

Uso de máquinas de estado, contadores y memorias

M. Aguilar Ibarra, C. Chávez Hernández, E. Díaz Hernández, D. Moreno Martínez

Departamento de ingeniería en sistemas computacionales, ESCOM IPN

jaguilari1800@alumno.ipn.mx, cchavez1900@alumno.ipn.mx, ediaz1900@alumno.ipn.mx, dmoren2002@alumno.ipn.mx

Resumen— El uso de máquinas de estado, contadores y memorias es fundamental en el diseño de sistemas digitales. Las máquinas de estado permiten modelar y controlar el comportamiento de sistemas complejos, definiendo estados y transiciones entre ellos. Los contadores se utilizan ampliamente para llevar un registro de eventos o para generar señales de temporización, mientras que las memorias permiten almacenar y recuperar datos de manera eficiente. En la práctica, estos elementos se emplean en una gran variedad de aplicaciones, desde controladores industriales hasta sistemas embebidos. Por ejemplo, las máquinas de estado se utilizan para gestionar protocolos de comunicación, los contadores se emplean en sistemas de monitoreo y las memorias son esenciales para el almacenamiento de datos en procesadores y microcontroladores. El diseño y la implementación adecuada de estos componentes es crucial para garantizar el correcto funcionamiento y el rendimiento de los sistemas digitales.

Palabras Clave —Contadores, Máquina de estado, Mealy, Memoria, Moore.

Abstract— The use of state machines, counters, and memories is essential in designing digital systems. State machines allow modeling and controlling the behavior of complex systems by defining states and transitions between them. Counters are widely used to keep track of events or generate timing signals, while memories allow efficient storage and retrieval of data. In practice, these elements find application in a wide variety of fields, from industrial controllers to embedded systems. For example, state machines are used to manage communication protocols, counters are employed in monitoring systems, and memories are essential for data storage in processors and microcontrollers. Proper design and implementation of these components are crucial to ensure the correct functioning and performance of digital systems.

Keywords -- Counters, Mealy and Moore Machines, Memory, State Machine.

I. INTRODUCCIÓN

El presente documento profundiza en el análisis de los conceptos relacionados con el empleo de máquinas de estado, contadores y memorias, los cuales son elementos fundamentales para el diseño de sistemas digitales. Durante el curso de "Diseño de Sistemas Digitales", se han abordado estos temas de manera teórica y práctica, lo que permite ahora realizar un estudio más exhaustivo de ellos.

Una máquina de estados es una representación de un sistema reactivo basado en eventos que pasa de un estado a otro si se cumple la condición que controla el cambio. Las máquinas de estados se utilizaban tradicionalmente para describir sistemas informáticos, pero su uso se ha extendido para modelar la lógica

compleja de sistemas dinámicos tales como aeronaves, automóviles, robots y teléfonos móviles.

Existen muchas formas de expresar una máquina de estados, aunque lo más frecuente es emplear un enfoque gráfico. Un diagrama de transición de estados muestra un número finito de estados con reglas que controlan cuando se pasa de un estado a otro. [1]

Existen dos tipos de máquinas de estado, siendo las máquinas de Mealy y las máquinas de Moore.

A. Máquina de Mealy

La máquina de Mealy se caracteriza por sus salidas que no solo dependen de sus estados, sino también de las entradas al sistema, que se representan definiendo salidas de la máquina, teniendo una estructura de la siguiente manera:

$$MMe = (Q, Ent, Sal, tran, res, q_0)$$

donde:

Q : es el conjunto de estados

Ent : es el alfabeto de entrada

Sal : es el alfabeto de salida

$tran: Q \times Ent \rightarrow Q$, es la función de transición

$res: Q \times Sal \rightarrow Q$, es la función de respuesta

$q_0 \in Q$: es el estado inicial

Al inicio de cualquier computación, la máquina se encuentra en el estado q_0 . Posteriormente cuando la máquina se encuentre en un estado $q_0 \in Q$, y recibe una literal de entrada $e \in Ent$, entonces emite el símbolo de salida $s=res(q, e)$ y transita el nuevo estado $p=tran(q, e)$.

Gráficamente, se presenta de la siguiente manera en la ilustración 1:





q_0 es el estado inicial. Si se está en q y llega e entonces se emite $s = res(q, e)$ y se transita a $p = tran(q, e)$.

Ilustración 1. Figura representativa de máquina de estados

Ejemplo. Repetición final de un mismo símbolo. Una máquina de Mealy que reconozca a las palabras en $(0+1)^*$ que terminan con la repetición de un mismo símbolo. Es decir, que reconozca a palabras en el alfabeto $L=(0+1)^*(00+11)$. Gráficamente, la máquina se representa en la ilustración 2:

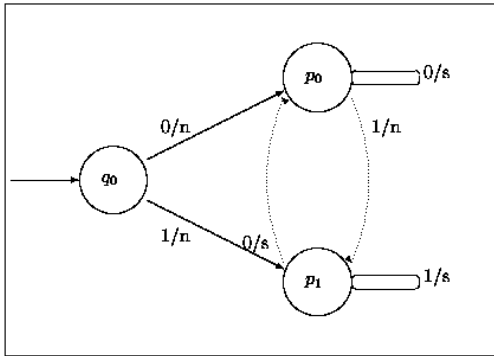


Ilustración 2. Ejemplo de máquina de Mealy con repetición final de un mismo símbolo

La interpretación de cada estado es natural:

q_0 : estado inicial

p_0 : estado de “haber llegado un 0”

p_1 : estado de “haber llegado un 1”

Se tiene una respuesta afirmativa cuando se permanece en un mismo estado. Las componentes de las máquinas son:

$$Q = \{q_0, p_0, p_1\}, Ent = \{0, 1\}, Sal = \{n, s\}. [2]$$

Y se muestran en la ilustración 3:

<i>tran</i>	0	1
q_0	p_0	p_1
p_0	p_0	p_1
p_1	p_0	p_1

<i>res</i>	0	1
q_0	n	n
p_0	s	n
p_1	n	s

Ilustración 3. Tabla de máquina de estados con repetición final de un mismo símbolo

B. Máquina de Moore

La máquina de Moore puede ser similar a la máquina de

Mealy, salvo que la respuesta solo depende del estado actual de la máquina y es independiente de la entrada. Precisamente, una máquina de Moore es una estructura con la siguiente forma:

$$MMo = (Q, Ent, Sal, tran, res, q_0)$$

donde:

Q : es el conjunto de estados

Ent : es el alfabeto de entrada

Sal : es el alfabeto de salida

$tran: Q \times Ent \rightarrow Q$, es la función de transición

$res: Q \times Sal \rightarrow Q$, es la función de respuesta

$q_0 \in Q$: es el estado inicial

Al inicio de cualquier computación, la máquina se encuentra en el estado q_0 . Posteriormente, cuando la máquina se encuentra en un estado $q_0 \in Q$, y recibe una literal de entrada $e \in Ent$, entonces transita el nuevo estado $p = tran(q, e)$ y emite e símbolo de salida $s = res(p)$.

Ejemplo. Se da un número $n \in \mathbb{N}$ en su representación binaria y se requiere calcular se residuo módulo 3. Teniendo su representación gráfica en la ilustración 4:

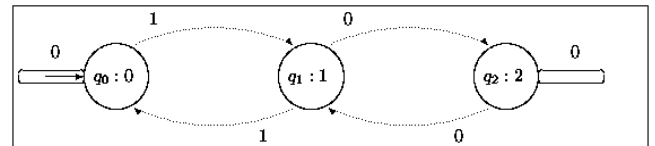


Ilustración 4. Representación de máquina de tres estados

Las funciones de transición y de respuesta quedan especificadas de la siguiente manera en la ilustración 5:

<i>tran</i>	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

<i>res</i>	
q_0	0
q_1	1
q_2	2

Ilustración 5. Tabla de máquina de tres estados

Por inducción en la longitud n de cualquier palabra $\sigma \in (0+1)^*$, que sea la representación en binario de un numero x_σ se puede ver que la respuesta final obtenida al aplicar σ es $x_\sigma \bmod 3$. En efecto para $n = 1$, con las palabras ‘0’ y ‘1’ se tienen las respuestas correctas 0 y 1. Sea $n > 0$. Al suponer que para una palabra σ , de longitud $n-1$, se tiene como respuesta final i , donde $x \equiv i \bmod 3$ y x es el número representativo en binario por σ . Para $s \in (0+1)$ el numero representado por la concatenación de σ con s , σs es $2x+s$, el cual es

congruente módulo 3 con $2i + smod\ 3$. Al tabular estos últimos valores se tiene lo siguiente en la ilustración 6: [3]

$i \backslash s$	0	1
0	0	1
1	2	0
2	1	2

Ilustración 6. Tabla de autómeta (0+1)*

C. Contadores

Un contador es un circuito en el que sus salidas siguen una secuencia fija que cuando termina vuelve a iniciar, o circuitos que reciben sus datos en forma serial y ordenados en distintos intervalos de tiempo.

Los pulsos de entrada pueden ser pulsos de reloj u originarse en una fuente externa y pueden ocurrir intervalos de tiempo fijos o aleatorios.

Existe una gran cantidad de contadores con diferentes características y funcionamiento, las cuales se pueden identificar por las funciones de sus entradas. [4]

D. Contadores asíncronos

Las salidas de cada flip-flop sirven de entrada CP para disparar otro flip-flop. El primer biestable tendrá una entrada de tipo asíncrono, es decir, que se acertará de forma aleatoria y cuando lo haga el circuito realizará una cuenta. Esto se muestra en la ilustración 7. El resto del tiempo, los flip-flops no cambiarán su estado presente.

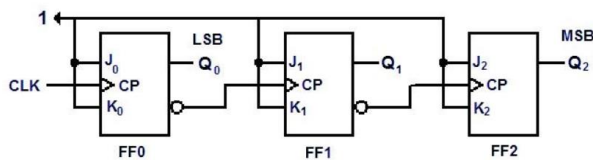


Ilustración 7. Ejemplo de contador asíncrono

E. Memorias

Una memoria es un sistema digital (de naturaleza combinacional y/o secuencial) capaz de almacenar información binaria organizada de forma tal que el acceso a la misma se realiza en forma parcial. Es decir que, en un momento dado, sólo se puede acceder a una porción de la memoria llamada: palabra. Esta palabra tiene una longitud de n bits.

F. Elementos de Memoria

Se le llama elemento de memoria (biestable), al dispositivo

elemental capaz de almacenar un bit. Existen varios biestables que se usan en la construcción de memorias.

- Circuito candado BIPOLAR
- Circuito candado MOSFET
- Circuito Cg de un transistor MOSFET
- Resistencia implementada con juntas PN (DIOSDOS, transistores BIPOLARES) o canales p o n de transistores MOSFET.
- Fusibles
- Capacitor con PUERTA flotante Cg de un transistor MOSFET

G. Modo de Funcionamiento

- Especificar la posición de la palabra a la que se pretende acceder (direccionamiento). Existen tres formas de hacerlo:
 - Suministrar una DIRECCION de m bits de forma que 2^m indica la cantidad de posiciones de la memoria
 - Acceder DIRECTAMENTE a una posición, determinada en función del primer o último dato
 - Suministrar una información (DATO) a fin de acceder a otra información (DATO) asociada a la ingresada, o bien verificar que la información suministrada esté o no en la memoria
- Suministrar la información (DATO) en caso de pretender ingresar (ESCRIBIR) en la memoria
- Suministrar las señales de Control necesarias para ingresar (ESCRIBIR) o extraer información (LEER)

La culminación de estos tres pasos da lugar a lo que se llama:

- Ciclo de lectura
- Ciclo de escritura

Estos ciclos consisten en una secuencia de señales a fin de realizar el acceso a la memoria en particular. [5]

H. Clasificación

La clasificación de memorias electrónicas se muestra en la ilustración 8:

MEMORIAS ELECTRÓNICAS	ACCESO	TIPO	TECNOLOGÍA		VOLATILIDAD	ELEMENTO DE MEMORIA
MEMORIAS ELECTRÓNICAS	ALEATORIAS (RAM)	ACTIVAS	RAM Estáticas (SRAM)	BIPOLARES	VOLATIL	Candado BIPOLAR
			RAM Dinámicas (DRAM)	MOS		Candado MOS
				FPM		Capacitor Cg
		PASIVAS	ROM		NO VOLATIL	Juntura PN o canal MOSFET
			PROM			Fusible
			RPRM	EPROM		Capacitor Cg con puerta flotante
	SERIE (SAM)	DINÁMICOS		EEPROM	VOLATIL	Capacitor Cg
				FLASH		Candados
	ASOCIATIVAS (CAM)					

Ilustración 8 Tabla de clasificación de memorias electrónicas

II. DESARROLLO

Diseño de la máquina de estado para controlar el contador de veinticuatro horas.

La idea de esta máquina de estado es: cada que el contador de veinticuatro horas se incremente en uno funcione como un contador BCD normal, hasta que las decenas se incrementen a dos, las unidades llegarán hasta tres.

Para controlar este comportamiento se decidió utilizar un contador Johnson de dos bits que se reinicie cuando sus dos salidas sean '1' (Ilustración 9) donde el bit uno representa que se ha contado 10 horas y el bit cero representa que ha contado 20 horas y por lo tanto las unidades únicamente contarán hasta 4 esta máquina utilizará la señal de reinicio del contador de unidades de hora como señal de reloj.

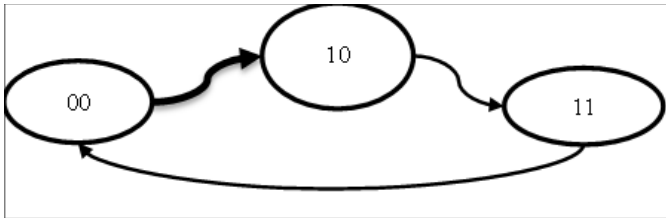


Ilustración 9. Diagrama de estados para contador de 24H

Implementación de la máquina de estado con un Flip - Flop D

Por la tabla de transición de estados Tabla 1 Transición de estados con Flip - Flop D se obtuvo las simplificaciones para las dos entradas D, de los bits de salida de la máquina de estado [Tabla 2 Simplificación para la entrada D del Flip - Flop 1 y Tabla 3 Simplificación para la entrada D del Flip - Flop 0] y finalmente las ecuaciones booleanas [Ecuación 1 Ecuación Booleana para D1].

El circuito consta de contadores BCD, compuertas lógicas, decodificadores BCD y la máquina de estado presenta anteriormente. El funcionamiento es que en cada ciclo de reloj de 1 Hz, el contador de segundos incrementará su valor y posteriormente enviará una señal en estado alto cuando llegue a una cuenta límite, que se representa con una compuerta AND el número límite al que se desea contar y en este caso es de 10, siendo esta señal usada para reiniciar el mismo contador y como

señal de reloj para el contador siguiente (ilustración 10) contador con reinicio en límite, que son las decenas de segundo y así sucesivamente para todos los contadores y de esta manera controlamos todo el conteo con un solo reloj de 1 Hz.

En cada valor de conteo binario es transferido a un decodificador BCD a 7 segmentos para mostrar una salida legible en el reloj. Lo último a resaltar del circuito es el conteo de unidades de hora, que es cuando las decenas de minuto se reinician en 6, es decir, se ha llegado al minuto 59 y empezará el conteo de las unidades de hora de has 9, reiniciándose en 10 y siendo esa señal de reinicio la señal de reloj del contador Johnson truncado para contar 20 horas, ya que, de lo contrario, el reloj terminaría en las horas 29, siendo ambas condiciones las de reinicio del mismo contador (ilustración 11).

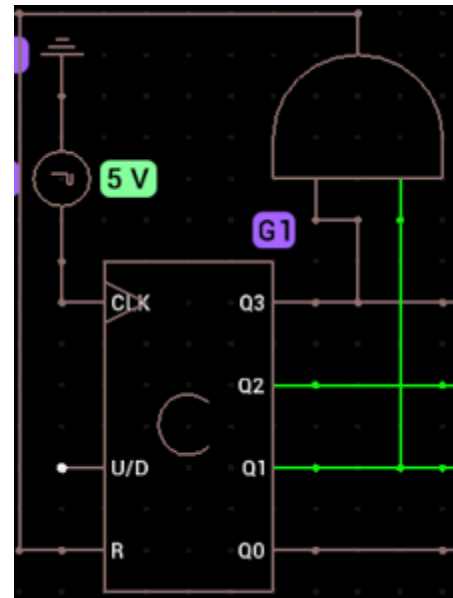


Ilustración 10. Contador con reinicio en límite

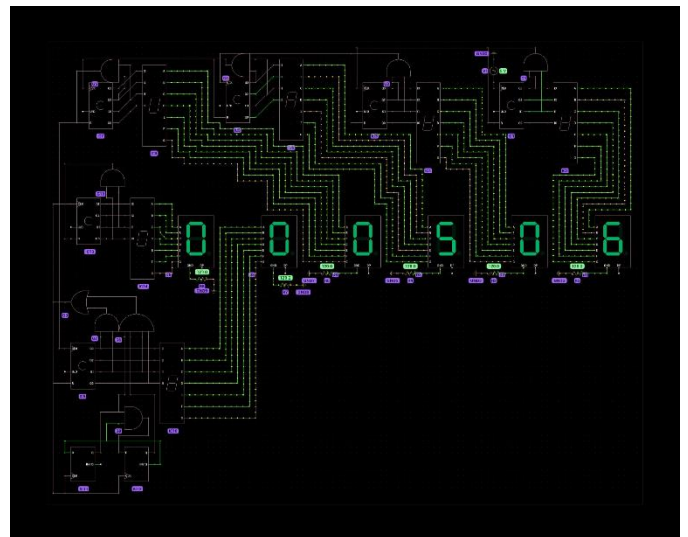


Ilustración 11. Circuito de reloj

I. RESULTADOS Y DISCUSIÓN

Estado actual	Estado siguiente	$D_1 D_0$
00	10	1 0
10	11	1 1
11	00	0 0

Tabla 1. Transición de estados con Flip - Flop D

Tabla 2.

		D_1	
		0	1
Q_1	0	0	1
	1	1	0

Simplificación para la entrada D del Flip - Flop 1

		D_0	
		0	1
Q_1	0	0	0
	1	1	0

Tabla 3. Simplificación para la entrada D del Flip - Flop 0

$$D_1 = \overline{Q_0}$$

Ecuación 1. Ecuación Booleana para D1

$$D_0 = Q_1 \overline{Q_0}$$

Ecuación 2 Ecuación Booleana para D0

III. CONCLUSIONES

Los resultados obtenidos en la práctica cumplen con el objetivo establecido. Al profundizar en lo que es un contador BCD (Binary-Coded Decimal), encontramos que este tipo de contador cuenta en código decimal desde 0000 hasta 1001 y luego vuelve a 0000. A diferencia de una cuenta binaria directa, donde el patrón es regular, el contador BCD no sigue un patrón uniforme debido a que se produce el retorno a cero desde 9 en lugar de 15 (aunque esta posibilidad existe). Es importante destacar que el BCD es un código, no un sistema de numeración completo, y su propósito principal es representar valores decimales en formato binario.

Además, es fundamental comprender las máquinas de estado, ya que las memorias funcionan de esa manera. Las memorias, tanto RAM como ROM, utilizan lógica combinacional y secuencial para almacenar y recuperar datos. También es importante entender algunos conceptos:

Lógica Combinacional

La lógica combinacional se basa en puertas lógicas (como AND, OR, NOT) para realizar operaciones específicas.

En el contexto de las memorias, la lógica combinacional se utiliza para direccionar las celdas de memoria y para realizar operaciones de lectura y escritura. [6]

Por ejemplo, al acceder a una dirección de memoria específica, la lógica combinacional determina qué celda se activa y cuál se lee

o escribe.

Lógica Secuencial

La lógica secuencial implica el uso de elementos de almacenamiento, como flip-flops o registros, para mantener estados y realizar operaciones en función de señales de reloj.

En las memorias, la lógica secuencial se utiliza para mantener los datos almacenados incluso después de que se retire la alimentación. Los registros de dirección y los registros de datos son ejemplos de componentes secuenciales en las memorias. [6]

En resumen, el contador BCD y las memorias son componentes esenciales en la arquitectura de sistemas digitales. Comprender su funcionamiento y cómo interactúan con la lógica combinacional y secuencial nos permite diseñar sistemas más eficientes y confiables.

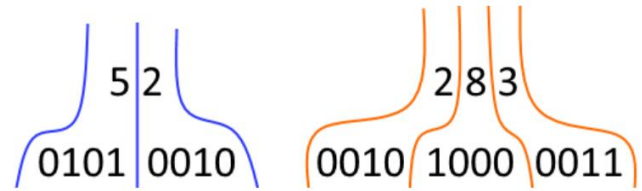


Ilustración 12. Conversión de decimal a BCD

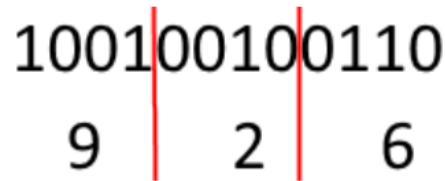


Ilustración 13. Conversión de BCD a decimal

AGRADECIMIENTOS (Heading 5)

Agradecemos a ISSI Integrated Silicon Solution Inc. por colocar una tabla de verdad tan clara en el datasheet de sus memorias SRAM.

REFERENCIAS

- [1] © 1994-2024 The MathWorks, Inc. Introducción a las state machine. Máquina de estados.
<https://la.mathworks.com/discovery/state-machine.html>
- [2] Guillermo Morales Luna, (27 de junio, 2000) Máquinas de Mealy, Máquinas secuenciales
<https://delta.cs.cinvestav.mx/~gmorales/ta/node49.html>
- [3] Guillermo Morales Luna, (27 de junio, 2000) Máquinas de Moore, Máquinas secuenciales
<https://delta.cs.cinvestav.mx/~gmorales/ta/node50.html>
- [4] González Ruben, Bardón Marcos, Salcines José Antonio, Electrónica Digital 1. Contadores.
<https://personales.unican.es/manzanom/planantiguo/edigitali/CONTG5.pdf>
- [5] Memorias electrónicas.
http://www1.frm.utn.edu.ar/tecnicad1/_private/Apuntes/PRESENTACION%20MEMORIAS.pdf
- [6] Thomas L. F. (2006). Fundamentos de Sistemas Digitales. Novena Edición. Pearson Educación.