (25 ctvs)	E0	E0	E0	E0	E0	E0	E0	E5	E5	E5
E6 (mayor a 25 ctvs)	E0	E0	E0	E0	E0	E0	E0	E6	E6	E6

Fig 12.Tabla de Estados

SALIDA										
ENTRADAS										
0 CTVS	5 CTVS	10 CTVS	15 CTVS	20 CTVS	25 CTVS	25 CTVS o más	C (coca)	Z (cerveza)	A (agua)	
N	N	N	N	N	B	B+C	B	B	B	B
N	N	N	N	B	B+C	B+C	B	B	B	B
N	N	N	B	B+C	B+C	B+C	B	B	B	B
N	N	B	B+C	B+C	B+C	B+C	B	B	B	B
N	B	B+C	B+C	B+C	B+C	B+C	B	B	B	B
B	B+C	B+C	B+C	B+C	B+C	B+C	B	B	B	B
B+C	B+C	B+C	B+C	B+C	B+C	B+C	B	B	B	B

Fig 13.Tabla de Estados

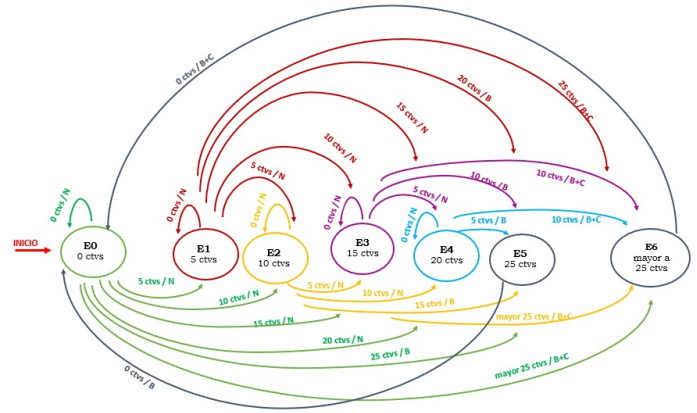


Fig 14.Diagrama de Estados

5. Construya una máquina de estados finito que modele un circuito de riego automático como el mostrado en la figura. El circuito deberá accionar la bomba en las siguientes condiciones: a. El circuito accionará la bomba solamente cuando la tierra esté seca, pero antes debe comprobar las siguientes condiciones:

- Para evitar que la bomba se estropee por funcionar en vacío, nunca se accionará la bomba cuando el depósito de agua esté vacío.
- Si hay restricciones en el riego (época de verano), sólo se podrá regar de noche.
- En el resto del año (si no hay restricciones) se podrá regar de día y de noche (si la tierra está seca).

b. Para la implementación del circuito se dispone de las siguientes entradas:

- S: Señal que indica si la tierra está seca: Tierra seca: S=1; Tierra húmeda: S=0
  - R: Señal que indica si hay restricciones en el riego (es verano): Hay restricciones: R=1 No hay restricciones: R=0
  - D: Señal que indica si es de día o de noche: Día: D=1; Noche: D=0
  - V: Señal que indica si el depósito de agua está vacío: Vacío: V=1; Hay agua: V=0
- c. Y la salida B, que accionará la bomba para regar: Bomba funcionando: B=1; Bomba apagada B=0.

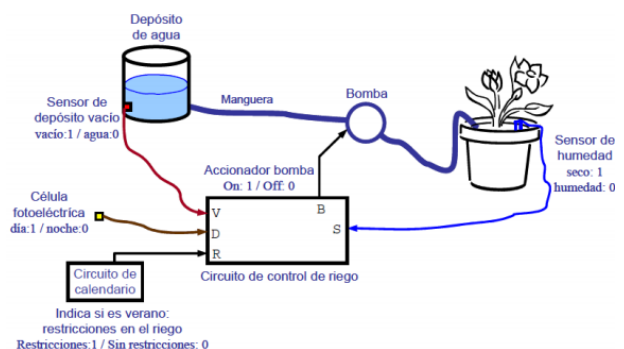


Fig 15. Circuito Control de Riego

Estados	TRANSICIÓN		SALIDA B	
	0	1	0	1
S	S	R	0	1
R	S	D	1	1
V	S	V	1	0
D	R	S	1	1

Fig 16.Tabla de Estados

Diagrama de estados del Sistema de Riego

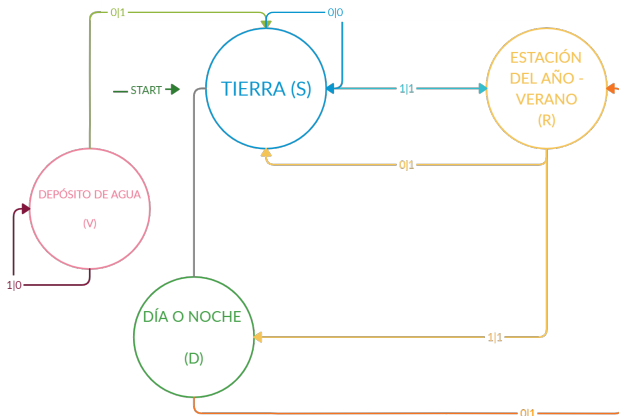


Fig 17. Diagrama de Estados

## V. RESULTADOS

Se realizó una máquina de estado finito que simula una máquina expendedora de bebidas en Thinkercad con un Arduino UNO que envía el código requerido a la Pantalla LCD que mostrará un mensaje INGRESE 150 Un dólar con 50 centavos y el SALDO que ingrese el usuario dependiendo de los botones que pulse ya que el primer botón o pulsador representa 100 Un dólar ,el segundo botón representa 50 cincuenta centavos y el tercer botón representa un ENTER, al confirmar el usuario pulsando un ENTER se reflejará en la pantalla LCD un mensaje que dice TOME SU BEBIDA ,GRACIAS además se tiene un foco Led que se enciende para indicar al usuario que ya ingresado los 150 ya que posteriormente se va a resetear la pantalla automáticamente.

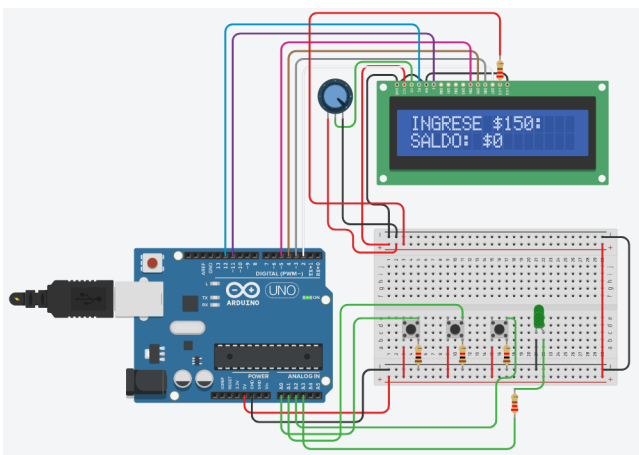


Fig 18. Simulación Máquina expendedora de alimentos

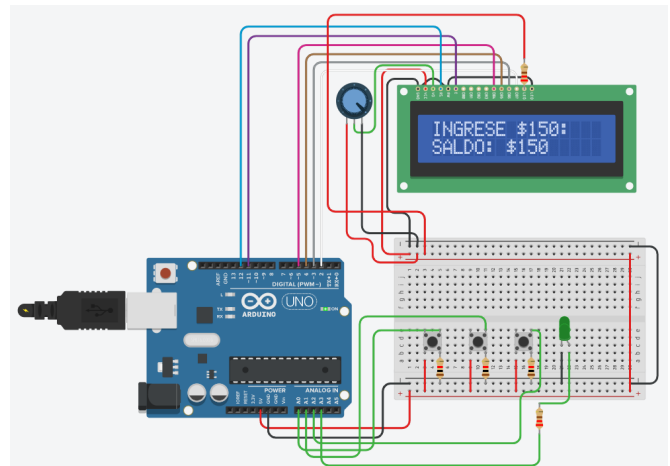


Fig 19. Simulación Máquina expendedora de alimentos

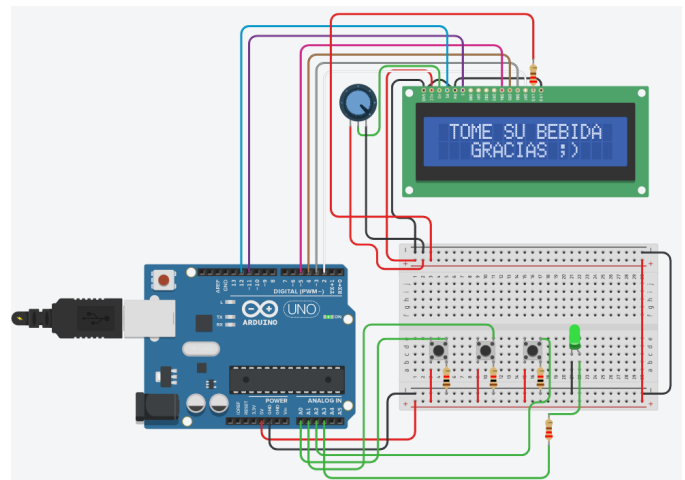


Fig 20. Simulación Máquina expendedora de alimentos

## VI. CONCLUSIONES

En conclusión:

- Las Maquinas de estado Finito al ser una modelación conceptual se puede afirmar gracias a la resolución de los problemas que pueden realizar procesos bien definidos en un tiempo discreto y que tienen relación con el mundo real ya que reciben una entrada, hacen un proceso y entregan una salida. Por lo tanto en la aplicación de circuitos digitales y en la ingeniería las máquinas de estado son capaces de resolver un problema si se tiene una solución algorítmica, es decir, puede ser descrito mediante una secuencia finita de pasos bien definidos.
- La resolución de cualquier problema de máquinas de Estado finito ya sea por el método de Moore o Mealy, tienen como punto crucial el saber identificar de manera correcta el número y tipo de estados sin obviar ningún paso, ya que en base a los estados se realiza la transición y se indica las respectivas salidas de cada sistema, por lo cual los estados son de suma importancia para identificar cualquier tipo de condición o característica especial que indique el problema en cuestión.
- Todas las máquinas de estado finito tienen un conjunto de estados, incluido el estado inicial, un alfabeto fuente y una función de transición que a cada pareja de estado y dato de entrada le asigna el estado siguiente y todos los estados de la máquina le dan unas capacidades de memoria limitada.

## REFERENCIAS

- [1] Kerttu, M., Lindgren, P., Thornton, M. y Drechsler, R. (nd). Estimación de la actividad de conmutación de máquinas de estados finitos para

síntesis de baja potencia. 2002 Simposio Internacional IEEE sobre Circuitos y Sistemas. Procedimientos (Cat. No 02CH37353). doi: 10.1109 / iscas.2002.1010389

- [2] Miroshnyk, M., Shkil, A., Kulak, E., Rakhlis, D., Filippenko, I. y Miroshnyk, A. (2020). Verificación de los sistemas de control FPGA mediante el análisis de la corrección de los diagramas de estado. 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT). doi: 10.1109 / postre50317.2020.9125067
- [3] Ming-Der Shieh, Wann-Shyang Ju y Ming-Hwa Sheu. (Dakota del Norte). Asignación de estado de bajo consumo para máquinas de estado finito asíncronas. Actas del 39º Simposio del Medio Oeste sobre circuitos y sistemas. doi: 10.1109 / mwscas.1996.593176
- [4] Rivas, D., Das, P., Saiz-Alcaine, J. y Ribas-Xirgo, L. (2018). Síntesis de controladores a partir de diagramas de máquina de pila de estados finitos. 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). doi: 10.1109 / etfa.2018.8502451
- [5] Paguayo, A. (2019, 2 mayo). MCI Capacitación. Máquinas de estado. <https://cursos.mcielectronics.cl/2019/06/18/maquinas-de-estado/>
- [6] Sánchez, E. (2019, 2 julio). Maquinas de Estado Finito y Maquinas de Turing. ISUU. [https://issuu.com/esau1409/docs/esau\\_sanchez\\_ci16669954\\_-\\_revista\\_digital](https://issuu.com/esau1409/docs/esau_sanchez_ci16669954_-_revista_digital)