

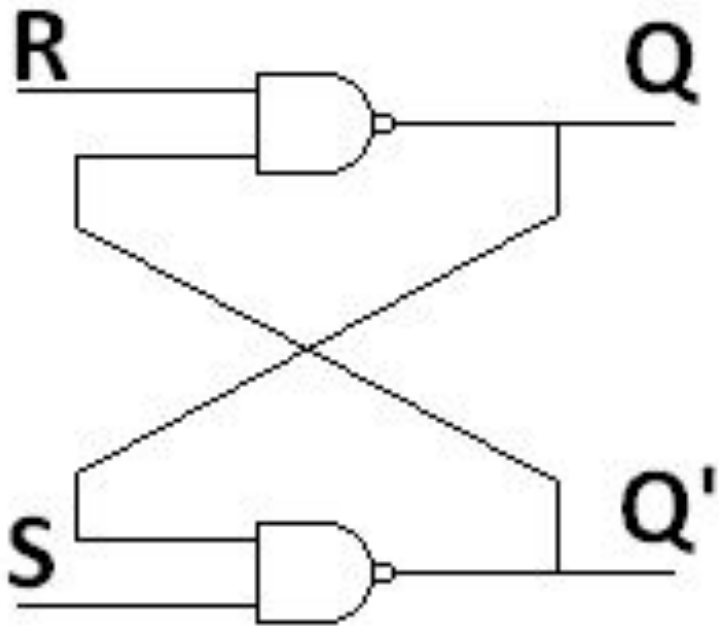
Universidad Autónoma de Querétaro

Facultad de Ingeniería
Ingeniería en Automatización

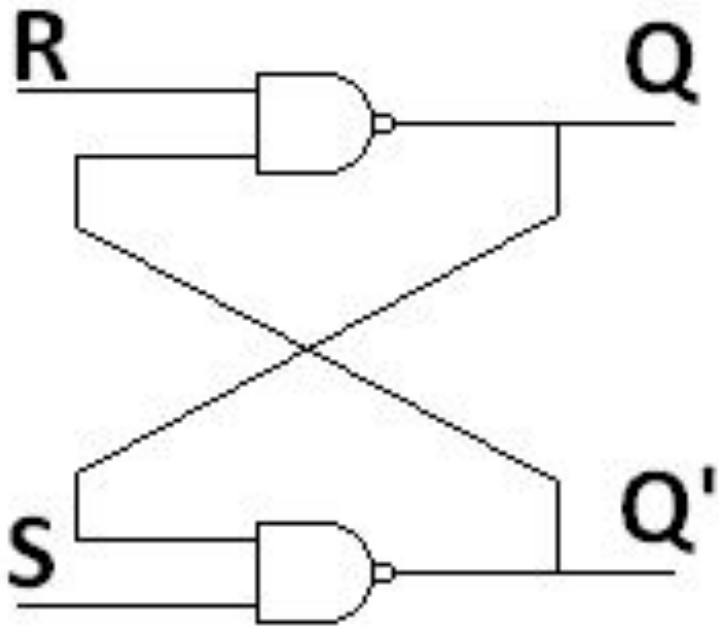


SDLR – Maquinas de estado

M en C. Marcos Romo Avilés



- Contraria a la lógica combinacional, donde las funciones de salida dependen únicamente de las entradas, en la lógica secuencial además de las entradas, las funciones de salida dependen del estado que guarda la máquina.
- La **lógica secuencial** es un tipo de circuito lógico cuya salida depende tanto de los valores de entrada actuales como de los valores de entradas anteriores (historial de entradas).
- La lógica secuencial posee un *estado (memoria)* que influye en el resultado de salida, mientras que la lógica combinacional no



R	S		
0	0	1*	1*
0	1	1	0
1	0	0	1
1	1		

- Candado SR, donde S es el set y R el Reset.
- Cuando se activa S la salida es forzada a un 1 lógico
- Cuando se activa R la salida es forzada a un 0 lógico

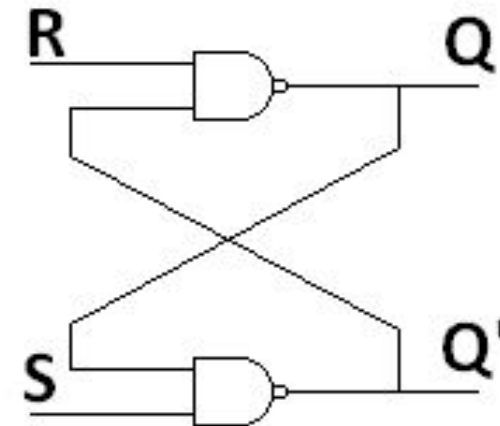
- **Elemento de memoria:** Es un dispositivo lógico formado con compuertas que mediante un anillo de retroalimentación es capaz de guardar información.
- **Estado:** En un sistema secuencial, el estado se refiere a una configuración particular del sistema en un momento dado. Es la condición o situación en la que se encuentra el sistema en un momento específico del tiempo.
- **Sistema secuencial:** Contiene uno o varios elementos de memoria. Por lo que cuenta con un estado actual y un estado futuro. Por lo tanto, la variable de tiempo se encuentra implicada.
- **Estado actual:** El estado actual es el estado en el que se encuentra el sistema en un momento dado. Es la condición presente del sistema, determinada por su historia de entradas y estados anteriores. Su símbolo es Q_p

- **Estado futuro:** El estado futuro es el estado que el sistema tomará después de algún cambio en las entradas o en el estado actual. Este estado se determina según las reglas de transición definidas por el diseño del sistema. Su símbolo es Q_n
- **Elemento de memoria asíncrono:** Un elemento de memoria es asíncrono si el cambio del estado depende exclusivamente de sus entradas y no tiene una señal de control maestra.
- **Elemento de memoria síncrono:** Un elemento de memoria síncrono es un tipo de elemento de memoria cuyo estado cambia de manera síncrona con una señal de reloj. El cambio de estado ocurre en momentos específicos determinados por la señal de reloj y sus entradas.

Elementos de memoria asíncronos

- Este circuito es conocido como el candado SR, donde S significa puesta (set) y R significa borrado (reset).
- De tal forma que cuando se activa S, la salida es forzada a 1 lógico y cuando R es activa, la salida es forzada a 0 lógico.
- Cuando las entradas sean iguales a 11, el próximo estado será igual al estado presente.
- La condición de las entradas en 00 es un estado no permitido en la operación normal del candado.
- * Indica que no está permitido.

R	S		
0	0	1*	1*
0	1	1	0
1	0	0	1
1	1		

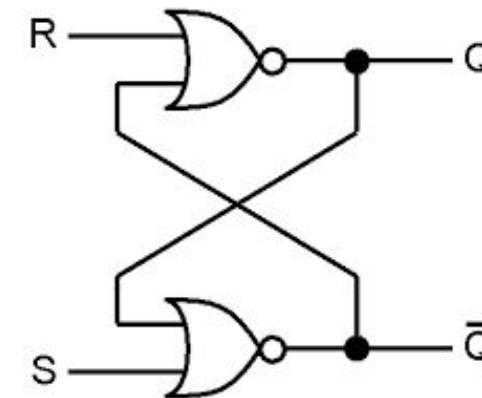


- Candado SR con compuertas NAND

Elementos de memoria asíncronos

- Otra versión del candado SR puede ser obtenida utilizando compuertas NOR en lugar de compuertas NAND, tal como se muestra en la figura.

R	S		
0	0		
0	1	0	1
1	0	1	0
1	1	0*	0*

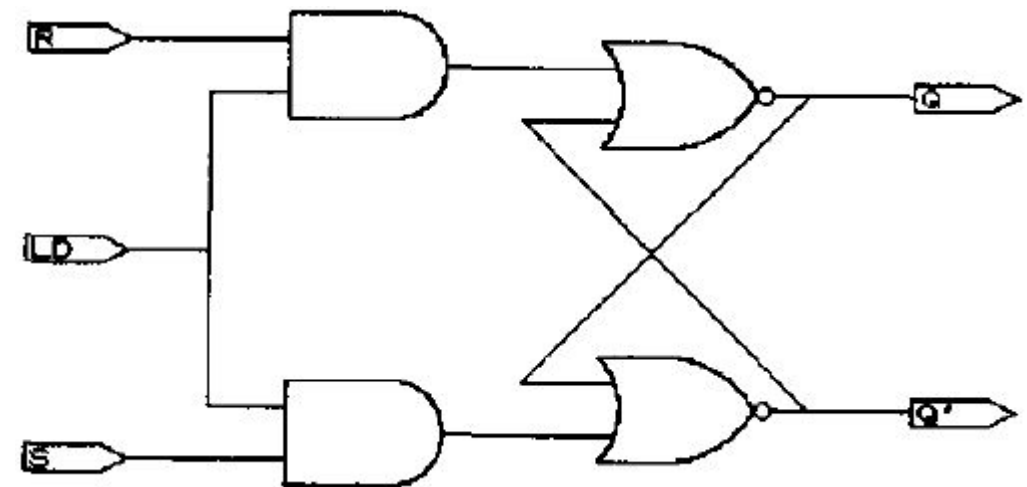


- Candado SR con compuertas NOR

Elementos de memoria síncronos

- Para añadir una señal de sincronía al candado SR es necesario agregar un par de compuertas, tal como se muestra en la figura.
- Cuando la señal LD del circuito es igual a 0, la entrada al candado es 00 lo cual, de acuerdo con la tabla, hace que se conserve el estado anterior, independientemente de los valores de S y R. Cuando LD es 1, entonces se permite el paso de S y R al candado el cual opera de manera normal.

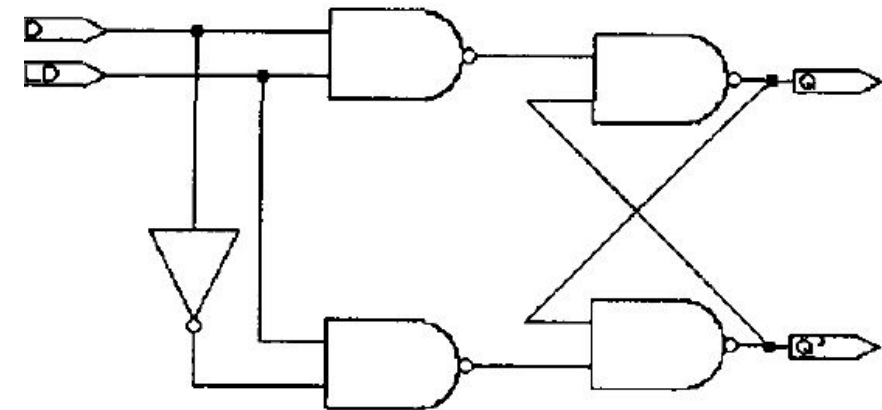
E	S	R		
0	X	X		
1	0	0		
1	0	1	0	1
1	1	0	1	0
1	1	1	0*	0*



- Candado SR síncrono

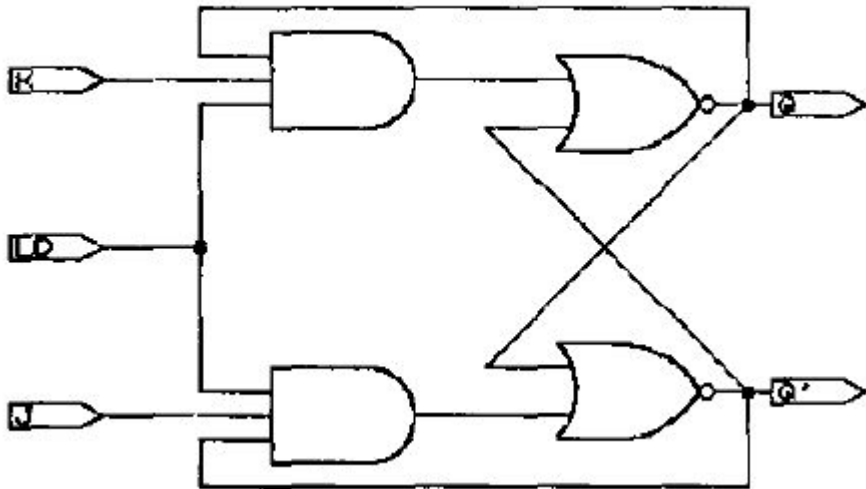
- Además del candado SR también existen otros tipos de candados cuyo funcionamiento es diferente y tienen aplicación en diversos problemas.
- Los candados más populares son los tipos: T, JK y D.

CLK	D		
0	X		
1	0		
1	1	1	0



- Candado D síncrono

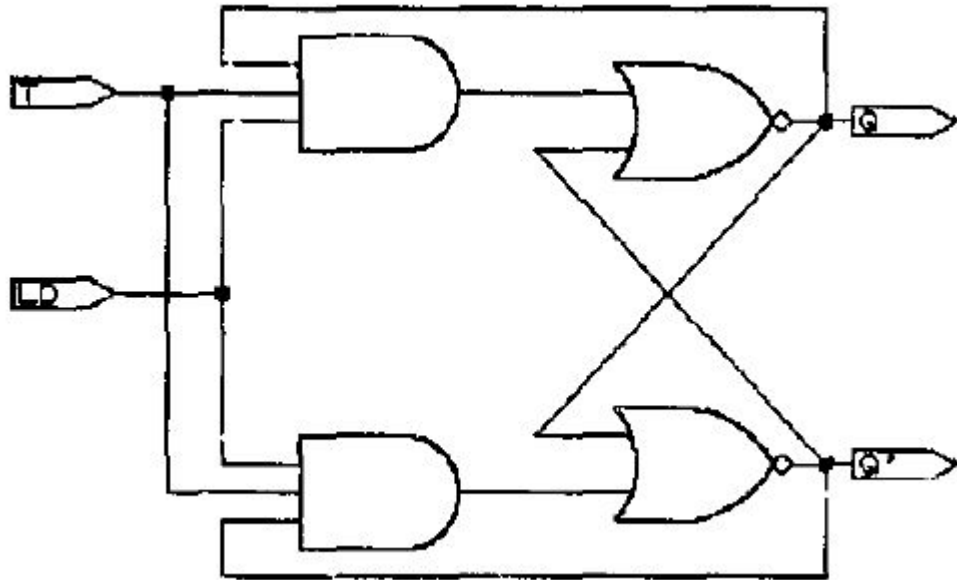
Elementos de memoria síncronos



- Candado JK síncrono

CLK	J	K		
0	X	X		
1	0	0		
1	0	1	0	1
1	1	0	1	0
1	1	1		

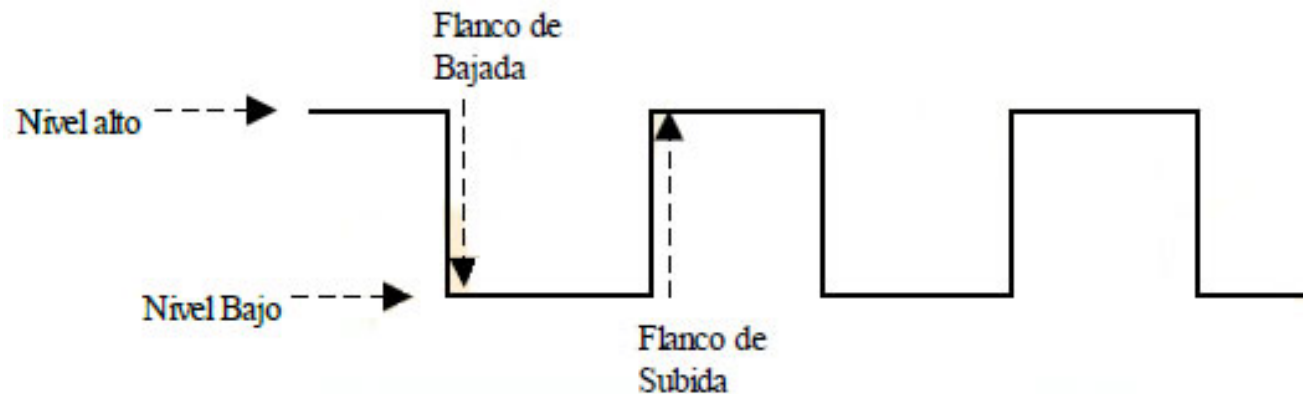
Elementos de memoria síncronos



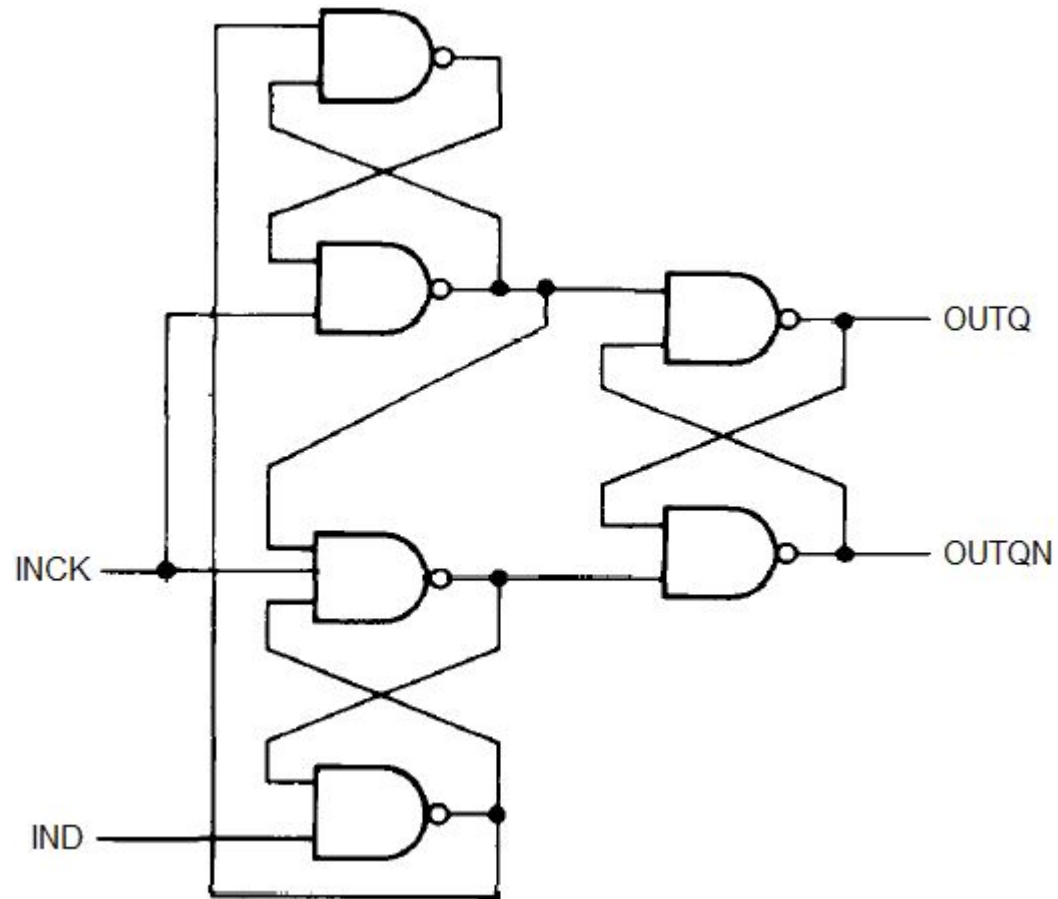
- Candado T síncrono

CLK	D		
0	X		
1	0		
1	1		

- El hecho de que las entradas de un candado síncrono puedan afectar el funcionamiento de este, mientras la señal de sincronía este activa puede ocasionar problemas en circuitos con retroalimentación, produciendo estados transitorios no deseados. La técnica más generalizada para evitar este problema es añadir circuitos extra a los candados para que la señal de sincronización se active solamente por un corto periodo de tiempo. Por ejemplo, en una transición de estados denomina flanco o borde de disparo.
- Los circuitos secuenciales activos al borde se llaman flip-flop.



Elementos de memoria síncronos

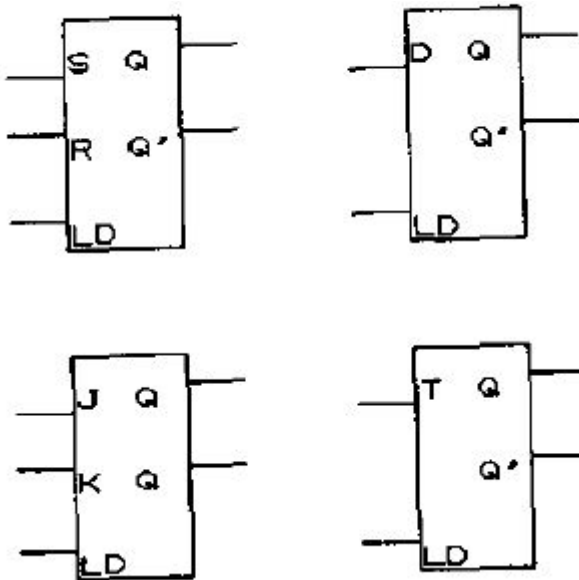


- Flip-flop tipo D

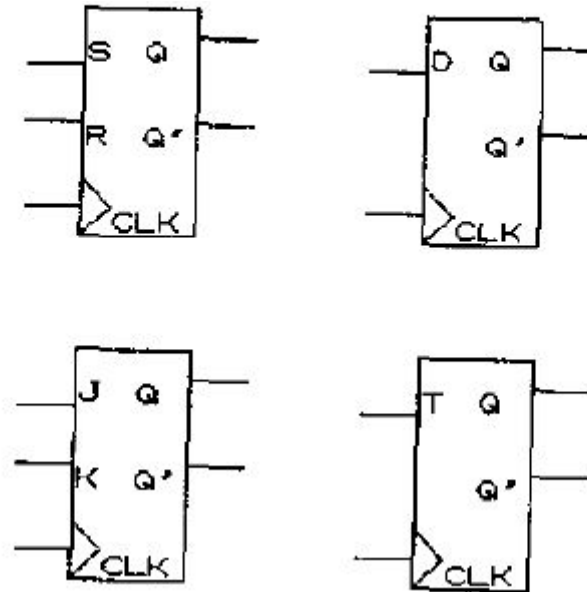
CLK	IND		
0	x		
1	x		
	0	0	1
	1	1	0

Elementos de memoria síncronos

Símbolos de los candados



Símbolos de los flip-flops



```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity Candado is
5      port(
6          D  : in  std_logic; -- Entrada de datos
7          LD : in  std_logic; -- Carga de datos
8          Q  : out std_logic  -- Salida
9      );
10     end Candado;
11
12     architecture Simple of Candado is
13     begin
14         process(D, LD)
15         begin
16             if (LD='1') then
17                 Q <= D;
18             end if;
19         end process;
20     end Simple;
```

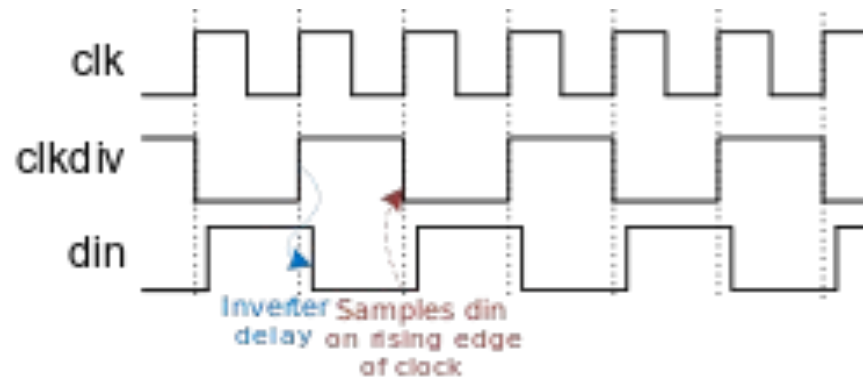
CLK	D		
0	X		
1	0		
1	1		

Elementos de memoria síncronos

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity Candado_RST is
5      port(
6          D : in  std_logic; -- Entrada de datos
7          LD : in  std_logic; -- Carga de datos
8          RST: in  std_logic; -- Borrado
9          Q : out std_logic  -- Salida
10     );
11  end Candado_RST;
12
13  architecture Simple of Candado_RST is
14  begin
15      process(D,LD,RST)
16      begin
17          if (RST='0') then
18              Q <= '0';
19          elsif (LD='1') then
20              Q <= D;
21          end if;
22      end process;
23  end Simple;
```

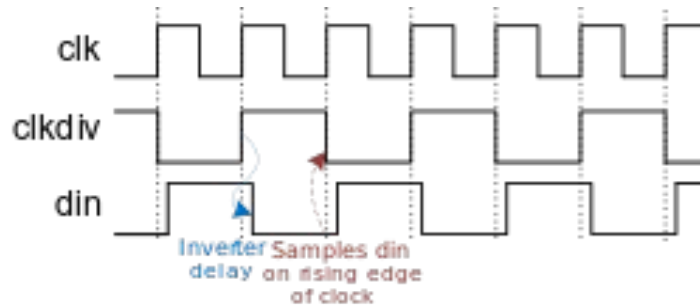
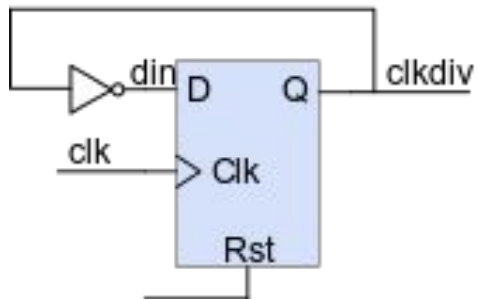
CLK	D		
0	X		
1	0		
1	1		

- Flip-Flops: Un método común es usar flip-flops tipo D o tipo T en una configuración en cascada. Cada flip-flop divide la frecuencia de la señal de entrada por dos, y encadenar N flip-flops resultará en una división de la frecuencia por
- Contadores: Los contadores digitales también se utilizan como divisores de frecuencia. Un contador puede contar hasta un valor predeterminado y luego reiniciarse, y la señal de reloj se divide basándose en el valor al que cuenta el contador.

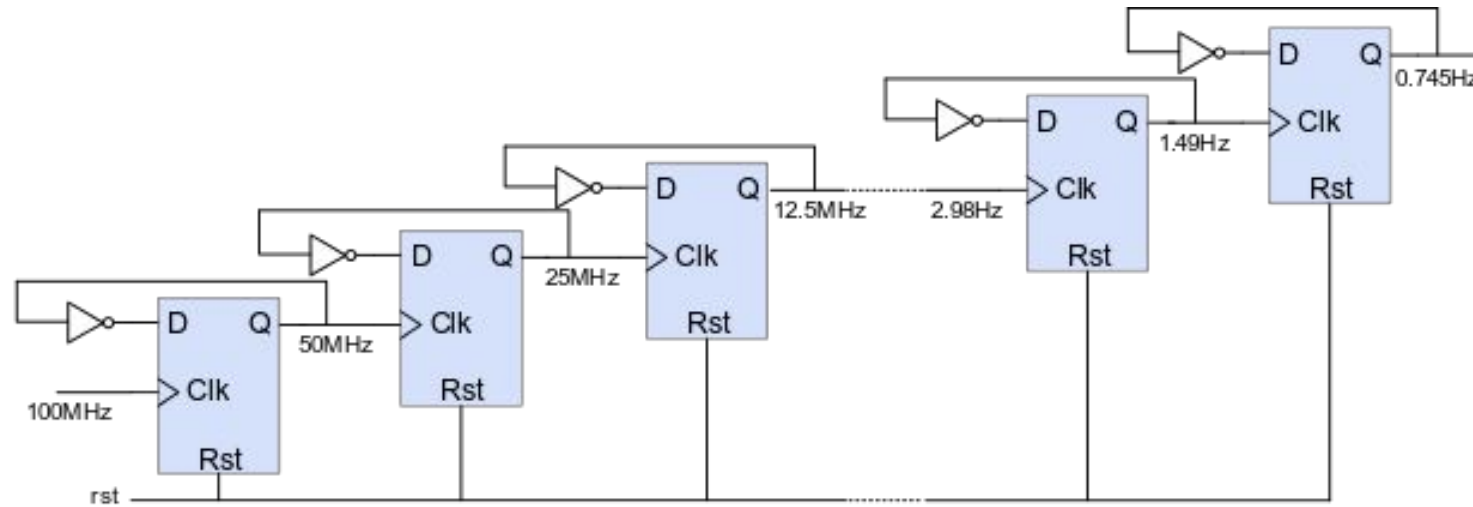




- El propósito de un divisor de frecuencia es generar una señal de reloj a una velocidad más lenta a partir de una señal de reloj más rápida. Esto es útil en una amplia gama de aplicaciones, incluyendo:
 - Sincronización de diferentes partes de un sistema digital que operan a distintas velocidades.
 - Generación de tasas de muestreo específicas para la conversión analógica a digital (ADC) o la conversión digital a analógica (DAC).
 - Creación de señales de temporización para la interfaz de usuario, como parpadeo de LEDs o tasas de refresco de pantalla.

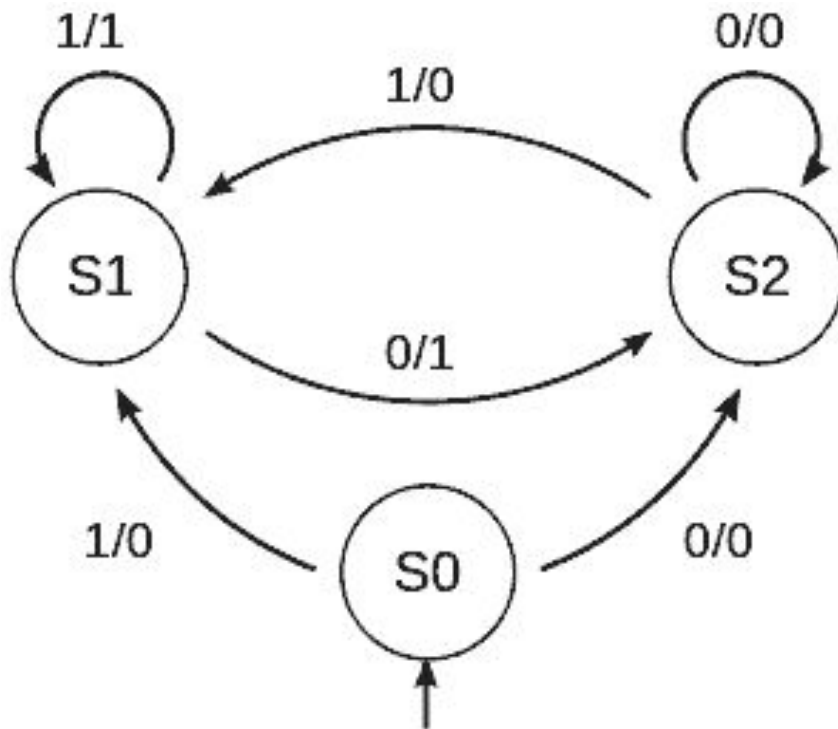


- Flip-Flops: Un método común es usar flip-flops tipo D o tipo T en una configuración en cascada. Cada flip-flop divide la frecuencia de la señal de entrada por dos, y encadenar N flip-flops resultará en una división de la frecuencia por 2^N
- Contadores: Los contadores digitales también se utilizan como divisores de frecuencia. Un contador puede contar hasta un valor predeterminado y luego reiniciarse, y la señal de reloj se divide basándose en el valor al que cuenta el contador.



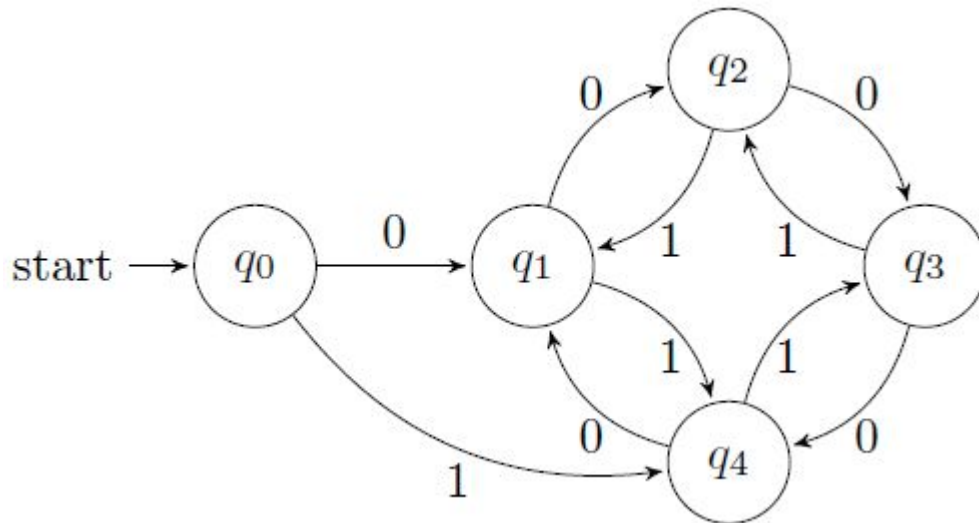
- Representar los candados mediante compuertas lógicas y cargarlos a la FPGA.
- Diseñar un divisor de reloj mediante flip-flops tipo D para obtener 100kHz.
 - Validar con un osciloscopio.
 - Se debe usar componentes.

Máquina de estados finitos



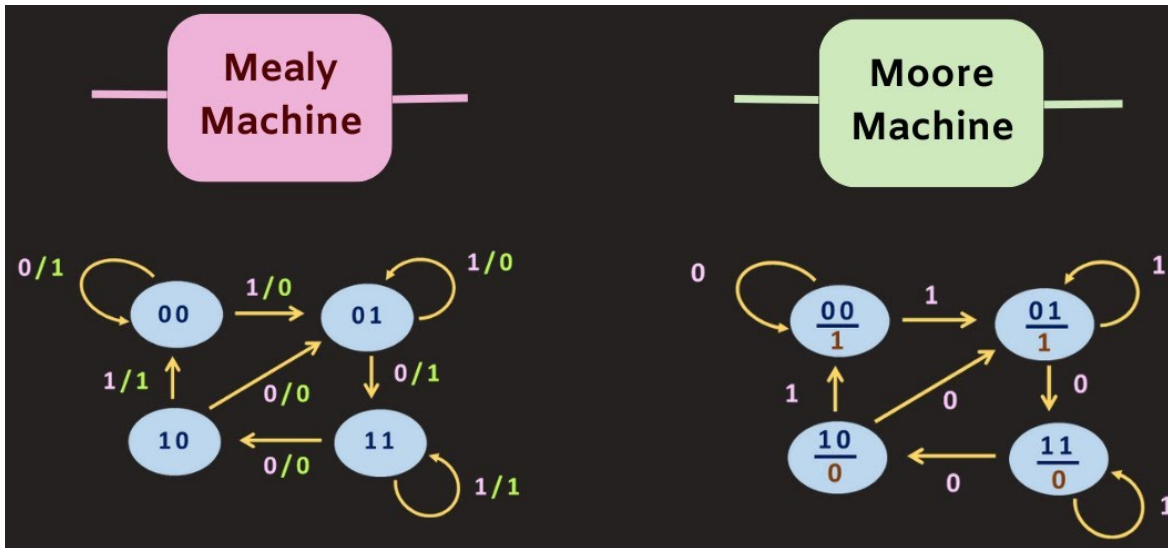
- Todo circuito secuencial puede ser descrito como una máquina de estado finito o FMS. Se dice que es de estados finitos porque el número combinaciones lógicas posibles (estados) que contiene es una cantidad finita y es la característica principal de los circuitos secuenciales.
- Mientras que la lógica combinacional se utiliza tablas de verdad, la lógica secuencial utiliza tabla de transiciones y el grafo.
- El grafo de una FMS es la gráfica del comportamiento de la maquina y contiene: Estados, Transiciones, Entradas y salidas.

Máquina de estados finitos



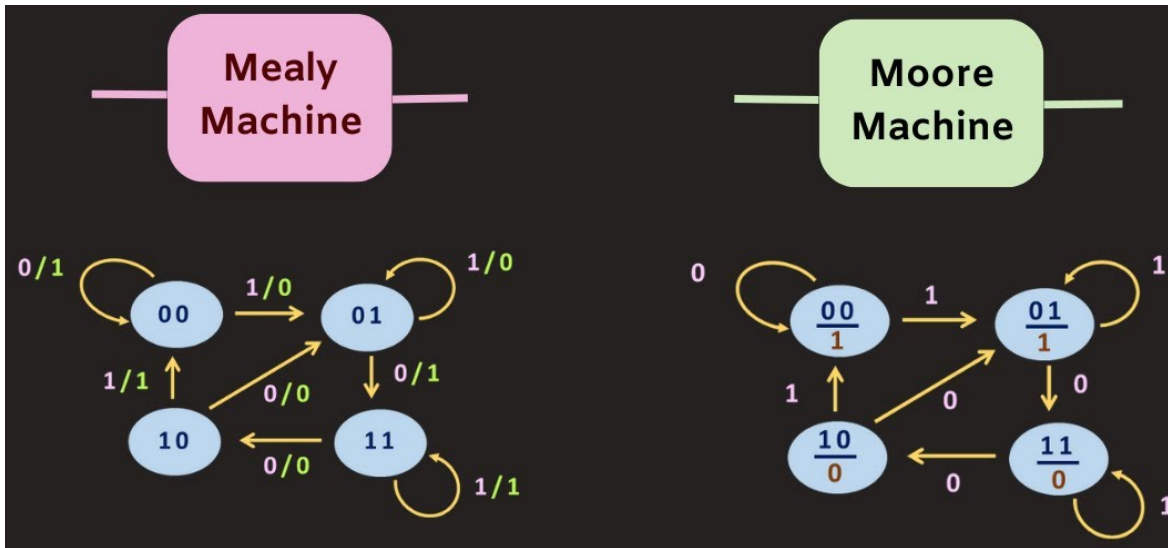
- Dentro del grafo los estados se representan con circunferencias etiquetadas con el nombre del estado.
- Las transiciones son la forma en que se dan los cambios de estado. Se simbolizan con flechas que parten del estado presente y llegan al próximo estado.
- Las entradas de una FMS son las que establecen las condiciones para realizar las transiciones y se representan sobre la transición con su valor lógico.

Máquina de estados finitos



- La salida de la máquina de estados define dos aspectos de esta, el tipo de maquina y las salidas propias.
- Por la forma en que se definen as salidas la FMS puede ser:
 - Máquina de Mealy
 - Máquina de More

Máquina de estados finitos

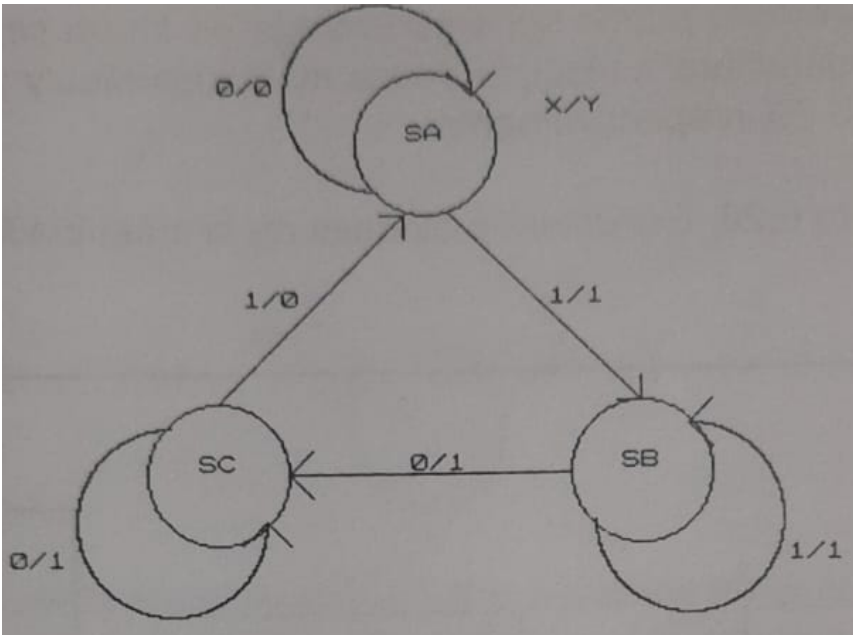


- En la máquina de Mealy, las salidas son funciones de las entradas y de los estados presentes, por lo tanto, se indica a un lado de las entradas, separadas por una diagonal.
- En la maquina Moore las salidas son función exclusiva de los estados presentes y se indica dentro del mismo estado, debajo de la etiqueta.
- Desde un punto de vista teórico ambas maquinas son equivalentes. Sin embargo, en la práctica las FMS no tienen el mismo comportamiento debido a los retardos de los elementos.



- En la máquina de Mealy, se pueden presentar señales transitorias no deseadas en la salida, debido a que cualquier cambio en la entrada las afecta.
- Mientras que las FMS Moore los cambios en las salidas solamente atienden a los cambios de estado y las entradas no tiene una conexión directa a la salida.
- En general las maquinas Moore son más rápidas que las maquinas Mealy.
- Además de la velocidad las maquinas Mealy presentan problemas de sincronía.

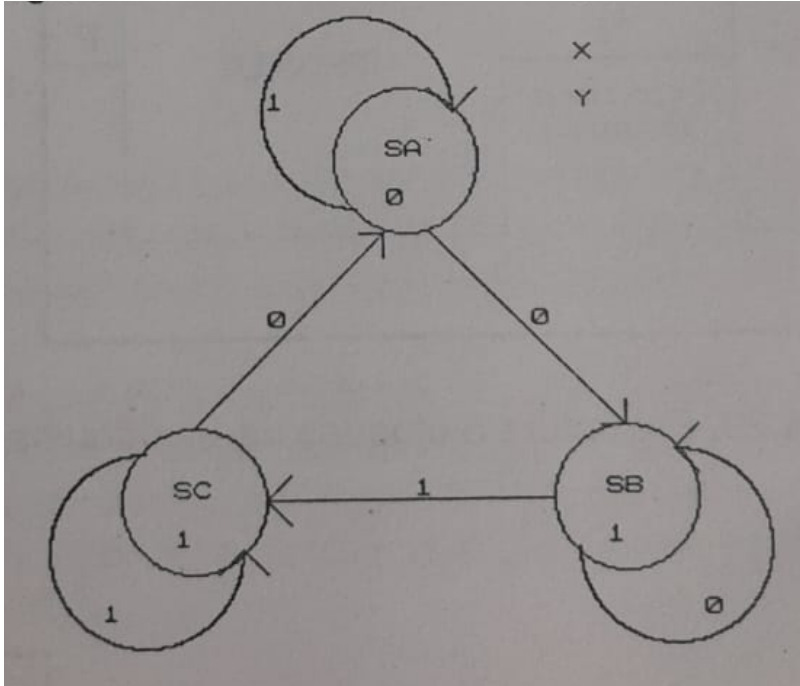
Máquina de estados finitos



- Máquina Mealy

- En la máquina de Mealy puede leerse como: En el estado presente SA si la entrada X es 0, el próximo estado será SA con salida 0.
- Si la entrada X es 1, el próximo estado será SB con salida 1.
- En el estado presente SB si la entrada X es 0, el próximo estado será SC con salida 1.
- Si la entrada X es 1 el próximo estado será SB con salida 1.
- En el estado presente SC cuando la entrada X es 0, el próximo estado será SC con salida 1.
- Si la entrada X es 1 el próximo estado será SA con salida 0.

Máquina de estados finitos



- Máquina Moore

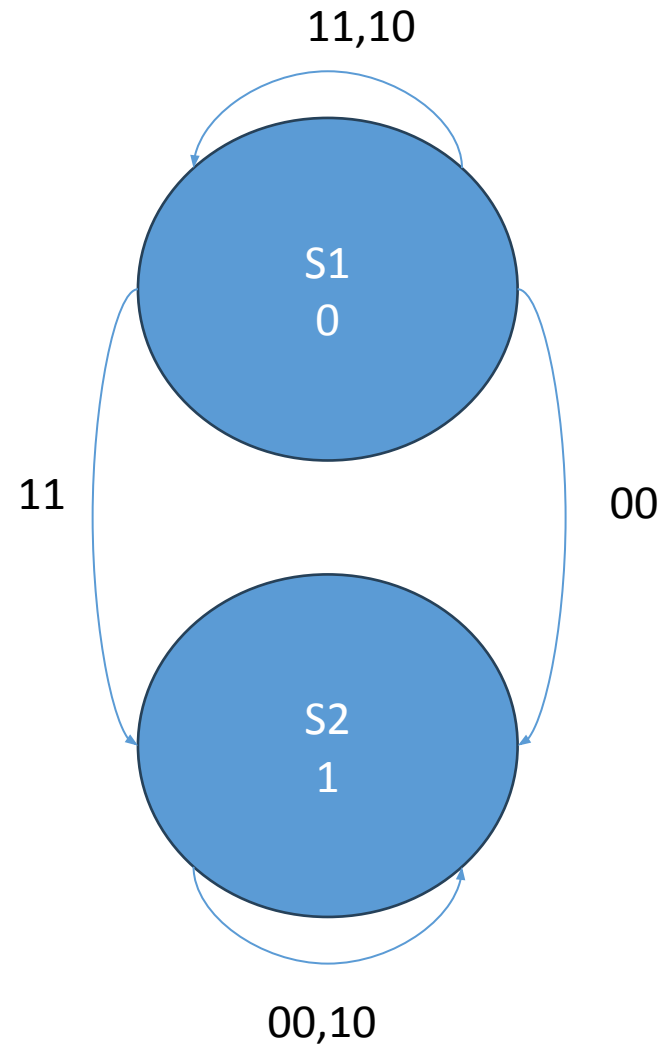
- En la máquina de Moore puede leerse como:
 - En el estado presente SA la salida es 0, si la entrada X es 0 el próximo estado será SB, si la entrada X es 1 el próximo estado sea SA.
 - En el estado presente SB la salida es 1, si la entrada X es 0, el próximo estado será SB, si la entrada X es 1 el próximo estado será SC.
 - En el estado presente SC la salida es 1, si la entrada X es 1 el próximo estado será SC, si la entrada X es 0 el próximo estado será SA.

- El proceso de diseño de máquinas secuenciales requiere de varios pasos. El proceso completo se puede resumir en:
 - Problema
 - Planteamiento del grafo
 - Síntesis síncrona o asíncrona (La tecnología actual esta orientada a la síntesis de FMS síncronas)
 - Realización
- Partiendo de un problema enunciado verbalmente, se procede a plantear el grafo que resuelve el problema en cuestión.
- La descripción en VHDL de FMS es muy simple debido a que VHDL contiene los elementos necesarios para la implementación directa.

- Problema: Controlar el llenado de un depósito de líquido mediante una bomba de tal manera que cuando el tanque se encuentre completamente vacío la bomba se encienda y llene el depósito hasta un nivel superior determinado. Una vez alcanza el límite superior, la bomba se apaga y no se vuelve a encender hasta que se haya vaciado el depósito nuevamente.
- Solución:
 - Paso 1 entender el problema.
 - Paso 2 Elegir la arquitectura de la máquina. Debido a las ventajas se implementará una maquina Moore.
 - Paso 3 Identificar las entradas: El depósito esta debajo del nivel mínimo (B) y el depósito está lleno (T). Definiremos ambas entradas como cero lógicos cuando no detectan la presencia de liquido y cuando 1 cuando detectan la presencia de líquido.

- Paso 4 Identificar las salidas: En el caso del problema la única salida será M, que activa o desactiva la bomba.
- Paso 5 Definir los estados necesarios para realizar el control.
 - Primero definimos un estado S1 para identificar el depósito totalmente lleno y por lo tanto el motor esta apagado ($M = 0$).
 - El sistema debe permanecer en S1 mientras el depósito siga lleno ($B=1, T=1$) cuando el depósito se esté vaciando ($B=1, T=1$).
 - Cuando el depósito este vacío ($B=0, T=0$) se debe cambiar a un estado S2 donde el motor inicie su operación. El estado S2 indica que el depósito se ha vaciado y se activa la bomba.
 - La bomba se queda encendida mientras el depósito siga vacío ($B=0, T=0$) o cuando se esté llenando ($B=1, T=0$).
 - Hasta que el depósito se llene nuevamente ($B=1, T=1$) se produce un cambio de estado

Máquina de estados finitos



- Paso 7 analizar que la FMS este completa: En este caso la maquina es completa y solo existe una condición de entrada que no se ha especificado ($B=0$ y $T=1$). Sin embargo, esta condición no se puede presentar en la realidad.

- Diseñar una máquina expendedora . El producto cuesta \$3.00 y acepta monedas de \$1.00 y \$2.00. La máquina entrega cambio en monedas de \$1.00. Suponga que solamente se puede introducir una moneda a la vez y que la maquina tiene monedas ilimitadas para dar cambio. También suponga que la señal de reloj tiene una frecuencia mucho mayor que la velocidad con las que se introducen las monedas.