Máquinas de estados finitos

Martín Vázquez

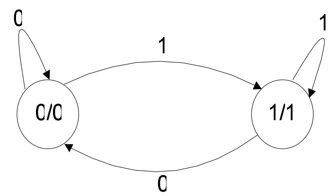
Maquinas de estados vistas en el curso

- Los circuitos con memoria (secuenciales) implementan máquinas de estados
- Observemos algunos *FF* estudiados en el curso. Ecuaciones de próximo estado
  - *D*: $Q^+ = D$
  - *SR*: $Q^+ = S + R'Q$
  - *JK*: $Q^+ = JQ' + K'Q$
  - *T*: $Q^+ = T \oplus Q$

Máquinas de estados conocidas

- Los cambios de estados se producen en el evento del *clock.*
  - *Donde* $Q = Q^{+}$, cuando *clk*↑ (flanco ascendente) o *clk*↓ (flanco descendente)
- Tablas y grafos de estados. Para *FF D*

Q	*Q*⁺ *X* = 0	1	*Output*
0	0	1	0
1	0	1	1



Máquinas de estados conocidas

- Para *FF JK*

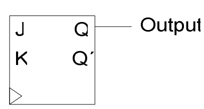


Diagrama de estados

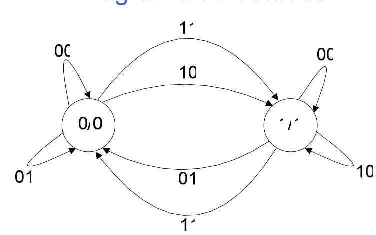


Tabla de estados

Q	*Q*⁺ *JK* = 00	01	10	11	*Output*
0	0	0	1	1	0
1	1	0	1	0	1

Concepto de autómata

- Autómata de Moore

Una máquina secuencial tipo Moore es una 5-tupla

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

con:

$Q \neq \varnothing$ *un conjunto finito de estados*
$I \neq \varnothing$ *un conjunto finito de entradas*
$O \neq \varnothing$ *un conjunto finito de salidas*
$\delta: QxI \rightarrow Q$ *función de transición de estado*
$\lambda: Q \rightarrow O$ *función de salida*

Concepto de autómata

- Autómata de Mealy

Una máquina secuencial de tipo Mealy es una 5-tupla

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

con:

$Q \neq \varnothing$ *un conjunto finito de estados*
$I \neq \varnothing$ *un conjunto finito de entradas*
$O \neq \varnothing$ *un conjunto finito de salidas*
$\delta: QxI \rightarrow Q$ *función de transición de estado*
$\beta: QxI \rightarrow O$ *función de salida*

Concepto de autómata

¿Que máquinas de estados implementan los *flip-flops D* y *JK*?

Diagrama de estados de *FF D*

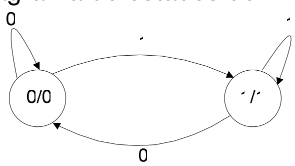
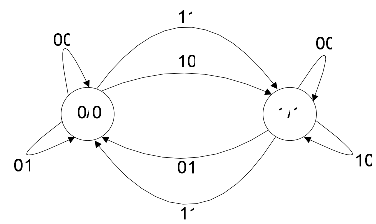


Diagrama de estados de *FF JK*

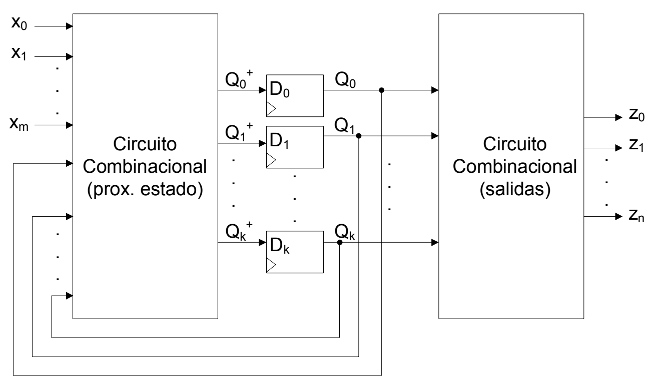


Implementan una máquina de Moore

Lo mismo sucede con *FF*s *T* y *SR*

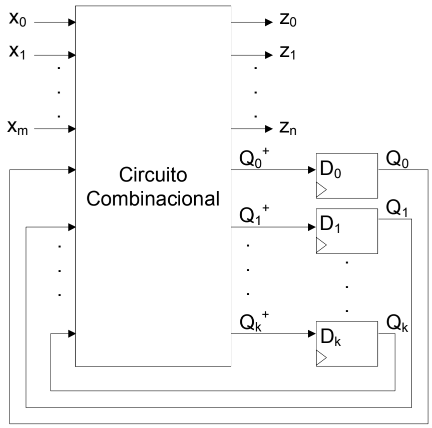
Implementación de máquinas de estados (modelo general)

- Moore



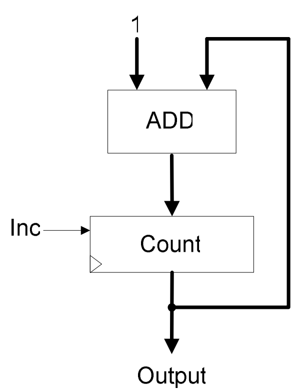
Implementación de máquinas de estados (modelo general)

- Mealy



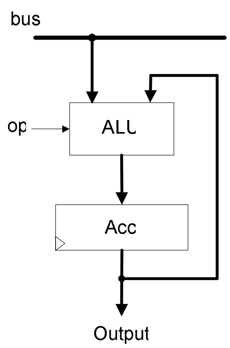
Más ejemplos de máquinas de estados conocidos

- Contador sincrónico basado en *FF D*. *Máquina de Moore*



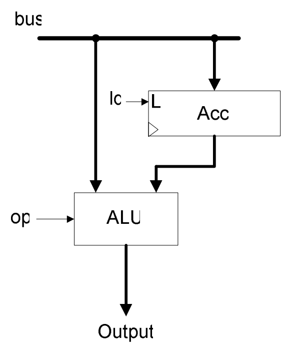
Ejemplos de diferentes modelos de ALU básica

Máquina de Moore



Acc ← a
Acc← Acc + b
Output ← Acc

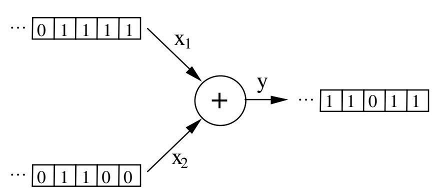
Máquina de Mealy



Acc ← a
Output ← Acc + b

Sumador binario serie

- 1 *Full Adder* de 1 bit
- 3 registros de desplazamiento de *N* bits: para *x*1, *x*2 e *y*.
- En cada ciclo *i*
  - se realiza $y_i = x1_i + x2_i$
  - se debe considerar el *carry* generado en ciclo anterior



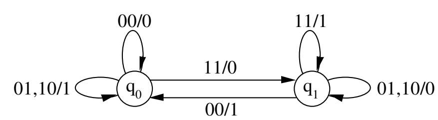
Sumador binario serie

- Modelo Mealy
  - estados asociados al acarreo de instancia anterior

Tabla de estados

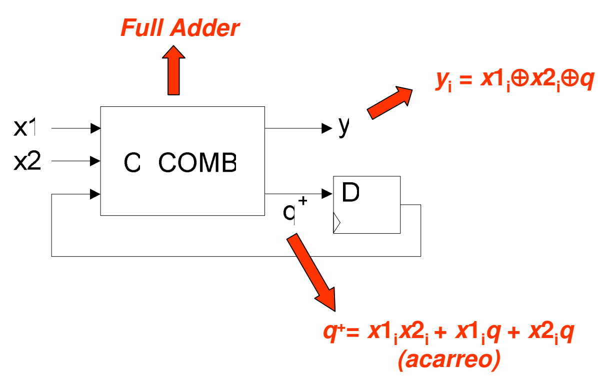
Q	Q^+				Output			
	x1 x2= 00	01	10	11	00	01	10	11
q_0	q_0	q_0	q_0	q_1	0	1	1	0
q_1	q_0	q_1	q_1	q_1	1	0	0	1

Diagrama de estados



Sumador binario serie

- Circuito que implementa el modelo de Mealy



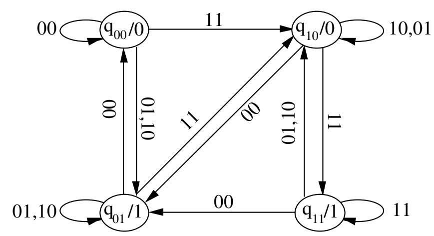
Sumador binario serie

- Modelo Moore
  - estados asociados al acarreo de instancia anterior y resultado

Tabla de estados

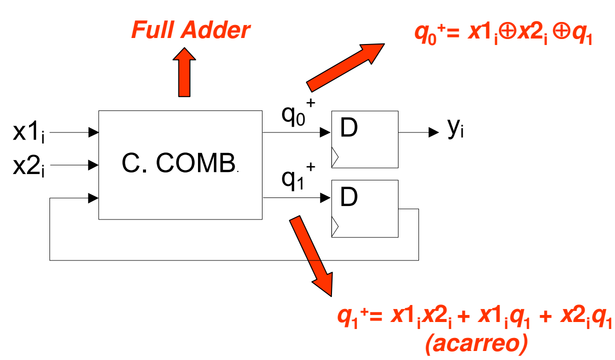
Q	*Q*+ *x1 x2* = 00	01	10	11	*Output*
q_{00}	q_{00}	q_{01}	q_{01}	q_{10}	0
q_{01}	q_{00}	q_{01}	q_{01}	q_{10}	1
q_{10}	q_{01}	q_{10}	q_{10}	q_{11}	0
q_{11}	q_{01}	q_{10}	q_{10}	q_{11}	1

Diagrama de estados



Sumador binario serie

- Circuito que implementa el modelo de Moore

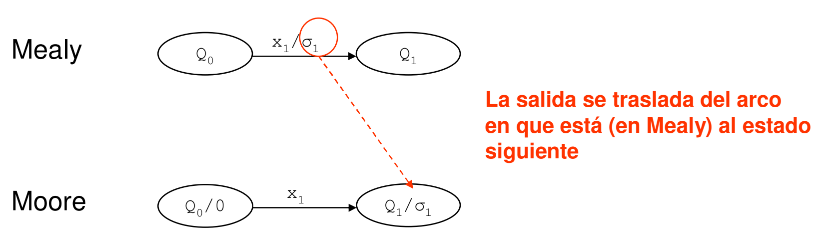


Máquinas de Mealy vs. Moore

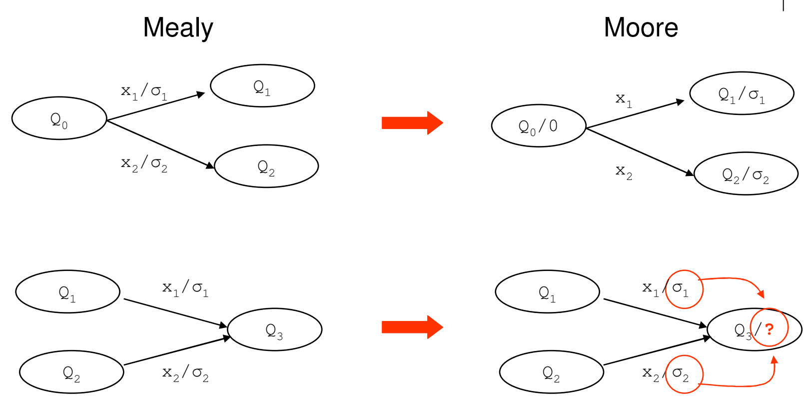
- Las máquinas de Moore producen salidas estables
- En las máquinas de Mealy, las salidas pueden que sean inestables. Las salidas dependen también de las entradas
- Para toda máquina de Mealy se puede encontrar una máquina Moore equivalente
- La cantidad de estados que poseen las máquinas de Moore, con frecuencia es superior a la cantidad de estados presente en máquinas de Mealy. Considerando máquinas equivalentes
- Dos sistemas secuenciales son **equivalentes**, si generan la misma secuencia de salida para la misma secuencia de estrada

Mealy → Moore

- Sea $x_i \in I$; $\sigma_i \in O$ y $Q_i \in Q$

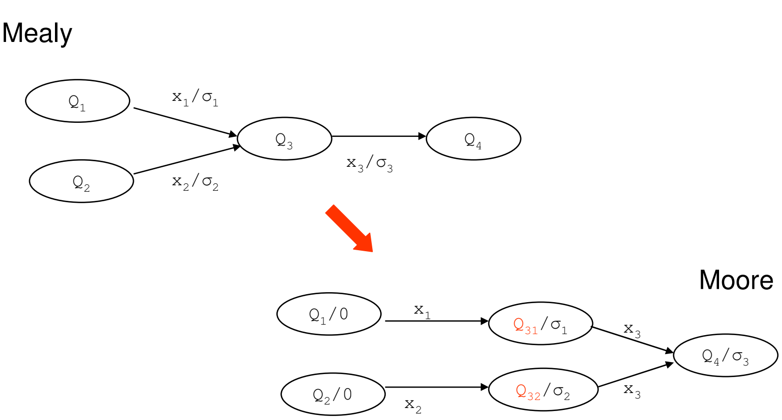


Mealy → Moore



¿Como queda el nuevo Q_3?
Q_3/σ_1 ó Q_3/σ_2

Mealy → Moore



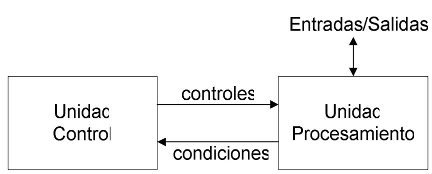
Cualquier estado con fuentes múltiples
se replica en estados equivalentes

Implementación de algoritmos

- La implementación de un algoritmo en un sistema digital se realiza mediante dos máquinas de estados
  - Unidad de Control
  - Unidad de Procesamiento o Operacional
- Concepto introducido por *Glushkov* en1965: *“Automaton theory and formal microprogram transformation”,* Kibernetica I, p.1-9
- Sistema microprogramado.
  - Máquina de control con algoritmo. Ejecuta μ-programa.
  - Máquina operacional o de procesamiento. Realiza las operaciones, μ-órdenes indicadas por la máquina de control

Modelo de *Glushkov*

- Modelo de *Glushkov* de un sistema μ-programado



- Los máquinas de estados pueden ser: Moore-Mealy, Mealy-Moore o Moore-Moore. Pero NO Mealy-Mealy ¿Porque?

Microinstrucciones

- Definimos μ-instrucciones con la siguiente forma:
  - Tipo IF: N_i z' σ_1 N_j
    z σ_2 N_k
  - Tipo EXE y/o GOTO: N_i t σ_1 N_j
- Donde
  - los z's son predicados elementales que provienen de la Unidad de Procesamiento
  - los σ's son μ-órdenes
  - t (*true*) significa siempre verdadero

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Problema: Obtener la cantidad de ocurrencias de la secuencia y (3 bits) en la cadena x (16 bits), mediante un sistema microprogramado
- Para la secuencia

x = 1001010101101001 e y = 101

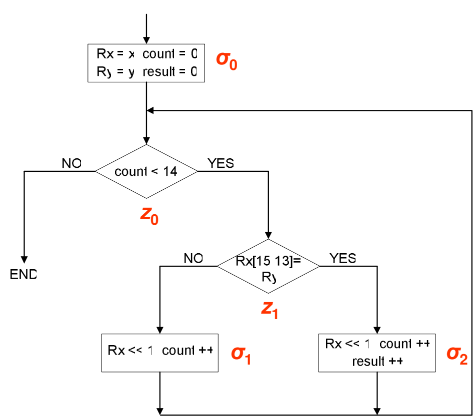
- Se obtiene el resultado $r = 4$
- Nótese que r está en [0, 14]

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Implementación del sistema digital microprogramado:
  - Definir el algoritmo
    - Identificar los recursos involucrados en Unidad de Procesamiento,
    - Identificar *μ*-órdenes (*σ*‘s) y predicados elementales (*z*’s)
  - Construir autómata de control
  - Construir *μ*–programa de control
    - codificación de estados y *μ*-órdenes
    - obtener funciones de salida (*σ*) y de próximo estado (*Q*+)
  - Sintetizar la unidad de control
    - PLA, AOI, ROM-MUX, etc

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Algoritmo



Ejemplo: Sistema microprogramado de identificación de subsecuencias

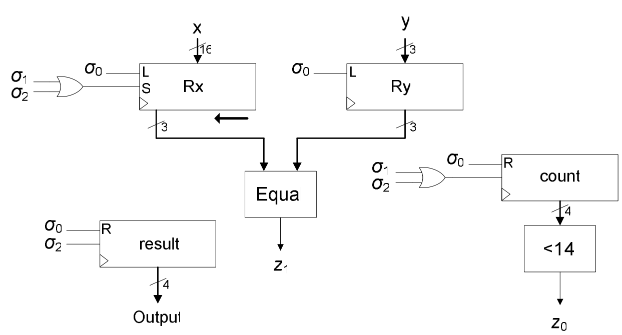
- La Unidad de procesamiento posee tres registros: *Rx* (16 bits), *Ry* (3 bits), *count* (4 bits) y *result* de 4 bits.

- Predicados elementales generados por la Unidad de Proceso
  - z_0=1 si *count*<14, sino z_0=0
  - z_1=1 si *Rx*[15:13] = *Ry*, sino z_1=0

- μ-órdenes generados por la Unidad de Control
  - σ_0: *Rx* = *x*, *Ry* = *y*, *count* = 0 y *result* = 0
  - σ_1: *count* = *count*+1 y *Rx* = *Rx*[14:0] & '0'
  - σ_2: *count* = *count*+1, *Rx* = *Rx*[14:0] & '0' y *result* = *result*+1

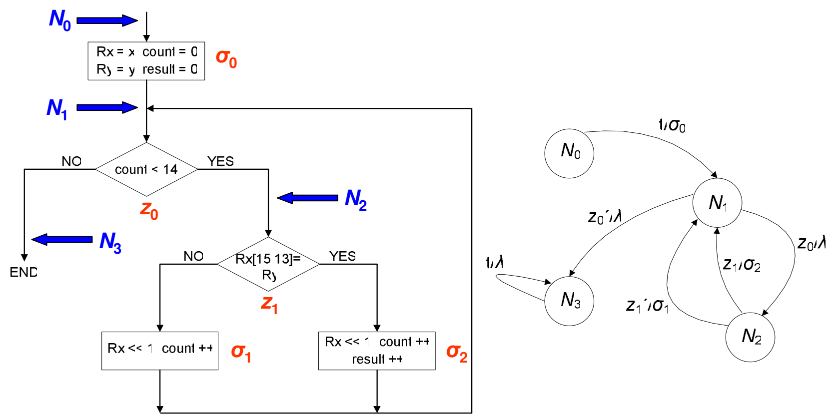
Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Unidad de Procesamiento u Operacional. *Máquina de Moore*



Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Construcción de autómata de control



Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Construcción de μ-programa de control
  - Codificación de μ-órdenes y estados
    - λ = (000), σ_0 = (001), σ_1 = (010), σ_2 = (100)
    - $N_i = (q_1\ q_0)$; N_0 = (00), N_1 = (01), N_2 = (10), N_3 = (11)
  - μ-programa de control asociado al autómata

N_0	t	σ_0	N_1
N_1	$z_0´$	λ	N_3
	z_0	λ	N_2
N_2	$z_1´$	σ_1	N_1
	z_1	σ_2	N_1
N_3	t	λ	N_3

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Obtención de funciones de salida (σ) y de próximo estado ($q_1^+ \, q_0^+$)
  - $\sigma_0 = q_1' q_0'$; $\sigma_1 = q_1 q_0' z_1'$ y $\sigma_2 = q_1 q_0' z_1$
  - Para $N_i^+ = (q_1^+ \, q_0^+)$

q_0 (columnas 2–3), q_1 (columnas 3–4), z_0 (filas 2–3), z_1 (filas 3–4)

01	11	11	01
01	10	11	01
01	10	11	01
01	11	11	01

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Para $N_i^+ = (q_1^+ \, q_0^+)$

q_0 (columnas 2–3), q_1 (columnas 3–4), z_0 (filas 2–3), z_1 (filas 3–4)

01	11	11	01
01	10	11	01
01	10	11	01
01	11	11	01

q_1^+

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$q_1^+ = q_0$

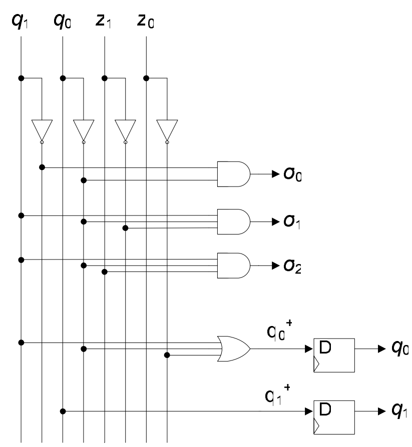
q_0^+

1	1	1	1
1	0	1	1
1	0	1	1
1	1	1	1

$q_0^+ = q_1 + q_0' + z_0'$

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Síntesis de Unidad de Control mediante compuertas básicas



Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Síntesis de Unidad de Control mediante PLA

AND

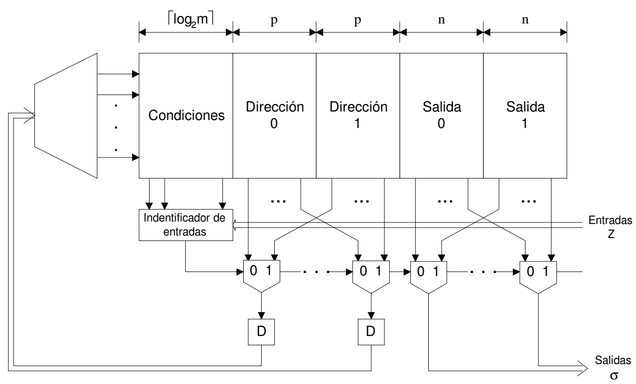
q_1	q_0	z_1	z_0
01	**01**	00	00
10	**01**	**01**	00
10	**01**	**10**	00
00	**10**	00	00
10	00	00	00
00	**01**	00	00
00	00	00	**01**

OR

σ_0	σ_1	σ_2	q_1^+	q_0^+
1	0	0	0	0
0	**1**	0	0	0
0	0	**1**	0	0
0	0	0	**1**	0
0	0	0	0	**1**
0	0	0	0	**1**
0	0	0	0	**1**

Ejemplo: Sistema microprogramado de identificación de subsecuencias

- Síntesis de Unidad de Control mediante ROM y MUX (genérico)



Microprogramación en procesadores

- Un procesador es un sistema μ-programado (o nanoprogramado, o…)
- En un procesador multiciclo
  - cada instrucción de su repertorio es interpretada por un μ-programa
  - la ejecución de cada instrucción involucra varios ciclos del procesador
  - cada μ-instrucción puede ser interpretada por un nanoprograma, y asi…
  - a modo de ejemplo, la instrucción *add* r_3,r_1,r_2 ($r_3 = r_1+r_2$) consume 3 ciclos de reloj. Ejecuta 3 microinstrucciones
    - obtener instrucción desde memoria de programa (*FETCH*)
    - obtener r_1 y r_2 desde el banco de registros y ejecutar suma en ALU
    - almacenar r_3 el resultado en el banco de registros