MÁQUINAS DE ESTADOS FINITOS

Arquitectura de Computadoras I
Prof. Marcelo Tosini
2010

INTRODUCCIÓN

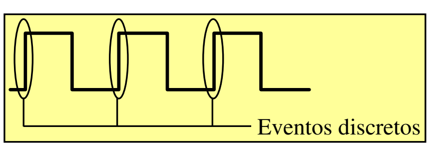
***Evento discreto**: ocurrencia de una característica en la evolución de una señal*

Por ejemplo.

** flanco de subida*

** paso por un cierto nivel*

** pulso, llegada de un dato*



		ESTADO	
		CONTINUO	DISCRETO
TIEMPO	CONTINUO	Sistemas analógicos	Sistemas Asíncronos
	DISCRETO	Sistemas por muestreo	Sistemas Síncronos

INTRODUCCIÓN

***Sistemas discretos**: sistemas que cambian de estado ante la ocurrencia de eventos discretos.*

- *El estado sólo puede adquirir un conjunto discreto de valores y*
- *Puede ser representado de forma simbólica en vez de numérica.*
- Tiempo continuo (sistemas asíncronos)
  - » El estado del sistema puede cambiar en cualquier instante ante la llegada de un evento. Ej.: accionamiento de un interruptor.
- Tiempo discreto (sistemas síncronos)
  - » El estado del sistema sólo cambia cada T seg en función del estado y entradas presentes en esos instantes de tiempo. Evento: señal de reloj.
  - » O bien con un evento de sincronización -> validación

CONCEPTO DE AUTÓMATA.

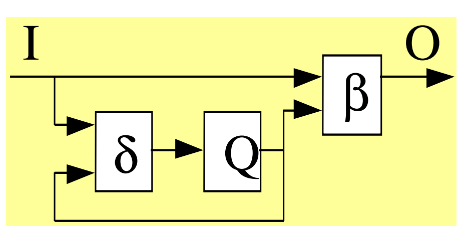
Autómata de MEALY

Una máquina secuencial de tipo MEALY es una 5-tupla

$$M=(Q,I,O,\delta,\beta)$$

con:

$Q \neq \emptyset$ un conjunto finito de estados
$I \neq \emptyset$ un conjunto finito de entradas
$O \neq \emptyset$ un conjunto finito de salidas
δ: Qx$I \rightarrow Q$ función de transición de estado
β: Qx$I \rightarrow O$ función de salida



CONCEPTO DE AUTÓMATA.

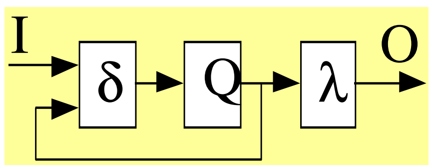
Autómata de MOORE

Una máquina secuencial de tipo MOORE es una 5-tupla

$$M=(Q,I,O,\delta,\lambda)$$

con:

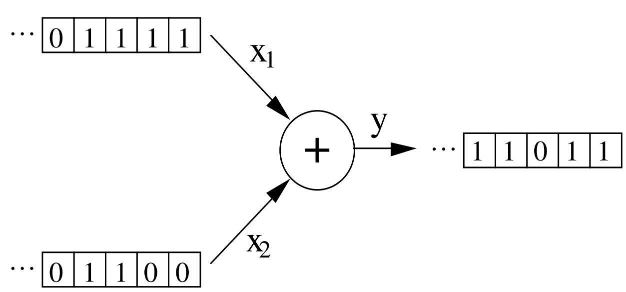
$Q \neq \emptyset$ un conjunto finito de estados
$I \neq \emptyset$ un conjunto finito de entradas
$O \neq \emptyset$ un conjunto finito de salidas
δ: $Q x I \rightarrow Q$ función de transición de estado
λ: $Q \rightarrow O$ función de salida



CONCEPTO DE AUTÓMATA.

Ejemplo: Sumador binario serie de 1 bit

- Dos entradas binarias **x1** y **x2**
- Una salida binaria **y**



CONCEPTO DE AUTÓMATA.

Modelo MEALY

- $Q = \{q0,q1\}$ donde
  - » q0 $\rightarrow$ estado de no acarreo
  - » q1 $\rightarrow$ estado de acarreo
- Función de transición de estado:
  - » $\delta(q0,11) = q1$ $\delta(q0,00/01/10) = q0$
  - » $\delta(q1,00) = q0$ $\delta(q1,10/01/11) = q1$
- Función de salida:
  - » $\beta(q0,00/11) = 0$ $\beta(q0,01/10) = 1$
  - » $\beta(q1,00/11) = 1$ $\beta(q1,01/10) = 0$

CONCEPTO DE AUTÓMATA.

Modelo MOORE

- $Q = \{q00,q01,q10,q11\}$ donde
  - » q00 $\rightarrow$ estado de no acarreo con salida y=0
  - » q01 $\rightarrow$ estado de no acarreo con salida y=1
  - » q10 $\rightarrow$ estado de acarreo con salida y=0
  - » q11 $\rightarrow$ estado de acarreo con salida y=1
- Función de transición de estado:
  - » $\delta(q00/q01\ ,00) = q00$ $\delta(q00/q01\ ,11) = q10$
  - » $\delta(q10/q11\ ,00) = q01$ $\delta(q10/q11\ ,11) = q11$
  - » $\delta(q00/q01\ ,01/10) = q01$ $\delta(q10/q11,01/10) = q10$
- Función de salida:
  - » $\lambda(q00/q10) = 0$ $\lambda(q01/q11) = 1$

REPRESENTACIÓN

Tabla de transición

- Representación tabular de las funciones de transición de estado y salida
- Ejemplo: Sumador binario serie de 1 bit

» Modelo MEALY

	00	01	11	10
q0	q0,0	q0,1	q1,0	q0,1
q1	q0,1	q1,0	q1,1	q1,0

>> Modelo MOORE

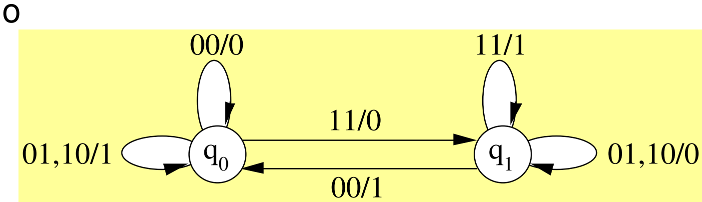
	00	01	11	10	O
q00	q00	q01	q10	q01	0
q01	q00	q01	q10	q01	1
q10	q01	q10	q11	q10	0
q11	q01	q10	q11	q10	1

REPRESENTACIÓN

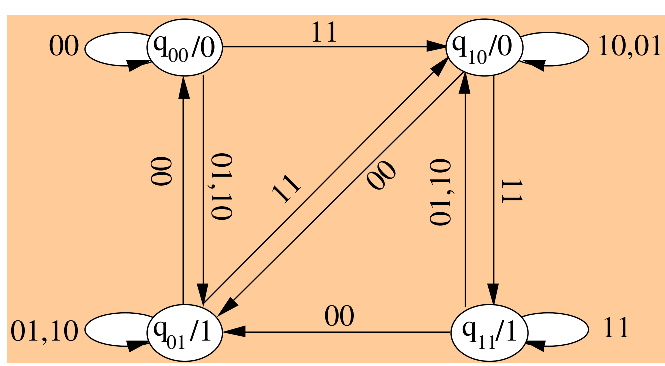
Diagrama de transición

- Grafo cuyos nodos representan estados y los arcos cambios de estado.
- Ejemplo: Sumador binario serie de 1 bit

» Modelo MEALY



» Modelo MOORE



DEFINICIONES

Equivalencia: Una máquina M es equivalente a otra M*, si para cualquier secuencia de entrada es posible encontrar algún estado inicial tal que la secuencia de salida sea la misma. Esta relación es denotada por **M = M***.

Compatibilidad: Una máquina M es compatible con otra M*, si para cualquier secuencia de entrada es posible encontrar un estado inicial tal que la secuencia de salida sea la misma, siempre y cuando ésta esté especificada. Esta relación se denota por **M ~ M***.

Máquina completamente especificada: es una máquina en la cual no existe ningún estado total para los que no esté especificada la función de próximo estado y/o de salida. En caso contrario, se dice que la **máquina** está **incompletamente especificada**.

EJEMPLOS

Máquina completamente especificada

A través de una línea serie va llegando bits, se desea realizar una máquina que indique la llegada de la secuencia 1 -> 0 -> 1.

Un sistema que permita detectar vehículos que circulan en dirección contraria por una autovía. Dicho sistema tendrá dos entradas e1 y e2 que serán las señales de dos células fotoeléctricas situadas a una distancia menor que la longitud del vehículo y la separación entre vehículos.

Máquina INcompletamente especificada

REDUCCIÓN DE AUTOMATAS

Autómatas completamente especificados

- Una vez construido un modelo:
  - » ¿Es posible reducir el número de estados?
    - – ↓ coste de una realización
    - – ↑ manejabilidad del modelo

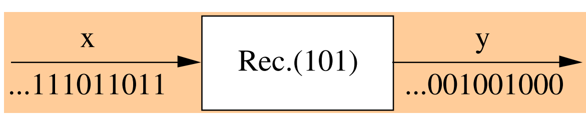
Estados equivalentes*: Dado un autómata de estados finitos A=(Q,I,O,δ,λ), dos estados qi, qj $\in$ Q se dicen equivalentes $\Leftrightarrow$ δ(qi,e) = δ(qj,e) $\forall$e $\in$ I y λ(qi) =λ(qj). (MEALY β(qi,e) = β(qj,e) $\forall$e $\in$ I)*

Dos estados equivalentes son INDISTINGUIBLES

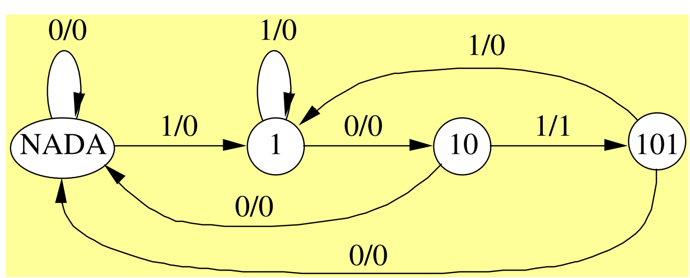
- El comportamiento del autómata a partir de cualquiera de los dos estados es el mismo.

REDUCCIÓN DE AUTÓMATAS

Ejemplo: Reconocedor de cadenas 101



- I: x={0,1}
- O: y={0,1} donde
  - » 0 → cadena no reconocida
  - » 1 → cadena reconocida



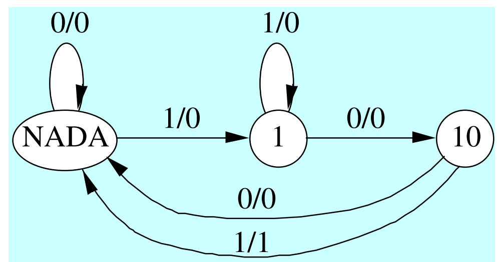
Estados:	NADA	nada reconocido
	1	subcadena 1 reconocida
	10	subcadena 10 reconocida
	101	cadena 101 reconocida

REDUCCIÓN DE AUTÓMATAS

	0	1	y
NADA	NADA	1	0
1	10	1	0
10	NADA	101	1
101	NADA	1	0

101 equivalente a **NADA**

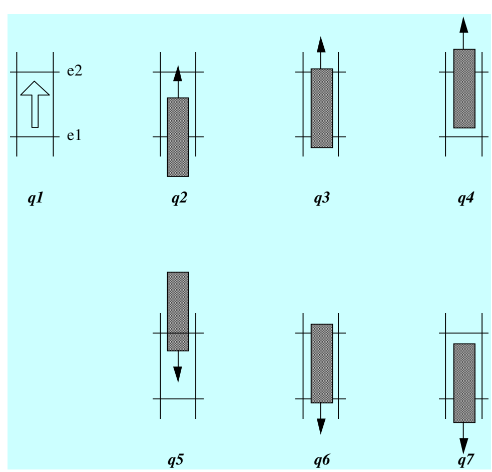
	0	1	y
NADA	NADA	1	0
1	10	1	0
10	NADA	NADA	1



REDUCCIÓN DE AUTÓMATAS

Autómatas incompletamente especificados

- Ejemplo: Detector de coches en sentido contrario
  - » Especificar un sistema que permita detectar vehículos que circulan en dirección contraria por una autovía. Dicho sistema tendrá dos entradas e1 y e2 que serán las señales de dos células fotoeléctricas situadas a una distancia menor que que la longitud del vehículo y la separación entre vehículos.



REDUCCIÓN DE AUTÓMATAS

	00	01	11	10	S
q1	q1	q5	-	q2	1
q2	-	-	q3	q2	1
q3	-	q4	q3	-	1
q4	q1	q4	-	-	1
q5	-	q5	q6	-	0
q6	-	-	q6	q7	0
q7	q1	-	-	q7	0

***Estados compatibles**: Dado un autómata de estados finitos A=(Q,I,O,δ,λ) incompletamente especificado, se dice que dos estados qi, qj $\in$ Q son compatibles* **(qi ~ qj)** $\Leftrightarrow$

(1) $\forall e \in I$ tal que $\delta(qi,e)$ y $\delta(qj,e)$ están especificadas $\Rightarrow \delta(qi,e) = \delta(qj,e)$ ó $\delta(qi,e) \sim \delta(qj,e)$

(2) $\lambda(qi) = \lambda(qj)$ si ambas están especificadas

Estados compatibles: Dos estados internos **Si** y **Sj** se denominan compatibles, si para cualquier secuencia de entrada, tomando ambos estados como iniciales, las dos secuencias de salida correspondientes resultan idénticas siempre que ambos estados de salida estén completamente especificados

Esta relación es denotada por **Si ~ Sj.** En caso contrario, los estados se denominan **incompatibles.**

REDUCCIÓN DE AUTÓMATAS

	00	01	11	10	S
q1	q1	q5	-	q2	1
q2	-	-	q3	q2	1
q3	-	q4	q3	-	1
q4	q1	q4	-	-	1
q5	-	q5	q6	-	0
q6	-	-	q6	q7	0
q7	q1	-	-	q7	0

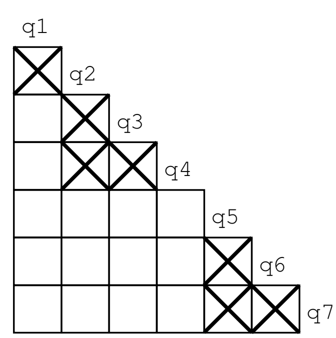
Compatibles

Incompatibles

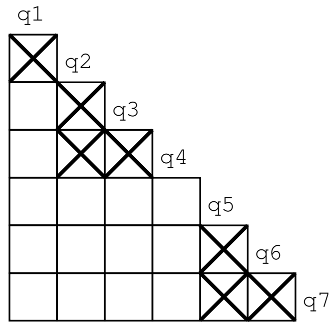
Cuando los estados son incompatibles, ya que existe combinaciones de entrada para las que el valor de las señales de salida es diferente, se identifica con un blanco ( ).

Cuando los estados son compatibles, ya que para todas las combinaciones de entrada tanto el valor de las salidas como los próximos estados son iguales, se identifica con una equis (X).

Cuando no se cumple ninguna de las condiciones anteriores, es decir, todas las combinaciones de entrada tienen el mismo valor de salida pero diferentes próximos estados, se identifica con la pareja de los próximos estados, ya que estos nos determinarán si realmente existe compatibilidad o no.



REDUCCIÓN DE AUTÓMATAS



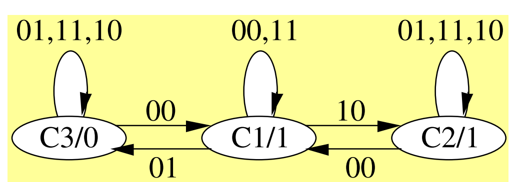
	C1	C2	C3
q1	x		
q2	x	x	
q3		x	
q4		x	
q5			x
q6			x
q7			x

6	(q6 q7)
5	(q6 q7) (q5 q6) (q5 q7) → (q5 q6 q7)
4	(q5 q6 q7)
3	(q5 q6 q7) (q3 q4)
2	(q5 q6 q7) (q3 q4) (q2 q4) (q2 q3) → (q5 q6 q7) (q2 q3 q4)
1	(q5 q6 q7) (q2 q3 q4) (q1 q2)

REDUCCIÓN DE AUTÓMATAS

- q1 → C1 (sistema en reposo)
- q2,q3,q4 → C2 (coche en sentido permitido)
- q5,q6,q7 → C3 (coche en sentido contrario)

	00	01	11	10	S
C1	C1	C3	-	C2	1
C2	C1	C2	C2	C2	1
C3	C1	C3	C3	C3	0



Conversión entre Mealy y Moore

Equivalencias entre máquinas de estados

Dos máquinas de estados *A* y *B* son *funcionalmente equivalentes* si y sólo si, para cualquier entrada, ellas realizan el mismo cálculo.

Dos máquinas de estados A y B son *semánticamente equivalentes* si y sólo si ellas son funcionalmente equivalentes y usan el mismo conjunto de entradas (I) y de salidas (O).

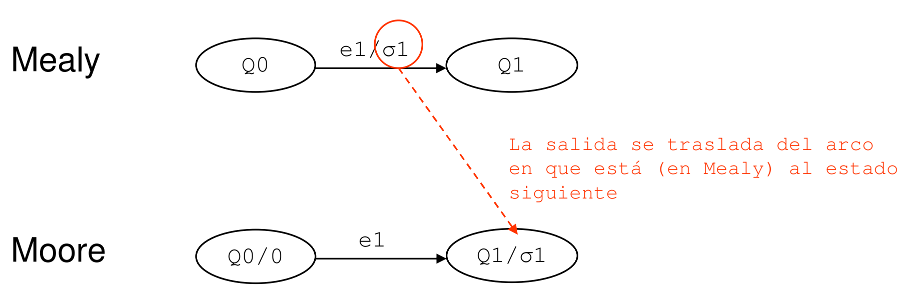
Dos máquinas de estados A y B son *equivalentes paso a paso* (pathwise) si y sólo si son semánticamente equivalentes y si y sólo si, para cualquier entrada (e), ellos ejecutan una idéntica secuencia de salidas y de cambios de estado.

Conversión entre Mealy y Moore

El método clásico para obtener un autómata de Moore similar a un autómata de Mealy dado, se basa en la búsqueda de un autómata equivalente paso a paso.

RECORDAR: En Moore las salidas no pueden depender de las entradas

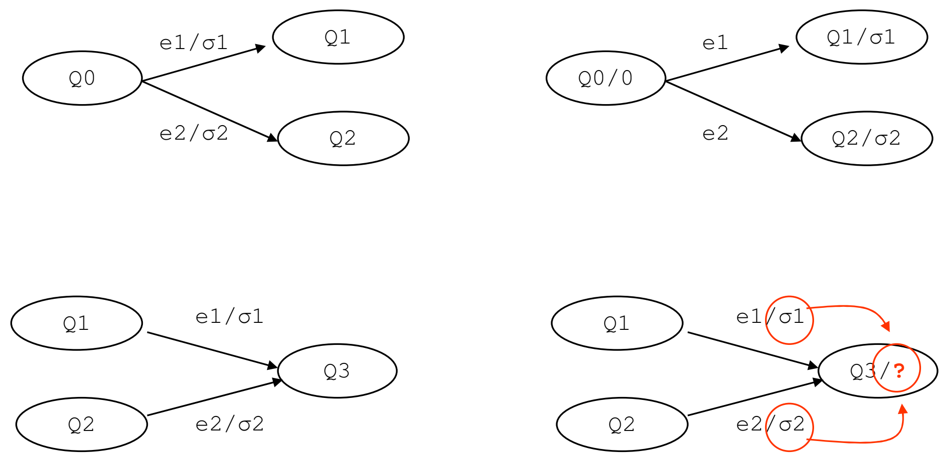
Sean $ei \in I$; $\sigma i \in O$ y $Qi \in Q$



Conversión entre Mealy y Moore

Otros casos de transformación

Mealy　　　　　　　　　　Moore



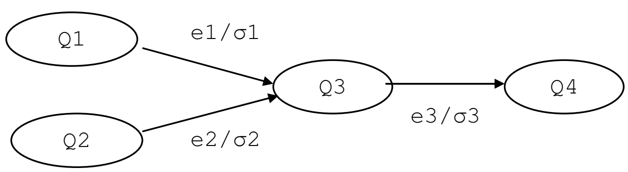
Como queda el nuevo estado Q3…

Q3/ σ1 ó Q3/σ2 ???

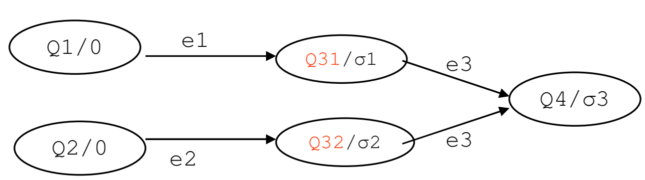
Conversión entre Mealy y Moore

Otros casos de transformación

Mealy



Moore



Cualquier estado con fuentes múltiples
se replica en estados equivalentes

Conversión entre Mealy y Moore

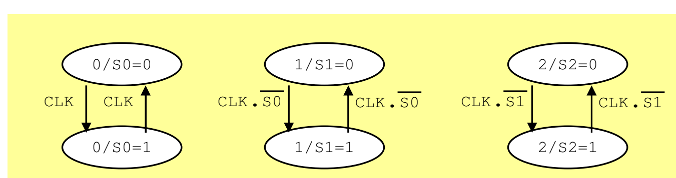
Como se convierte una máquina Moore en Mealy???

Unidades de control y de cálculo…

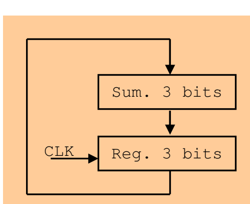
Las máquinas de estado pueden usarse para modelar la parte de control y la parte de cálculo de cualquier sistema.

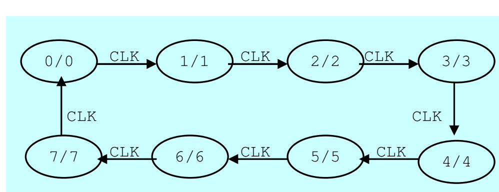
El modelado de la parte de cálculo implica la descripción del proceso de cálculo de los distintos operadores utilizados (operaciones aritméticas y lógicas: sumas, productos, desplazamientos, comparadores, etc).

Usualmente la descripción de los operadores de cálculo NO se realiza como máquina de estado pues la representación de todos los estados de datos puede ser costosa…



Ejemplo: Contador de 3 bits…





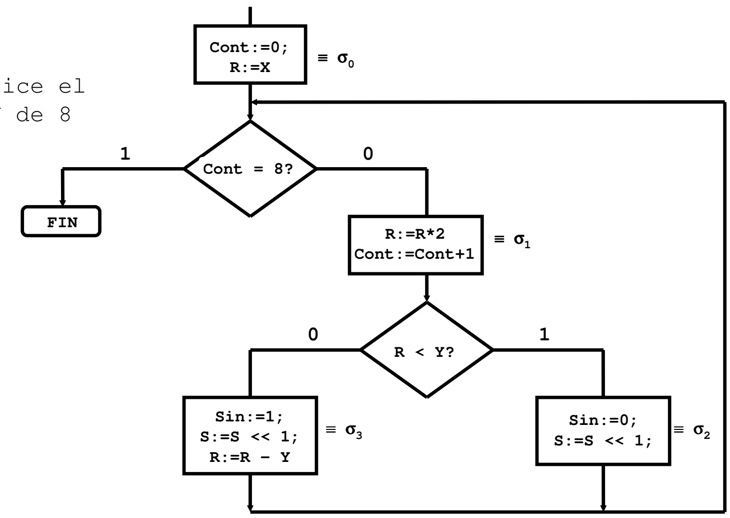
La unidad de control (UC)

La unidad de control modela el algoritmo que controla el sistema que se está implementando.

Sus salidas son las señales que controlan las operaciones de cálculo del sistema (operadores aritméticos, lógicos, operaciones en registros, etc.)

Pueden ser modeladas como autómatas de Mealy o Moore indistintamente.

Ejemplo: una máquina que realice el cociente de dos números X e Y de 8 bits con X < Y.



La unidad de control (UC)

Prueba del algoritmo

Sea X = 3 e Y = 4

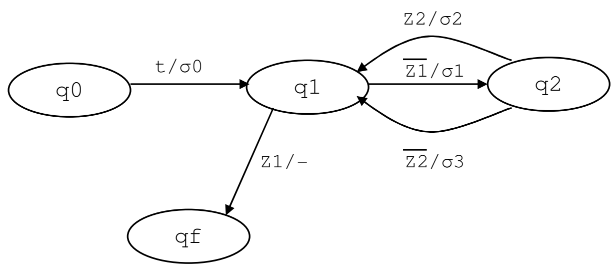
entonces

S = X / Y = 3 / 4 = 0,75

Cont	X	Y	R	Sin	S		sigma
----	0011	0100	----	----	----		
000	0011	0100	0011	----	0000		0
						Cont<>4	
001			0110				1
						R>=Y	
			0010	1	0001		3
						Cont<>4	
010			0100				1
						R>=Y	
			0000	1	0011		3
						Cont<>4	
011			0000				1
						R<Y	
				0	0110		2
						Cont<>4	
100			0000				1
						R<Y	
				0	1100		2
						Cont=4	
X/Y =	**3 / 4 =**	**0011 /**	**0100**	**=**	**1100**	**= 0,1100**	**= 0,75**

La unidad de control (UC)

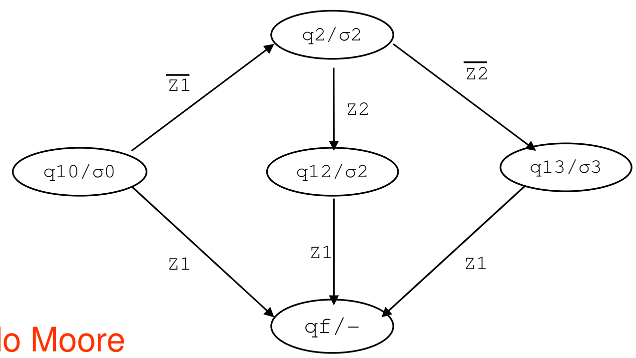
Modelo Mealy



con:

Z1 = 1 si Cont = 8
= 0 en otro caso

Z2 = 1 si R < Y
= 0 en otro caso



Modelo Moore

Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

4 estados (q0..q3) se direccionan con 2 bits y1,y0

$\overline{y1}\ \overline{y0} = q0$

$\overline{y1}\ y0 = q1$

$y1\ \overline{y0} = q2$

$y1\ y0 = q3$

Determinación de las ecuaciones de salida

$$\sigma j = \underset{\forall Qi \supset \sigma j}{\vee} mi \ . \ Qi = N(m,Q)$$

$$\sigma 0 = 1 \ . \ \overline{y1} \ . \ \overline{y0}$$
$$\sigma 1 = \overline{z1} \ . \ \overline{y1} \ . \ y0$$
$$\sigma 2 = z2 \ . \ y1 \ . \ \overline{y0}$$
$$\sigma 3 = \overline{z2} \ . \ y1 \ . \ y0$$

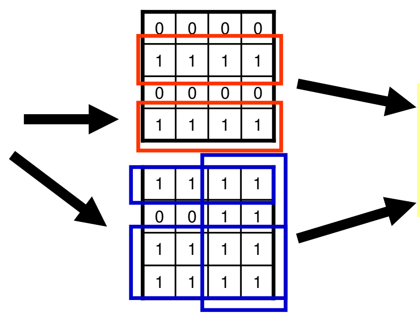
Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

Determinación de las ecuaciones de salida

y_1y_0/ z_1z_2	00	01	10	11	
0 0	01	01	01	01	Q_0
0 1	10	10	11	11	Q_1
1 0	01	01	01	01	Q_2
1 1	11	11	11	11	Q_3

$y1^+y0^+ = Qj^+$

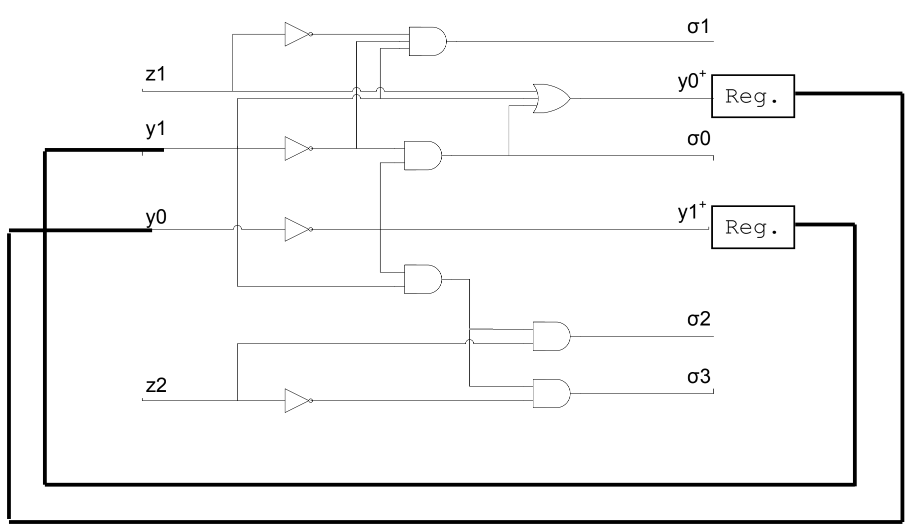


$$y_1^+ = y_o$$

$$y_0^+ = z_1 + \overline{y_1}\,\overline{y_0} + y_1$$

Implementación de máquinas de estado

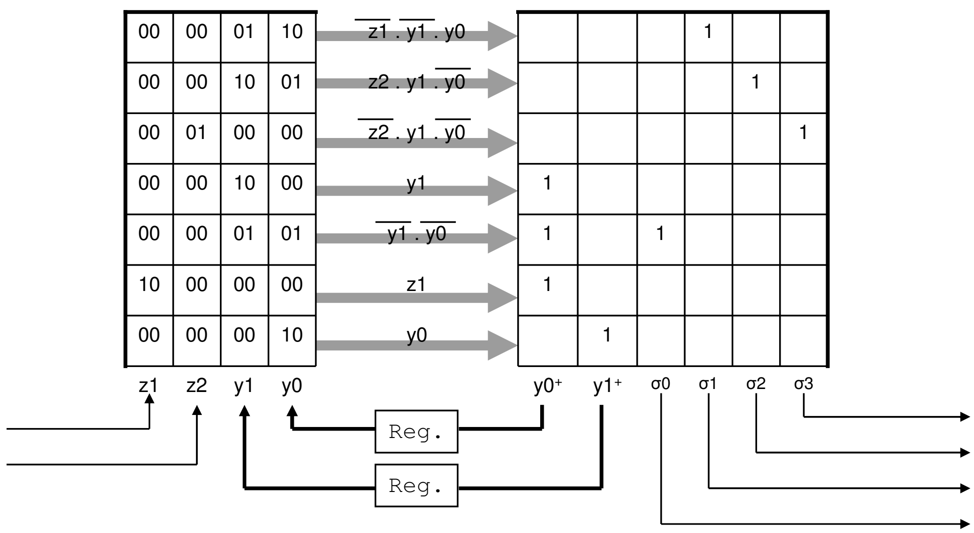
- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux



Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

Se utilizan las mismas ecuaciones del caso anterior



Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

Se puede representar la maquina de estados como una secuencia de operaciones de la forma:

Qi $\overline{\mathbf{Zi}}$ **σi Qm**

Zi σj Qn

con

Qi : estado actual
Qm, Qn : estados siguientes
Zi : i-esima entrada al autómata
σi, σj : salidas

con la siguiente dinámica:

1) si **Zi** = 1 entonces se realiza la acción **σi** y se salta al estado **Qm**

2) si Zi = 0 entonces se realiza la acción **σj** y se salta al estado **Qn**

Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

Bajo esta definición el algoritmo de división se puede escribir como sigue:

q0	t	$\sigma 0$	q1
q1	$\overline{z1}$	$\sigma 1$	q2
	z1	λ	q3
q2	$\overline{z2}$	$\sigma 3$	q1
	z2	$\sigma 2$	q1
q3	t	λ	q3

De este modo se puede almacenar cada estado en una celda de una memoria ROM con los campos necesarios para almacenar los parámetros:

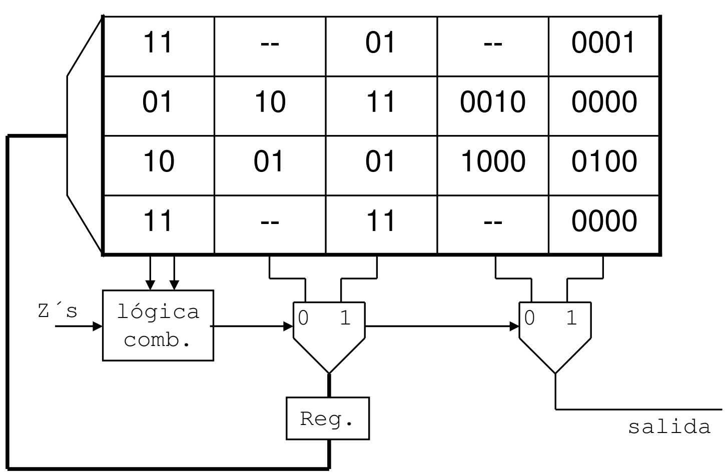
σ i : acción si se cumple Zi
σj : acción si NO se cumple Zi
qm : estado destino si se cumple Zi
qn : estado destino si NO se cumple Zi
Ci : indicador de que en este estado se debe evaluar la entrada Zi

```
con     t = true
        λ = no acción
```

Implementación de máquinas de estado

- Mediante lógica de compuertas
- Mediante arreglos homogéneos de compuertas (PLA´s)
- Mediante ROM+Mux

C	$\overline{\text{dir}}$	dir	$\overline{\sigma}$	σ
11	--	01	--	0001
01	10	11	0010	0000
10	01	01	1000	0100
11	--	11	--	0000




```
codificaciones utilizadas:

C:      01    z1
        10    z2
        11    t

σ       0001  σ0
        0010  σ1
        0100  σ2
        1000  σ3
        0000  λ

--      no importa
```