

8 Circuitos Secuenciales

⊗ En la *lógica combinatorial* los circuitos producen una respuesta instantánea, es decir, las *salidas* se pueden calcular a partir de la combinación de los valores de las *entradas* en el mismo instante. La lógica combinatorial no sirve para construir circuitos que con capacidad de *memoria*, es decir, funciones lógicas cuya salida en el instante presente depende de entradas en el pasado. Es entonces, cuando los circuitos secuenciales aparecen y cobran relevancia conceptos que no eran tan trascendentes para los circuitos combinatoriales, algunos de estos conceptos son: *instante presente*, *instante siguiente*, *estado*, *retroalimentación*, *tiempo de propagación*, *sincronización*, *memoria*, *secuencia*, *conteo*, etc. Obsérvese que el principal concepto involucrado en todos los anteriores es el *tiempo*.

8.1.- INTRODUCCIÓN

Los circuitos considerados hasta aquí, tienen la característica de que su salida depende solamente de la combinación presente de valores de las entradas, es decir, a una misma combinación de entrada responden siempre con la misma salida. Debido a esto, estos circuitos se denominan **combinacionales**.

Los circuitos combinatoriales tienen muchas limitantes debido a que no son capaces de reconocer el orden en que se van presentando las combinaciones de entradas con respecto al tiempo, es decir, no pueden reconocer una **secuencia** de combinaciones, ya que no poseen una manera de almacenar información pasada, es decir no poseen **memoria**.

⌘ Un circuito cuya salida depende no solo de la combinación de entrada, sino también de la historia de las entradas anteriores se denomina **Circuito Secuencial**. La historia de las entradas anteriores en un momento dado se encuentra resumida en el **estado** del circuito, el cual se expresa en un conjunto de **variables de estado**.

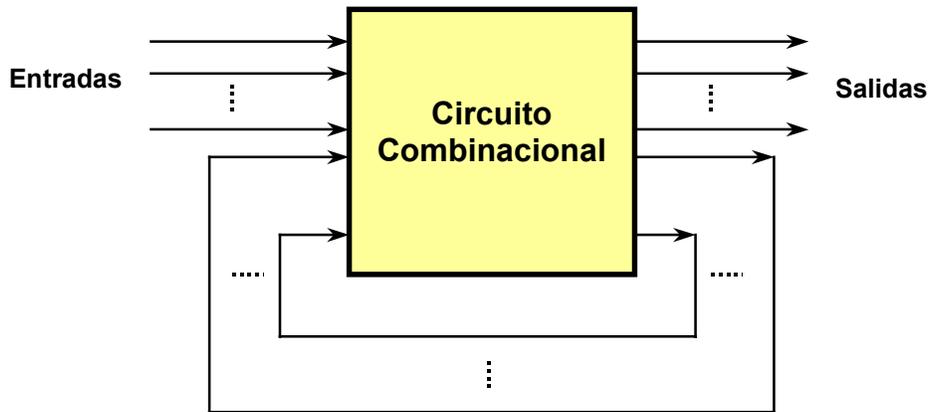
El circuito secuencial debe ser capaz de mantener su estado durante algún tiempo, para ello se hace necesario el uso de **dispositivos de memoria**. Los dispositivos de memoria utilizados en circuitos secuenciales pueden ser tan sencillos como un simple retardador (inclusive, se puede usar el retardo natural asociado a las compuertas lógicas) o tan complejos como un circuito completo de memoria denominado **multivibrador biestable o Flip Flop**.

Como puede verse entonces, en los circuitos secuenciales entra un factor que no se había considerado en los combinatoriales, dicho factor es *el tiempo*. De hecho, los circuitos secuenciales se clasifican de acuerdo a la manera como manejan el tiempo en **circuitos secuenciales síncronos y circuitos secuenciales asíncronos**.

⌘ En un **circuito secuencial asíncrono**, los cambios de estado ocurren al ritmo natural marcado por los retardos asociados a las compuertas lógicas utilizadas en su implementación, es decir, estos circuitos no usan elementos especiales de memoria, pues se sirven de los retardos propios (tiempos de propagación) de las compuertas lógicas usados en ellos. Esta manera de operar puede ocasionar algunos problemas de funcionamiento, ya que estos retardos naturales no están bajo el control del diseñador y además no son idénticos en cada compuerta lógica.

⌘ Los **circuitos secuenciales síncronos**, sólo permiten un cambio de estado en los instantes marcados por una señal de sincronismo de tipo oscilatorio denominada **reloj**. Con ésto se pueden evitar los problemas que tienen los circuitos asíncronos originados por cambios de estado no uniformes en todo el circuito.

Un **circuito secuencial** puede entenderse simplemente como un circuito combinacional en el cual las salidas dependen tanto de las entradas como de las salidas en instantes anteriores, esto implica una **retroalimentación de las salidas** como se muestra en diagrama de la siguiente figura

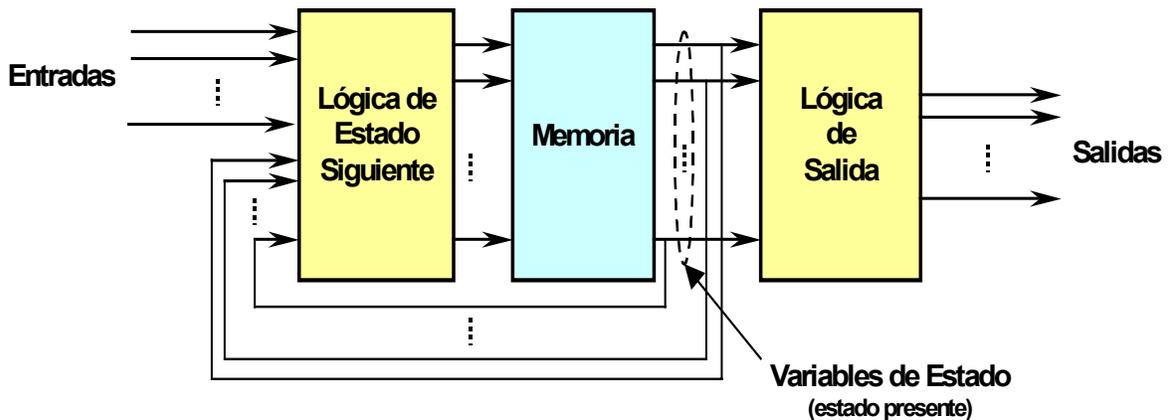


📖 **Modelo de Moore.-** Un modelo más completo de lo que puede ser un circuito secuencial es el denominado Modelo de Moore, cuya estructura se presenta en la siguiente figura.

El modelo de Moore consiste en dos bloques (circuitos) de lógica combinacional mas un bloque de **memoria**

- ❑ **La lógica de estado siguiente** que define la manera de generar las variables de estado a partir de las entradas
- ❑ **La Lógica de salida** que define la manera en que se obtienen las salidas del circuito a partir de las variables de estado

Este modelo tiene la particularidad de que *las salidas sólo son función de las variables de estado*, es decir, del estado presente. Por ello, cuando en un circuito, las salidas solo dependen de las variables de estado, se les llama **Salidas tipo Moore**.



📖 **Modelo de Mealy.-** En un modelo más completo, las salidas en instantes anteriores están expresadas por un conjunto de **variables de estado**, de manera que las salidas actuales dependen tanto de las

El Flip Flop Set Reset FF-SR

El FF-SR es un dispositivo con dos entradas (Set y Reset) y una variable de estado o salida (Q) capaz de “guardar” un bit de información y funciona como sigue:

- Si su entrada Set se activa su estado Q se pone en Alto
- Si su entrada Reset se activa su estado Q se pone en Bajo
- Si no se activa ni Set ni Reset su estado no cambia
- Por supuesto, no se permite activar Set y Reset simultáneamente.

Diagrama de Bloques

Aunque el FF-SR posee dos entradas (S y R) y sólo una salida (Q), es común la implementación que provee además de Q su versión complementada \bar{Q} , como se muestra en la figura siguiente

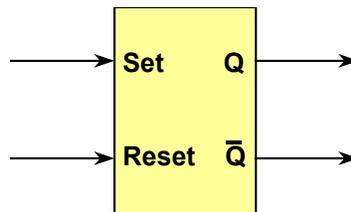


Tabla de Funcionamiento

Los fabricantes de los circuitos integrados usan una tabla de funcionamiento para describir la operación de un circuito de una manera compacta, dicha tabla de funcionamiento no es otra cosa que una tabla de verdad como la usada para circuitos combinacionales, en la cual se ha introducido la información del tiempo que en el caso de circuitos secuenciales se vuelve esencial. Enseguida se ilustrará el uso de esta tabla para describir de manera compacta el funcionamiento del FF-SR.

Entradas en t_n		Salida en t_{n+1}
S	R	Q^+
0	0	Q_0
0	1	0
1	0	1
1	1	no válida

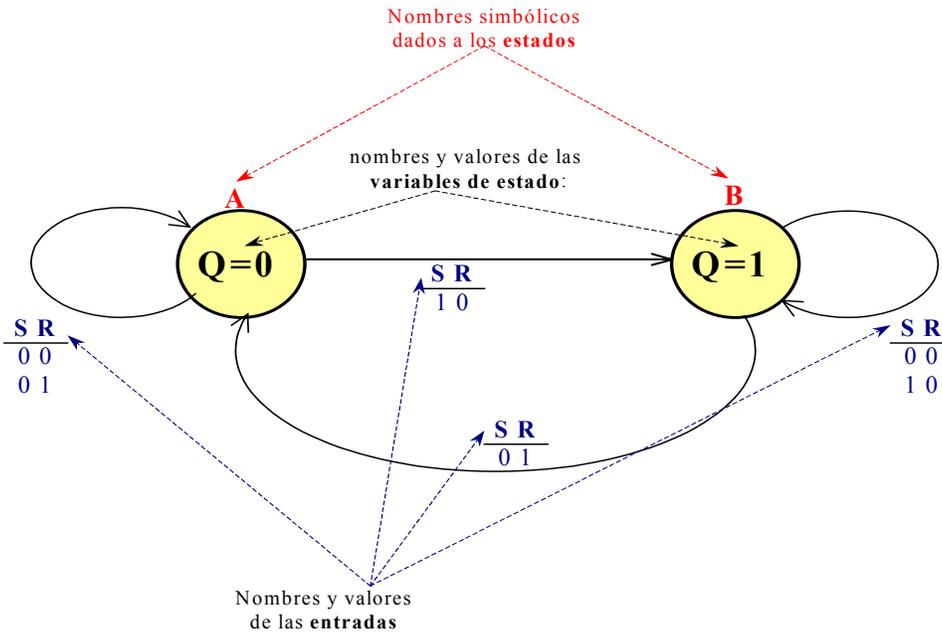
En donde se ha utilizado la siguiente notación:

- t_n = instante en el cual se aplican las entradas.
- t_{n+1} = instante después que el circuito responde.
- Q_0 = salida Q en el instante t_n
- Q^+ = salida en el instante t_{n+1}

☞ No es difícil notar que la tabla de funcionamiento es una tabla de verdad con la **variable introducida** Q_0 .

Diagrama de Estado Clásico

La misma información especificada por la tabla de funcionamiento puede ser representada de varias maneras diferentes, por ejemplo, el siguiente diagrama es una alternativa gráfica que tiene la particularidad de enfatizar el número y nombre de los estados del circuito, por ello se le llama diagrama de estado o de estado clásico. Así, para el FF-SR:



Obsérvese que el diagrama de estado clásico incluye información separada de la siguiente manera:

- Nombres simbólicos dados a los estados (opcional)
- Nombres y valores que las variables de estado toman en cada estado.
- Nombres y valores de las variables de entrada
- Transiciones posibles de un estado a otro y condiciones (sobre las variables de entrada) para producir dicha transición.
- En algunas variantes de diagrama de estado se incluye también información sobre las **variables de salida** que no se muestran en el ejemplo, dado que para el FF-SR la variable de estado Q coincide con la variable de salida.

Tabla de Excitación

La información que guarda el diagrama de estado clásico se puede representar en forma de tabla colocando **todas las transiciones posibles** de un estado a otro como variables independientes de la tabla y las entradas como variables dependientes, es decir, se genera un renglón de la tabla por cada transición y anotando los valores necesarios de las entradas para producir dicha transición. Así, para el ejemplo del FF-SR se obtiene

Transiciones posibles		Entradas que producen la transición	
Q _o	Q ⁺	S	R
0	0	0	*
0	1	1	0
1	0	0	1
1	1	*	0

Tabla de Estado o Tabla Característica

Esta es otra manera de organizar en forma de tabla el comportamiento del circuito secuencial, Se trata básicamente de la misma tabla de funcionamiento ya descrita, salvo que ahora no se introduce ninguna variable de manera que el estado presente (Q_o) se trata como si fuera otra entrada. Para el ejemplo del FF-SR tendremos

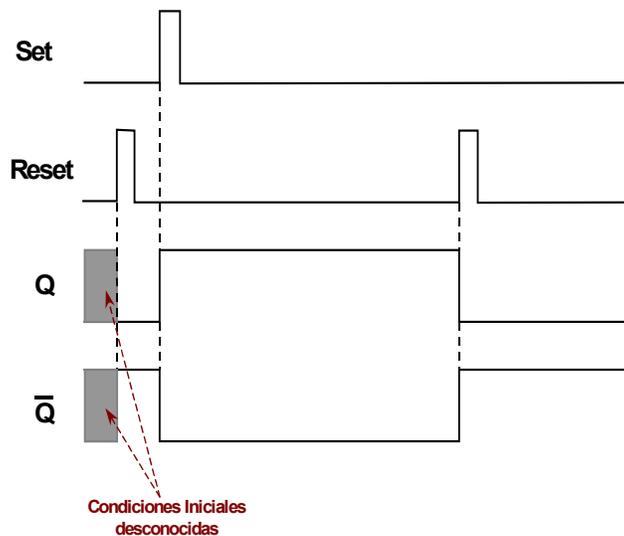
Estado	Estado Presente Q_0	Entradas en t_n S R	Estado Siguiete Q^+
A	0	0 0	0
A	0	0 1	0
A	0	1 0	1
A	0	1 1	no válida
B	1	0 0	1
B	1	0 1	0
B	1	1 0	1
B	1	1 1	no válida

Diagramas de tiempo

Los diagramas de tiempo son representaciones gráficas de la evolución de los valores que toman las variables de interés en un circuito digital, de la manera como se podrían ver en la pantalla de un osciloscopio.

Los diagramas de tiempo no son una herramienta propia de los circuitos secuenciales, ya que estos también son útiles para circuitos combinacionales como se ilustró en los capítulos anteriores, sin embargo, en el caso de los circuitos secuenciales, la información de tiempo es más crucial por esto los diagramas de tiempo cobran una mayor importancia que en el caso combinacional.

Es importante mencionar que estos diagramas no son únicos para un circuito dado, de hecho, pueden poseer información incompleta o en ocasiones redundante. Así, para el ejemplo del FF-SR un posible diagrama de tiempo sería como en la siguiente figura



Diseño de un circuito secuencial

De acuerdo a las estructuras planteadas para los circuitos secuenciales se puede ver que éstos se pueden diseñar con las herramientas descritas para los circuitos combinacionales, pero tomando en cuenta la retroalimentación del estado presente.

Diseño del Flip Flop Set Reset.

Como ejemplo introductorio, consideraremos el problema de diseñar el Flip Flop-SR. En este caso la salida Q^+ depende del estado anterior Q_0 y de las entradas S y R, es decir,

$$Q^+ = f(Q_0, S, R)$$

Es decir, el diseño lo plantearemos como si se tratara de un circuito combinacional, pero considerando Q_0 como si fuera una entrada más. Esta función la podemos plantear por medio de la siguiente tabla de verdad, obtenida de la tabla de estado descrita anteriormente:

Entradas en t_n			Salida en t_{n+1}
Q_0	S	R	Q^+
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	*
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	*

El Mapa de Karnaugh correspondiente es el siguiente

		$Q_0 S$			
		00	01	11	10
R	0	0 ⁰	1 ²	1 ⁶	1 ⁴
	1	0 ¹	* ³	* ⁷	0 ⁵

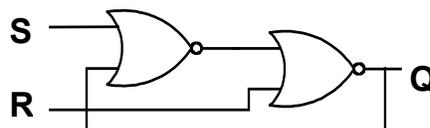
De donde podemos obtener la expresión siguiente. (Aunque no es un procedimiento común, la experiencia a demostrado que se puede obtener una implementación más sencilla despreciando las condiciones sin cuidado), entonces

$$Q^+ = S\bar{R} + Q_0\bar{R}$$

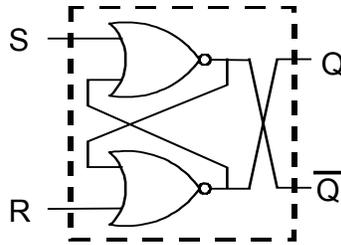
Para implementar con sólo compuertas NOR negamos dos veces la expresión para obtener

$$Q^+ = \bar{\bar{R}}(\bar{S + Q_0}) = \bar{R} + \overline{(S + Q_0)}$$

Con lo cual podemos implementar el FF-SR con sólo dos compuertas NOR, como sigue



Esta implementación además tiene la ventaja de que también produce la función negada \bar{Q} a la salida de la primera compuerta NOR, de manera que una mejor manera de dibujar este circuito es como se muestra en la siguiente figura



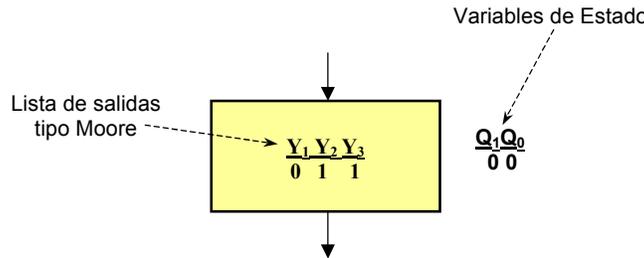
Diagramas de Flujo de Estado

Una herramienta alternativa para la representación, análisis y diseño de circuitos secuenciales son los Diagramas de Flujo de Estado, también conocidos como diagramas ASM (Algorithmic State Machine), los cuales no son más que una manera diferente de dibujar un diagrama de estado clásico, con símbolos muy similares a los usados en los *Diagramas de Flujo* usados para especificar programas de computadora como los que se describen a continuación

Símbolos usados en los diagramas ASM

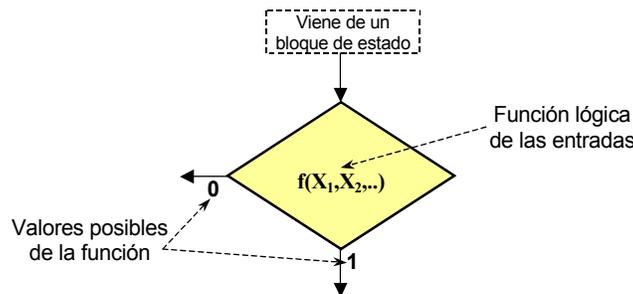
Bloque de Estado.- El diagrama deberá tener un bloque de estado por cada posible estado presente del circuito.

En este bloque se especifica una lista de las salidas que dependen de este estado (Salidas tipo Moore) y los valores que toman en dicho estado. También se deberá especificar a un lado del bloque los valores que toman las variables de estado.



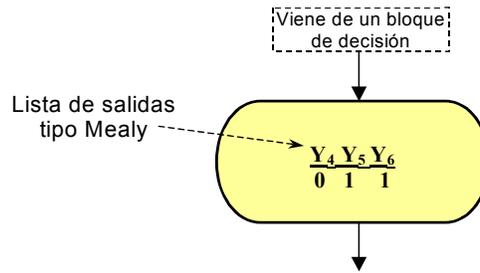
Bloque de decisión.- Los bloques de decisión son los que establecen las condiciones para que ocurra un cambio de estado, es decir, definen las trayectorias posibles y las condiciones para pasar de un estado a otro.

Dentro del bloque se deberá especificar la expresión lógica (en términos de las entradas) que decide cual es la trayectoria a seguir y en cada salida del bloque se deberá especificar el valor de la expresión para seguir por dicha salida.



Bloque de Salida.- Este bloque siempre viene de un bloque de decisión para especificar salidas cuyo valor depende del estado y de las entradas de dicho bloque de decisión (Salidas tipo Mealy).

Se especifica dentro de este bloque la lista de salidas que dependen del estado y de las entradas así como los valores que toman dichas salidas.



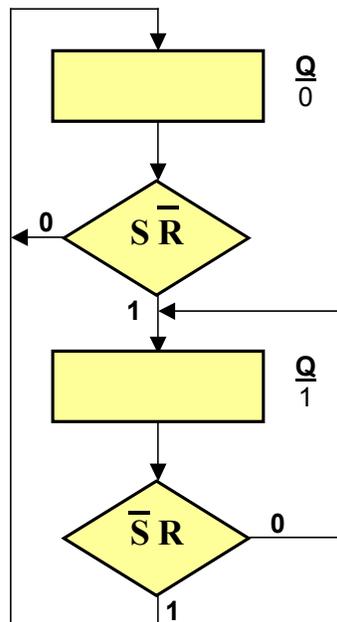
Ejemplo

Para dibujar el diagrama de flujo de estado de un circuito secuencial es importante hacer una lista preliminar que contenga la siguiente información. (Los valores se especifican entre paréntesis para el caso del FF-SR).

- Número de Estados (2)
- Número de variables de estado (1)
- Nombres de las variables de Estado (Q), Entradas (S,R) y Salidas (Q) y tipo (Moore).

Obsérvese que en el caso del FF-SR la única salida corresponde a la única variable de estado, por lo tanto es de tipo Moore.

El número de estados indica directamente el número de bloques de estado, de manera que el diagrama queda como sigue



Técnicas de Análisis y Diseño de Circuitos Secuenciales Asíncronos.

Basados en las herramientas descritas hasta aquí (especialmente los diagramas de flujo de estado), se pueden plantear los procedimientos de diseño y de análisis para circuitos secuenciales asíncronos como sigue

ANÁLISIS

- 1) Especificar el circuito lógico (diagrama de compuertas lógicas)
- 2) Obtener las ecuaciones de estado siguiente
- 3) Obtener la tabla característica parcial para cada estado
- 4) Dibujar el **diagrama de flujo de estado**

DISEÑO

- 1) Especificar la función a realizar por el circuito mediante un **diagrama de flujo de estado**
- 2) Obtener la tabla característica
- 3) Obtener los Mapas de Karnaugh
- 4) Obtener las ecuaciones de estado siguiente simplificadas
- 5) Implementar el circuito.

☞ **Observación.-** Si en lugar de usar **Diagramas de Flujo de Estado** se usan **Diagramas de Estado Clásico**, los procedimientos no cambian de manera considerable, salvo que el uso de diagramas de estado clásico nos lleva a tablas de verdad y MK completos, mientras que el uso de Diagramas de Flujo de Estado nos lleva a Tablas de Verdad y **MK reducidos (Con Variables Introducidas)**, con lo cual el procedimiento de diseño es más abreviado.

Ejemplo: Diseño del FF Tipo Latch (Cerrojo)

El FF Tipo Latch es una variante de Flip Flop, con dos entradas (G,D) y una salida Q. El funcionamiento de este FF se puede describir en palabras como sigue

- Mientras la entrada G esté en bajo, la salida Q no cambia.
- Cuando la entrada G está en alto, la salida Q toma el valor de la entrada D.

A continuación se expresa este funcionamiento en un diagrama de flujo de estado

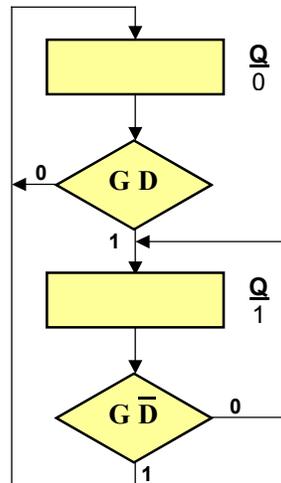


Diagrama de bloques del FF-Latch

Para el diseño es conveniente también tener en mente el diagrama de bloques en donde se enfatiza cuales son entradas y cuales son salidas

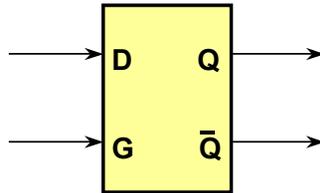
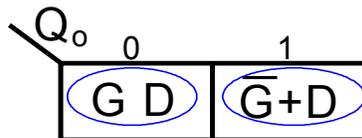


Tabla Característica del FF Tipo Latch

A partir del diagrama de flujo de estado es fácil obtener la siguiente tabla característica con variables introducidas, en donde se puede notar que hay un renglón por cada bloque estado del diagrama de flujo y este renglón a su vez se divide en tantos renglones como transiciones (trayectorias) posibles hay de un estado a otro o a sí mismo.

Estado presente Q_n	Condiciones de Entrada para producir un cambio (C.E.)	Estado Siguiete Q^+	$Q^+(C.E)$
0	\overline{GD}	0	GD
	GD	1	
1	$G\overline{D}$	0	$\overline{G} + D$
	\overline{GD}	1	

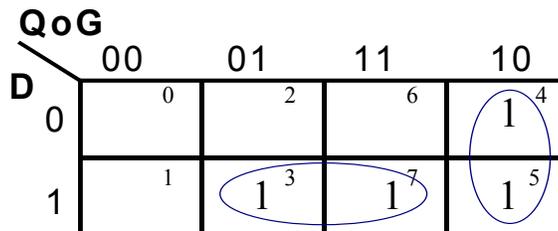
De aquí tenemos el siguiente mapa de Karnaugh reducido:



Reduciendo el Mapa con variables introducidas anterior obtenemos

$$Q = \overline{Q}_0GD + Q_0\overline{G} + Q_0D$$

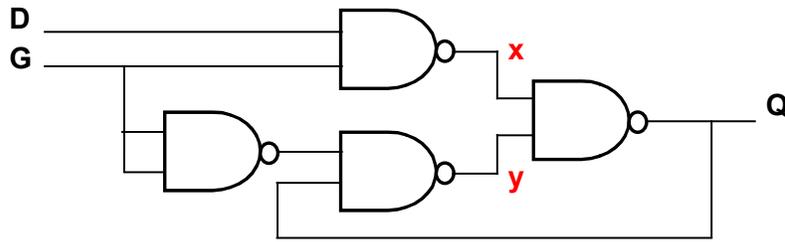
Sin embargo, esta expresión se puede simplificar aún más usando un MK normal, como sigue



Con lo que obtenemos la expresión

$$Q = GD + Q_0\overline{G}$$

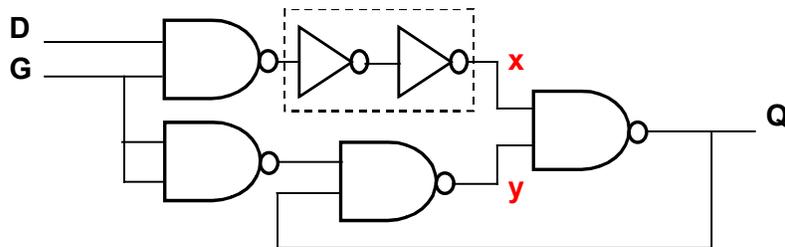
y su implementación con compuertas NAND será como sigue



Sin embargo en esta implementación ocurre un problema, que es común al diseñar circuitos asíncronos con retroalimentación:

Este circuito no trabaja como se espera si después de la combinación D=1, G=1 (Estado Q=1) se presenta la combinación D=1, G=0, esto se debe a que el retardo de G a X es menor que el retardo de G a Y. Este problema se puede corregir de dos maneras:

Una manera consiste en usar dos inversores en el trayecto de G a X para adicionar dos retardos de propagación, de tal manera que cuando D=1 y G baje a 0 la línea X permanecerá en bajo hasta que la línea Y esté en bajo, para asegurar que Q se quede en 1, o sea:



Otra manera de resolver el problema es añadir el término de consenso a la ecuación mínima de Q⁺, es decir, implementar la ecuación redundante:

$$Q = GD + Q_0 \bar{G} + Q_0 D$$

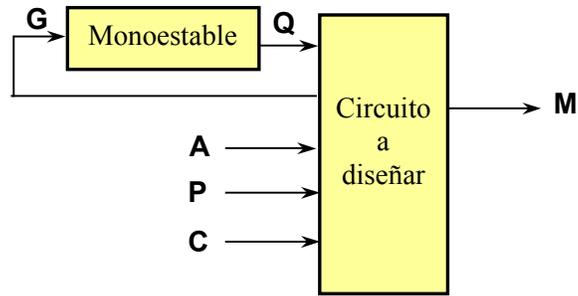
Debido a los problemas que se pueden tener al tratar de implementar un circuito secuencial asíncrono por la diferencia de retardos entre diferentes rutas del circuito, en la mayoría de los casos es preferible diseñar los circuitos secuenciales de manera síncrona. Este tema se tratará más adelante, antes veamos otros ejemplos de diseño asíncrono.

Ejemplo.

Diseñar un circuito secuencial para controlar el arranque de un motor bajo las siguientes especificaciones de funcionamiento

- Si el motor está apagado, se encenderá durante 15 segundos al presionar un botón de arranque A, siempre y cuando un switch de permiso C esté activado.
- Si el motor está encendido se apagará al presionar un botón de paro P o al terminar los 15 segundos.

1er paso: Diagrama de bloques.- El circuito a diseñar tiene tres entradas (A, P y C) y una salida (M), sin embargo, requiere de un circuito (tipo monoestable) capaz de generar una señal de 15 segundos (para proporcionar el retardo), esto hace necesario una salida (G) y una entrada (Q) adicionales como se muestra en la siguiente figura



2º paso: Diagrama de Flujo de Estado

De acuerdo a las especificaciones del circuito, podemos dos estados: Motor apagado y Motor encendido (esperando que transcurran los 15 segundos o el botón de paro). Con esto obtenemos el siguiente diagrama de dos estados.

