

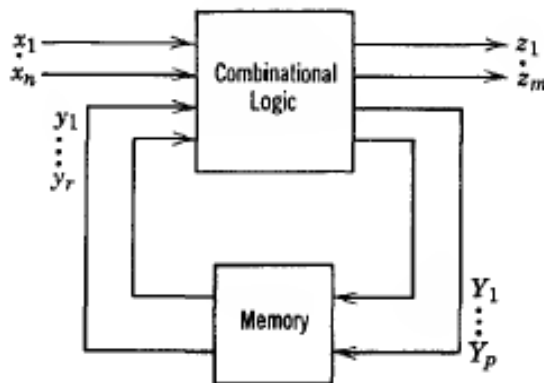


UNIDAD 7: SECUENCIALES SINCRÓNICOS.

(86:44) Técnica Digital Avanzada- Unidad 7.
Profesor: Ing. Miguel Antonio Martínez.

Circuitos Secuenciales.

El modelo general de un circuito secuencial responde a la siguiente figura:



Este modelo se compone de dos partes, una sección lógica combinacional y una sección de memoria. La memoria almacena información sobre sucesos pasados, que se requieren para el funcionamiento adecuado del circuito. Esta información se representa en la forma de **r salidas binarias** y_1, y_2, \dots, y_r , conocidas como **variables de estado**. Cada una de las 2^r combinaciones de las variables de estado define un **estado de la memoria** y, en general, cada estado corresponde a una combinación particular de eventos pasados.

La sección de lógica combinacional recibe como entradas a las variables de estado y las **n entradas del circuito** x_1, x_2, \dots, x_n . Genera **m salidas** z_1, z_2, \dots, z_m y **p variables de excitación** Y_1, Y_2, \dots, Y_p , que especifican el siguiente estado que debe adoptar la memoria.

La red lógica combinacional se puede definir por medio de ecuaciones de Boole de las formas estándar, es decir:

$$z_i = f_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad (i = 1, 2, \dots, m)$$

$$Y_j = f_j(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad (j = 1, 2, \dots, p)$$

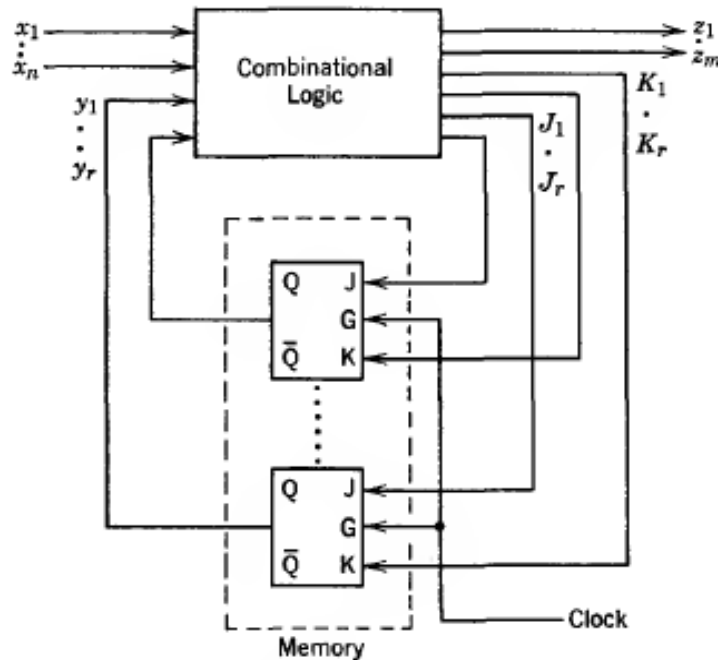
Hay ecuaciones independientes del tiempo, esto es, que son válidas en cualquier momento en que todas las entradas y las salidas sean estables, como sucede para cualquier circuito combinacional.

No se han especificado todavía la relación entre las entradas y las salidas de la porción que denominamos **memoria**. Esta relación dependerá si este bloque está integrado por elementos de memoria en sí como ser Flip Flops, o si se trata simplemente de un bloque realimentador con retardos de tiempo.

En la siguiente figura se muestra un circuito secuencial con **modalidad de reloj**. La memoria se compone de **r flip flops**, que pueden ser de cualquier tipo (en este ejemplo JK). Las excitaciones consisten en **r señales J** y **r señales K**,

Todas las entradas y salidas de la sección lógica combinacional son señales de **nivel**, en otras palabras, señales que pueden asumir valores 1 o 0 durante periodos arbitrarios de tiempo, dado que las salidas y las excitaciones pueden cambiar en cualquier momento dado, en respuesta a las variaciones de las entradas del circuito.

Sin embargo, las variables de estado cambiarán solo cuando los flip flops sean disparados por un pulso de reloj. Este pulso de reloj no contiene ninguna información en el sentido de que determine cual cambio de estado deberá efectuarse, sino que simplemente marca el tiempo o **sincroniza** el cambio de estado. Por eso a estos circuitos secuenciales de modalidad de reloj se los suele llamar **Circuitos Secuenciales Sincrónicos**.



Los valores de las señales J y K en el instante del pulso de reloj, determinan la transición que se llevará a cabo. Los tiempos de llegada de los pulsos de reloj están indicados por t_v ($v = 1, 2, 3, \dots$). Todas las variables del circuito deben estar estables en el tiempo t_v . El pulso de reloj que llega en t_v dará origen a una transición de las variables de estado de y_i^v a y_i^{v+1} , colocando así al circuito en un nuevo estado. Las variables de excitación y salida se ajustarán, a su vez, a este cambio de estado, al igual que a cualquier cambio de entrada, llegando a nuevos niveles estables, antes del siguiente pulso de reloj, en t_{v+1} . Este comportamiento del circuito se puede resumir en las siguientes ecuaciones:

$$z_i^v = f_{z_i}(x_1^v \cdots x_n^v, y_1^v \cdots y_r^v) \quad (i = 1, 2, \dots, p)$$

$$J_j^v = f_{J_j}(x_1^v \cdots x_n^v, y_1^v \cdots y_r^v) \quad (j = 1, 2, \dots, r)$$

$$K_j^v = f_{K_j}(x_1^v \cdots x_n^v, y_1^v \cdots y_r^v) \quad (j = 1, 2, \dots, r)$$

$$y_j^{v+1} = \bar{K}_j^v y_j^v + J_j^v \bar{y}_j^v \quad (j = 1, 2, \dots, r)$$

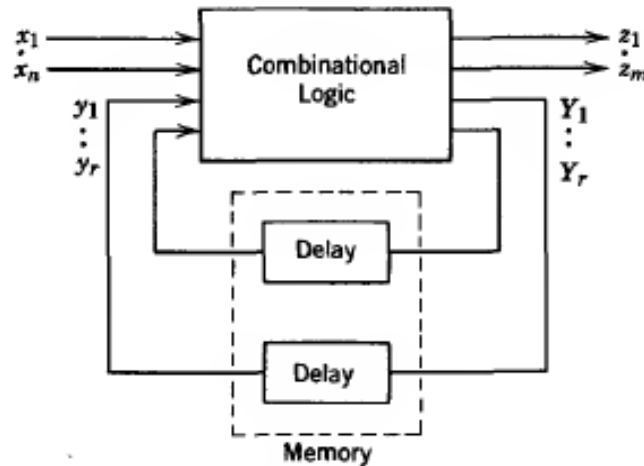
Dentro de los circuitos secuenciales síncronos existen otros llamados **de modalidad de pulso**. En estos no hay una línea de reloj independiente. Los pulsos que disparan los cambios de estado aparecen a la entrada del circuito.

Los circuitos con modalidad de reloj y pulso requieren de señales de clock para controlar las transiciones de estado. Ahora consideraremos circuitos donde todas las entradas son niveles, es decir, señales que pueden permanecer en cualquier valor (cero o uno) durante periodos arbitrarios y pueden cambiar en tiempos arbitrarios también.

Estos se denominan circuitos secuenciales con **modalidad de nivel** y más general, **Circuitos Secuenciales Asíncronos**.

Si, además, a las entradas se impone la restricción de que solo una puede cambiar a la vez, y que no pueden ocurrir cambios subsecuentes hasta que no se hayan estabilizado todas las salidas y las variables de estado, se dice que el circuito de modalidad de nivel opera con **modalidad fundamental o en modo fundamental**.

El modelo básico de este tipo de circuito se muestra en la siguiente figura:



Como se ve la memoria está formada por bloques de retardos de tiempo. No se trata de un retardo físico insertado en la realimentación, sino que este bloque representa una concentración de los retardos inherentes distribuidos en la lógica combinacional que desarrolla las variables de excitación. Con este tipo de memoria, las excitaciones constituyen por sí mismas los siguientes valores de las variables de estado. Las excitaciones se identificarán con las letras mayúsculas **Y**, en tanto que los valores teóricamente retardados de las excitaciones, que sirven como entradas a la red lógica combinacional, se denominan **secundarios** y se identifican con las letras minúsculas **y**. Las ecuaciones para las variables **z** y **Y** son las mismas que para el modelo general y son:

$$z_i = f_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad (i = 1, 2, \dots, m)$$

$$Y_j = f_j(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad (j = 1, 2, \dots, p)$$

La ecuación que describe el cambio de estado es la siguiente:

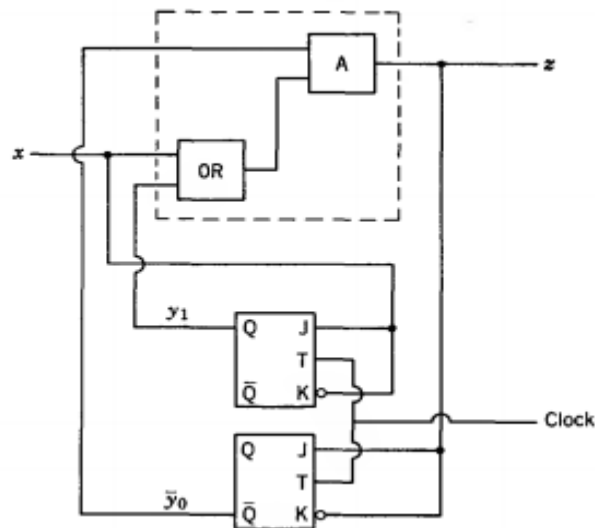
$$y_j(t + \Delta t) = Y_j(t) \quad (j = 1, 2, \dots, r)$$

Análisis de un circuito secuencial sincrónico.

Trabajando con secuenciales sincrónicos y con dispositivos electrónicos en general se pueden hacer dos cosas. El **análisis** que es el estudio del funcionamiento de un dispositivo ya armado. En cambio, la **Síntesis** es el diseño propiamente dicho a partir de especificaciones técnicas.

Para entrar en el tema de circuitos secuenciales sincrónicos empezaremos por el análisis de un circuito.

Veamos el circuito de la siguiente figura:



Este circuito tiene el bloque combinacional formado por una compuerta OR y una compuerta AND y la realimentación tiene los elementos de memoria que son dos flip flop del tipo JK que llamamos FF0 y FF1 representado por las salidas y_1 y y_0 .

Las ecuaciones de entrada a los elementos de memoria se pueden ver fácilmente siguiendo el bloque combinacional, dichas expresiones son las siguientes:

$$J_{y_1} = x \quad K_{y_1} = \bar{x}$$

$$J_{y_0} = z = (x + y_1)\bar{y}_0 \quad K_{y_0} = \overline{(x + y_1)\bar{y}_0}$$

Teniendo en cuenta que los dos FF son del tipo JK y sabiendo que la ecuación característica de dicho FF es la siguiente:

$$Q_{\text{siguiente}} = J\bar{Q} + \bar{K}Q$$

Así podemos determinar las ecuaciones del siguiente estado, de la siguiente forma:

$$y_1^{v+1} = x^v \cdot \bar{y}_1^v + \bar{x}^v \cdot y_1^v = x^v$$

$$y_0^{v+1} = [(x^v + y_1^v)\bar{y}_0^v] \cdot \bar{y}_0^v + [(x^v + y_1^v)\bar{y}_0^v]y_0^v$$

$$= (x^v + y_1^v)\bar{y}_0^v$$

En este ejemplo las ecuaciones quedan bastante simplificadas por el hecho de que J y K de cada FF es uno negado del otro.

De acuerdo con las últimas ecuaciones se podría tabular el próximo estado y el valor de las salidas $(y_1^{v+1}, y_0^{v+1}, z^v)$ para todas las combinaciones de valores posibles de x^v, y_0^v y y_1^v . Esta tabla se denomina **Tabla de transición** y se muestra en la siguiente figura:

y_1^v	y_0^v	y_1^{v+1}, y_0^{v+1}		z^v			
		x^v	0	1	0	1	
0	0	0	0	1	1	0	1
0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	0
1	0	0	1	1	1	1	1

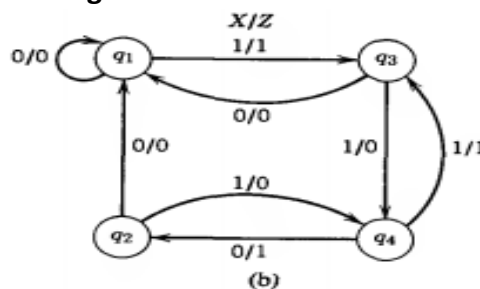
Las tablas de transición de la forma indicada tienden a hacerse muy voluminosas al aumentar el número de variables. Para simplificar se adopta una notación abreviada como sigue. En primer lugar, los estados se representan mediante la letra **q**, a la cual se le asignan arbitrariamente subíndices decimales. Por ejemplo, $y_1 y_0 = 0 0$ se representa mediante **q₁**, $y_1 y_0 = 0 1$ como **q₂** y así sucesivamente.

Ahora podemos relacionar la salida y las porciones del próximo estado de la tabla de transición. Esto da lugar a lo que se llama **Tabla de Estado**. La vemos en la siguiente figura:

$y_1^v y_0^v$	Present state	x^v	
	q^v	0	1
0 0	q_1	$q_1, 0$	$q_3, 1$
0 1	q_2	$q_1, 0$	$q_4, 0$
1 1	q_3	$q_1, 0$	$q_4, 0$
1 0	q_4	$q_2, 1$	$q_3, 1$

q^{v+1}, z^v

Esto se puede interpretar de la siguiente manera, si el circuito está en el estado **q₄** y recibe una entrada **cero**, después del pulso de reloj la salida será **uno** y pasará al estado **q₂**. La misma información contenida en la tabla de estado se puede representar gráficamente por medio el **Diagrama de Estado** como se ve en la siguiente figura:

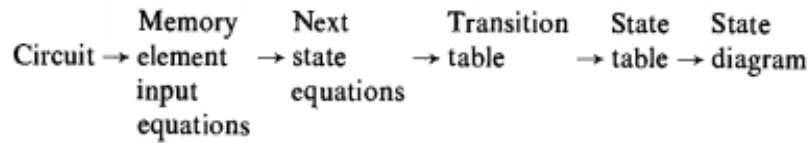


Hay un círculo para cada estado y una flecha que sale de cada estado para cada una de las entradas. Las flechas terminan en los estados siguientes apropiados. Las entradas se indican a la izquierda de cada flecha y el valor de la salida a la salida, o sea X/Z.

Hasta ahora hemos visto el proceso de análisis, ahora haremos al revés. Empezaremos por las especificaciones técnicas y terminaremos en el circuito terminado.

Procedimiento de Diseño.

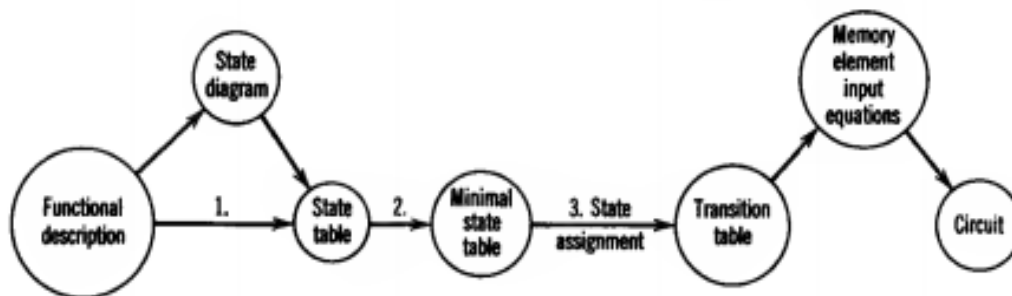
El proceso de análisis de la sección anterior siguió una secuencia de etapas que transformaron sucesivamente la descripción del circuito como se ve en el siguiente diagrama:



El proceso de síntesis es semejante a un análisis en sentido inverso. La principal distinción es que la síntesis principia antes de la existencia de un **diagrama de estado**. Este diagrama se debe obtener de alguna otra descripción del problema, generalmente de las especificaciones del mismo. O sea, de lo que queremos que el circuito realice. A partir del diagrama obtenemos la **tabla de estado**. Luego veremos procedimientos para que esta tabla sea lo más reducida posible, o sea que sea equivalente con el número mínimo de estados.

La complejidad de las ecuaciones de entrada de los elementos de memoria variará en función de la asignación de combinaciones de valores de salida de los FF a los estados de la tabla de estado. El intento de optimizar esta correspondencia se conoce con el nombre de problema de **asignación de estados**.

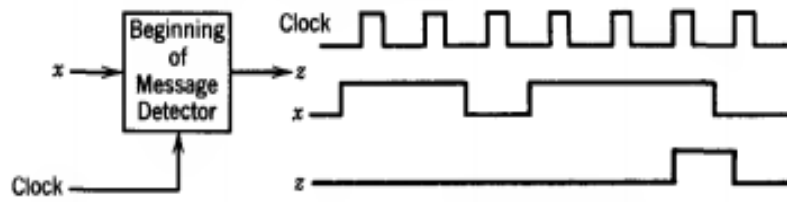
Los pasos para la síntesis de un circuito secuencial se muestran en la siguiente figura:



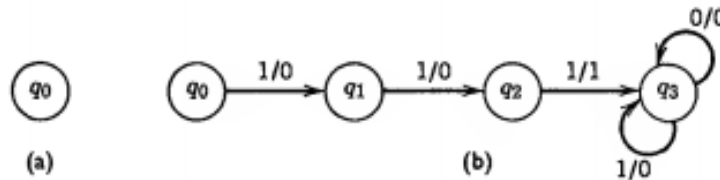
Como se ve en la figura un método para obtener la tabla de estado incluye el uso de un diagrama de estado. Este método es muy conveniente para determinados circuitos que poseen un estado de fácil identificación, conocido como **estado de restauración**. Cuando se tiene un estado de esta índole, se tiene un mecanismo por el cual el circuito puede volver al estado de restauración a partir de cualquier estado. Veremos un ejemplo de diseño.

Ejemplo:

Se trata de una línea de comunicación serie donde el principio del mensaje se identifica con tres unos consecutivos en una línea de entrada que llamaremos **x**. Los datos de esta línea se han sincronizado con una fuente de pulsos de reloj. Se debe diseñar un circuito secuencial con modalidad de reloj que tenga una salida en valor **1**, solo en el tiempo de reloj que coincida con el tercero de una secuencia de **tres unos** en la línea **x**. El circuito servirá para advertirle al sistema receptor sobre la iniciación de un mensaje. Se proporcionará un mecanismo de restauración o borrado independiente, para colocarlo en un estado **q₀**. La especificación del circuito es la siguiente:

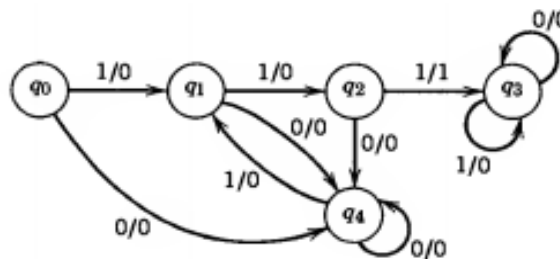


Se designa al estado de restauración como q_0 . Básicamente el circuito debe llevar la cuenta del número de unos consecutivos que haya recibido. Esto se logra pasando a un estado diferente con cada entrada en 1. Un diagrama preliminar sería el siguiente:



Como muestra la figura el circuito pasa al estado q_1 con el primer 1 de entrada, y al estado q_2 con el segundo. Las salidas asociadas con estas dos entradas son 0, tal como señala. La tercera entrada consecutiva en 1 genera un 1 y hace que el circuito pase a q_3 . Una vez que se encuentra en q_3 , el circuito permanecerá en ese estado emitiendo salidas 0, hasta que externamente se lo restaure a q_0 . Es posible que aparezca una entrada en cero durante cualquier periodo de reloj, interrumpiendo quizá la secuencia de unos. Para esta eventualidad, se proporciona un el estado q_4 . Mientras el circuito se encuentre en q_4 una entrada de 1 puede ser la primera de una secuencia de 3 unos. De donde un 1 hará que el circuito cambie de estado q_4 a q_1 . El lector intuitivo pudo haber notado que el circuito podría volver a q_0 con entradas en cero, lo cual habría eliminado la necesidad del estado q_4 . Cuando hay dudas no hay problema en definir un nuevo estado. Si los estados nuevos corresponden a la especificación del sistema, luego se reducirán cuando simplifiquemos la tabla de estado.

En nuestro ejemplo se han considerado todas las secuencias de entrada posibles, por lo tanto, el diagrama de estado queda de la siguiente manera:



Es fácil pasar del diagrama de estado a la tabla de estado, que queda de la siguiente manera:

	x^v	
	0	1
q^v		
q_0	$q_4, 0$	$q_1, 0$
q_1	$q_4, 0$	$q_2, 0$
q_2	$q_4, 0$	$q_3, 1$
q_3	$q_3, 0$	$q_3, 0$
q_4	$q_4, 0$	$q_1, 0$
	q^{v+1}, z	

(d) State Table

Volveremos a este ejemplo más adelante, ahora nos abocaremos a como simplificar una tabla de estados, pero antes debemos dar algunas definiciones teóricas.

Relaciones de equivalencia.

Los procedimientos que vimos proporcionan medios para obtener una tabla de estados para una especificación dada, pero la tabla resultante no es única y puede no ser la óptima. Puesto que el número de FF depende de la cantidad de estados, se puede decir que la tabla de estados mínima es aquella con el número mínimo de estados. En algunos casos se verá que diferentes estados en una tabla efectúan la misma función, o quizás no sea posible distinguir entre dos o más estados por medio de mediciones de la salida. En estos casos se dicen que los estados son **equivalentes** y se pueden sustituir por uno solo. Primero explicaremos el concepto de **relación de equivalencia**.

Cuando un par ordenado de elementos (x, y) posee una propiedad que los relaciona, se puede decir que x está R -relacionado con y . Este concepto se simboliza con " $x R y$ ". La relación R se define como el conjunto de todos los pares ordenados que posean una propiedad particular, sea cual fuese.

Si $x R x$ para cada x del conjunto se dice que la relación es **reflexiva**.

Si $y R x$ siempre que $x R y$ entonces la relación es **simétrica**.

Si $x R y$ y $y R z$ implican que $x R z$ entonces R es una relación **transitiva**.

Una relación que satisfaga los tres criterios anteriores, o sea que sea reflexiva, simétrica y transitiva se denomina **Relación de Equivalencia**.

Una relación de equivalencia en un conjunto lo dividirá siempre en subconjuntos separados conocidos como **Clases de equivalencia**.

Estados equivalentes y circuitos.

Para definir el concepto de estados equivalentes y circuitos secuenciales equivalentes, es necesario introducir una nueva notación. Las tablas de estado comprenden representaciones tabulares de dos funciones, la de salida y la del siguiente estado. Si las salidas se especifican para combinación de entradas estados presentes, los circuitos se clasifican como **completamente especificados**. La función del siguiente estado se denotará con el símbolo δ y la función de salida será λ . Por ejemplo, para la tabla de la siguiente figura (hemos tomado valores de x arbitrarios, recordar que en nuestros diseños x toma valores binarios).

	x^y				
Present State		0	1	3	2
q_1		$q_3, 0$	$q_1, 0$	$q_2, 0$	$q_2, 0$
q_2		$q_3, 0$	$q_3, 0$	$q_4, 0$	$q_4, 0$
q_3		$q_3, 0$	$q_1, 1$	$q_1, 3$	$q_1, 2$
q_4		$q_4, 0$	$q_4, 0$	$q_2, 0$	$q_2, 0$
		q^{y+1}, z			

Se ve que:

$$\lambda(q_2, 3) = 0$$

$$\delta(q_2, 3) = q_4$$

Si un circuito se encuentra en un estado inicial, responderá a una secuencia de entradas, con una secuencia específica de salidas. Considere el caso en cual el circuito correspondiente a la tabla anterior recibe en sus entradas la secuencia 0 2 3 0 0 1. Para determinar la secuencia de salida en respuesta a esta secuencia de entrada, se requiere determinar también la secuencia del siguiente estado, entonces tenemos lo siguiente:

$$\lambda(q_1, 0) = 0 \quad \text{and} \quad \delta(q_1, 0) = q_3$$

$$\lambda(q_3, 2) = 2 \quad \text{and} \quad \delta(q_3, 2) = q_1$$

$$\lambda(q_1, 3) = 0 \quad \text{and} \quad \delta(q_1, 3) = q_2$$

$$\lambda(q_2, 0) = 0 \quad \text{and} \quad \delta(q_2, 0) = q_3$$

$$\lambda(q_3, 0) = 0 \quad \text{and} \quad \delta(q_3, 0) = q_3$$

$$\lambda(q_3, 1) = 1 \quad \text{and} \quad \delta(q_3, 1) = q_1$$

La secuencia de salida y el estado final se pueden resumir ahora como una función del estado inicial y la secuencia de entrada, como sigue:

$$\lambda(q_1, 023001) = 020001$$

$$\delta(q_1, 023001) = q_1$$

Tomando este ejemplo vemos que, si conocemos la secuencia de salida para un estado inicial y tenemos una secuencia de entrada dada, no **nos interesa la secuencia de estados internos**. Para nosotros lo único importante **es la respuesta de salida de un circuito a una secuencia de entradas**. En cierto sentido, **la función de un circuito secuencial se puede tomar como la traducción de una secuencia de señales en una segunda secuencia**.

Ahora definiremos la equivalencia de dos estados dentro de un circuito secuencial:

Definición. Sean S y T dos circuitos completamente especificados sujetos a las mismas secuencias de entrada posibles. Sea (X_1, X_2, \dots, X_n) un conjunto de valores posibles del conjunto de entrada S, de una longitud arbitraria. Los estados $p \in T$ y $q \in S$ son indistinguibles (equivalentes), lo cual se expresa como $p \equiv q$, si y solo si $\lambda_T(p, X_1, X_2, \dots, X_n) = \lambda_S(q, X_1, X_2, \dots, X_n)$ para cada secuencia posible de entrada.

Esta definición también es válida para dos estados diferentes de un solo circuito. Expresado de una forma menos formal, p y q son equivalentes si no existe ninguna forma de distinguirlos, basándose en cualquiera de las secuencias de salida que se inician en ese estado.

Veamos ahora las siguientes tablas:

S	X = 0	X = 1
q_1	$q_3, 0$	$q_2, 1$
q_2	$q_1, 1$	$q_2, 0$
q_3	$q_1, 0$	$q_2, 1$

T	X = 0	X = 1
p_1	$p_1, 0$	$p_2, 1$
p_2	$p_1, 1$	$p_2, 0$

Veremos primero la tabla para el circuito S. Suponemos que se encuentra inicialmente en el estado q_1 . Una entrada en 0 hará que el circuito pase a q_3 con una salida en 0. En tanto continúen las entradas en 0, el circuito repetirá un ciclo entre q_1 y q_3 , produciendo salidas en 0. Si se registra una entrada 1, el circuito pasará a q_2 y la salida será 1. En este caso, una entrada 0 enviará a circuito nuevamente a q_1 .

Se observará que, si el circuito se encontrara inicialmente en el estado q_3 , se producirán exactamente las mismas secuencias de salida que se delinearon antes. De hecho, estas dos secuencias de entrada son suficientes para demostrar que los estados q_1 y q_3 son equivalentes. Ahora veremos que estas secuencias de entrada producen las mismas secuencias de salida en el circuito T, por lo tanto, S y T son circuitos equivalentes.

El problema básico consiste en encontrar una tabla de estado óptima, o sea el problema sería dada una tabla de estado, encuentre una equivalente con el menor número de estados posibles. Viendo el ejemplo anterior el circuito S tiene tres estados, por lo tanto, necesitará dos FF, mientras que el circuito T cuenta con dos estados, por lo tanto, requerirá solo un FF.

Determinación de clases de estados indistinguibles.

El método básico para encontrar la tabla de estados óptima consiste en dividir dicha tabla en el número más pequeño posible de clases de equivalencia de estados indistinguibles.

Se dará la siguiente definición, se hará que los estados de un circuito secuencial se dividan en clases separadas, $p \equiv q$ denota que los estados p y q están dentro de la misma clase de partición. Esta partición se compone de clases de equivalencias de estados indistinguibles (dos estados indistinguibles deben estar en la misma clase), si y

solo si se satisfacen las dos condiciones siguientes, para cada par de estados **p** y **q** en la misma clase y cada entrada individual **X**:

$$(1) \lambda(p, X) = \lambda(q, X)$$

$$(2) \delta(p, X) \triangleq \delta(q, X)$$

O sea, dos **estados son equivalentes** si:

- 1) Producen las mismas salidas para todas las combinaciones de entrada.
- 2) Tienen el mismo estado siguiente o los estados siguientes de ambos son equivalentes.

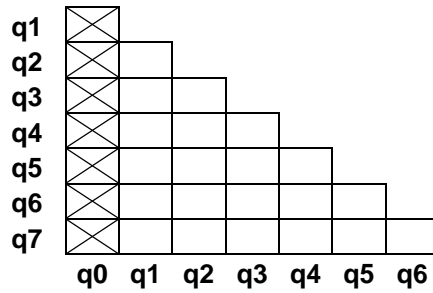
Hay varios métodos para encontrar las clases de equivalencia, como ser el de Huffman/Mealy. En este apunte explicaremos el método de las **tablas de implicación**, basado en un trabajo de Paul y Unger.

Para explicar el método empezaremos con una tabla de estados como la de la figura:

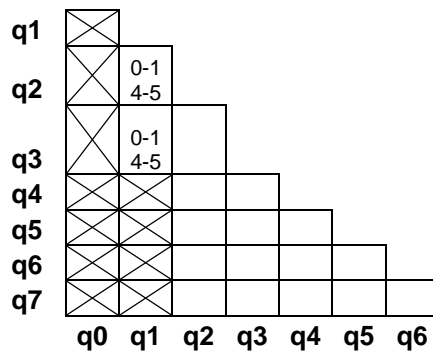
		x^v	
		0	1
q^v	q_0	$q_0, 1$	$q_4, 0$
	q_1	$q_0, 0$	$q_4, 0$
	q_2	$q_1, 0$	$q_5, 0$
	q_3	$q_1, 0$	$q_5, 0$
	q_4	$q_2, 0$	$q_6, 1$
	q_5	$q_2, 0$	$q_6, 1$
	q_6	$q_3, 0$	$q_7, 1$
	q_7	$q_3, 0$	$q_7, 1$
		q^{v+1}, x^v	

q1	X						
q2	X						
q3							
q4							
q5							
q6							
q7							
	q0	q1	q2	q3	q4	q5	q6

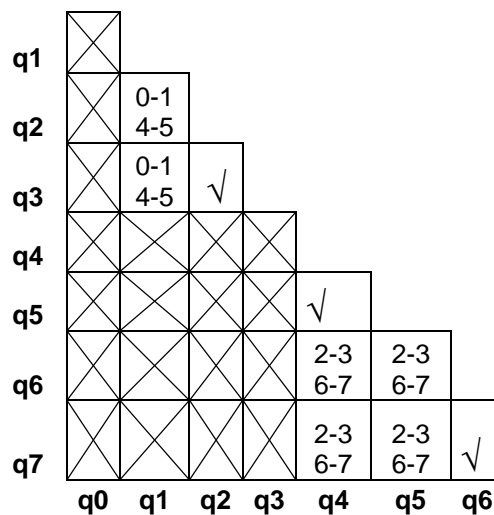
Al tener un cuadrado por cada par de estados podemos comparar todos contra todos. Vamos poniendo **X** en los estados que no son equivalentes, por ejemplo, por tener salidas distintas y por lo tanto no pueden ser equivalentes. En nuestro caso por ejemplo vemos que q_0 tiene salidas distintas a todos los demás estados, por lo tanto, no puede ser equivalente con ninguno, o sea:



Después seguimos comparando, por ejemplo, para que el estado q_1 y q_2 sean equivalentes, lo deberían ser los estados q_0 y q_1 como así también q_4 y q_5 . Anotamos estos y seguimos:



Llegamos a comparar q_2 y q_3 , ahí vemos que estos estados son equivalentes, ya que tienen los mismos estados siguientes y las mismas salidas para los mismos valores de entrada, en ese caso ponemos una tilde. Finalmente, la tabla queda de la siguiente manera:



La siguiente etapa es hacer una segunda **pasada** a través de la tabla para ver si algunos conjuntos implicados quedan excluidos como pares posibles de equivalencia. En general

si existe una **X** en un cuadrado con coordenadas **j-k** entonces cualquier cuadrado que contenga a **j-k** como conjunto implicado puede excluirse por **X**. Este es el caso de los estados q_1 con q_2 y q_1 con q_3 , estos implican a los estados q_0 y q_1 que no son equivalentes, por lo tanto, se marca con X. Siguiendo el análisis se ve que tanto q_2 con q_3 y q_6 con q_7 , son equivalentes. Por lo tanto, la tabla final queda:

q1	X						
q2	X	0-1 4-5					
q3	X	0-1 4-5	√				
q4	X	X	X	X			
q5	X	X	X	X	√		
q6	X	X	X	X	√	√	
q7	X	X	X	X	√	√	√
	q0	q1	q2	q3	q4	q5	q6

Ahora las clases de equivalencia se determinan como sigue, en primer lugar, se anotan los estados correspondientes a las columnas de la tabla de implicación en orden inverso. A continuación, se comprueba cada columna de la tabla final de implicación para ver si no se ha cruzado ningún cuadrado.

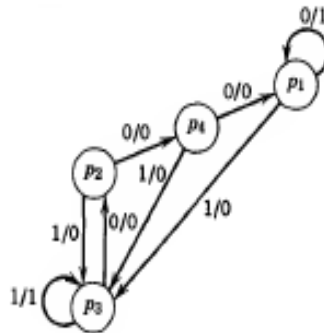
6	(6,7)
5	(6,7) (5,6) (5,7)
4	(5,6,7) (4,7) (4,6) (4,5)
3	(4,5,6,7)
2	(4,5,6,7) (2,3)
1	(4,5,6,7) (2,3)
0	(4,5,6,7) (2,3) (0) (1)

Vamos anotando los cuadrados equivalentes, en la columna 6 se encuentra que 6 y 7 son equivalentes. En la columna 5 encontramos que 5 y 6 como así también 5 y 7. Como 6 y 7 eran equivalentes se anota (5,6,7). Y Así sucesivamente, al final incorporamos los que quedan solos como (0) y (1). Nos quedaron 4 clases de equivalencia. Si llamamos **a** a (0), **b** a (2,3), **c** a (4,5,6,7) y **d** a (1), nos queda la siguiente tabla de estado:

p^v	x^v		Class	Equivalent States of S
	0	1		
p_1	$p_1, 1$	$p_3, 0$	a	q_0
p_2	$p_4, 0$	$p_3, 0$	b	q_2, q_3
p_3	$p_2, 0$	$p_3, 1$	c	q_4, q_5, q_6, q_7
p_4	$p_1, 0$	$p_3, 0$	d	q_1

p^{v+1}, z^v

Y podemos dibujar el diagrama de estado reducido de la siguiente forma:



Ahora nos queda el último paso que es el de asignación de estados.

Asignación de Estados.

Una vez que se ha obtenido la tabla mínima de estado, queda todavía por diseñar un circuito que realice dicha tabla. Debe reconocerse que el problema de la economía no se ha solucionado con la minimización del número de estados. Es cierto que la cantidad de elementos de memoria requeridos por una tabla de estados de esta índole es mínima, pero también requiere una lógica combinacional para desarrollar las ecuaciones de salida y entrada de cada elemento de memoria. La forma de elegir la mejor realización del circuito es, desafortunadamente, un problema difícil por sí mismo. Se debe recordar que cada uno de los estados correspondientes a una de las 2^r combinaciones de las r variables de estado. El primer problema en el diseño de un circuito para realizar una tabla de estado dado consiste en decidir cuál de las 2^r combinaciones se asignará a cada estado. Si el número de estados m satisface:

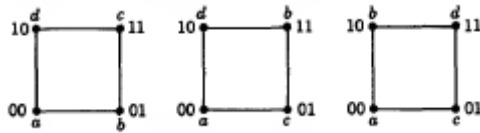
$$2^{r-1} < m \leq 2^r$$

Entonces, se requerirán r variables de estado. Se tendrán:

$$\frac{2^r!}{(2^r - m)!}$$

formas para asignar las 2^r combinaciones de variables de estados a los m estados.

El problema de asignar combinaciones de variables de estado a 4 estados, por ejemplo, **a, b, c y d** se puede representar como la asignación de 4 estados a los cuatro vértices de un cubo-2. En la siguiente figura se muestra las tres asignaciones distintas en esta forma. Cualesquiera otras asignaciones posibles se agregarían a las rotaciones o inversiones de estas tres asignaciones.



Solo a modo de ejemplo mostramos una tabla donde se ve la cantidad de asignaciones distintas posibles compatibles con la cantidad de estados de la tabla de estado mínima.

No. of states m	No. of state variables r	No. of distinct assignments
2	1	1
3	2	3
4	2	3
5	3	140
6	3	420
7	3	840
8	3	840
9	4	10,810,800

Para circuitos que solo tienen dos estados, no existe elección y por lo tanto no tenemos problema de asignación. Para tres o cuatro estados, lo más sencillo es intentar las tres asignaciones posibles y ver cual produce el circuito más económico. Para más de cuatro estados, la enumeración completa es obviamente imposible, de manera que es necesario desarrollar métodos para escoger una buena asignación. Empezaremos tomando un ejemplo de cuatro estados e intentando las tres asignaciones posibles, para ver si se pueden obtener mayores conocimientos respecto a las características de una buena asignación. La tabla de estado presentada es la del ejemplo inicial de detector de comienzo de mensaje, el que tenía que detectar tres unos consecutivos. De este ejemplo vemos las tres asignaciones posibles en la siguiente figura:

q^v	x^v			
	0	1	0	1
a	a	c	0	0
b	a	d	0	1
c	a	b	0	0
d	d	d	0	0

(a)

q^v	x^v		
	$y_1 y_0^v$	0	1
a	0 0	00	11
b	0 1	00	10
c	1 1	00	01
d	1 0	10	10

(b)

q^v	x^v		
	$(y_1 y_0)^v$	0	1
a	00	00	01
c	01	00	11
b	11	00	10
d	10	10	10

(c)

q^v	x^v		
	$(y_1 y_0)^v$	0	1
a	00	00	01
c	01	00	10
d	11	11	11
b	10	00	11

(d)

Como tenemos una sola salida la obviamos para mayor claridad. Tenemos que elegir qué tipo de FF elegimos. Presentamos las tablas de transición de los FF comerciales:

JK	Q^v		$Q^v \rightarrow Q^{v+1}$	J	K
	0	1			
00	0	1	0 → 0	0	×
01	0	0	0 → 1	1	×
10	1	1	1 → 0	×	1
11	1	0	1 → 1	×	0
	Q^{v+1}				

(a) (b)

$Q^v \rightarrow Q^{v+1}$	S	C	$Q^v \rightarrow Q^{v+1}$	T	$Q^v \rightarrow Q^{v+1}$	D
0 → 0	0	×	0 → 0	0	0 → 0	0
0 → 1	1	0	0 → 1	1	0 → 1	1
1 → 0	0	1	1 → 0	1	1 → 0	0
1 → 1	×	0	1 → 1	0	1 → 1	1

(c) (d) (e)

Eliendo FF tipo JK y calculando las transiciones para las tres asignaciones tenemos que una es:

y_1^v	x^v		y_1^{v+1}	$y_1 y_0^v$	x^v		J_{y_1}	K_{y_1}
	0	1			0	1		
a	0	0	1	00	0	1	×	×
b	0	0	1	01	0	1	×	×
c	1	0	0	11	×	×	1	1
d	1	1	1	10	×	×	0	0

(a) (b) (c)

y_0^v	x^v		y_0^{v+1}	$y_1 y_0^v$	x^v		J_{y_0}	K_{y_0}
	0	1			0	1		
a	0	0	1	00	0	1	×	×
b	1	0	0	01	×	×	1	1
c	1	0	1	11	×	×	1	0
d	0	0	0	10	0	0	×	×

(d) (e) (f)

Estos mapas nos dan las siguientes ecuaciones:

$$J_{y_1} = x \quad J_{y_0} = x\bar{y}_1 \quad K_{y_1} = y_0$$

$$K_{y_0} = \bar{x} + \bar{y}_1 \quad z = x\bar{y}_1 y_0$$

Hacemos lo mismo para las otras dos asignaciones, la segunda nos queda:

	y_1^v	x^v		$(y_1 y_0)^v$	x^v		$(y_1 y_0)^v$	x^v	
			0	1			0	1	
a	0	0	0	0	00	0	0	0	00
c	0	0	1	1	01	0	1	1	01
b	1	0	1	1	11	×	×	×	11
d	1	1	1	1	10	×	×	×	10
			y_1^{v+1}		J_{y_1}				K_{y_1}
	y_0^v	x^v			$(y_1 y_0)^v$	x^v			$(y_1 y_0)^v$
			0	1			0	1	
a	0	0	1	1	00	0	1	1	00
c	1	0	1	1	01	×	×	×	01
b	1	0	0	0	11	×	×	×	11
d	0	0	0	0	10	0	0	0	10
			y_0^{v+1}		J_{y_0}				K_{y_0}

$$J_{y_1} = xy_0 \quad K_{y_1} = \bar{x}y_0 \quad z = xy_1y_0$$

$$J_{y_0} = x\bar{y}_1 \quad K_{y_0} = \bar{x} + y_1$$

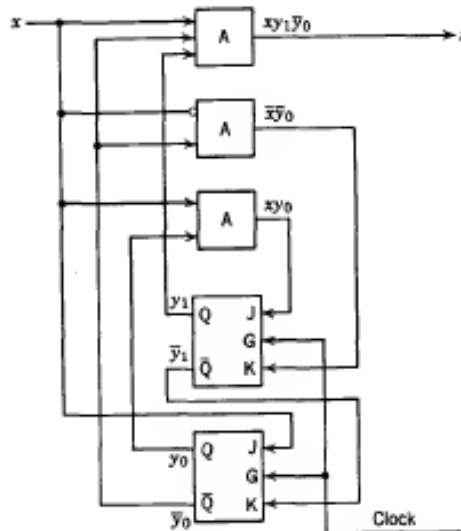
Y finalmente para la tercera asignación nos queda:

	y_1^v	x^v		$(y_1 y_0)^v$	x^v		$(y_1 y_0)^v$	x^v	
			0	1			0	1	
a	0	0	0	0	00	0	0	0	00
c	0	0	1	1	01	0	1	1	01
d	1	1	1	1	11	×	×	×	11
b	1	0	1	1	10	×	×	×	10
			y_1^{v+1}		J_{y_1}				K_{y_1}
	y_0^v	x^v			$(y_1 y_0)^v$	x^v			$(y_1 y_0)^v$
			0	1			0	1	
a	0	0	1	1	00	0	1	1	00
c	1	0	0	0	01	×	×	×	01
d	1	1	1	1	11	×	×	×	11
b	0	0	1	1	10	0	1	1	10
			y_0^{v+1}		J_{y_0}				K_{y_0}

$$J_{y_1} = xy_0 \quad K_{y_1} = \bar{x}\bar{y}_0 \quad z = xy_1\bar{y}_0$$

$$J_{y_0} = x \quad K_{y_0} = \bar{y}_1$$

Vemos que la asignación 1 y 3 tiene el mismo costo básico de compuertas, pero se prefiere la tres porque utiliza solo compuertas AND. En la siguiente figura se muestra el circuito final:



De la experiencia de probar asignaciones podemos definir algunas reglas que tienen importancia decreciente.

Regla 1a. Se deben examinar los renglones de la tabla de estado que tengan anotaciones idénticas para el estado siguiente en cada columna. Estos renglones deben recibir asignaciones adyacentes.

b) Se verifican los renglones de la tabla de estado que tienen las mismas anotaciones del siguiente estado, pero en diferente orden de columnas. A estos renglones se les deben dar asignaciones adyacentes.

c) Los renglones con anotaciones idénticas para el estado siguiente en algunas, pero no en todas las columnas, deben recibir asignaciones adyacentes, en donde los renglones que tengan más columnas idénticas asuman la máxima prioridad.

Regla 2. Las notaciones de estado siguiente para un renglón dado deben recibir asignaciones adyacentes.

Regla 3. Las asignaciones deben hacerse de tal manera que se simplifiquen los mapas de salida.

Nuevamente es importante observar que estas reglas se han enunciado en orden decreciente de importancia para producir buenas asignaciones.

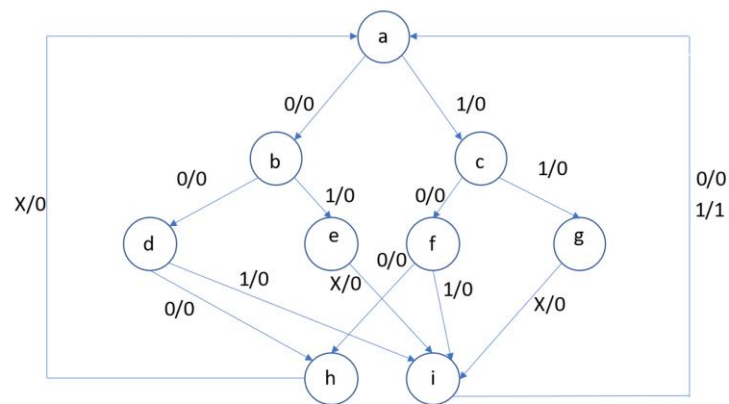
Ejemplo de diseño:

Se desea diseñar un circuito que reciba por una única línea Y, en forma secuencial y sincronizada con el flanco descendente del reloj, los bits de peso 1, 2, 4 y 8 (en ese orden) de un dígito decimal codificado en BCD natural. Luego de recibir los bits de un dígito se pasa inmediatamente a recibir los correspondientes a otro. Este circuito poseerá una única línea de salida Z que, en momento de recibir el bit de peso 8 de cada dígito decimal, indique con un 1 cuando la combinación recibida no corresponda a una de las combinaciones válidas del código BCD natural, señalando así un error.

Antes de empezar a resolver el problema daremos algunas recomendaciones para realizar el diagrama de estados inicial. Hay que recordar que no existen reglas mecánicas para efectuar esta tarea, la que debe realizarse basándose en el sentido común y la experiencia previa. Algunos de estos tips son:

- Si el funcionamiento prevé un estado inicial, comenzar el diagrama de estados a partir de él.
- Analizar los estados sucesivos por los que pasa el circuito, y las salidas correspondientes, cubriendo todas las posibilidades de las entradas y creando, en la medida de lo necesario, nuevos estados. Tratar de no crear nuevos estados equivalentes a **otros ya colocados**.
- **Sin embargo, no preocuparse si por un análisis algo superficial se hubiera creado un nuevo estado sin advertir que existía uno equivalente. Los estados equivalentes pueden ser reconocidos en un paso posterior, como lo vimos anteriormente en este apunte.**

Volviendo al problema, un diagrama de estados que satisface las especificaciones técnicas expresadas en el enunciado de este problema es el indicado en el siguiente gráfico:



Como se observa en el gráfico, recién cuando arriba el cuarto bit (el más significativo de cada palabra) se puede dilucidar si la palabra corresponde a una combinación válida del código BCD. En lugar de los estados **h** e **i** se podría poner 8 estados y de ahí, volver al estado inicial **a**. Esto no hubiera sido problema porque nos falta simplificar este diagrama, aunque intuitivamente pudimos resolver esto con los dos estados enunciados.

Ahora pasamos del diagrama de estados a la tabla de estados, como vemos en la siguiente figura:

Estado	Estado futuro, salida	
	Y=0	Y=1
a	b,0	c,0
b	d,0	e,0
c	f,0	g,0
d	h,0	i,0
e	i,0	i,0
f	h,0	i,0
g	i,0	i,0
h	a,0	a,0
i	a,0	a,1

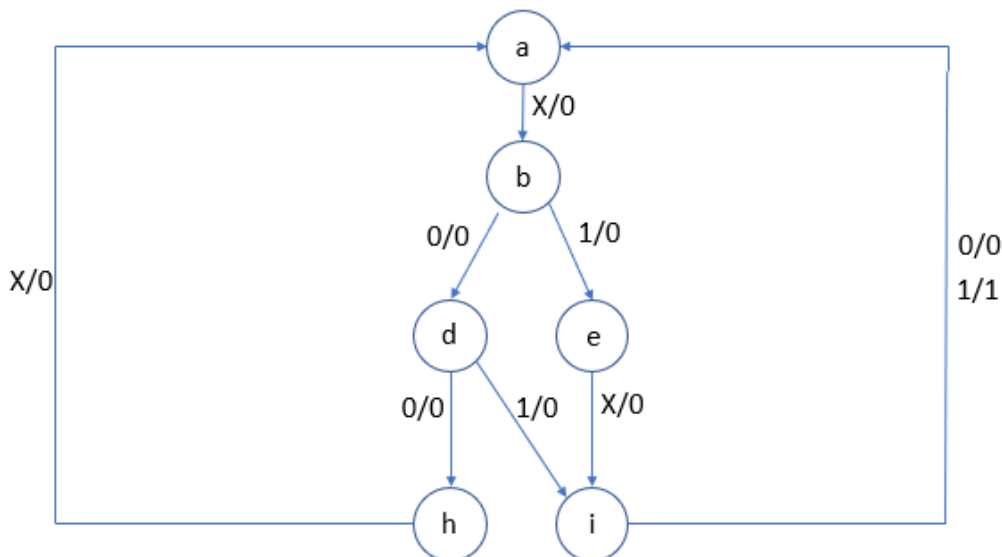
Ahora buscamos los estados equivalentes, podríamos usar la tabla de implicaciones u otro método. En este caso en forma visual se puede observar que los estados **e** y **g**, como así también los estados **d** y **f**, son equivalentes. Eliminando los equivalentes, reemplazamos el **e** por el **g** y el **d** por el **f**, llegamos a:

Estado	Estado futuro, salida	
	Y=0	Y=1
a	b,0	c,0
b	d,0	e,0
c	d,0	e,0
d	h,0	i,0
e	i,0	i,0
h	a,0	a,0
i	a,0	a,1

En esta tabla, recién ahora, se advierte que los estados **b** y **c** también son equivalentes. Resolviendo el caso de estos dos estados llegamos a la tabla final:

Estado	Estado futuro, salida	
	Y=0	Y=1
a	b,0	b,0
b	d,0	e,0
d	h,0	i,0
e	i,0	i,0
h	a,0	a,0
i	a,0	a,1

Podemos rehacer el diagrama de estado final habiendo eliminado todos los estados equivalentes, o sea el gráfico que represente la última tabla mostrada más arriba:



Ahora tenemos que hacer la asignación de estados, para lo cual aplicamos las reglas y los criterios enunciados anteriormente. Por lo tanto, como ya lo dijimos, hay infinidad de asignaciones posibles. Nosotros elegimos una que respeta los criterios de asignación ya mencionados. No haremos acá una descripción detallada de la asignación, pero podemos siguiendo las reglas dadas que los estados **d** y **e** como así también **h** e **i** deberían tener asignaciones adyacentes. Obviamente tomamos el estado **a** como inicial para el cual la salida de los Flip Flop están en **0**. Una asignación posible respetando estas reglas es el siguiente:

Estado	Q _C	Q _B	Q _A
a	0	0	0
b	0	0	1
d	0	1	0
e	1	1	0
h	0	1	1
i	1	1	1
no asignado	1	0	0
no asignado	1	0	1

Ahora rehacemos la tabla de estados designando ahora a estos estados con la asignación decidida en el gráfico anterior, llegamos a:

			Y=0				Y=1			
Q _C	Q _B	Q _A	Q _C *	Q _B *	Q _A *	Z	Q _C *	Q _B *	Q _A *	Z
0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	1	0	0	1	1	0	0
0	1	0	0	1	1	0	1	1	1	0
0	1	1	0	0	0	0	0	0	0	0
1	0	0	X	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X	X
1	1	0	1	1	1	0	1	1	1	0
1	1	1	0	0	0	0	0	0	0	1

De más está decir que a cada estado le asignamos 3 bits, esto se debe a que para numerar esta cantidad de estados con 2 bits no nos alcanza. Con tres bits podemos numerar 8 estados, o sea en este caso nos quedarán dos estados no especificados.

Lo que queda es elegir qué tipo de Flip Flop queremos usar, en nuestro caso usamos FF JK, en la siguiente figura vemos la tabla de transiciones de este tipo de FF y también del tipo T:

		Flip-flop JK		Flip-flop T
Q	Q*	J	K	T
0	0	0	X	0
0	1	1	X	1
1	0	X	1	1
1	1	X	0	0

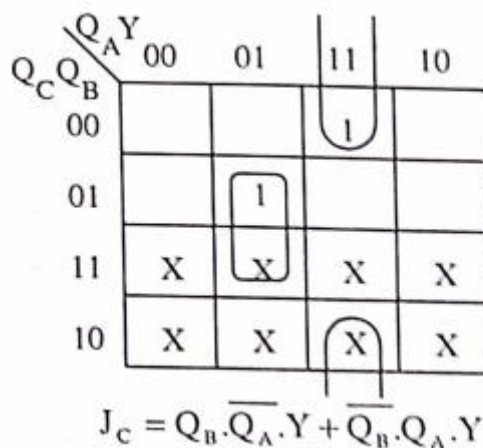
Aplicando esta tabla de transiciones para el flip flop tipo JK. Nos queda la tabla de estados con las transiciones de los FF de la siguiente manera:

			Y=0				Y=1				Y=0				Y=1							
Q _C	Q _B	Q _A	Q _C *	Q _B *	Q _A *	Z	Q _C *	Q _B *	Q _A *	Z	J _C	K _C	J _B	K _B	J _A	K _A	J _C	K _C	J _B	K _B	J _A	K _A
0	0	0	0	0	1	0	0	0	1	0	0	X	0	X	1	X	0	X	0	X	1	X
0	0	1	0	1	0	0	1	1	0	0	0	X	1	X	X	1	1	X	1	X	X	1
0	1	0	0	1	1	0	1	1	1	0	0	X	X	0	1	X	1	X	X	0	1	X
0	1	1	0	0	0	0	0	0	0	0	0	X	X	1	X	1	0	X	X	1	X	1
1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	1	1	0	1	1	1	0	X	0	X	0	1	X	X	0	X	0	1	X
1	1	1	0	0	0	0	0	0	0	1	X	1	X	1	X	1	X	1	X	1	X	1

Ahora tenemos que simplificar esta tabla, para ello usamos mapas de Karnaugh. No obstante, a simple vista, se ve que:

$$J_A=1 \quad K_A=1 \quad J_B=Q_A \quad K_B=Q_A \quad K_C=Q_A$$

Para las demás variables hacemos los mapas nombrados:

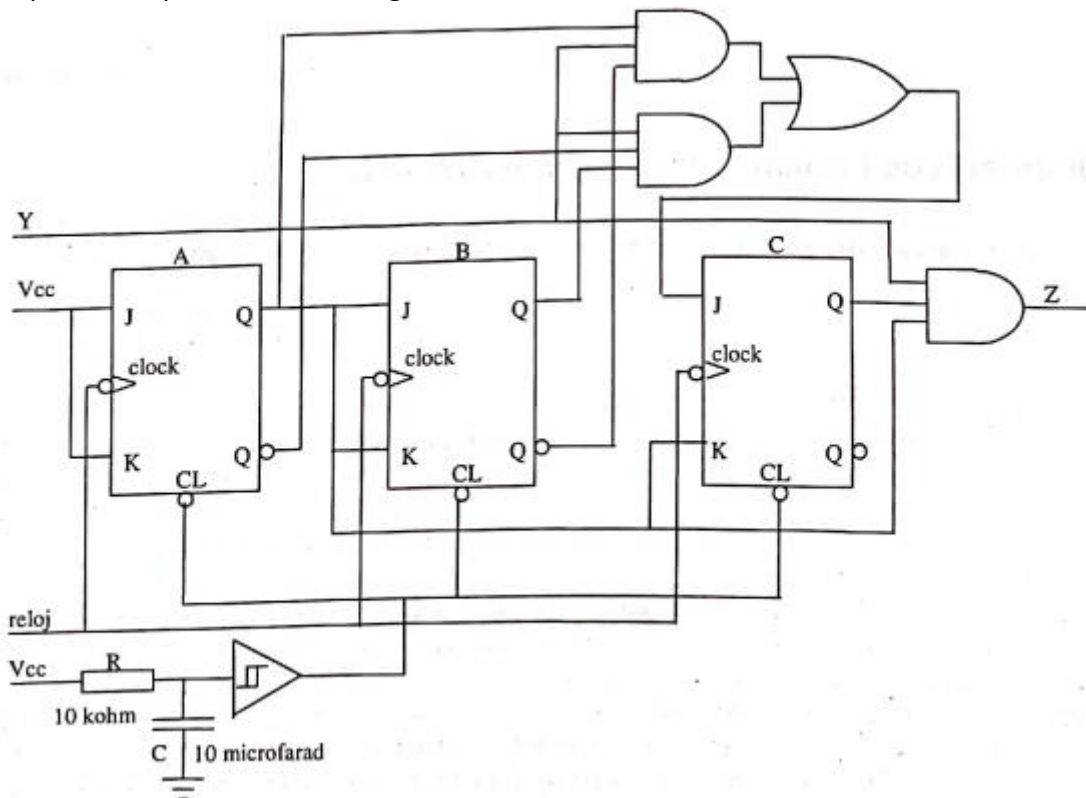


Y el mapa para la salida Z del circuito queda:

	Q _A Y	00	01	11	10
Q _C Q _B	00				
	01				
	11			1	
	10	X	X	X	X

$Z = Q_C \cdot Q_A \cdot Y$

Por lo tanto, del circuito final que funciona de acuerdo con las especificaciones del problema planteado es el siguiente:

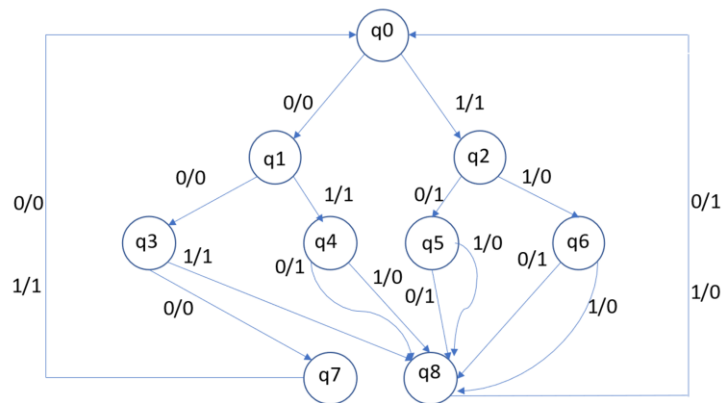


Otro ejemplo:

Diseñar un circuito secuencial síncrono que recibe por una sola línea de entrada **X** uno tras otros dígitos de 4 bits. Tiene una salida **Z** que calcula el **complemento a dos** de la entrada. Luego de computar los cuatro bits vuelve a leer los otros cuatro siguientes.

Para empezar a resolver este problema recordemos que la regla práctica para hallar el complemento al módulo de un número binario es recorrer el número de derecha a izquierda hasta encontrar el primer uno, ese uno queda como está y después se cambian unos por ceros y ceros por unos. Así, por ejemplo, el complemento al módulo del número 010101 será 101011.

Realizamos el diagrama de estados (recordar en este punto que van a existir tantos diagramas distintos como personas lo hagan). En nuestro caso nos queda:



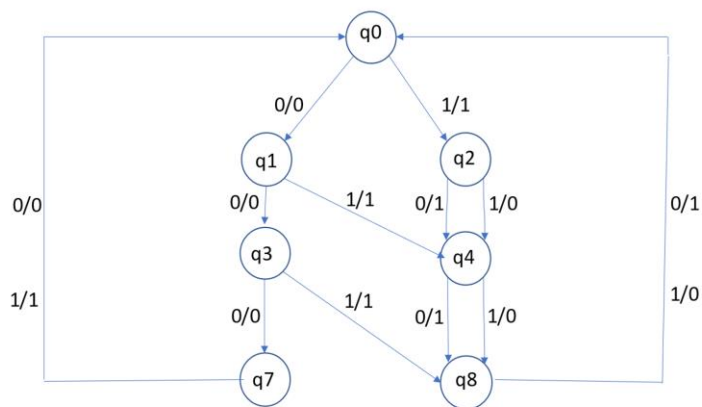
Pasando este gráfico a una tabla de estados nos queda:

	X=0	X=1
q0	q1/0	q2/1
q1	q3/0	q4/1
q2	q5/1	q6/0
q3	q7/0	q8/1
q4	q8/1	q8/0
q5	q8/1	q8/0
q6	q8/1	q8/0
q7	q0/0	q0/1
q8	q0/1	q0/0

Buscando estados equivalentes (usar el método que más guste). Se ve claramente que los estados **q4**, **q5** y **q6** son equivalentes. Los reemplazamos por el estado **q4** y llegamos a la siguiente tabla de estados reducida:

	X=0	X=1
q0	q1/0	q2/1
q1	q3/0	q4/1
q2	q4/1	q4/0
q3	q7/0	q8/1
q4	q8/1	q8/0
q7	q0/0	q0/1
q8	q0/1	q0/0

Esta tabla reducida se corresponde con el siguiente diagrama de estados:



Recordando las reglas empíricas nombradas anteriormente, una asignación de estados posibles es la siguiente:

q0	000
q1	111
q2	101
q3	100
q4	110
q7	001
q8	011

Ahora realizamos una tabla con el estado actual y el estado siguiente respetando la asignación de estados reflejada en la tabla anterior, y nos queda:

Qc	Qb	Qa	X	Q'c	Q'b	Q'a	Z
----	----	----	---	-----	-----	-----	---

0	0	0	0	1	1	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	X	X	X	X
0	1	0	1	X	X	X	X
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	0
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	0
1	1	0	0	0	1	1	1
1	1	0	1	0	1	1	0
1	1	1	0	1	0	0	0
1	1	1	1	1	1	0	1

Suponiendo que usamos Flip Flop tipo D, recordar que la ecuación característica de este dispositivo es:

$$Q^{n+1} = D_n$$

Como el próximo estado corresponde a el valor en D, en la tabla anterior cada entrada D de los flip flop coincide con el próximo estado. Como se ve en la misma la variable C es la más significativa. Resolviendo los mapas de Karnaugh, nos queda:

$$Q'c = (C \oplus A)'$$

$$Q'b = B.A' + C.X + C'.A'.X' + C.B'.A$$

$$Q'a = A'$$

$$Z = A'.X'.B + X'.C.B' + A'.X.B' + X.C'.B' + A.X.B.C + A.X'.C.B'$$

Si usáramos otro tipo de asignación como por ejemplo la siguiente:

q0	000
q1	001
q2	011
q3	010
q4	110
q7	101
q8	111

Las ecuaciones finales nos darían de la siguiente manera:

$$Q'c = B.A' + C'.B + C'.A.X$$

$$Q'b = C.A' + A'.X + C'.A$$

$$Q'a = A'$$

$$Z = A'.X + A.X.C' + C.A.X' + A.B.X'$$