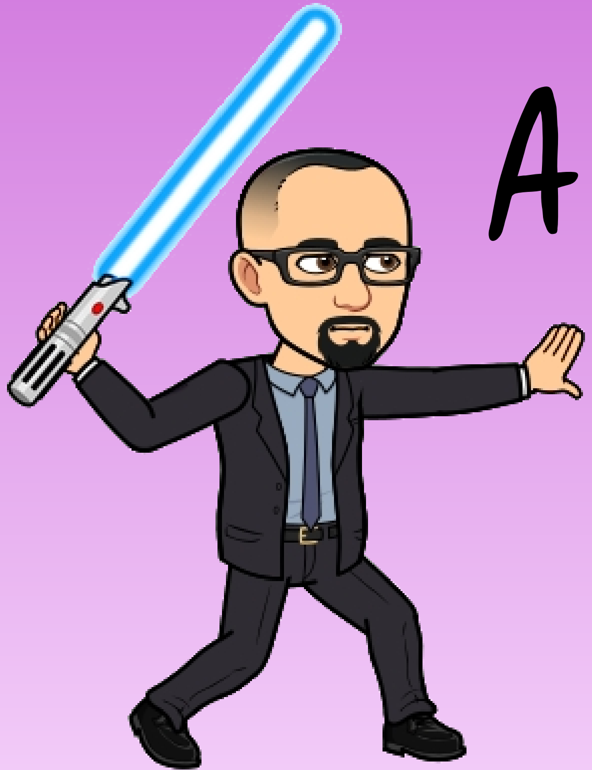
Circuitos Secuenciales

Autómatas

Apunte N° 8



Autómatas

En el desarrollo de sistemas digitales combinacionales las salidas del circuito en un momento dado dependen únicamente de los valores de las entradas en ese momento. En un sistema secuencial además del valor de las entradas en un instante dado hay que tener en cuenta también el estado anterior por el que ha pasado el sistema, por lo que ha estos circuitos se les suele llamar sistemas con memoria. Para implementar esa memoria se suele utilizar biestables (Flip-Flop) ó memorias ROM.

Un autómata es una máquina secuencial síncrona (controlada por una señal de reloj) que se puede encontrar en uno de entre un número posible de estados, recibe una serie de entradas binarias y en función de estas entradas y del estado particular en el que se encuentra, genera una o varias salidas binarias determinadas. Se le llama finito por que el número de estados en el que puede encontrarse el autómata tiene que quedar perfectamente determinado, de ahí que a estos sistemas se les llame también deterministas. A este tipo de autómatas se los suele llamar también máquina de estado finito (FSM: finite state machine).

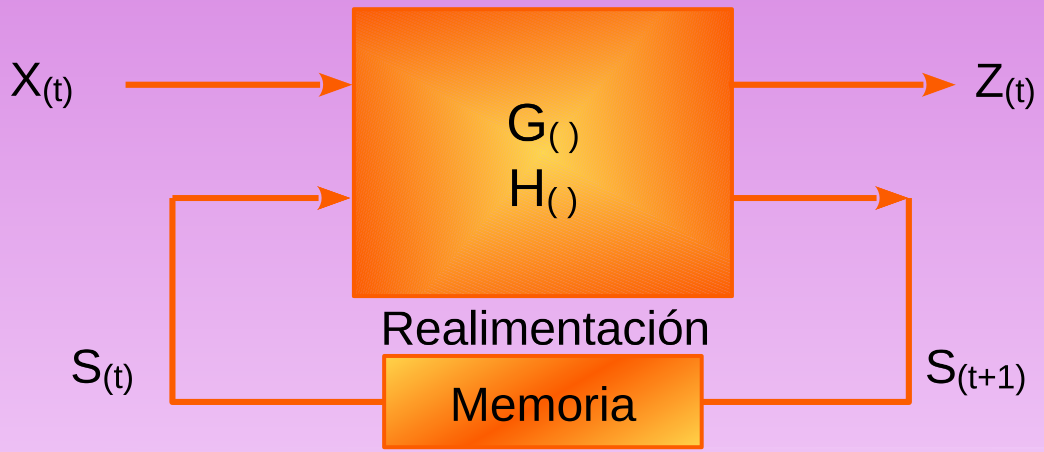
El desarrollo de circuitos con autómatas secuenciales es una herramienta muy potente y cuya comprensión abre el camino a otras técnicas de desarrollo como son por ejemplo en los Microcontroladores los RTOS (Sistemas Operativos en Tiempo Real).

Básicamente existen dos tipos de autómatas finitos: el autómata de Mealy y el autómata de Moore.

Máquinas de estado finito

En los sistemas secuenciales la salida Z en un determinado instante de tiempo t_i depende de X (la entrada) en ese mismo instante de tiempo t_i y en todos los instantes temporales anteriores. Para ello es necesario que el sistema disponga de elementos de memoria que le permitan recordar la situación en que se encuentra (estado).

Como un sistema secuencial es finito, tiene una capacidad de memoria finita y un conjunto finito de estados posibles lo que define a las máquina de estado finito (FSM: finite state machine)



$X_{(t)}$: entrada actual
$Z_{(t)}$: salida actual
$S_{(t)}$: estado actual
$S_{(t+1)}$: estado próximo
$G_{()}$: función de salida
$H_{()}$: función de transición

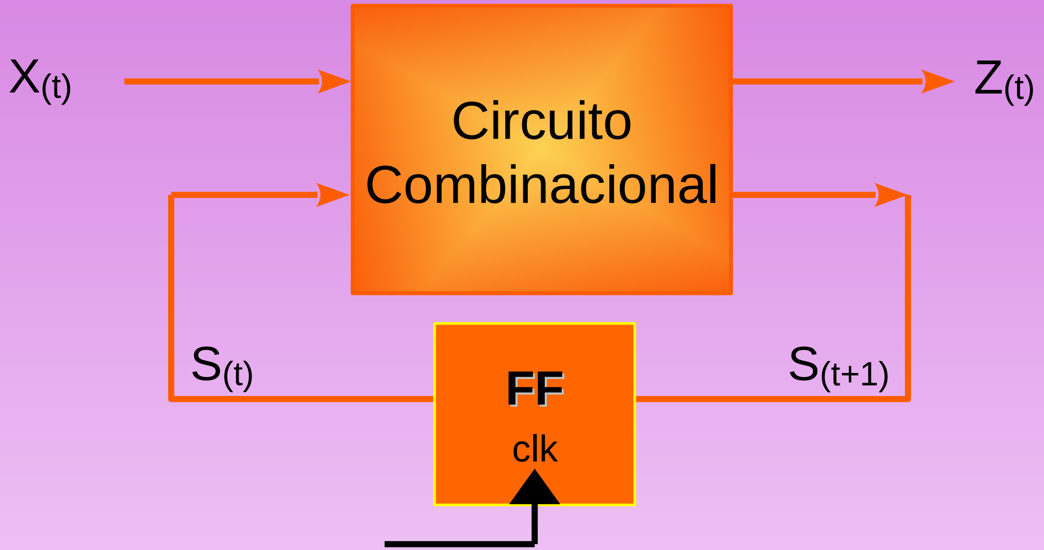
$Z_{(t)}=G(X_{(t)},S_{(t)})$
$S_{(t+1)}=H(X_{(t)},S_{(t)})$

Máquinas de estado finito

- Partiendo de un estado determinado, las máquinas de estados deterministas generan siempre la misma secuencia de salida para la misma secuencia de entrada.
- Dos máquinas de estados son equivalentes si generan las mismas secuencias de salida para las mismas secuencias de entrada.
- Las máquinas de estados se pueden optimizar: máquinas equivalentes con menor número de estados.
- El estado cambia según la secuencia de entrada, por lo que representa el conjunto de entradas pasadas.
- Las máquinas de estados pueden ser incompletamente especificadas: próximo estado no definido para un estado actual y entrada dados.
- Los circuitos secuenciales síncronos implementan máquinas de estados finitos empleando funciones combinacionales y biestables.
- El cambio de estado se controla mediante una señal de reloj. Ej: biestables disparados por flanco.

Máquinas de Mealy

- El estado siguiente depende de la entrada y del estado actual.
- La salida depende de la entrada y del estado actual.
- Esto implica que un autómata de Mealy, estando en un determinado estado, puede evolucionar hacia estados siguientes distintos y producir salidas distintas si se introduce una ó varias entradas binarias distintas.

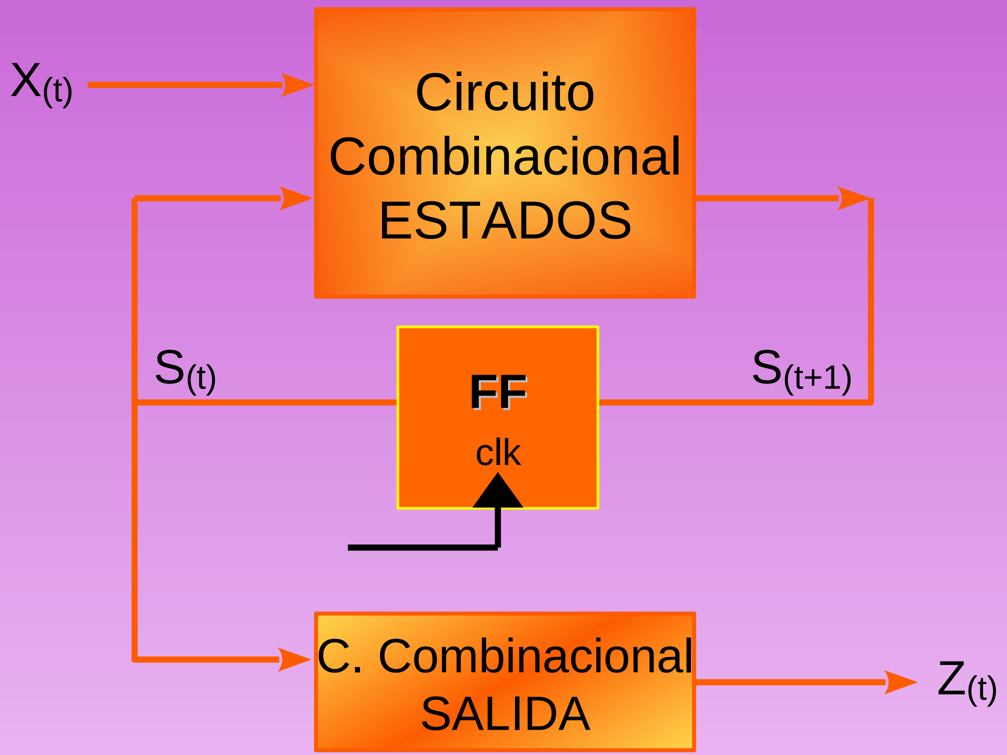


Máquinas de Moore

- El estado siguiente depende de la entrada y del estado actual.
- La salida depende exclusivamente del estado actual.
- Esto implica que un autómata de Moore, estando en un determinado estado, produce siempre la misma salida, independientemente de cuál sea la entrada ó entradas de datos en ese estado.

Toda máquina de Moore es un caso particular de una máquina de Mealy.

Una máquina de Moore suele emplear mas estados internos y, consecuentemente mas Flip-Flops que uno de Mealy. Por el contrario, en una de Mealy el circuito combinacional que da las salidas del sistema suele ser mas complejo.



Metodología

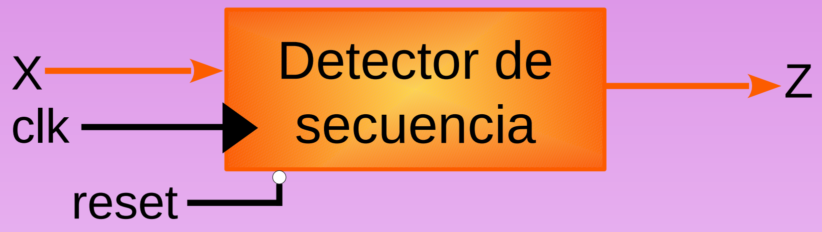
1) Identificación de entradas y salidas
2) Diagrama de transición de estados
3) Comprobación del diagrama
4) Determinación del número de biestables
5) Asignación de estados
6) Tablas de transición de estados, salidas y excitación del autómata
7) Minimización de las funciones lógicas
8) Diseño del circuito

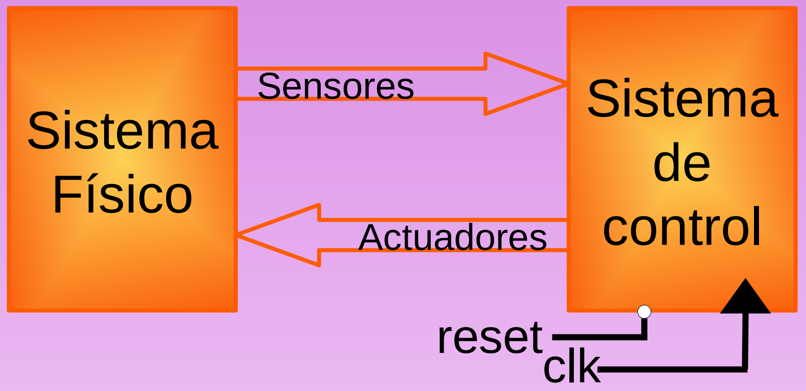
Ejemplo: Detector de secuencia

Diseñar un sistema secuencial con una entrada serie que detecte si los tres últimos datos recibidos coinciden con la secuencia 1 0 0

1) Identificación de entradas y salidas

- **Determinar las señales que entran y salen del circuito que se quiere diseñar**
- **El reloj y el reset deben ir siempre, y no se consideran**
- **En los sistemas de control, los sensores son entradas al circuito y los actuadores son las salidas.**



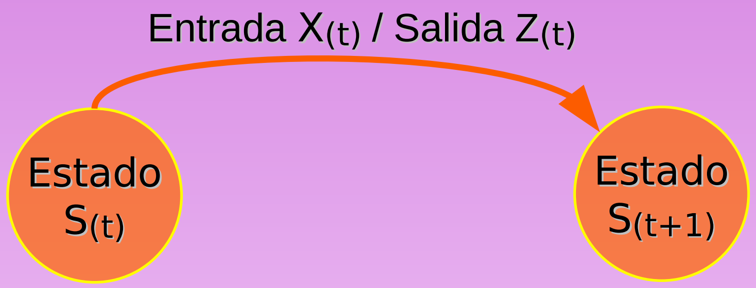


Ejemplo: Detector de secuencia

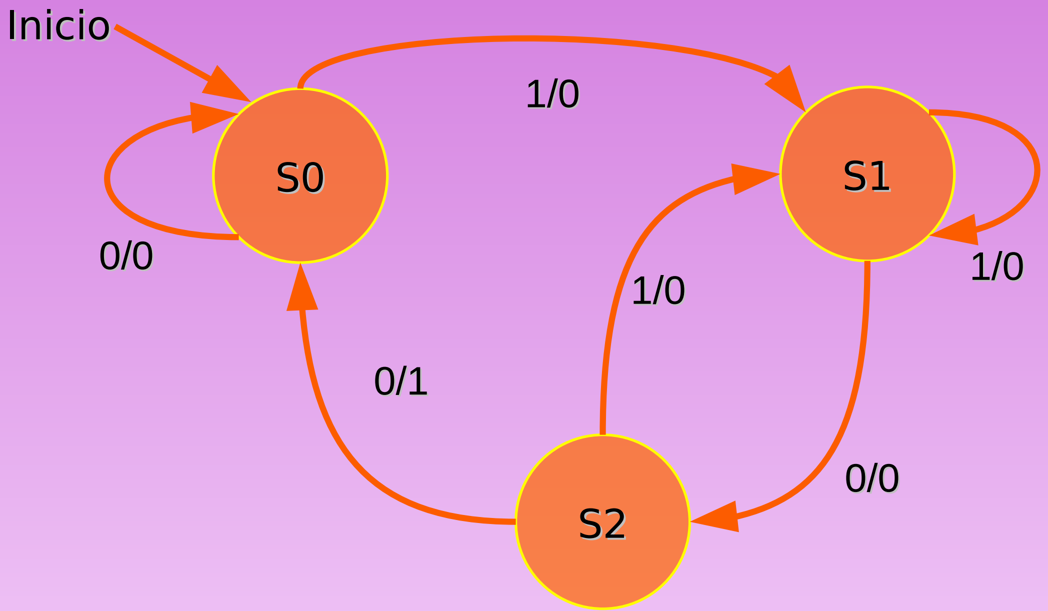
2) Diagrama de transición de estados para FSM de Mealy

- Comenzar por el estado de reposo o inicio (al encender el sistema)
- De cada estado deben salir 2^E transiciones (E = nro de entradas)
- En los sistemas físicos se considera la señal de reloj mucho más rápida que las señales de entrada

FSM de Mealy



La salida asociada a la transición



Ejemplo: Detector de secuencia

3) Comprobación del diagrama

- De cada estado deben salir 2^E transiciones (E = nro de entradas), salvo aquellas que sean imposibles
- Todas las transiciones tiene que tener valores de entrada distintos

4) Determinación del número de biestables

- N estados ⇒ n biestables tal que $2^n \geq N$
- Ejemplo (detector de secuencia 100): 3 estados ⇒ 2 biestables

5) Asignación de estados

- A cada estado se le asigna una combinación de valores de los biestables

Estado	Q1	Q0
S0	0	0
S1	0	1
S2	1	0

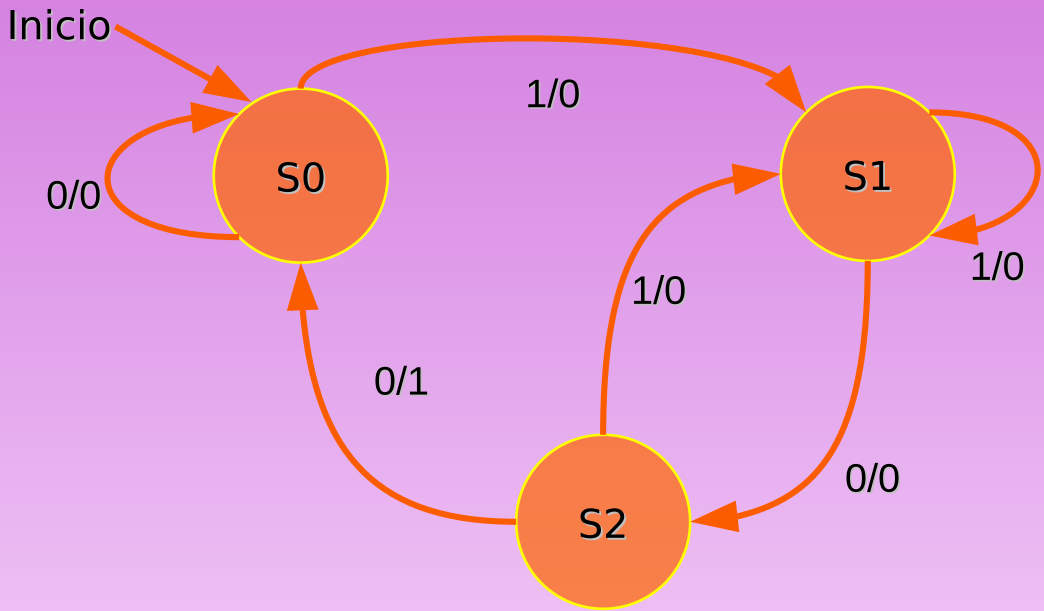
Ejemplo: Detector de secuencia

6) Tablas de transición de estados, salidas y excitación del autómata

Transición de estados
Ecuación de estado:
$S_{(t+1)} = H(S_{(t)}, X_{(t)})$

Salidas del circuito:
Moore: $Z_{(t)} = G(S_{(t)})$
Mealy: $Z_{(t)} = G(S_{(t)}, X_{(t)})$

	Q1	Q0	Entrada X	Q1'	Q0'	Salida Z
S0	0	0	0	0	0	0
	0	0	1	0	1	0
S1	0	1	0	1	0	0
	0	1	1	0	1	0
S2	1	0	0	0	0	1
	1	0	1	0	1	0
	1	1	X	X	X	X
	$\underbrace{}_{S_{(t)}}$		$\underbrace{}_{X_{(t)}}$	$\underbrace{}_{S_{(t+1)}}$		$\underbrace{}_{Z_{(t)}}$



Ejemplo: Detector de secuencia

6) Tablas de excitación del autómata

- Se construye a partir de la tabla de transición de estados.
- ¿Qué valor tiene que tomar el biestable para estando en $Q_{(t)}$ pase al estado $Q_{(t+1)}$?
- La implementación se puede hacer con biestables D, RS, T, JK.
- Basándonos en las tablas de verdad de los biestables se obtienen los valores que deben tomar las entradas para pasar del estado $Q_{(t)}$ al $Q_{(t+1)}$

D	$Q_{(t+1)}$
0	0
1	1

D: facilita el diseño, reduce el número de conexiones.

T	$Q_{(t+1)}$
0	$Q_{(t)}$
1	$\overline{Q_{(t)}}$

T: más conveniente en aplicaciones específicas (contadores)

RS: más simple que el JK pero menos flexible.

R	S	$Q_{(t+1)}$
0	0	$Q_{(t)}$
0	1	1
1	0	0
1	1	-

JK: reduce el coste de la parte combinacional.

JJ	KK	$Q_{(t+1)}$
0	0	$Q_{(t)}$
0	1	0
1	0	1
1	1	$\overline{Q_{(t)}}$

Ejemplo: Detector de secuencia

6) Tablas de excitación del autómata

- Basándonos en las tablas de verdad de los biestables se obtienen los valores que deben tomar las entradas de los biestables para pasar del estado Q(t) al estado Q(t+1)

$Q_{(t)}$	$Q_{(t+1)}$	D
0	0	0
0	1	1
1	0	0
1	1	1

$Q_{(t)}$	$Q_{(t+1)}$	T
0	0	0
0	1	1
1	0	1
1	1	0

$Q_{(t)}$	$Q_{(t+1)}$	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

$Q_{(t)}$	$Q_{(t+1)}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Ejemplo: Detector de secuencia

6) Tablas de excitación del autómata

- Se completa la tabla de transición de estados con los valores necesarios en las entradas de los biestables para realizar la transición de estado.

$\underbrace{\text{Q1}}$	$S_{(t)}$ Q0	Entrada X	$S_{(t+1)}$ Q1'	Q0'	Salida Z	entradas en JK para pasar de $S_{(t)}$ a $S_{(t+1)}$ J1	K1	J0	K0
0	**0**	0	0	**0**	0	0	X	**0**	**X**
0	0	1	0	1	0	0	X	1	X
0	1	0	1	0	0	1	X	X	1
0	1	1	0	1	0	0	X	X	0
1	0	0	0	0	1	X	1	0	X
1	0	1	0	1	0	X	1	1	X
1	1	0	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X

Para pasar de Q0 = 0 → Q0' = 0
Se necesita J0 = 0 y K0 = X

$Q_{(t)}$	$Q_{(t+1)}$	J	K
0	**0**	**0**	**X**
0	1	1	X
1	0	X	1
1	1	X	0

Ejemplo: Detector de secuencia

7) Minimización de las funciones lógicas

- Se obtienen las ecuaciones lógicas de las J, K y las salidas (Z), en función del estado actual (Q1 y Q0) y las entradas (X)
- Para minimizarlas se emplean los mapas de Karnaugh

Q1	Q0	Entrada X	Q1’	Q0’	Salida Z	J1	K1	J0	K0
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	1	0	0	1	X	X	1
0	1	1	0	1	0	0	X	X	0
1	0	0	0	0	1	X	1	0	X
1	0	1	0	1	0	X	1	1	X
1	1	0	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X

\ X Q1Q0\	0	1
00		
01	1	
11	X	X
10	X	X

$J1 = Q0 \cdot \overline{X}$

\ X Q1Q0\	0	1
00		1
01	X	X
11	X	X
10		1

$J0 = X$

\ X Q1Q0\	0	1
00	X	X
01	X	X
11	X	X
10	1	1

$K1 = 1$

\ X Q1Q0\	0	1
00	X	X
01	1	
11	X	X
10	X	X

$K0 = \overline{X}$

\ X Q1Q0\	0	1
00		
01		
11	X	X
10	1	

$Z = Q1 \cdot \overline{X}$

Ejemplo: Detector de secuencia

8) Diseño del circuito

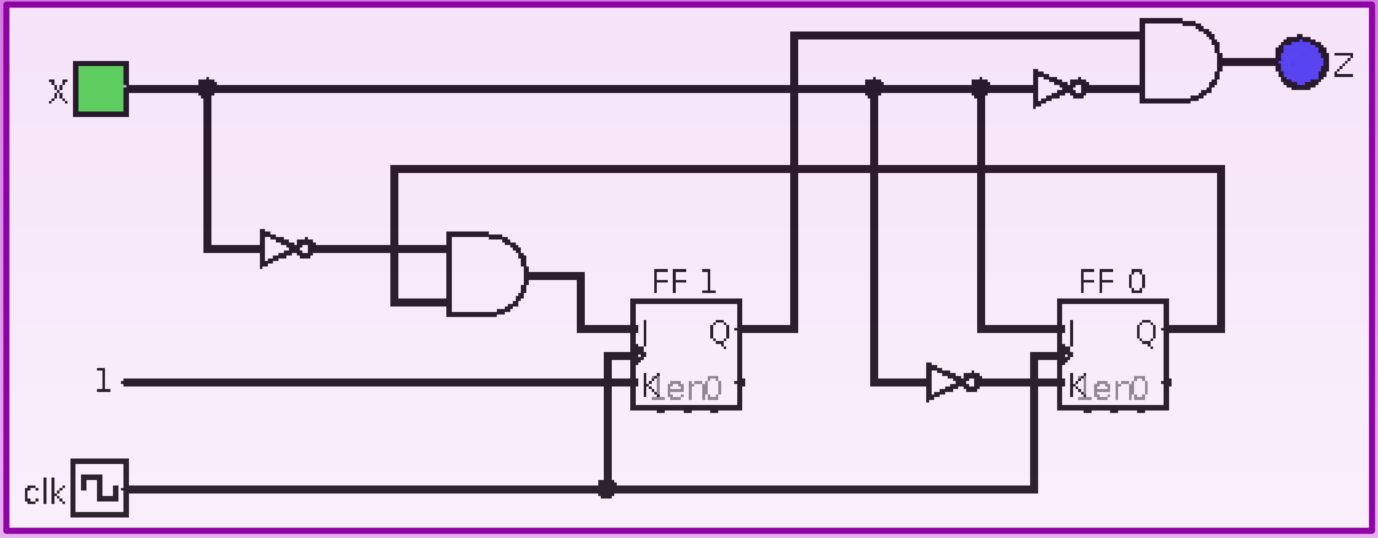
$J1 = Q0 \cdot \overline{X}$

$K1 = 1$

$J0 = X$

$K0 = \overline{X}$

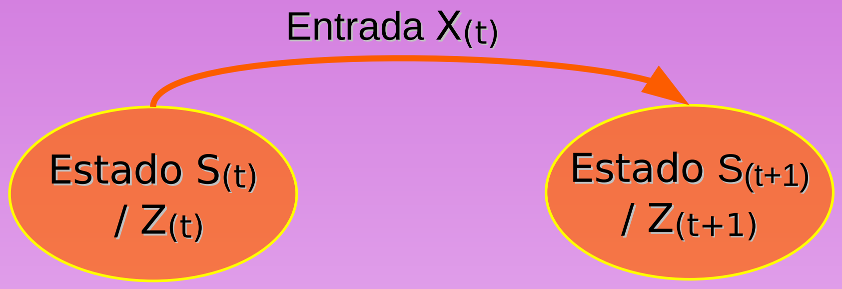
$Z = Q1 \cdot \overline{X}$



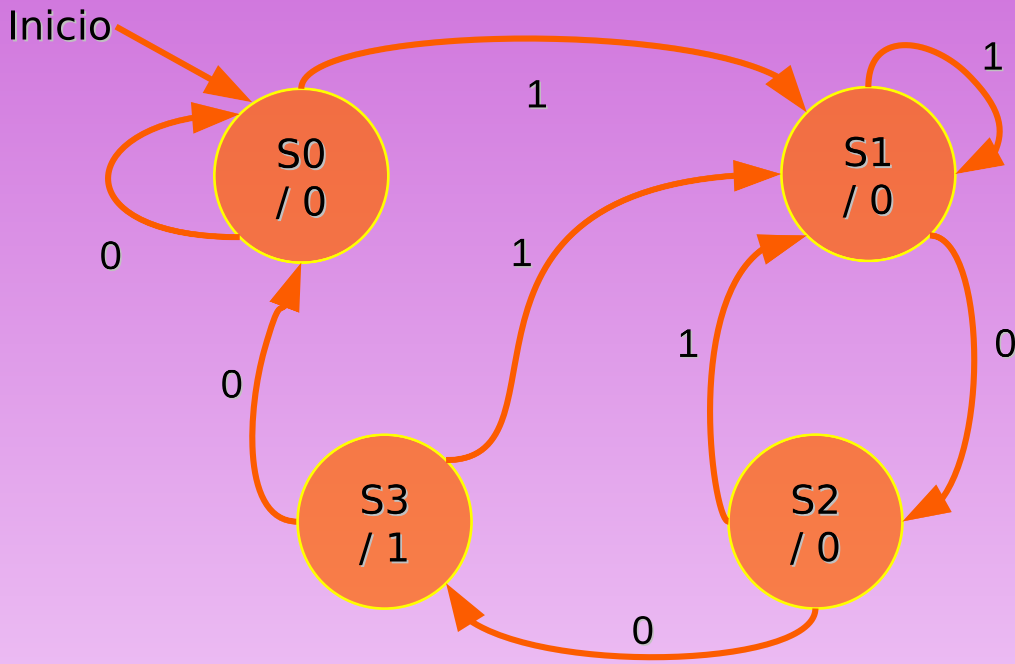
Ejemplo: Detector de secuencia

2) Diagrama de transición de estados para FSM de Moore

FSM de Moore



La salida asociada al estado



Ejemplo: Detector de secuencia

3) Comprobación del diagrama

- De cada estado deben salir 2^E transiciones (E = nro de entradas), salvo aquellas que sean imposibles
- Todas las transiciones tiene que tener valores de entrada distintos

4) Determinación del número de biestables

- N estados ⇒ n biestables tal que $2^n \geq N$
- Ejemplo (detector de secuencia 100): 3 estados ⇒ 2 biestables

5) Asignación de estados

- A cada estado se le asigna una combinación de valores de los biestables

Estado	Q1	Q0
S0	0	0
S1	0	1
S2	1	0
S3	1	1

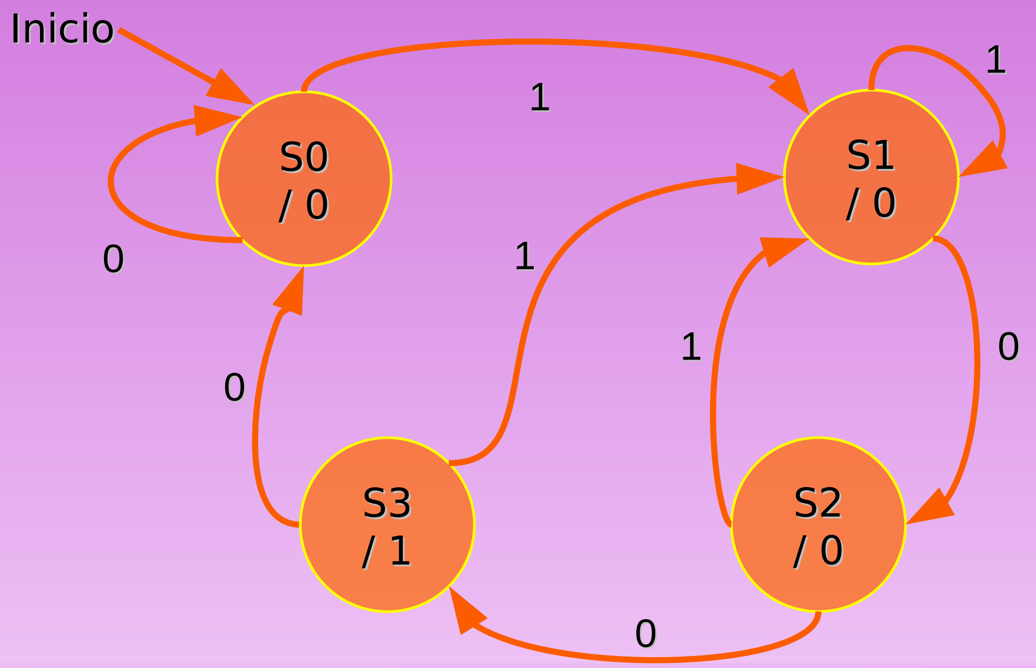
Ejemplo: Detector de secuencia

6) Tablas de transición de estados, salidas y excitación del autómata

Transición de estados
Ecuación de estado:
$S_{(t+1)} = H(S_{(t)}, X_{(t)})$

Salidas del circuito:
Moore: $Z_{(t)} = G(S_{(t)})$
Mealy: $Z_{(t)} = G(S_{(t)}, X_{(t)})$

	Q1	Q0	Entrada X	Q1'	Q0'	Salida Z
S0	0	0	0	0	0	0
	0	0	1	0	1	0
S1	0	1	0	1	0	0
	0	1	1	0	1	0
S2	1	0	0	0	0	1
	1	0	1	0	1	0
S3	1	1	0	0	0	1
	1	1	1	0	1	1
	$S_{(t)}$		$X_{(t)}$	$S_{(t+1)}$		$Z_{(t)}$



Ejemplo: Detector de secuencia

6) Tablas de excitación del autómata

- Se completa la tabla de transición de estados con los valores necesarios en las entradas de los biestables para realizar la transición de estado.

$\underbrace{}_{S_{(t)}}$ $\underbrace{}_{S_{(t+1)}}$ entradas en JK para pasar de $S_{(t)}$ a $S_{(t+1)}$

Q1	Q0	Entrada X	Q1'	Q0'	Salida Z	J1	K1	J0	K0
0	**0**	0	0	**0**	0	0	X	**0**	**X**
0	0	1	0	1	0	0	X	1	X
0	1	0	1	0	0	1	X	X	1
0	1	1	0	1	0	0	X	X	0
1	0	0	1	1	0	X	0	1	X
1	0	1	0	1	0	X	1	1	X
1	1	0	0	0	1	X	1	X	1
1	1	1	0	1	1	X	1	X	0

Para pasar de Q0 = 0 → Q0' = 0
Se necesita J0 = 0 y K0 = X

$Q_{(t)}$	$Q_{(t+1)}$	J	K
0	**0**	**0**	**X**
0	1	1	X
1	0	X	1
1	1	X	0

Ejemplo: Detector de secuencia

7) Minimización de las funciones lógicas

- Se obtienen las ecuaciones lógicas de las J, K y las salidas (Z), en función del estado actual (Q1 y Q0) y las entradas (X)
- Para minimizarlas se emplean los mapas de Karnaugh

Q1	Q0	Entrada X	Q1’	Q0’	Salida Z	J1	K1	J0	K0
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	1	0	0	1	X	X	1
0	1	1	0	1	0	0	X	X	0
1	0	0	1	1	0	X	0	1	X
1	0	1	0	1	0	X	1	1	X
1	1	0	0	0	1	X	1	X	1
1	1	1	0	1	1	X	1	X	0

\ X Q1Q0\	0	1
00		
01	1	
11	X	X
10	X	X

$J1 = Q0 \cdot \overline{X}$

\ X Q1Q0\	0	1
00		1
01	X	X
11	X	X
10	1	1

$J0 = Q1 + X$

\ X Q1Q0\	0	1
00	X	X
01	X	X
11	1	1
10	0	1

$K1 = Q0 + X$

\ X Q1Q0\	0	1
00	X	X
01	1	
11	1	0
10	X	X

$K0 = \overline{X}$

\ X Q1Q0\	0	1
00		
01		
11	1	1
10		

$Z = Q1 \cdot Q0$

Ejemplo: Detector de secuencia

8) Diseño del circuito

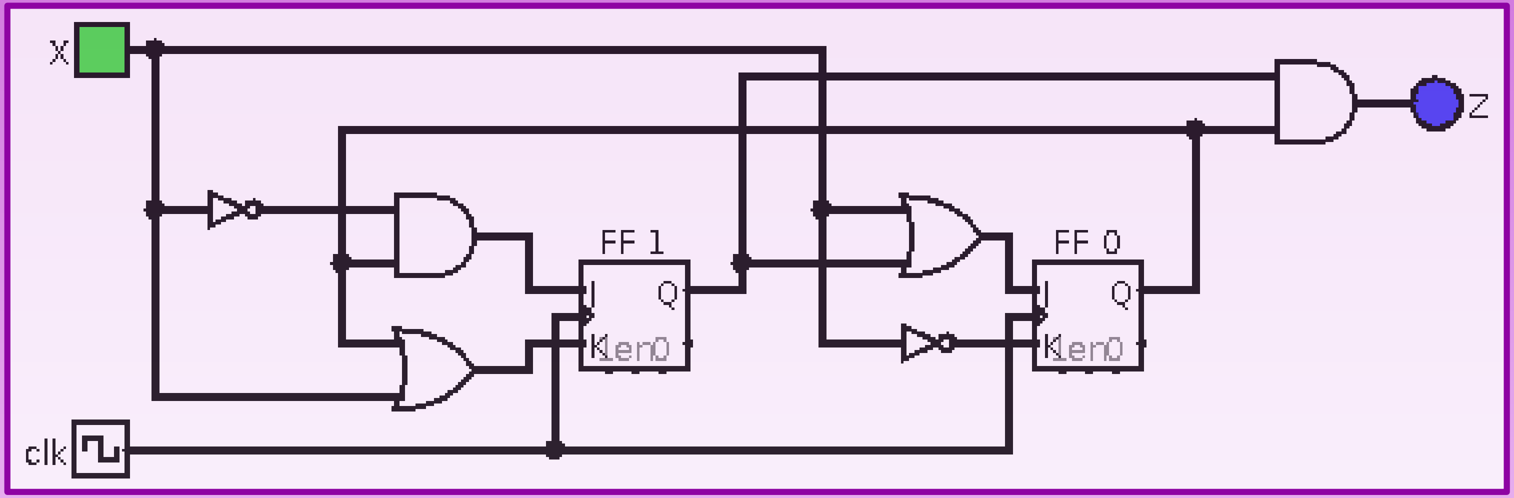
$J1 = Q0 \cdot \overline{X}$

$K1 = Q0 + X$

$J0 = Q1 + X$

$K0 = \overline{X}$

$Z = Q1 \cdot Q0$



Bibliografía y Licencia

- Acha, Santiago, Castro, Manuel, Rioseras, Miguel, “Electrónica Digital Introducción a la Lógica Digital” 2da Ed. (Ra-Ma 2010)
- Floyd, Thomas, “Fundamentos de sistemas digitales” 9na Ed. (Pearson 2006)
- Gonzalez Gomez, Juan,“Circuitos y Sistemas Digitales” (Madrid 2002)

- Autor:

Matías E. García

Prof. & Tec. en Informática Aplicada

www.profmatiasgarcia.com.ar

info@profmatiasgarcia.com.ar