

Sistemas Digitales

Tema 7:

*Diseño de circuitos secuenciales
mediante diagramas de estados.
Síntesis de máquinas de estados y
análisis de circuitos*

Autores: Ángel Luis Álvarez Castillo
Carmen Coya Párraga

Universidad Rey Juan Carlos

Escuela Técnica Superior de Ingeniería
de Telecomunicación



1

EVALUACIÓN

Cuestiones específicas

Al terminar el tema 2 el alumno debería ser capaz de responder:

- ¿Qué es un circuito ASIC? Qué ventajas e inconvenientes ofrecen los dispositivos configurables frente a los ASIC?
- Explicar las posibilidades de ambigüedad en la descripción gráfica de un diagrama de estados.
- ¿Por qué resulta aconsejable trabajar con máquinas de estados síncronas en lugar de asíncronas?
- ¿Por qué se busca que todos los biestables lleven la misma señal de reloj, qué problema hay para conseguirlo y cómo se resuelve esto en una FPGA?
- ¿Cuál es la diferencia entre una máquina de Mealy y de Moore? ¿Cómo se detecta esto en un diagrama de estados y en un circuito?
- Cómo se convierte una máquina de Mealy en una de Moore y viceversa?
- En máquinas de estados pequeñas el uso de flip-flops JK puede resultar ventajoso frente al de flip-flops D ¿Por qué razón?
- Indicar diversas alternativas de actuación al construir una máquina de estados, cuando el nº de biestables escogidos m hace que $2^m > n^\circ$ de estados descritos.
- Utilizar un procedimiento sistemático para sintetizar un circuito a partir de una máquina de estados.
- Utilizar un procedimiento sistemático para deducir el diagrama de estados a partir de un circuito secuencial.

2

3	<p style="text-align: center;"><u>Apartado 2.1 <i>Concepto de máquina de estados o autómeta</i></u></p> <ul style="list-style-type: none"> • Def.: Máquina de estados o autómeta. <p>Es un circuito secuencial, que incluye una parte combinacional, otra parte de memoria (biestables en general), y unas salidas con un propósito determinado.</p> <p>Diferencia entre un combinacional y un secuencial: la parte de memoria. Ésta se halla constituida por biestables, cada uno con sus entradas de excitación y su salida $q_i(t)$. El nº de biestables (n) determina el espectro de valores posibles (2^n) asumidos por el conjunto de las $q_i(t)$.</p> <ul style="list-style-type: none"> • Def.: Variable de estado. <p>Cada salida $q_i(t)$ de los biestables.</p> <ul style="list-style-type: none"> • Def.: Estado interno. <p>Cada conjunto de variables actuales, [$q_1(t)$, $q_2(t)$, ... $q_n(t)$], constituye un ESTADO o ESTADO INTERNO del circuito. n variables dan lugar a 2^n posibles estados internos.</p> <p>Notas:</p> <ul style="list-style-type: none"> - Veremos que cada estado [$q_1(t)$, $q_2(t)$, ... $q_n(t)$] contiene la información suficiente para predecir la evolución del sistema, a partir de unas entradas. - Veremos que identificar estados del problema real, y definir estados del circuito que se correspondan con ellos, es el paso decisivo en el diseño de autómatas.

4	<p style="text-align: center;"><u>Apartado 2.1 <i>Concepto de máquina de estados o autómeta</i></u></p> <ul style="list-style-type: none"> • <u>¿Hay diversos tipos de autómatas? Clasificación.</u> <p>Por el nº de estados:</p> <ul style="list-style-type: none"> - Autómeta finito: el nº máximo de estados internos es finito. El nº de entradas $X(t)$, salidas $Z(t)$, entradas de excitación de los biestables $Y(t)$ y estados $Q(t)$ es limitado y concreto. - Autómeta infinito: un tipo de autómeta teórico, con nº posible de estados internos infinito. <p>Por su sincronismo:</p> <ul style="list-style-type: none"> - Autómatas asíncronos: no disponen de señal de reloj externa. Las salidas siguientes dependerán continuamente del estado de las entradas y de la salida actual (con mediación del estado interno). Requieren un conocimiento detallado de los retardos y tiempos de propagación. - Autómatas síncronos: todos los biestables llevan entrada de sincronización. <p>Por su estructura interna:</p> <ul style="list-style-type: none"> - Autómatas de Moore: las entradas generan primero cambios en el estado interno, y el estado interno genera la salida: $Z(t) = g[Q(t)]$ - Autómatas de Mealy: las entradas generan cambios a la vez en el estado interno y en las salidas. $Z(t) = g[X(t), Q(t)]$

Apartado 2.1 *Concepto de máquina de estados o autómeta*

- ¿Cómo describimos un circuito secuencial?

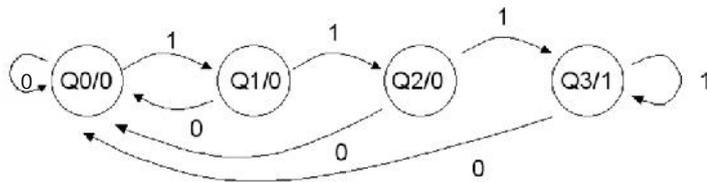
Varios niveles

Gráfico - comportamental: Diagrama de estados.

- Def.: Diagrama de estados

Es un gráfico que describe de forma completa el comportamiento del autómeta (la formulación de una situación real). Incluye dos elementos:

- Círculos o nodos, que equivalen a un estado interno, y reciben un nombre ("Q0", "pepe", etc).
- Flechas (arco dirigido) para indicar la evolución de unos a otros a impulsos de reloj. Incluyen dígitos o variables lógicas que reflejan las entradas al sistema y, en algún caso, las salidas correspondientes.



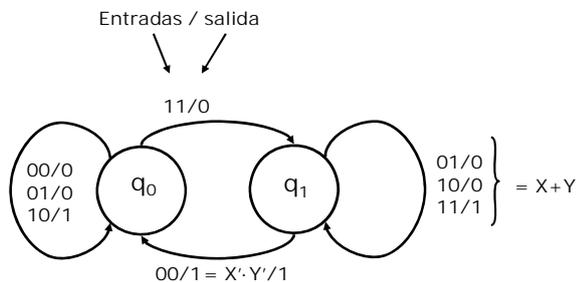
Esta máquina de estados describe un detector de la secuencia "111".

5

Apartado 2.1 *Concepto de máquina de estados o autómeta*

Las expresiones de transición que acompañan a las flechas que SALEN de un estado deben verificar:

- *Ser mutuamente excluyentes*: ningún par de expresiones de transición deben ser iguales a 1 para la misma combinación de entrada.
- *Todas incluidas*: Todas las posibles combinaciones de entradas deben estar contempladas en las flechas que salen del nodo.



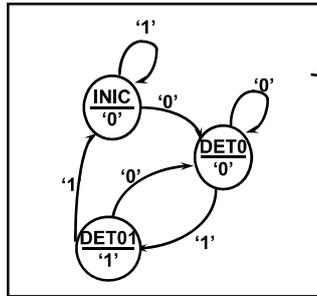
6

Apartado 2.1

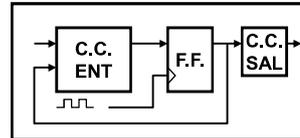
Concepto de máquina de estados o autómeta

Codificado (HDL) - comportamental: descripción algorítmica que entienden los programas de descripción de hardware:

Estilo A



La salida sólo depende del estado actual



```

entity DETECTA is
port( CLK : in std_logic;
      RN : in std_logic;
      DIN : in std_logic;
      SAL : out std_logic);
end DETECTA;
  
```

7

Apartado 2.1

Concepto de máquina de estados o autómeta

```

architecture MOORE of DETECTA is
type ESTADO is (INIC, DET0, DET01);
signal ACTUAL, SIGUE : ESTADO;
begin
SEQ: process(CLK, RN)
begin
if RN = '0' then
ACTUAL <= INIC;
elsif CLK'event and CLK = '1' then
ACTUAL <= SIGUE;
end if;
end process SEQ;
...
  
```

```

...
COMB: process(ACTUAL, DIN)
begin
case ACTUAL is
when INIC =>
SAL <= '0';
if DIN = '1' then
SIGUE <= INIC;
else
SIGUE <= DET0;
end if;
when DET0 =>
SAL <= '0';
if DIN = '1' then
SIGUE <= DET01;
else
SIGUE <= DET0;
end if;
end case;
end process COMB;
...
  
```

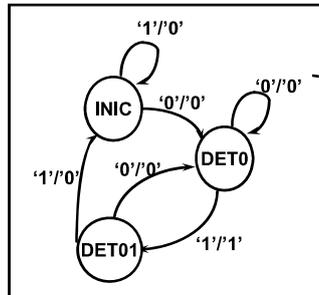
La salida sólo depende del estado actual

8

Apartado 2.1 Concepto de máquina de estados o autómeta

Codificado (HDL) - comportamental:

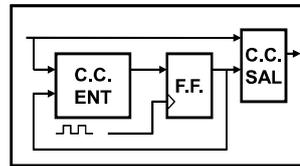
Estilo B



La salida depende del estado actual y la entrada (asociada a la transición)

```

entity DETECTA is
port(CLK : in std_logic;
      RN : in std_logic;
      DIN : in std_logic;
      SAL : out std_logic);
end DETECTA;
  
```



9

Apartado 2.1 Concepto de máquina de estados o autómeta

```

architecture MEALY of DETECTA is
type ESTADO is (INIC, DET0, DET01);
signal ACTUAL, SIGUE : ESTADO;
begin
SEQ: process(CLK, RN)
begin
if RN = '0' then
ACTUAL <= INIC;
elsif CLK'event and CLK = '1' then
ACTUAL <= SIGUE;
end if;
end process SEQ;
...
  
```

La salida depende del estado actual y la entrada (asociada a la transición)

```

...
COMB: process(ACTUAL, DIN)
begin
case ACTUAL is
when INIC =>
if DIN = '1' then
SIGUE <= INIC;
SAL <= '0';
else
SIGUE <= DET0;
SAL <= '0';
end if;
when DET0 =>
if DIN = '1' then
SIGUE <= DET01;
SAL <= '1';
else
SIGUE <= DET0;
SAL <= '0';
end if;
end case;
end process COMB;
...
  
```

10

viii

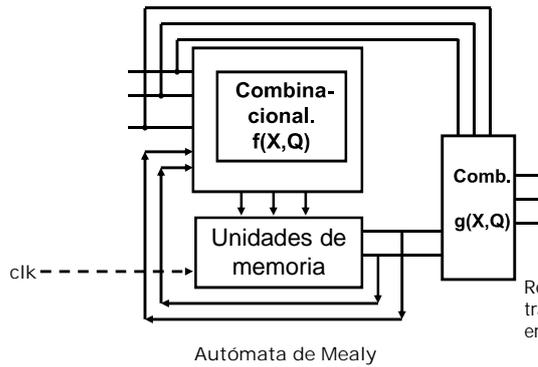
Apartado 2.1 Concepto de máquina de estados o autómeta

Nivel lógico. RTL:

Tablas de estados/salidas: incluyen transiciones, excitaciones y salidas.

En un autómeta se manejan tres funciones:

- Funciones de salida: $Z(t)=g[X(t), Q(t)]$ (si es de Moore, la $X(t)$ no aparece)
 - Excitaciones de las entradas de los biestables: $Y(t)=f[X(t), Q(t)]$
 - Funciones de transición de los biestables: $Q(t+1)= h[X(t), Q(t)]$ (ya conocidas).
- Estas dos últimas funciones permitirán construir la TABLA DE TRANSICIONES DEL CIRCUITO.



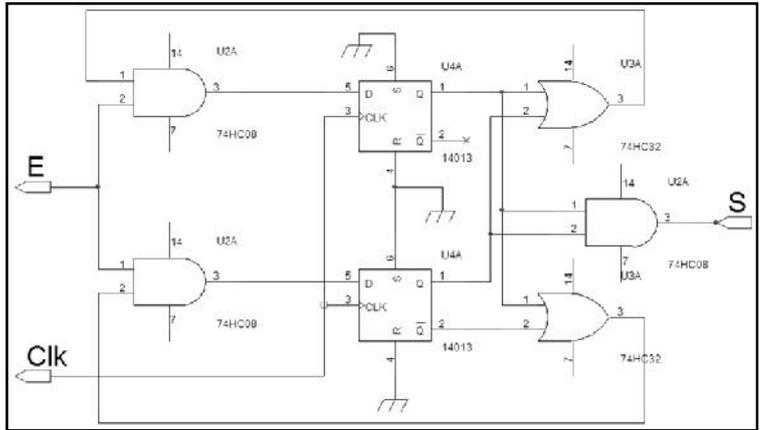
Nota:
Recordar que las funciones de transición de los biestables no eran un circuito comb. como tal

11

Apartado 2.1 Concepto de máquina de estados o autómeta

Nivel estructural:

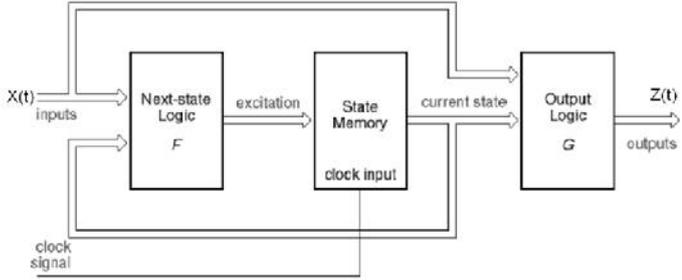
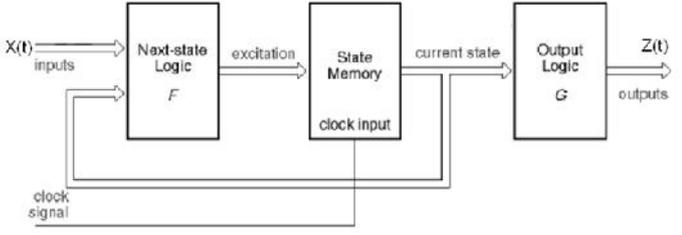
Es el circuito que refleja el comportamiento del autómeta. Pero ocurre que cuando planteamos un problema en términos secuenciales, esto es lo último que conocemos...



Este circuito es, precisamente, el detector de la secuencia "111".

12

	<p><i>Apartado 2.1 Concepto de máquina de estados o autómeta</i></p> <hr/> <ul style="list-style-type: none"> • <u>Más sobre máquinas de estados.</u> • Def.: Estados equivalentes Aquellos que para cada una de las entradas $X(t)$ posibles producen salidas $Z(t)$ iguales y conducen a estados $Q(t+1)$ iguales o equivalentes. Nota: Detectar estados equivalentes es importante para simplificar circuitos. • Def.: Autómetas equivalentes Aquellos en los que por cada estado en uno de los autómetas existe al menos un estado equivalente en el otro. Nota: <u>Todo autómeta de Mealy posee un autómeta de Moore equivalente, y viceversa.</u> Esto no significa que sea igual de sencillo un tipo de autómeta u otro.
13	

	<p><i>Apartado 2.1 Concepto de máquina de estados o autómeta</i></p> <hr/>  <p style="text-align: center;"> Autómeta de Mealy   Autómeta de Moore </p> 
14	

Apartado 2.1 Concepto de máquina de estados o autómeta

- TABLA DE ESTADOS de los autómetas de Mealy y Moore.

$Q(t) \backslash X(t)$	X_1	X_2	...	X_n
q_1	$f(q_1, X_1) / f(q_1, X_1)$	$f(q_1, X_2) / f(q_1, X_2)$...	$f(q_1, X_n) / f(q_1, X_n)$
q_2	$f(q_2, X_1) / f(q_2, X_1)$	$f(q_2, X_2) / f(q_2, X_2)$...	$f(q_2, X_n) / f(q_2, X_n)$
...
q_m	$f(q_m, X_1) / f(q_m, X_1)$	$f(q_m, X_2) / f(q_m, X_2)$...	$f(q_m, X_n) / f(q_m, X_n)$

$Q(t) \backslash X(t)$	X_1	X_2	...	X_n	$Z(t)$
q_1	$f(q_1, X_1)$	$f(q_1, X_2)$...	$f(q_1, X_n)$	$g(q_1)$
q_2	$f(q_2, X_1)$	$f(q_2, X_2)$...	$f(q_2, X_n)$	$g(q_2)$
...
q_m	$f(q_m, X_1)$	$f(q_m, X_2)$...	$f(q_m, X_n)$	$g(q_m)$

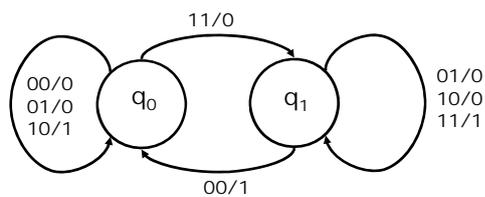
Nota:

Hay una correspondencia inmediata entre el diagrama y la tabla de estados.

15

Apartado 2.1 Concepto de máquina de estados o autómeta

- Ejemplos.



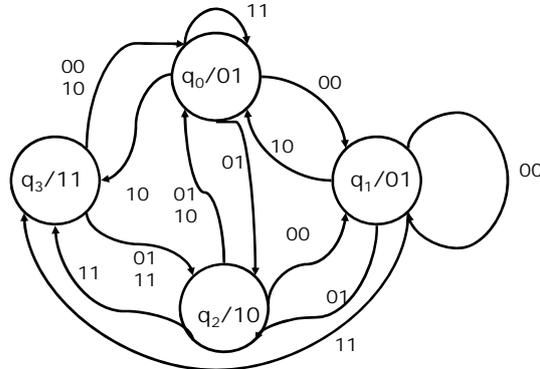
$Q(t) \backslash X(t)$	00	01	10	11
q_0	$q_0/0$	$q_0/1$	$q_1/1$	$q_1/0$
q_1	$q_0/1$	$q_1/0$	$q_1/0$	$q_1/1$

16

Apartado 2.1

Concepto de máquina de estados o autómeta

• Ejemplos.



Q(t) \ X(t)	00	01	10	11	Z ₁ Z ₂
q ₀	q ₁	q ₂	q ₃	q ₀	01
q ₁	q ₁	q ₂	q ₀	q ₃	10
q ₂	q ₁	q ₀	q ₀	q ₃	10
q ₃	q ₀	q ₂	q ₀	q ₂	11

17



Apartado 2.2

Diseño de máquinas de estados

• ¿Cómo diseñar un circuito secuencial?

Dos pasos generales:

- 1) Formular el problema en términos de un diagrama de estados (es decir, a un nivel comportamental).

Es un proceso creativo.

Recordar que un estado de la máquina es una situación de la memoria, que sirve para decidir cual será la situación siguiente al recibir la entrada.

- 2) A partir de aquí, la implementación del circuito es un proceso sistematizado, aunque conlleva ciertas decisiones:
 - Codificar los estados de la forma mas apropiada.
 - Asumir la estrategia adecuada para los estados no asignados.
 - Elegir los biestables adecuados a las especificaciones.

18

19	<p style="text-align: center;"><u>Apartado 2.2</u> <i>Diseño de máquinas de estados</i></p>
	<ul style="list-style-type: none"> • <u>Procedimiento para diseñar un circuito secuencial.</u> <p>Ejemplo:</p> <p>Por la única entrada de un circuito se recibe un tren de bits aleatorios. Construir el circuito que detecte, con un nivel alto a la salida, la secuencia "111".</p> <ol style="list-style-type: none"> 1. Enunciado del problema: detectar nº de entradas y de salidas. Situaciones del problema que requerirán memoria. 2. Obtención del diagrama de estados <ul style="list-style-type: none"> - Definición de los estados. En este ejemplo <ol style="list-style-type: none"> 1. Q0 estado inicial 2. Qi los demás estados, con i es el número de "1" alcanzado - Formulación del diagrama de estados <div style="text-align: center;"> </div>

20	<p style="text-align: center;"><u>Apartado 2.2</u> <i>Diseño de máquinas de estados</i></p>
	<ol style="list-style-type: none"> 3. Número de biestables necesarios "m". Se debe cumplir siempre que: $2^m \geq$ número de estados. <ul style="list-style-type: none"> - En el ejemplo necesitamos dos biestables 4. Codificación de los estados en función de los valores de los biestables (Q1, Q0) <ul style="list-style-type: none"> - En el ejemplo E0=(00), E1=(01), E2=(10) y E3=(11) - Normas de asignación: <ul style="list-style-type: none"> • Códigos con más 0s para los estados más referenciados. • Minimizar el número de biestables que cambian su estado en las transiciones de estado. - Estrategias con los estados no asignados <ol style="list-style-type: none"> 1. Riesgo mínimo: Evitar transiciones accidentales a ellos 2. Coste mínimo: Considerarlos como indeterminados en la tabla de transiciones (máxima minimización)

5. Tabla de transiciones/excitaciones de los biestables estudiados (sumario de todos los casos)

Q(t)	Q(t+1)	D(t)	K(t)	J(t)	T(t)
0	0	0	1	X	0
0	1	1	0	X	1
1	0	0	X	0	1
1	1	1	X	1	0

- Escogemos biestables tipo D.

21

6. Tabla de excitaciones, transiciones y salidas

X(t)	Q(t)		Q(t+1)		D(t)		Z(t)
	Q1(t)	Q0(t)	Q1(t+1)	Q0(t+1)	D1	D0	
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1

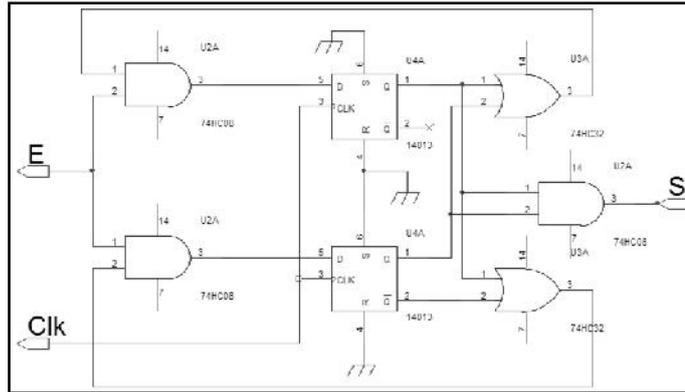
Punto de partida

Por inspección del diagrama de flujo

Se deduce de la tabla de excitación.

22

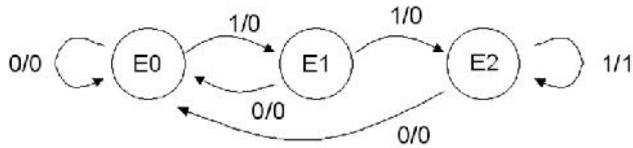
7. Simplificación (por Karnaugh) de las tres funciones: D0, D1 y S
 - $D1 = E \cdot (Q0 + Q1)$, $D0 = E \cdot (Q0' + Q1)$ y $S = Q0 \cdot Q1$
8. Realización del circuito



23

Resolver el problema anterior utilizando una máquina de Mealy, y biestables tipo J-K

- Diagrama de estados: aquí cada entrada lleva aparejada una salida



- Número de biestables necesarios para 3 estados: $m=2$
- Asignación de estados $E0=(00)$, $E1=(01)$ y $E2=(11)$ (nótese el criterio de minimizar el cambio de bits de uno a otro)
- Tabla del biestable J-K, por especificación del enunciado (ver ejemplo anterior)

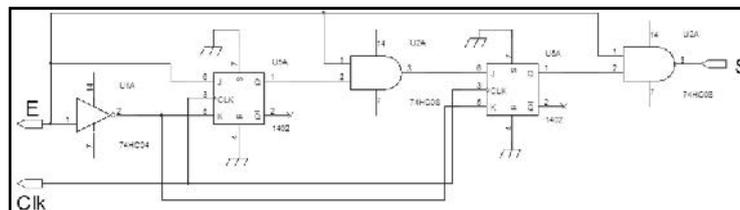
24

5. Tabla de excitaciones, transiciones y salidas

X(t)	Q(t)		Q(t+1)		J-K(t)				Z(t)
	Q1	Q0	Q1	Q0	J1	K1	J0	K0	S
0	0	0	0	0	0	X	0	X	0
0	0	1	0	0	0	X	X	1	0
0	1	0	X	X	X	X	X	X	X
0	1	1	0	0	X	1	X	1	0
1	0	0	0	1	0	X	1	X	0
1	0	1	1	1	1	X	X	0	0
1	1	0	X	X	X	X	X	X	X
1	1	1	1	1	X	0	X	0	1

25

- 6. Simplificación (por Karnaugh)
 - $J0=E$, $K0=K1=E'$, $J1=E \cdot Q0$ y $S=E \cdot Q1$
- 7. Realización del circuito



26

(ix)

Apartado 2.3

Análisis de circuitos secuenciales

- Ecuaciones de excitación

$$D0 = Q0 \cdot EN' + Q0' \cdot EN$$

$$D1 = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

- Ecuaciones características

Notación: $Q(t+1) \equiv Q^*$

$$Q0^* = D0$$

$$Q1^* = D1$$

- Sustituir las ecuaciones de excitación en las ecuaciones características, y obtener las ecuaciones de transición

$$Q0^* = Q0 \cdot EN' + Q0' \cdot EN$$

$$Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

- Ecuación de la salida (MAX): $MAX = Q1 \cdot Q0 \cdot EN$

29

Apartado 2.3

Análisis de circuitos secuenciales

Q1 Q0	EN	
	0	1
00	00	01
01	01	10
10	10	11
11	11	00
	Q1*	Q0*

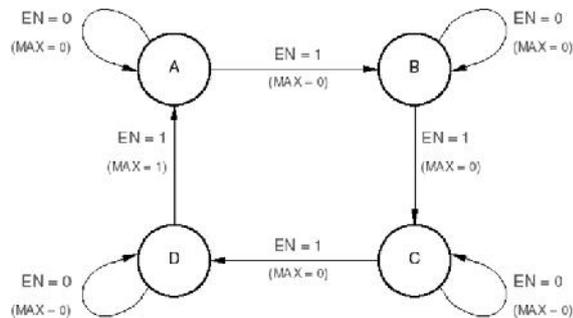
Tabla de transición

S	EN	
	0	1
A	A	B
B	B	C
C	C	D
D	D	A
	S*	

Tabla de estado

S	EN	
	0	1
A	A, 0	B, 0
B	B, 0	C, 0
C	C, 0	D, 0
D	D, 0	A, 1
	S*, MAX	

Tabla de estado/salida



30