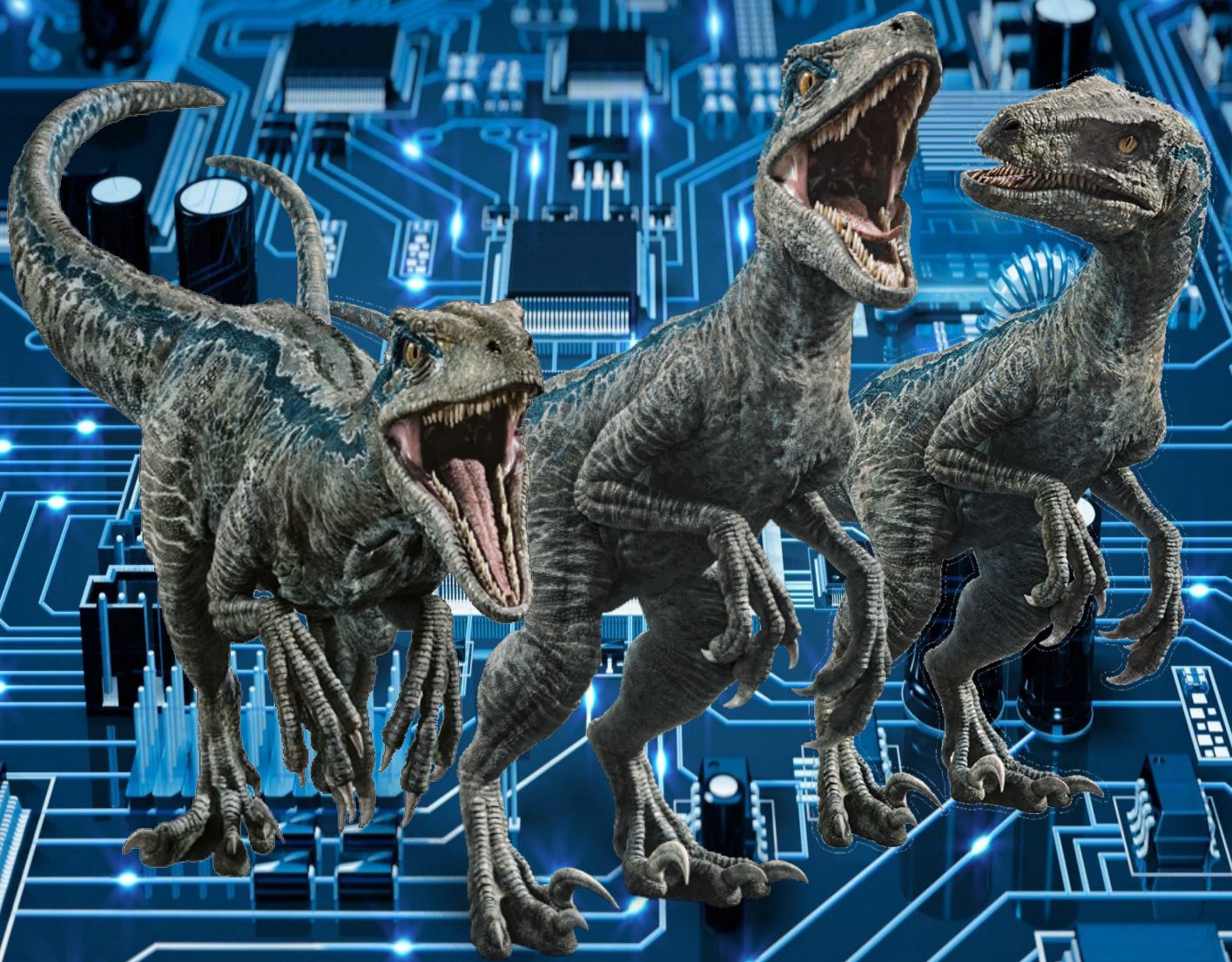


ARQUITECTURA DE SISTEMAS DIGITALES DE CONTROL

Un Enfoque Práctico



Nahur M. Meléndez Araya

©Nahur M. Meléndez Araya

© ARQUITECTURA DE SISTEMAS DIGITALES DE CONTROL: Un Enfoque Práctico

Registro de Propiedad Intelectual EN TRAMITE

IMPRESO EN CHILE/PRINTED IN CHILE

Derechos Reservados para todos los países.

Prohibida su reproducción total o parcial.

Fotocopiar libros es un delito severamente

penado por la ley y lesiona los legítimos

derechos de autor.

ARQUITECTURA DE SISTEMAS DIGITALES DE CONTROL

Un Enfoque Práctico

Nahur M. Meléndez Araya

2021

Acerca del Autor

NAHUR M. MELÉNDEZ ARAYA es Magíster en Ingeniería Informática e Ingeniero en Computación e Informática de la Universidad Católica del Norte, Antofagasta, Chile. Actualmente, es académico del Departamento de Ingeniería Informática y Ciencias de la Computación de la Universidad de Atacama. Es autor de publicaciones nacionales e internacionales, sobre temas de Ingeniería de Requisitos para productos de software, Computación aplicada e Informática Educativa. Ha participado como conferencista en paneles, foros, seminarios, workshop y grupos de análisis a nivel ejecutivo, académicos y niveles de gobierno en temas relacionados con la alta tecnología.

CAPITULO 1

Aspectos generales sobre circuitos secuenciales

En los circuitos combinacionales la salida en un determinado momento depende únicamente de los valores de las señales de entrada en el mismo instante. Sin embargo, en el mundo real la mayor parte de los sistemas con los que nos enfrentamos tienen una dimensión adicional: el funcionamiento de los mismos depende no únicamente de sus entradas actuales, sino también de la historia por la que han pasado. Así, los circuitos secuenciales surgen para solucionar las limitaciones intrínsecas de los combinacionales.

Los **circuitos secuenciales** son sistemas que, además de entradas y salidas, también tienen estados que recuerdan la historia pasada por el circuito. Utilizan la información del estado conjuntamente con una combinación lógica de sus entradas de datos para determinar el futuro estado del sistema y sus salidas. Por tanto, una de sus características es que las mismas entradas en estados diferentes dan lugar a salidas distintas, ya que estas dependen también del estado.

Muchos de los sistemas digitales prácticos se realizan siguiendo la filosofía de los circuitos secuenciales (circuitos de control, sistemas de alarma y seguridad, relojes, etc.). Podemos considerar como ejemplo una máquina expendedora (simplificada), que representaríamos como se muestra en la figura 1.

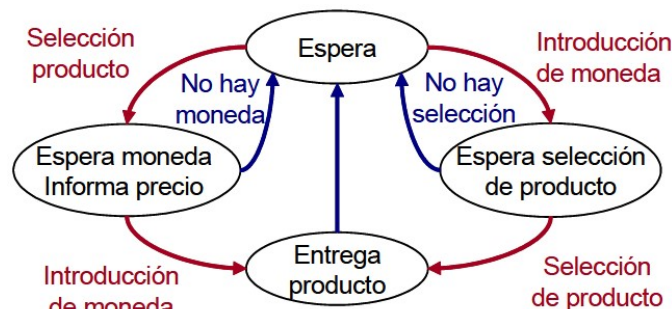


Figura 1. Representación de una máquina expendedora.

En la figura 1 pueden verse varios estados, en los que el sistema espera monedas o la selección de productos, o entrega el producto. Estos estados memorizan la última acción del usuario, de forma que el sistema puede responder a las nuevas acciones de forma diferente dependiendo de la historia pasada: si seleccionamos producto, la salida no será la misma si ya le hemos entregado el dinero o todavía no.

Este puede ser un ejemplo sencillo de máquina secuencial, y a partir de ahora formalizaremos el concepto y estudiaremos la forma de analizar y diseñar este tipo de circuitos.

Estructura General y Funcionamiento

La estructura general de un circuito secuencial responde al diagrama de bloques de la figura 2 (modelo de Huffman). Como se puede observar, incluye un circuito combinacional y un bloque de memoria (formado por flip-flops, por ejemplo). Existen “u” entradas físicas cada una de las cuales la identificaremos por la letra “X” (X_0, X_1, \dots, X_{u-1}), “v” líneas de salida, que identificaremos por la letra “Z” (Z_0, Z_1, \dots, Z_{v-1}), y “p” elementos de memoria que identificaremos por la letra M (M_0, M_1, \dots, M_{p-1}). Cada elemento de memoria tiene unas entradas que vamos a denominar E_i y una salida que llamaremos Q_i que almacenará el estado actual del circuito. Además existe una señal auxiliar, la señal de reloj (CLK), que juega un papel fundamental en los sistemas secuenciales síncronos. Esta última señal no existe en los sistemas secuenciales asíncronos.

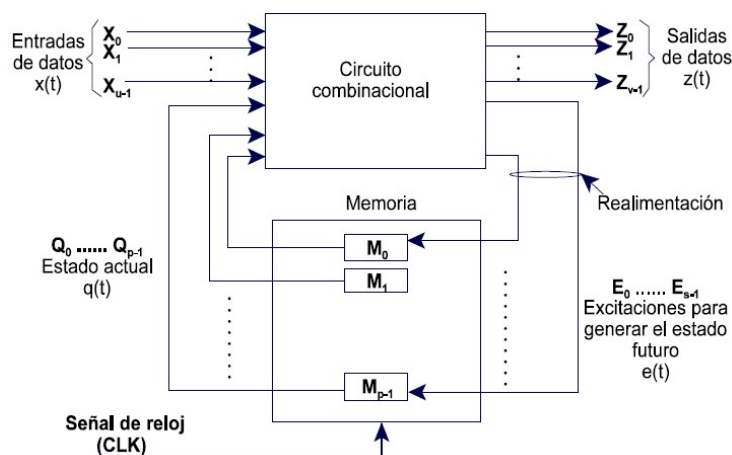


Figura 2. Estructura general de un circuito secuencial

Como se puede observar en la figura 2, al **circuito combinacional** llegan dos tipos de datos: las entradas de datos del circuito ($X_0(t), X_1(t), \dots, X_{u-1}(t)$) y el estado actual, es decir, las salidas de la memoria ($Q_0(t), \dots, Q_{p-1}(t)$). Las salidas del circuito combinacional son las salidas de datos ($Z_0(t), Z_1(t), \dots, Z_{v-1}(t)$) y las entradas de los elementos de memoria ($E_0(t), \dots, E_{s-1}(t)$), (si bien en ciertos casos algunas de estas variables pueden ser comunes). El hecho de que parte de la información de salida del circuito combinacional se introduzca de nuevo a su entrada, previo paso por la memoria, se conoce como **realimentación**.

Tipos de Circuitos Secuenciales

Los sistemas secuenciales se pueden clasificar en dos grandes bloques: **síncronos y asíncronos**. Esta clasificación se hace atendiendo a los tipos de elementos de memoria (M_i) utilizados. La diferencia entre los sistemas secuenciales síncronos y asíncronos está en que en los primeros los cambios de estado son controlados por una señal de referencia común (señal de reloj) y en los segundos no.

Las salidas de un circuito secuencial asíncrono, ante un cambio en las señales de entrada de datos, alcanzan sus niveles definitivos transcurrido un tiempo que será función de los tiempos de propagación del circuito combinacional y de los elementos de memoria (las salidas de datos $-Z_i-$ se estabilizarán cuando se establezcan las salidas de los elementos de memoria, $Q_i(t)$). Estos tiempos pueden variar entre un valor mínimo y un máximo, y, por tanto, introducen un cierto grado de incertidumbre en los tiempos que tardan en propagarse los cambios de las variables de entrada a las salidas, y unas salidas pueden cambiar antes que otras. Hay que advertir que pueden existir situaciones en las que no se alcance la estabilidad (estados **metaestables**). En el diseño de circuitos se trata siempre de evitar la existencia de este tipo de estados. Dos técnicas usuales para ello son, una impidiendo excitaciones marginales y otra evitando, o al menos reduciendo, la propagación de señales metaestables).

Según esto, en los circuitos secuenciales **asíncronos** después de producirse un cambio en las entradas hay que esperar un tiempo hasta que se establezcan los valores de las salidas y sólo después de que se alcance esta estabilización se podrá realizar otro cambio en sus entradas, ya que en caso contrario no se verán reflejados estos cambios en las salidas. Evidentemente, el tiempo mínimo entre dos cambios consecutivos en las entradas de datos es función de los tiempos de propagación del circuito combinacional y de los elementos de memoria.

Los circuitos síncronos eliminan la incertidumbre producida por los tiempos de propagación utilizando una señal de sincronismo o señal de reloj. Los cambios en las entradas y en los estados internos sólo tienen efecto sobre el resto del sistema en instantes discretos de tiempo a los que denominaremos $t, t+1, t+2, \dots$. Estos instantes suelen coincidir con los flancos de subida o bajada de la señal de reloj. Así, si nos fijamos en el circuito de la figura 2, las salidas que adopten los elementos “Mi” de memoria después de un flanco de reloj (CLK), serán función únicamente de los valores que tenían sus entradas en el instante anterior al flanco de CLK, aunque estas hayan variado varias veces desde el último flanco de reloj.

Finalmente queremos indicar que, si bien parece que los circuitos síncronos presentan ventajas frente a los asíncronos, no siempre es así. De hecho, al realizar el diseño de un circuito VLSI (*Very Large Scale Integration*), es necesario establecer una comparación rigurosa entre las alternativas síncrona y asíncrona, a fin de valorar las limitaciones y restricciones de ambos modos de operación. Actualmente existe un auge e interés por el diseño de circuitos VLSI asíncronos o **autotemporizados**, ya que el aumento de la velocidad de operación y la complejidad de los circuitos digitales conllevan grandes problemas para conseguir que las señales de reloj que se propagan por caminos diferentes tengan idénticos tiempos de propagación. Esto hace que la distribución de la señal de reloj dentro de los circuitos digitales sea una tarea compleja, siendo una de las razones de esta complejidad los retrasos en dicha señal (lo que se conoce como *clock skew*), además de los problemas de interferencias que puede sufrir (*cross-talk*). No es, sin embargo, objeto de este capítulo el estudio de todos estos detalles.

Aspectos Básicos sobre Circuitos Secuenciales Síncronos

Si bien la palabra **autómata** puede tener diferentes significados (es frecuente relacionar este término con las máquinas capaces de imitar los movimientos de los seres vivos, equipos electrónicos de control industrial, etc.), en nuestro caso se utiliza para definir sistemas electrónicos capaces de tratar información. Así, un **autómata** puede ser un dispositivo con unos terminales de entrada y salida de datos (ver figura 1) y una señal de reloj (CLK), que transforma los datos de entrada en otros datos de salida, en función de la historia pasada por él, de forma sincronizada con la señal CLK. En este contexto, ejemplos de autómatas sencillos son: contadores digitales, sumadores, registros, unidades de control, etc. Un ejemplo de autómata más complejo sería un ordenador.

Queremos indicar que la realización física de los sistemas secuenciales síncronos (autómatas) puede llevarse a cabo utilizando diferentes alternativas tecnológicas, si bien actualmente lo más frecuente es utilizar Dispositivos Lógicos Programables (PLDs).

Como hemos visto, un **circuito secuencial síncrono** es un circuito secuencial (es decir, un sistema que tiene entradas, salidas y estados que recuerdan la historia pasada por el circuito, y cuya salida depende tanto de la entrada como del estado en que se encuentre), en que las transiciones entre estados se producen en momentos discretos, que denominaremos t , $t+1$, $t+2$, etc. Estos momentos coinciden con los flancos activos de subida o bajada de los pulsos de una señal de reloj.

El modelo matemático que describe un sistema secuencial síncrono es la **máquina de estados finitos o autómata de estados finitos**. El calificativo de “estados finitos” se debe a que la memoria de que dispone el sistema secuencial para recordar en qué estado está es finita y, por tanto, solamente podrá almacenar un número finito de estados diferentes.

Con todo esto, si a una **máquina de estados finitos**, que denominaremos abreviadamente MEF, se le introducen datos a lo largo del tiempo, la salida, $z(t)$, de la MEF, en un instante t será función de la entrada en ese instante, $x(t)$ y de los estados internos $q(t)$. Si la memoria tiene p elementos (FF_0, \dots, FF_{p-1}), siendo p un número finito, el número máximo de estados que se pueden almacenar será 2^p .

Observando la figura 2, se puede ver que las entradas de datos $x(t)$, junto con las variables de estado $q(t)$, constituyen las entradas del circuito combinacional. La salida del circuito combinacional está formada por las salidas de datos $z(t)$ y por las excitaciones de los biestables $e(t)$, para que se produzca la transición al siguiente estado $q(t+1)$ con el siguiente flanco de reloj.

El funcionamiento del autómata es tal que si recibe una nueva entrada en t , $x(t)$, $\{X\}$, (ésta se hace efectiva mediante la señal de reloj en el instante t) y se encuentra en el estado $q(t)$, $\{Q\}$ (salidas de los FF), entonces la salida del autómata en el instante t será función de $x(t)$ y $q(t)$, esto es, $z(t) = g[x(t), q(t)]$ y pasará a un nuevo estado que será función de $x(t)$ y $q(t)$ en el instante $t+1$, esto es, denominando al nuevo estado $q(t+1) = h[f(x(t), q(t))]$. A g se le denomina función de salida, a f función de excitación y h es la función de los biestables, es decir, la transición que producirá el biestable con la excitación aplicada. En resumen:

$$\begin{aligned}q(t+1) &= h[f(x(t), q(t))] \\ z(t) &= g[x(t), q(t)]\end{aligned}$$

Además, también es necesario proporcionar el estado inicial a partir del cual empezará a evolucionar el autómata. Si no se indica lo contrario, a partir de ahora consideraremos q_0 como estado inicial. En resumen una máquina de estados finitos se puede definir matemáticamente como un conjunto de elementos:

$$MEF = \{x, z, q, q_0, f, g, h\}$$

siendo:

- x : Datos de entrada
- z : Datos de salida
- q : Estados internos (variables de estado)
- q_0 : Estado inicial
- f : Función de excitación de los biestables
- h : Función de los biestables
- g : Función de salida

Tipos de circuitos secuenciales síncronos

Las dos realizaciones básicas de los autómatas finitos son las **máquinas de Mealy y de Moore**. En las figuras 3 y 4 se muestran los diagramas de bloques de estas dos realizaciones básicas, donde se han reflejado las funciones de excitación f y de salida g . Obsérvese que en ambos casos existe un bloque de memoria (realizado, por ejemplo, por flip-flops -FF-) que memoriza el estado actual (q) del autómata, y una realimentación de esos estados hacia el bloque combinacional que implementa la **función de salida** g . La diferencia entre ambas realizaciones reside en que **en el autómata de Moore la salida $z(t)$ es función únicamente del estado actual, $q(t)$, mientras que en un autómata de Mealy la salida actual depende tanto del estado actual como de las entradas actuales $x(t)$.**

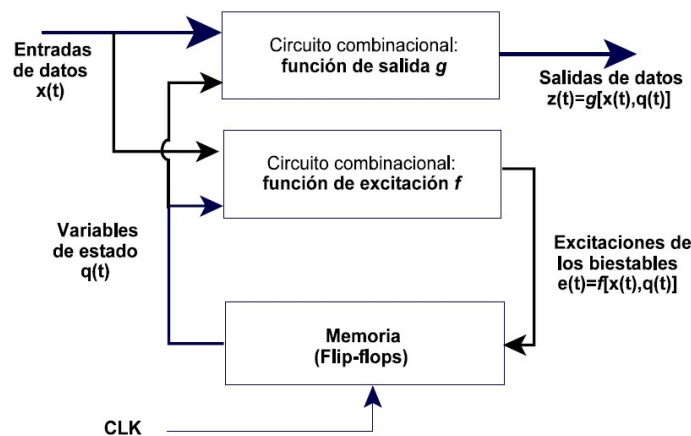


Figura 3. Autómata de Mealy: $z(t) = g[x(t), q(t)]$.

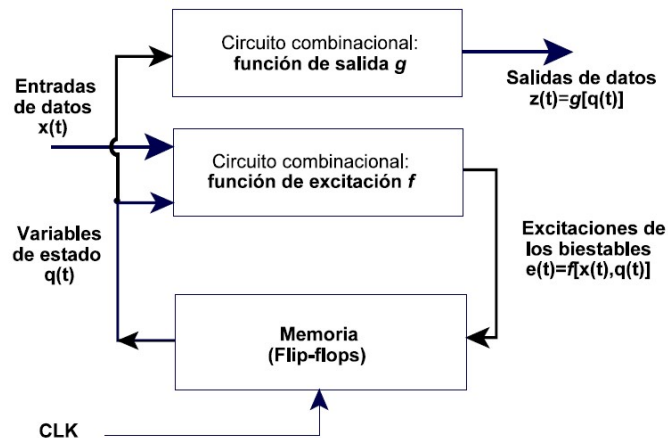


Figura 4. Autómata de Moore: $z(t) = g[q(t)]$.

Todos los circuitos secuenciales síncronos pueden implementarse tanto como autómatas de Moore como de Mealy, por lo cual, en cada caso concreto debemos decidir cual será la implementación óptima considerando que:

- En general, un circuito implementado como autómata de Moore necesita más estados que su equivalente como autómata de Mealy, por lo que el circuito resultante será de mayor complejidad cuando se diseñe como autómata de Moore.
- En los autómatas de Mealy las modificaciones en las entradas provocan cambios en la salida en el momento en el que se producen (más el tiempo de retardo propio de cualquier circuito combinacional), independientemente de la señal de reloj, mientras que en los autómatas de Moore las salidas solamente cambian cuando se produce un flanco de reloj y cambia el estado.
- Solamente utilizaremos autómatas de Mealy cuando:
 - los cambios en las entradas del circuito estén sincronizados con la señal de reloj, o
 - los cambios en otros momentos no afecten negativamente al funcionamiento del sistema global.

Para representar el comportamiento de los circuitos secuenciales se utilizan tablas de estados o tablas de transiciones (en un formato similar a las tablas de verdad) y grafos (también conocidos como diagramas de estados), donde aparecen los diferentes estados (n° finito) y las salidas y estados siguientes para cada estado y combinación de las entradas (que también son un n° finito). **Estas tablas y grafos son distintos para los autómatas de Mealy y de Moore.**

Un grafo es una representación gráfica de un autómata en la que

- Los estados se muestran dentro de círculos.
- Cada transición entre estados se representa con un arco que empieza en su estado inicial y acaban en su estado final. De cada estado deben salir tantas flechas como posibles combinaciones tengan las entradas, ya que deben estar contempladas las transiciones para todas ellas.
- Sobre cada arco se indica la entrada que condiciona dicha transición y la salida para esa entrada y ese estado actual, con una barra inclinada entre ellas, $x(t)/z(t)$.

En la figura 5 se muestra la forma general de un **diagrama de estados o grafo para el caso de un autómata de Mealy**. En él se ha representado con una línea más gruesa el círculo correspondiente al estado inicial. Considérese la figura como una parte de un grafo mayor, de forma que las transiciones que aparentemente llegan o van a “ningún” estado, realmente son transiciones al resto del grafo.

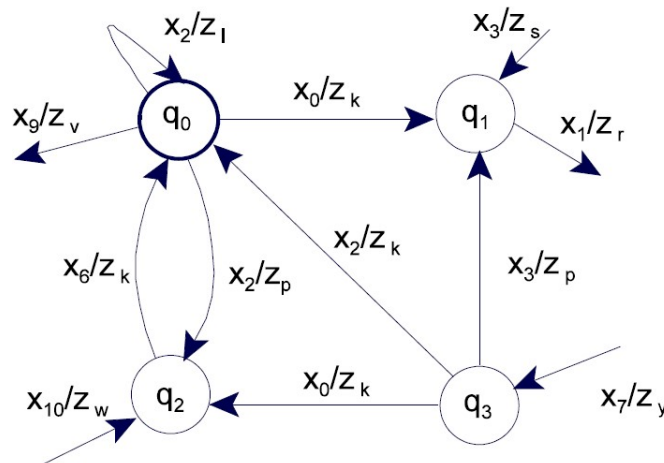


Figura 5. Ejemplo que refleja la forma general de representar los diagramas de estados o grafos de los autómatas de Mealy

La interpretación que se debe hacer de un grafo es la siguiente: en cada momento estamos en un estado determinado (uno de los círculos), y las entradas tienen un valor (entrada actual). Miraremos cual es la flecha que sale de ese estado condicionada por la entrada actual, y nos dará la salida que tiene el sistema en ese momento. Cuando llegue un flanco de reloj, la transición se hará efectiva, y pasaremos al estado siguiente, volviendo a empezar el proceso.

En los **grafos de los autómatas de Moore**, los estados y las salidas que les corresponden se muestran dentro de círculos. Las transiciones entre estados se indican con arcos que conectan dos estados y sobre estos arcos se indican las entradas que condicionan las transiciones. En la figura 6 se muestra la forma general de un diagrama de estados o grafo para el caso de un autómata de Moore. Considérese la figura como una parte de un grafo mayor, de forma que las transiciones que aparentemente llegan o van a “ningún” estado, realmente son transiciones al resto del grafo.

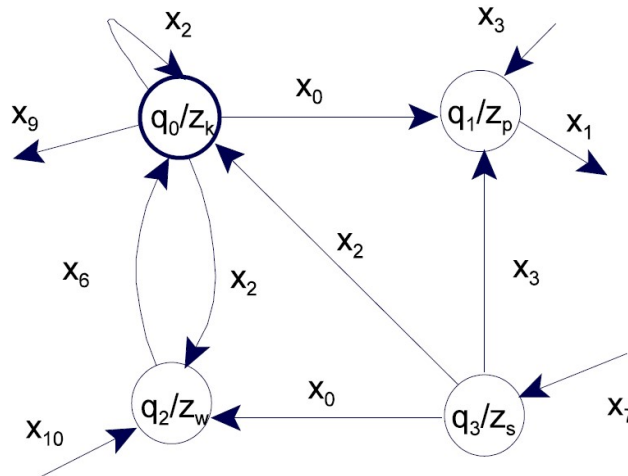


Figura 6. Forma general de representar los grafos en los autómatas de Moore.

La interpretación que se debe hacer de un grafo es la siguiente: en cada momento estamos en un estado determinado (uno de los círculos), y las entradas tienen un valor (entrada actual). La salida que acompaña al estado actual dentro del círculo será la salida que tenga el sistema en este momento. Cuando llegue un flanco de reloj, se hará efectiva la transición condicionada por la entrada actual al sistema, y pasaremos al estado siguiente correspondiente con sus valores de salida asociados, volviendo a empezar el proceso.

Análisis de circuitos secuenciales síncronos

El análisis tiene por objetivo, dado un circuito secuencial, obtener unas tablas en las que se muestren, para cada combinación digital de sus entradas y cada posible estado, el estado siguiente al que pasa y el valor de las salidas, esto es, la tabla de estados o grafo que define una máquina de Mealy o Moore. Conocer el proceso de análisis facilita el aprendizaje de la síntesis de circuitos secuenciales síncronos, ya que el diseño consiste, a grandes rasgos, en aplicar los mismos pasos en orden inverso.



Ejemplo: Supóngase el circuito mostrado en la Figura 7. Como se puede observar, es un sistema con una sola entrada (X_0) y salida (Z_0), y dos flip-flops de tipo JK activos en el flanco de subida. Además, se puede ver que la salida es función de la entrada (X_0) y de las salidas de los flip-flops (estado), por lo tanto, la realización de este circuito responde a una máquina de Mealy

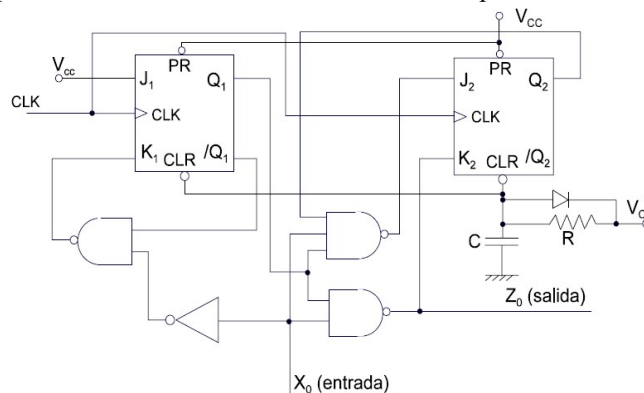


Figura 7. Autómata de Mealy correspondiente al ejemplo

Para obtener las ecuaciones lógicas de las señales que se están aplicando a las entradas de los flip-flops (excitaciones de los biestables). Estas ecuaciones son las funciones lógicas de las entradas de los flip-flops (RS -caso de flip-flops RS- JK -caso de flip-flops JK- D -caso de flipflops D- o T -caso de flip-flops T-) en función de las salidas de los propios flip-flops (estado actual), y las entradas del sistema.

En nuestro ejemplo, las ecuaciones lógicas de las entradas de los flip-flops son:

$$J_1^t = 1$$

$$K_1^t = \overline{Q_1^t} \cdot \overline{X_0^t}$$

$$J_2^t = \overline{Q_1^t} \cdot \overline{X_0^t} \cdot Q_2^t$$

$$K_2^t = \overline{Q_1^t} \cdot X_0^t$$

En este el ejemplo, al tratarse de flip-flops JK, las ecuaciones lógicas de los próximos estados son:

$$Q_1(t+1) = J_1(t)\overline{Q_1(t)} + \overline{K_1(t)}Q_1(t) = 1 \cdot \overline{Q_1(t)} + \overline{\overline{Q_1(t)} \cdot \overline{X_0(t)}} \cdot Q_1(t)$$

$$Q_2(t+1) = J_2(t)\overline{Q_2(t)} + \overline{K_2(t)}Q_2(t) = \overline{Q_1(t)} \cdot \overline{X_0(t)} \cdot \overline{Q_2(t)} \cdot \overline{Q_2(t)} + \overline{\overline{Q_1(t)} \cdot X_0(t)} \cdot Q_2(t)$$

Simplificando estas expresiones se obtiene:

$$Q_1(t+1) = \overline{Q_1(t)}$$

$$Q_2(t+1) = \overline{Q_2(t)} + Q_1(t) \cdot X_0(t) \cdot Q_2(t)$$

El tercer paso consiste en obtener las funciones lógicas que relacionan las variables de salida del sistema con el estado actual (salidas de los flip-flops) y las entradas del sistema.

Estado actual Q ₂ (t) Q ₁ (t)	Entrada actual X ₀ (t)		Entrada actual X ₀ (t)	
	0	1	0	1
0 0	1 1	1 1	1	1
0 1	1 0	1 0	1	0
1 0	0 1	0 1	1	1
1 1	0 0	1 0	1	0
Q ₂ (t+1) Q ₁ (t+1) Próximos estados			Z ₀ (t) Salida actual	

Asignar un nombre (q0, q1, q2, etc.) a cada combinación binaria de las salidas de los flipflops y reescribir la tabla obtenida como tabla de estados

Estado actual q(t)	Entrada actual x(t)	
	0	1
q ₀	q ₃ /1	q ₃ /1
q ₁	q ₂ /1	q ₂ /0
q ₂	q ₁ /1	q ₁ /1
q ₃	q ₀ /1	q ₂ /0
q(t+1)/z(t) Próximo estado/Salida actual		

A partir de la tabla se obtiene el grafo de la figura 8

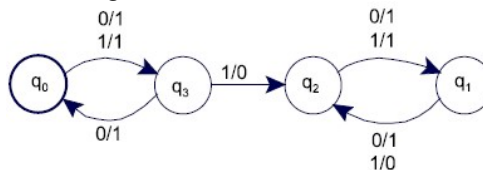


Figura 8. Grafo correspondiente al ejemplo