Circuitos Electrónicos Digitales

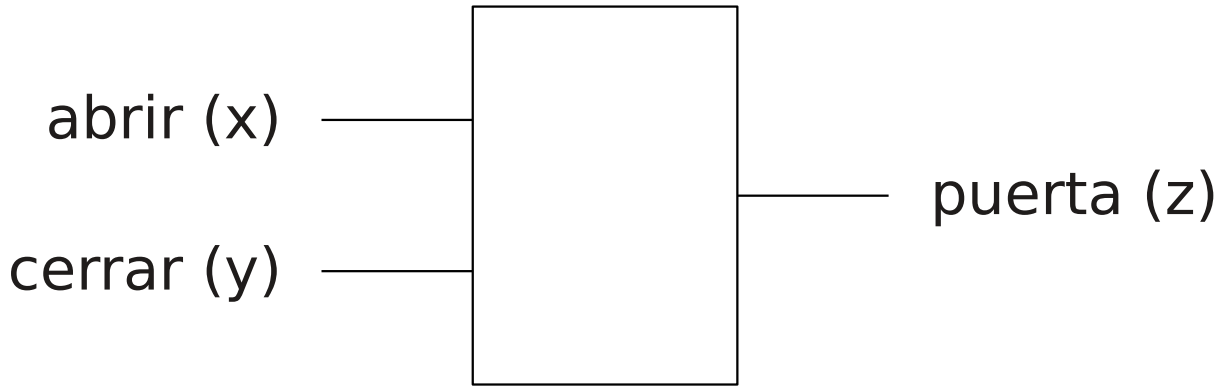
Análisis y diseño de circuitos secuenciales

Contenidos

1. Introducción
2. Biestables
3. Máquinas de estados finitos y circuitos secuenciales síncronos (CSS)
4. Diseño de CSS
5. Análisis de CSS

Introducción

- Diseñar un sistema de control de una puerta de garaje con dos pulsadores (no interruptores) situados a cierta distancia:
  - ⇨ x: abre la puerta
  - ⇨ y: cierra la pueta



Introducción

- Muchos problemas prácticos no pueden resolverse sólo mediante definición de funciones de conmutación.
- Se necesita que la acción del sistema tenga en cuenta las entradas y el estado del sistema.
- Para almacenar un estado son necesarios nuevos elementos de circuito: elementos de memoria.
- En este tema
  - ⇨ Elementos de memoria (biestables).
  - ⇨ Concepto de estado y de circuito secuencial.
  - ⇨ Técnicas de diseño y análisis de circuitos secuenciales.

Biestables

- Introducción
- Biestables
  - ⇨ Introducción
  - ⇨ Biestable SR asíncrono
  - ⇨ Biestables síncronos. Señal de reloj
  - ⇨ Otros biestables síncronos
  - ⇨ Entradas asíncronas de los biestables
  - ⇨ Consideraciones temporales
- Máquinas de estados finitos y circuitos secuenciales síncronos (CSS)
- Diseño de CSS
- Análisis de CSS

Biestables

- Los biestables son circuitos electrónicos que pueden asumir uno de dos estados estables que muestran en sus salidas
- Son el elemento básico de los dispositivos de memoria
- Poseen una o más entradas de control que hacen que conmute entre ambos estados estables
- Con n biestables se pueden “recordar” 2^n estados

Biestable SR. Representación formal

Símbolos

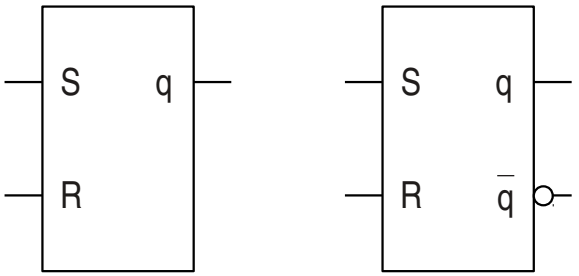


Diagrama de estados

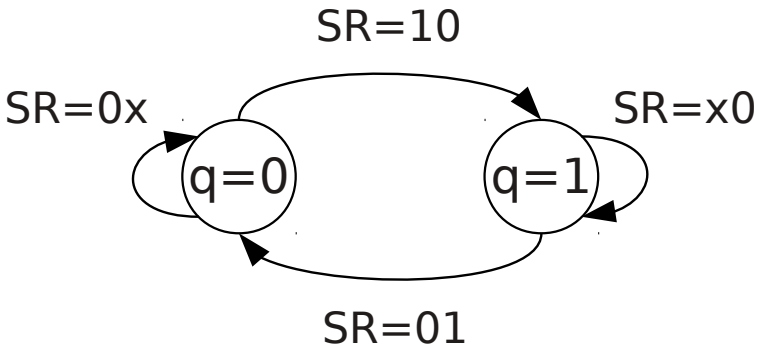


Tabla de estados

q \ SR	00	01	11	10
0	0	0	-	1
1	1	0	-	1

Q

Tabla de excitación

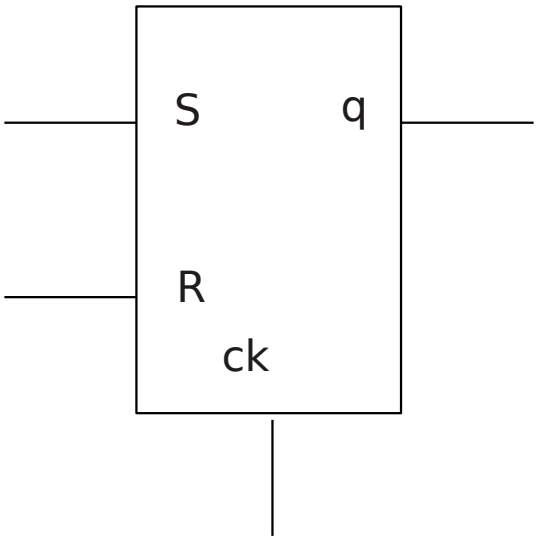
q → Q	SR
0 → 0	0x
0 → 1	10
1 → 0	01
1 → 1	x0

Biestables síncronos

- En circuitos reales con miles (o millones) de biestables es muy útil que todos cambien de estado a la vez: simplificación del proceso de diseño.
- Los cambios de estado se producen “sincronizados” con una “señal de reloj” (CK)
- Tipos de sincronización:
  - ⇨ Por nivel: cuando CK tiene un valor determinado, alto (1) o bajo (0).
  - ⇨ Por flanco: cuando CK cambia de 0 a 1 (flanco de subida) o de 1 a 0 (flanco de bajada).
  - ⇨ Maestro-esclavo: dos biestables consecutivos disparados por niveles opuestos.
- Flanco: más conveniente.
  - ⇨ Determina de forma precisa el instante de cambio
  - ⇨ Minimiza errores en los circuitos

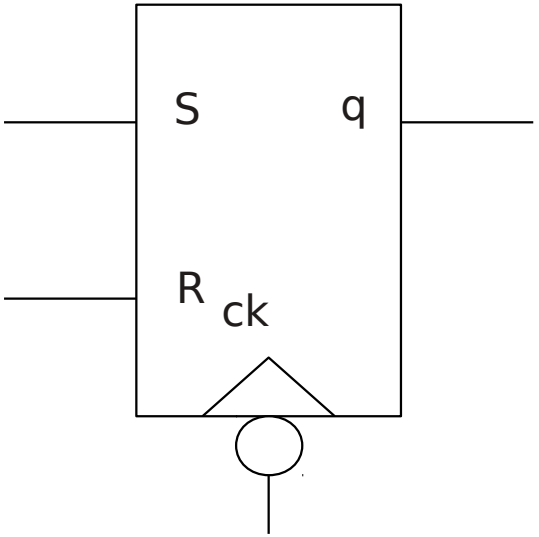
Biestables síncronos

Disp. por nivel



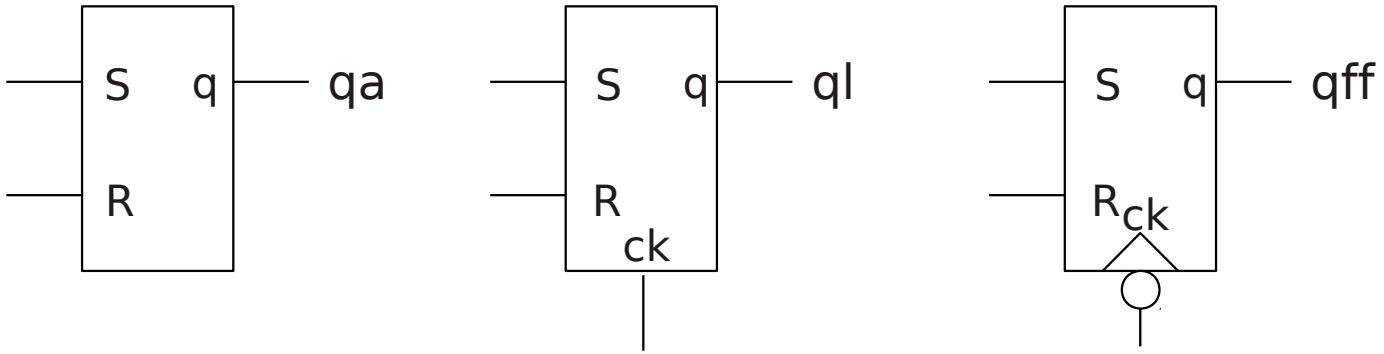
El cambio de estado sólo se produce cuando ck=1 (nivel alto) o ck=0 (nivel bajo)

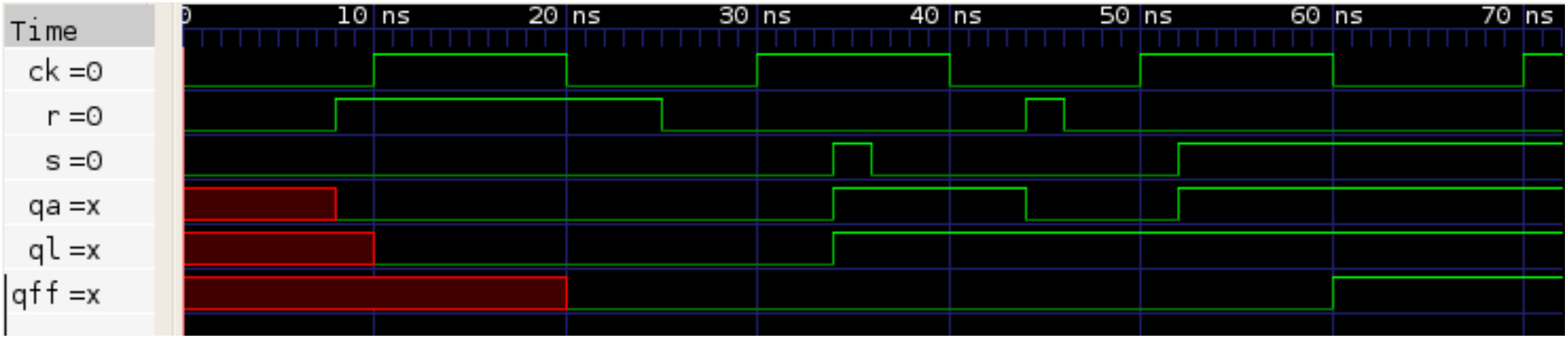
Disp. por flanco



El cambio de estado sólo se produce cuando ck cambia de 1 a 0 (flanco de bajada) o de 0 a 1 (flanco de subida). Mejor precisión en el cambio de estado

Biestables síncronos





Otros biestables síncronos

- SR
- JK
  - ⇨ Similar a SR: J~S, K~R
  - ⇨ Función de complemento para J=K=1
- D
  - ⇨ Una única entrada que indica el próximo estado.
  - ⇨ Fácil de usar e implementar.
- T
  - ⇨ Una única entrada que permite complementar el estado.
  - ⇨ Útil en aplicaciones especiales.

Biestable JK

Símbolos

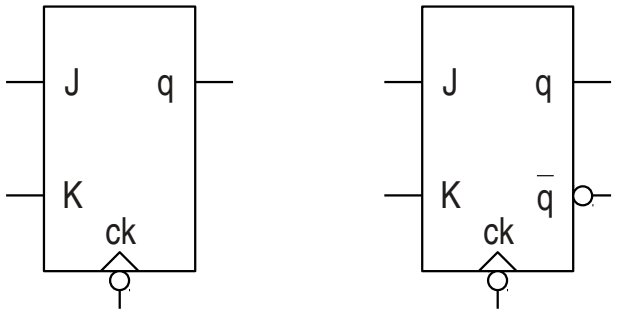


Diagrama de estados

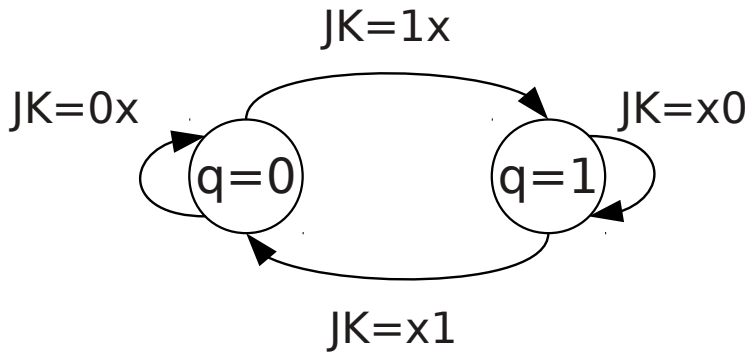


Tabla de excitación

q → Q	JK
0 → 0	0x
0 → 1	1x
1 → 0	x1
1 → 1	x0

Tabla de estados

q \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q

Biestable D

Símbolos

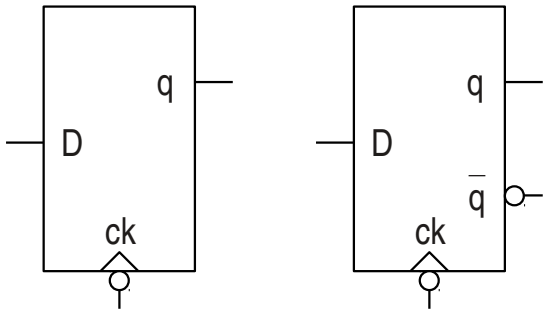


Diagrama de estados

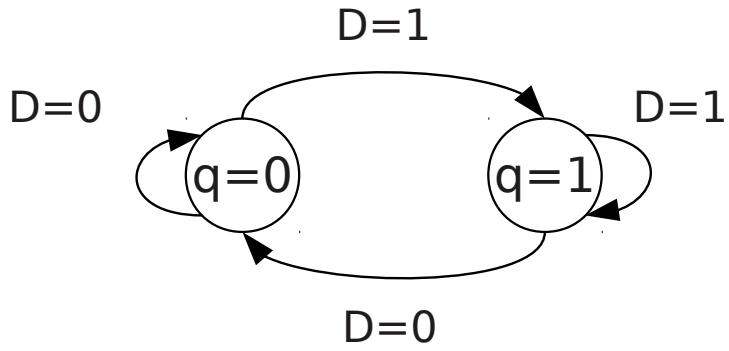


Tabla de excitación

q → Q	D
0 → 0	0
0 → 1	1
1 → 0	0
1 → 1	1

Tabla de estados

q \ D	0	1
0	0	1
1	0	1

Q

Biestable T

Símbolos

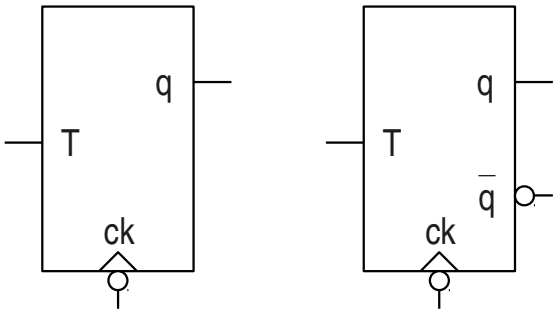


Diagrama de estados

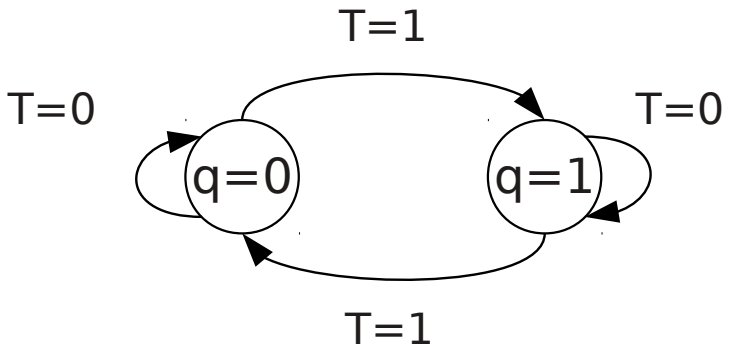


Tabla de excitación

q → Q	T
0 → 0	0
0 → 1	1
1 → 0	1
1 → 1	0

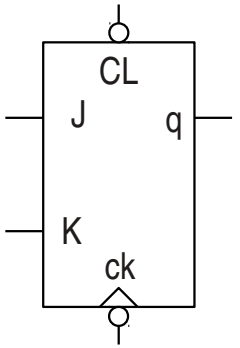
Tabla de estados

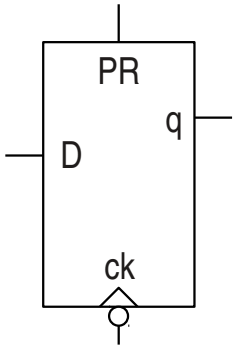
q \ T	0	1
0	0	1
1	1	0

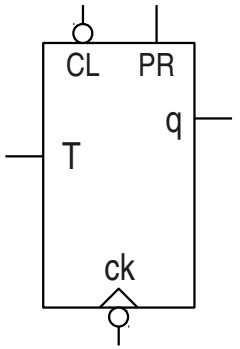
Q

Entradas asíncronas de los biestables

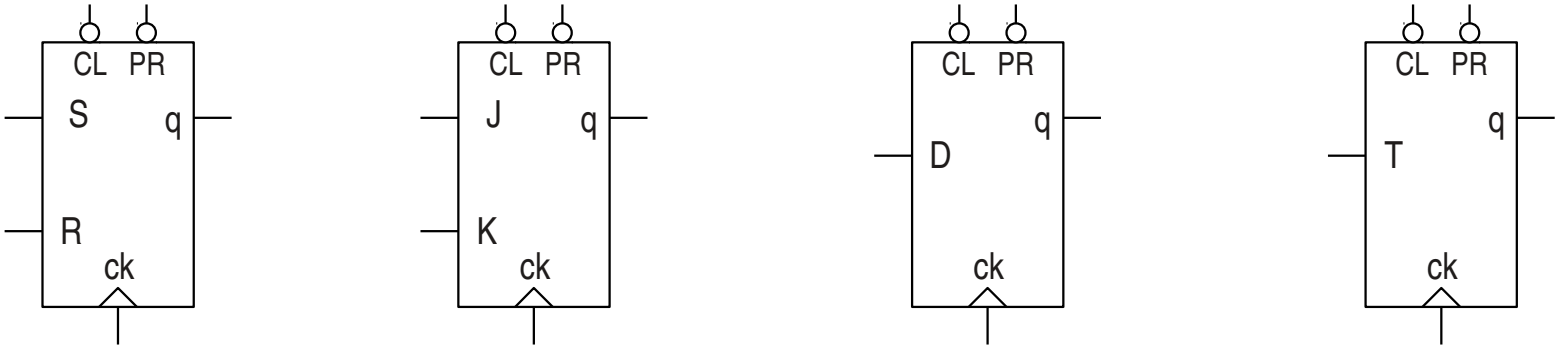
- Permiten cargar un estado determinado de forma sencilla
  - ⇨ CL (clear): puesta a cero
  - ⇨ PR (preset): puesta a uno
- Operan inmediatamente cuando se activan:
  - ⇨ Activas en nivel bajo (0)
  - ⇨ Activas en nivel alto (1)
- Las entradas asíncronas tienen prioridad sobre las síncronas (J, K, D, T, ...)
- Resuelven el problema de la iniciación en los circuitos digitales complejos
  - ⇨ millones de biestables
  - ⇨ necesidad de partir de un estado conocido

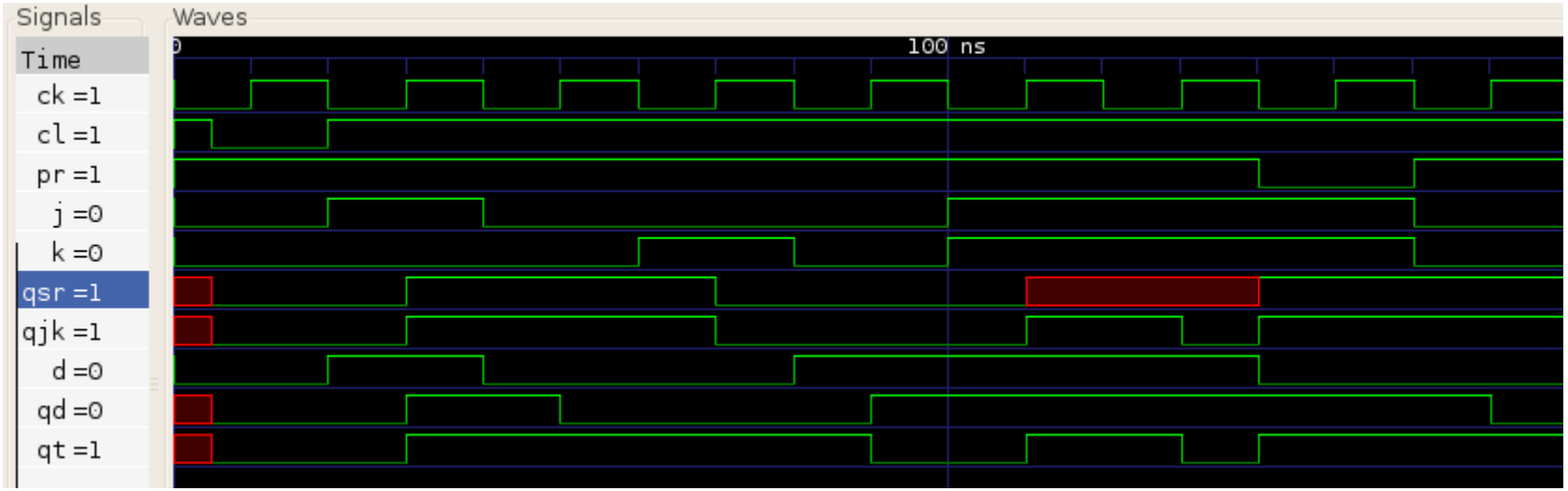






Entradas asíncronas de los biestables



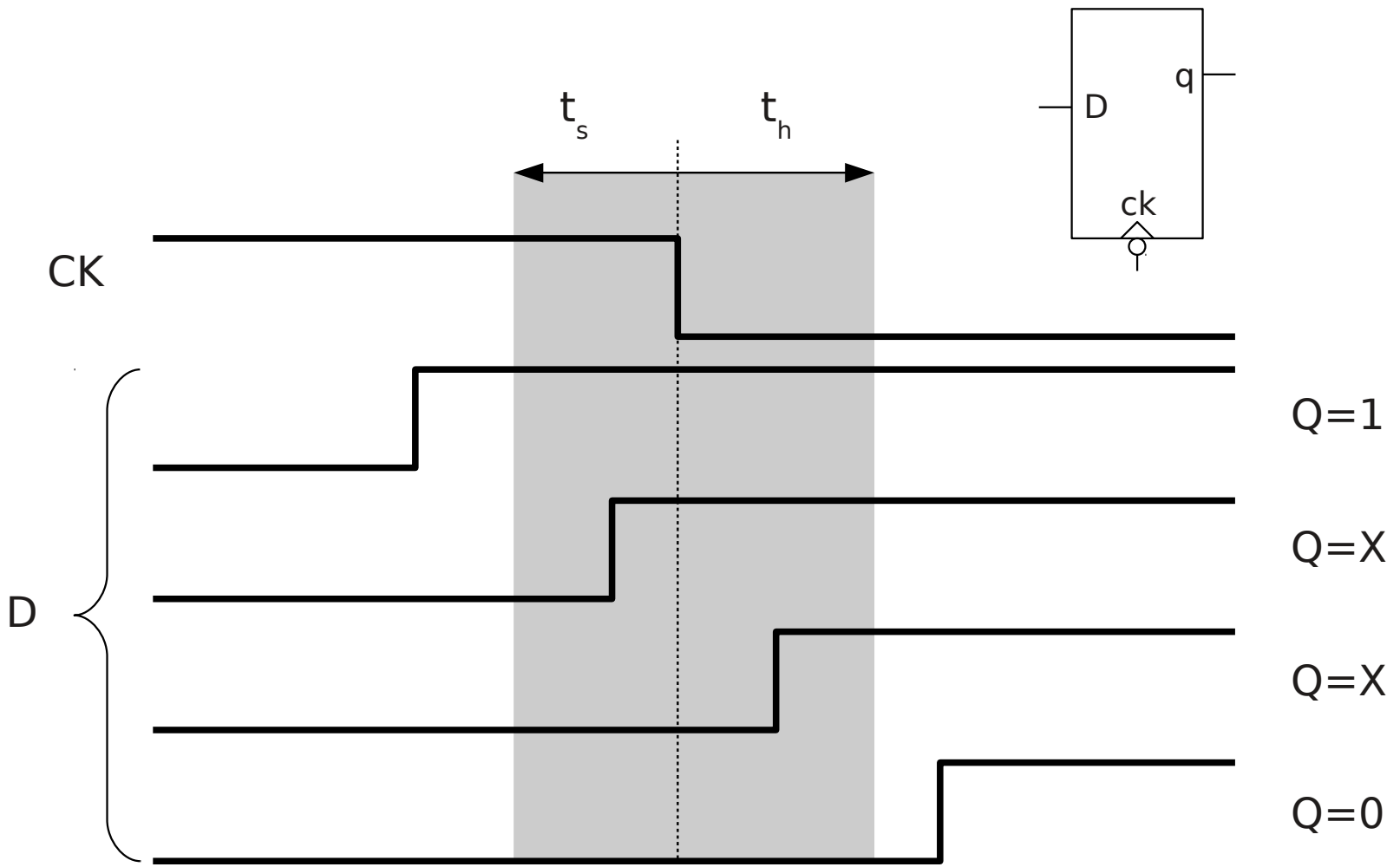


S=J, R=K, T=D

Consideraciones temporales

- Las entradas síncronas no deben cambiar en un entorno del flanco activo de la señal de reloj para evitar cambios de estado no predecibles.
- Tiempo de set-up (ts)
  - ⇨ Las entradas deben estar fijas desde un tiempo antes del flanco
- Tiempo de hold (th)
  - ⇨ Las entradas deben permanecer fijas un tiempo después del flanco.

Consideraciones temporales



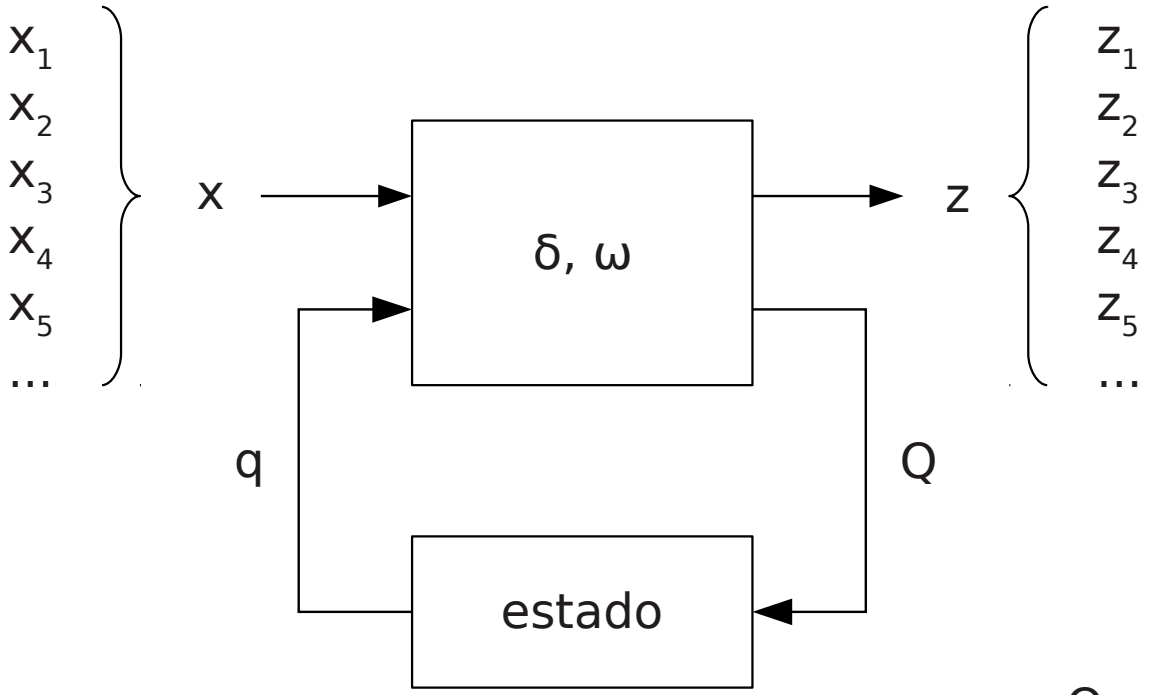
Máquinas de estados finitos y CSS

- Introducción
- Biestables
- Máquinas de estados finitos y circuitos secuenciales síncronos (CSS)
  - ⇨ Concepto de máquina de estados
  - ⇨ Circuitos secuenciales síncronos
  - ⇨ Representaciones formales
  - ⇨ Aplicaciones
- Diseño de CSS
- Análisis de CSS

Concepto de máquinas de estados

- "Máquina determinista de estados finitos"
- Componentes
  - ⇨ Conjunto finito de estados ($q \in S$)
  - ⇨ Conjunto de símbolos de entrada ($x \in \Sigma$)
  - ⇨ Conjunto de símbolos de salida ($z \in \Gamma$)
  - ⇨ Función de próximo estado (δ): $Q = \delta(q, x)$
  - ⇨ Función de salida (ω):
    - ✔ Modelo Mealy: $z = \omega(q, x)$
    - ✔ Modelo Moore: $z = \omega(q)$
- Operación
  - ⇨ Por la entrada llegan símbolos en secuencia. Para cada símbolo de entrada la máquina genera un símbolo de salida.
  - ⇨ Tras cada símbolo de entrada la máquina puede pasar a un nuevo estado.

Concepto de máquinas de estados



S_1
S_2
S_3
S_4
...

$Q = \delta(q, x)$
$z = \omega(q, x)$

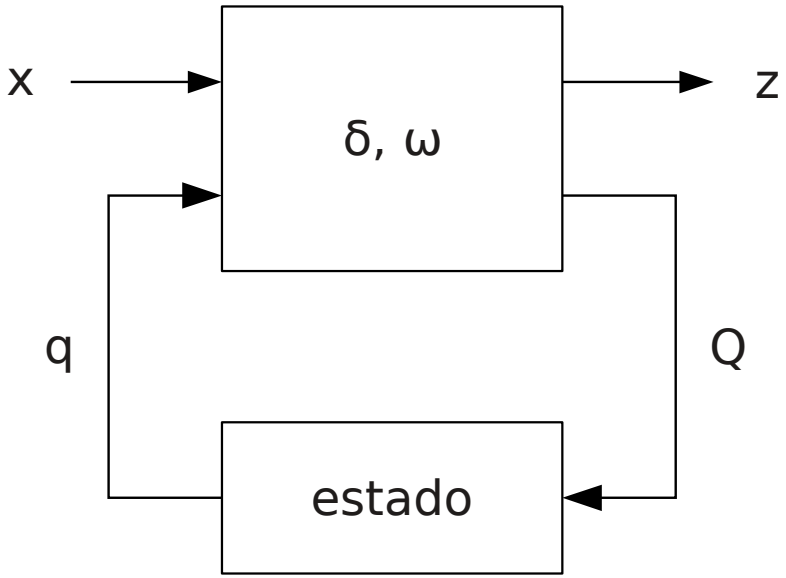
Concepto de máquinas de estados. Propiedades

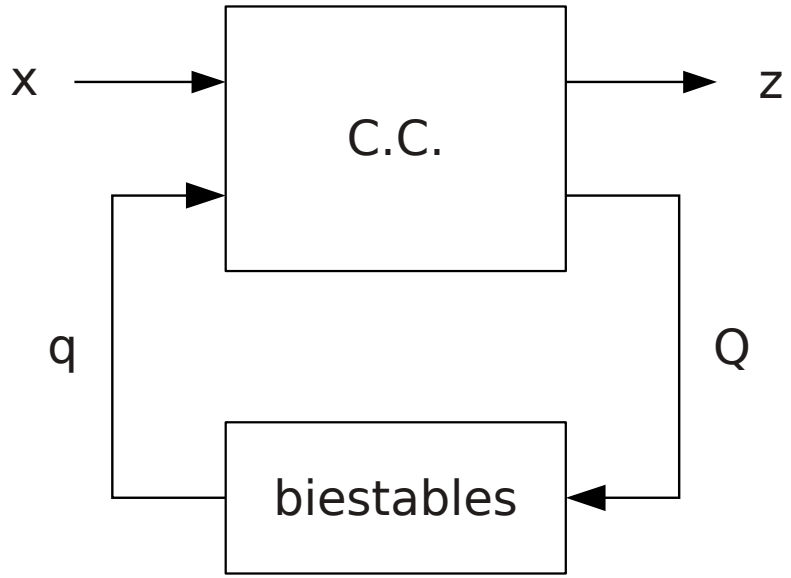
- Partiendo de un estado determinado, las máquinas de estados deterministas generan siempre la misma secuencia de salida para la misma secuencia de entrada.
- Dos máquinas de estados son equivalentes si generan las mismas secuencias de salida para las mismas secuencias de entrada.
- Las máquinas de estados 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.

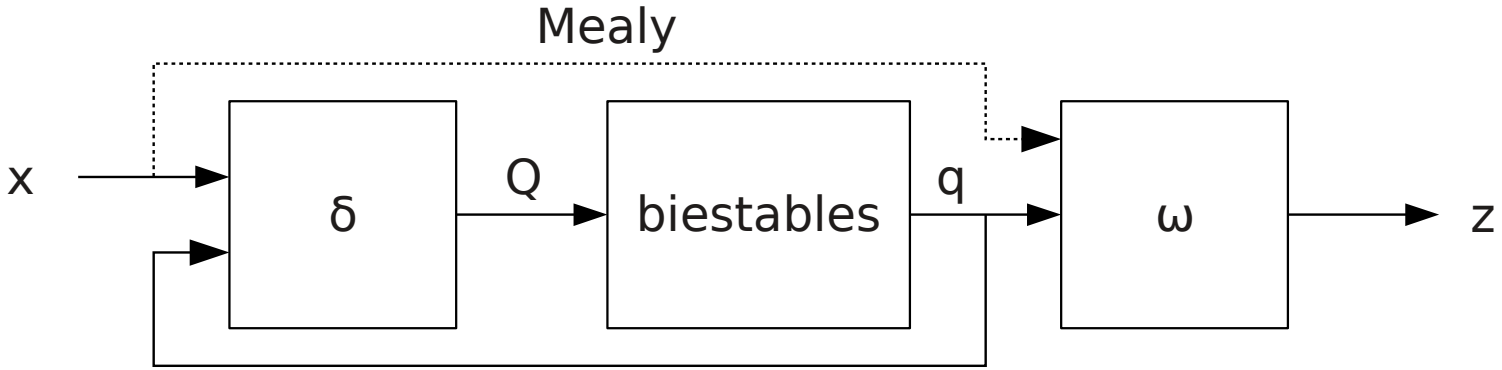
Circuitos secuenciales síncronos

- Las máquinas de estados finitos son un buen instrumento para modelar circuitos digitales con memoria.
- Los circuitos digitales con memoria son una tecnología adecuada para implementar máquinas de estados finitos.
  - ⇨ Entradas/salidas: señales digitales de 1 o más bits.
  - ⇨ Estado: valor almacenado en los biestables
  - ⇨ Función de próximo estado: funciones combinacionales que actúan sobre las entradas de los biestables
  - ⇨ Función de salida: función combinacional
- 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.

Circuitos secuenciales síncronos





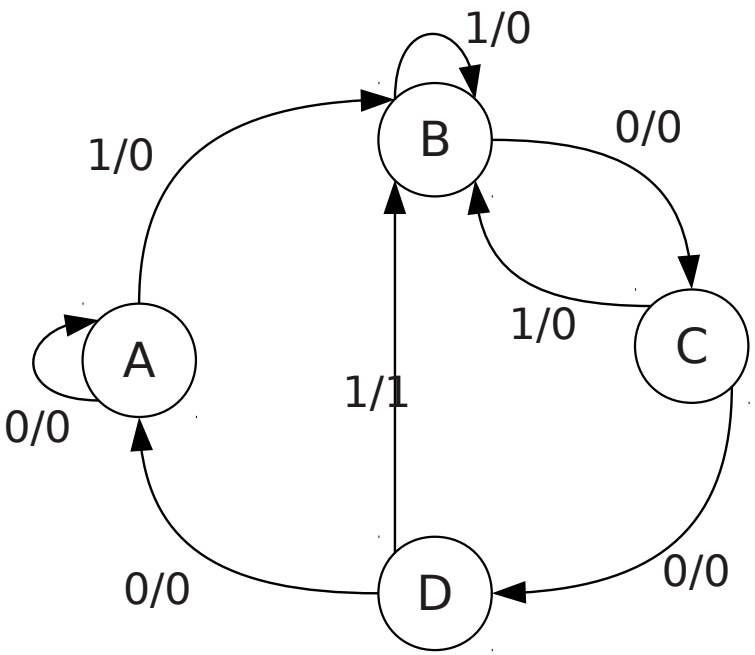


Representaciones formales

1. Diagramas de estados

2. Tabla de estados

Diagrama de estados. Mealy

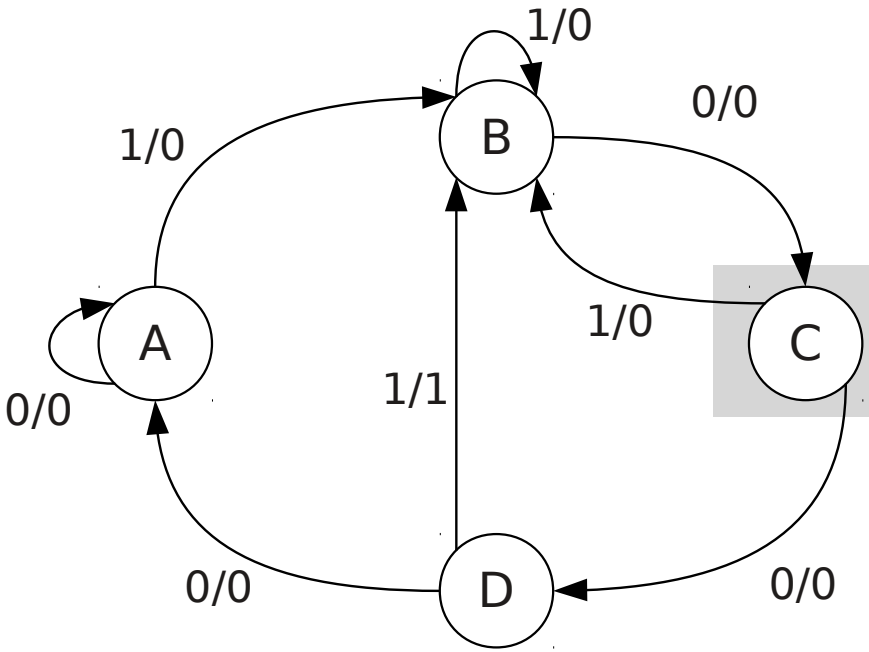


- Nodos
  - ⇨ Representan los estados. Se nombran de forma más o menos indentificativa. Ej. {A, B, C, ...}, {S0, S1, S2, ...}, {espera, comienzo, recibiendo, ...}
- Arcos
  - ⇨ Indican las posibles transiciones desde cada estado (S).
  - ⇨ Se nombran con x/z:
    - ✔ x: valor de entrada que provoca la transición desde el estado S.
    - ✔ z: valor de salida generado en el estado S cuando la entrada vale x.

Tabla de estados. Mealy

- Información equivalente al diagrama de estados en forma de tabla de doble entrada (filas y columnas)
  - ⇨ Posibles estados en filas
  - ⇨ Posibles valores de entradas en columnas
  - ⇨ Próximo estado y salida en cada celda
- Cada nodo del diagrama y los arcos que salen de él se corresponden a una fila de la tabla de estados.
- Pasar del diagrama de estados a la tabla de estados y viceversa es inmediato.

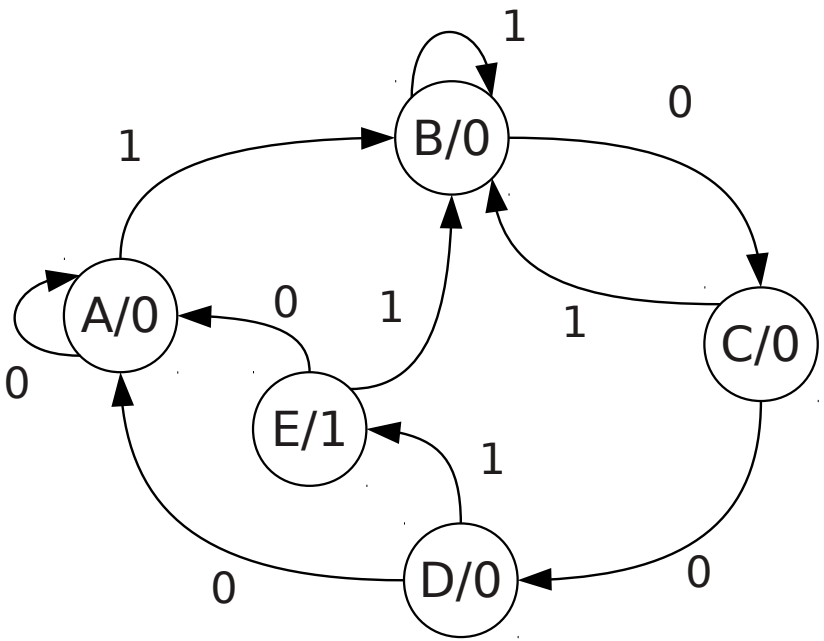
Tabla de estados. Mealy



S \ x	0	1
A	A,0	B,0
B	C,0	A,0
C	D,0	B,0
D	A,0	B,1

Q,z

Diagrama de estados. Moore

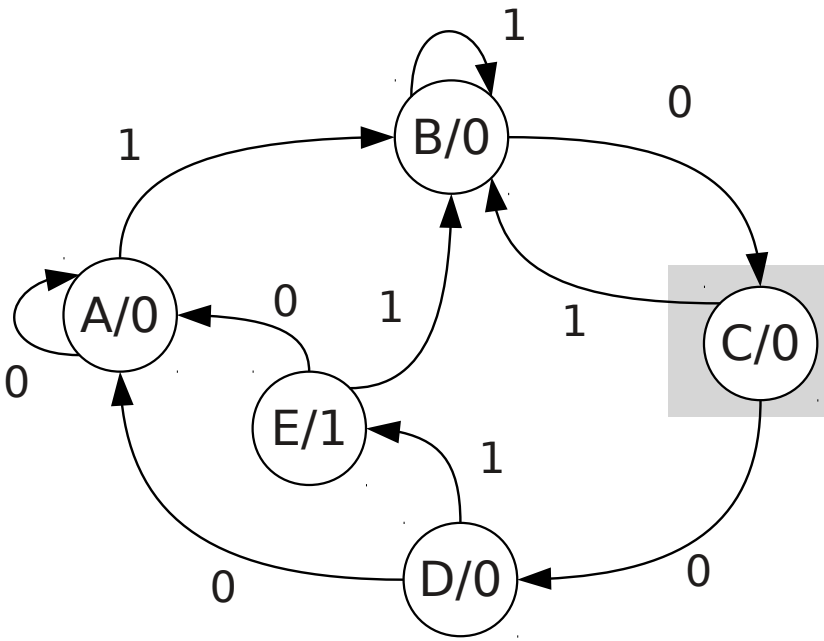


- Nodos
  - ⇨ Representan los estados. Se nombran de forma más o menos indentificativa. Ej. {A, B, C, ...}, {S0, S1, S2, ...}, {espera, comienzo, recibiendo, ...}
  - ⇨ Cada estado lleva asociado un valor de salida correspondiente.
- Arcos
  - ⇨ Indican las posibles transiciones desde cada estado (S).
  - ⇨ Se nombran con x: valor de entrada que provoca la transición desde el estado S.

Tabla de estados. Moore

- Información equivalente al diagrama de estados en forma de tabla de doble entrada (filas y columnas)
  - ⇨ Posibles estados en filas
  - ⇨ Posibles valores de entradas en columnas
  - ⇨ Salida asociada al estado en la última columna (opcionalmente misma salida para cada entrada al estilo Mealy)
- Cada nodo del diagrama y los arcos que salen de él se corresponden a una fila de la tabla de estados.
- Pasar del diagrama de estados a la tabla de estados y viceversa es inmediato.

Tabla de estados. Moore



S \ x	0	1	z
A	A	B	0
B	C	A	0
C	D	B	0
D	A	E	0
E	A	B	1
	Q		

Aplicaciones de los circuitos secuenciales síncronos

- Detectores de secuencia
  - ⇨ La salida se activa sólo en caso de que aparezca una determinada secuencia a la entrada.
- Generadores de secuencia
  - ⇨ La salida genera una secuencia fija o variable en función de la entrada.

Aplicaciones de los circuitos secuenciales síncronos

- Unidades de control
  - ⇨ Las entradas modifican el estado y el estado define la actuación sobre un sistema externo (control de una barrera, control de temperatura, control de presencia, control de nivel de líquidos, etc.)
- Procesamiento secuencial
  - ⇨ La secuencia de salida es el resultado de aplicar alguna operación a la secuencia de entrada (cálculo de la paridad, suma de una constante, producto por una constante, codificación/decodificación secuencial en general).

Diseño de CSS

- Introducción
- Biestables
- Máquinas de estados finitos y circuitos secuenciales síncronos (CSS)
- Diseño de CSS
  - ⇨ Objetivos y procedimientos
  - ⇨ Procedimiento de diseño manual
  - ⇨ Procedimiento con herramientas de diseño
- Análisis de CSS

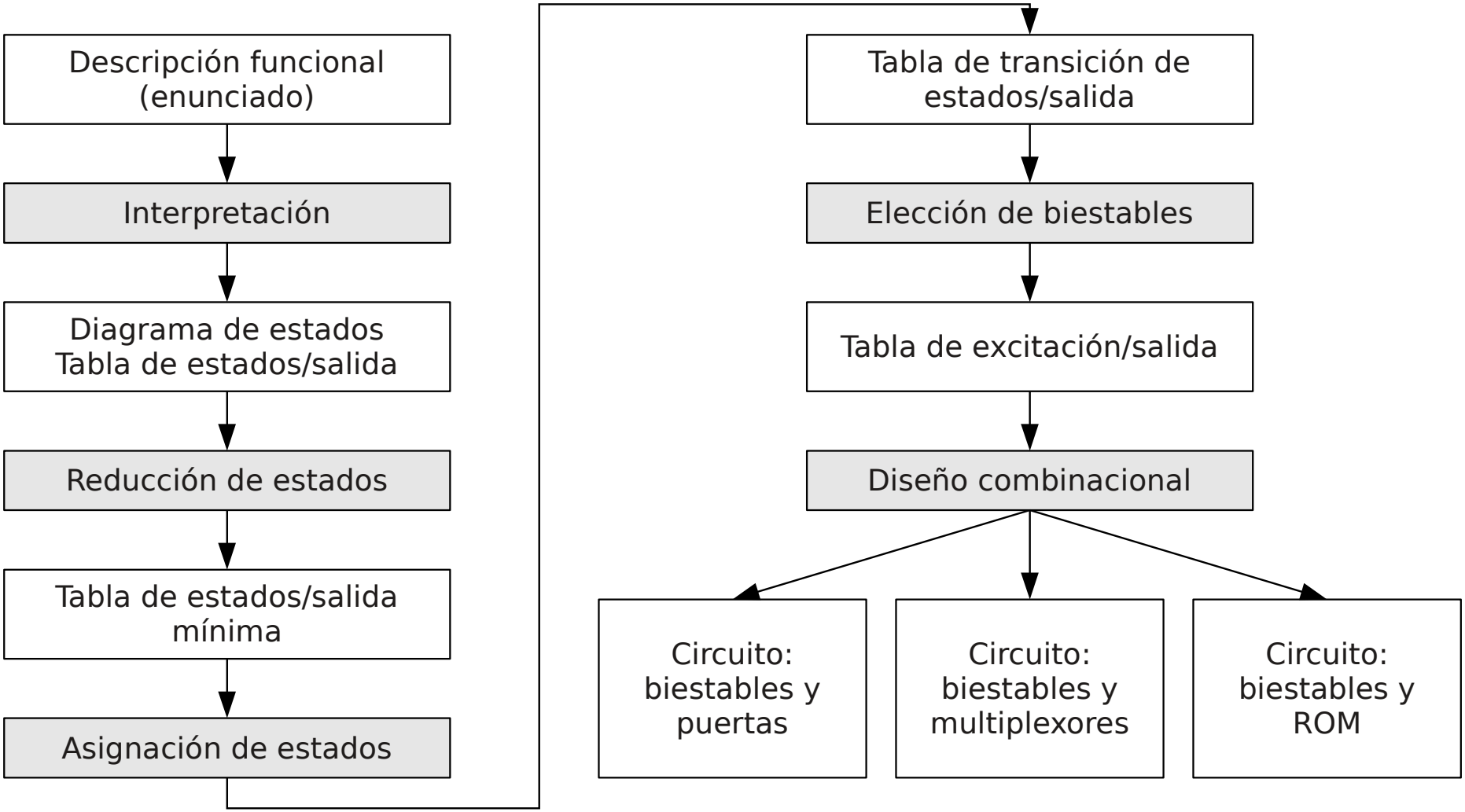
Objetivo

- Objetivo
  - ⇨ Definir una máquina de estados que resuelva un problema dado.
  - ⇨ Implementar la máquina de estados mediante un circuito secuencial síncrono.
- Coste
  - ⇨ Habitualmente, el proceso de diseño va dirigido por consideraciones de coste y de optimización de recursos.
  - ⇨ Ejemplo de criterios
    - ✔ Minimización del número de elementos de memoria
    - ✔ Minimización de componentes
    - ✔ Frecuencia de operación
    - ✔ Consumo de energía
  - ⇨ Compromiso entre diferentes criterios

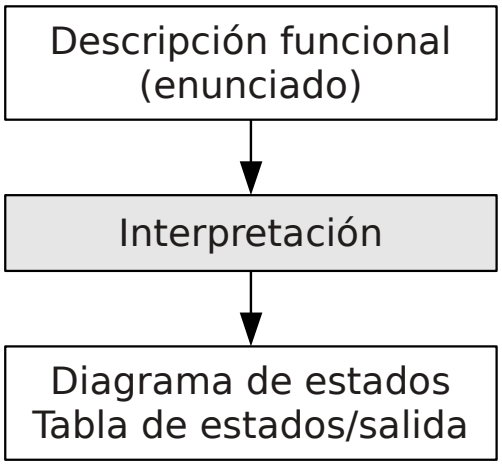
Procedimientos

- Procedimiento manual
  - ⇨ Realizable con lápiz y papel.
  - ⇨ Comienza describiendo el problema formalmente mediante un diagrama o tabla de estados.
  - ⇨ A partir del diagrama de estados se van obteniendo diversas representaciones hasta llegar al circuito digital.
- Procedimiento con herramientas de diseño
  - ⇨ Emplea herramientas informáticas.
  - ⇨ A partir del enunciado del problema o el diagrama de estados, se hace una descripción formal en un LDH.
  - ⇨ Se emplean herramientas de simulación para comprobar que la descripción del sistema es correcta.
  - ⇨ Se emplean herramientas de síntesis automática para obtener el circuito final.

Procedimiento manual



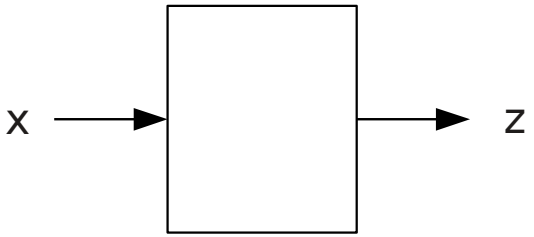
Interpretación



- Es la fase más importante del diseño
- Es la fase menos sistemática
- Procedimiento/consejos
  - ⇨ Definir claramente entradas y salidas.
  - ⇨ Elegir Mealy o Moore según características del problema (sincronización de la salida)
  - ⇨ Identificar y definir los estados adecuados de la forma más general posible
  - ⇨ Establecer las transiciones y salidas necesarias
  - ⇨ Capturar todos los detalles del problema en la máquina de estados
  - ⇨ Comprobar el diagrama con una secuencia de entrada típica

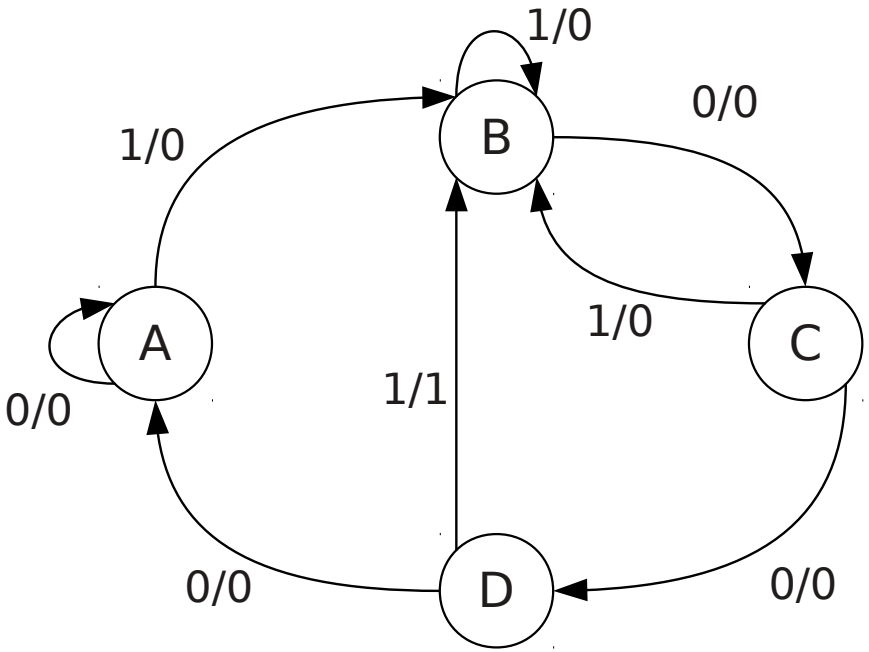
Interpretación

- Ejemplo
  - ⇨ Diseñe un circuito con una entrada x y una salida z que detecte la aparición de la secuencia "1001" en la entrada. Cuando esto ocurre se activará la salida (z=1). El último "1" de una secuencia puede considerarse también el primer "1" de una secuencia posterior (detector con solapamiento).




```
x: 00100111000011101001001001010011...
z: 00000100000000000000100100100010...
```

Interpretación

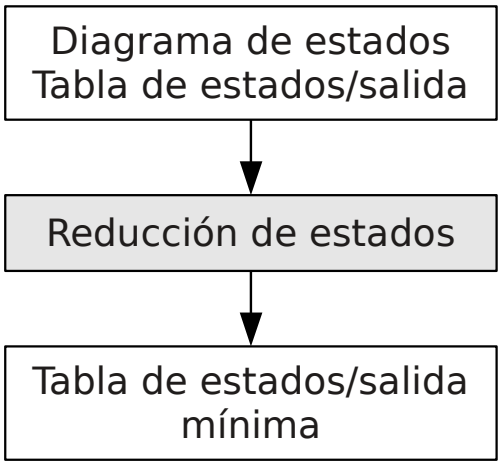


S \ x	0	1
A	A,0	B,0
B	C,0	A,0
C	D,0	B,0
D	A,0	B,1

Q,z

A: esperando llegada primer bit "1"
B: bit 1º correcto, esperando "0"
C: bit 2º correcto, esperando "0"
D: bit 3º correcto, esperando "1"

Reducción de estados



- Objetivo:
  - ⇨ Eliminación de estados redundantes.
  - ⇨ Reducción del coste en biestables y lógica combinacional.

Estados equivalentes:

Dos estados p y q son equivalentes si cualquier secuencia de entrada aplicada partiendo del estado p genera exactamente la misma salida que la misma secuencia aplicada partiendo del estado q.

Dos estados p y q son equivalentes si y sólo si:
Los próximos estados de p y q son idénticos o equivalentes para todos los valores de las entradas
Los valores de salida son los mismos para todos los valores de las entradas.

En una tabla de estados mínima no hay estados equivalentes.

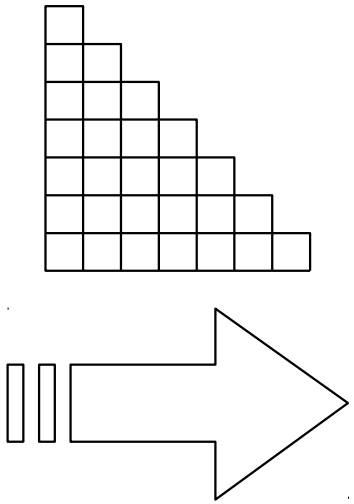
Reducción de estados. Procedimiento

- A partir de la tabla de estados se identifican los estados que pueden ser compatibles comparando todas las posibles parejas de estados.
- Tabla de estados compatibles: ayuda a identificar estados compatibles y las condiciones necesarias para la compatibilidad.
- Una vez identificadas todas la compatibilidades en la tabla de estados compatibles se agrupan los estados compatibles (clases de equivalencia).
- Se genera una nueva tabla de estados eligiendo un representante de cada clase de equivalencia.

Reducción de estados. Ejemplo 1

S \ x	0	1
A	B,0	C,0
B	D,0	E,0
C	G,0	E,0
D	H,0	F,0
E	G,0	A,0
F	G,1	A,0
G	D,0	C,0
H	H,0	A,0

NS, z



S \ x	0	1
a	b,0	a,0
b	d,0	a,0
d	h,0	f,0
f	b,0	a,0
h	h,0	a,0

NS, z

Reducción de estados. Ejemplo 2

S \ x	0	1
A	A,0	B,0
B	C,0	A,0
C	D,0	B,0
D	A,0	B,1

Q,z

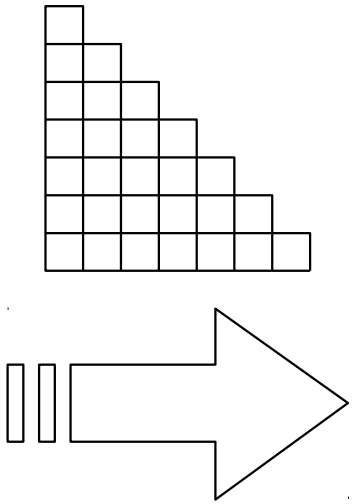
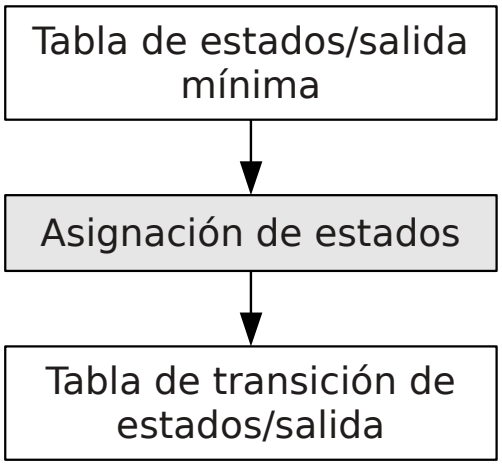


Tabla de estados minima

Asignación de estados



- Objetivo:
  - ⇨ Asignar valores binarios a los estados (codificación de estados) para su almacenamiento en biestables.
- Elección:
  - ⇨ Afecta al resultados final: número de componentes, tamaño, velocidad de operación, consumo de energía.
  - ⇨ Elección diferente según el objetivo (criterio de coste)
- Opciones
  - ⇨ Algoritmos complejos
  - ⇨ Asignación arbitraria
  - ⇨ Un biestable por estado (códificación one-hot)

Asignación de estados

Tabla de estados/salida

S \ x	0	1
A	A,0	B,0
B	C,0	A,0
C	D,0	B,0
D	A,0	B,1

Q,z

Asignación de estados

S	q_1q_0
A	00
B	01
C	11
D	10

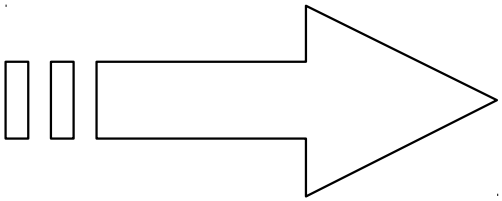
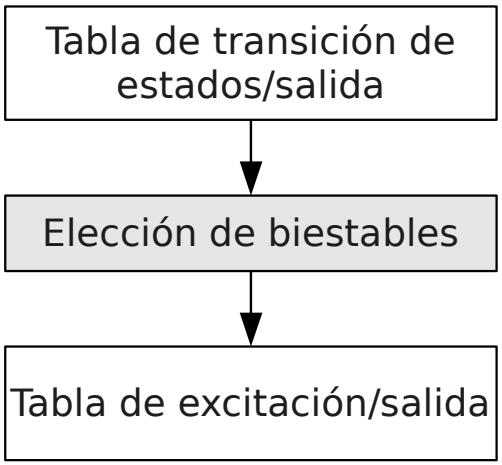


Tabla de transición de estados/salida

q_1q_2 \ x	0	1
00	00,0	01,0
01	11,0	00,0
11	10,0	01,0
10	00,0	01,1

Q,z

Elección de biestables



- Objetivo
  - ⇨ Seleccionar qué tipo de biestables almacenarán los bits del estado codificado.
- Opciones
  - ⇨ JK: reduce el coste de la parte combinacional.
  - ⇨ RS: más simple que el JK pero menos flexible.
  - ⇨ D: facilita el diseño, reduce el número de conexiones.
  - ⇨ T: más conveniente en aplicaciones específicas (contadores)

Elección de biestable. Ej: JK

Tabla de transición de estados/salida

q_1q_2 \ x	0	1
00	00,0	01,0
01	11,0	00,0
11	10,0	01,0
10	00,0	01,1

Q_1,Q_2,z

Tabla de excitación

q → Q	JK
0→ 0	0x
0 → 1	1x
1 → 0	x1
1→ 1	x0

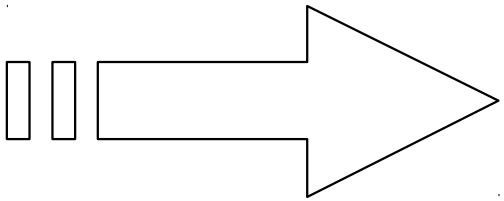


Tabla de excitación/salida

q_1q_2 \ x	0	1
00	0x,0x,0	0x,1x,0
01	1x,x0,0	0x,0x,0
11	x0,0x,0	0x,x0,0
10	0x,0x,0	0x,1x,1

J_1K_1,J_2K_2,z

Elección de biestable. Ej: D

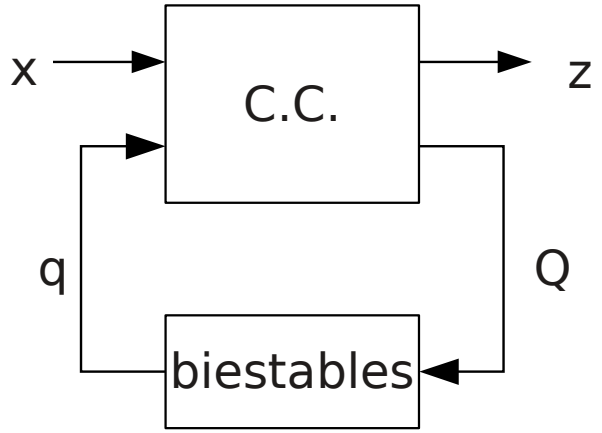
- En el biestable D:
  - ⇨ Q = D
  - ⇨ D = Q

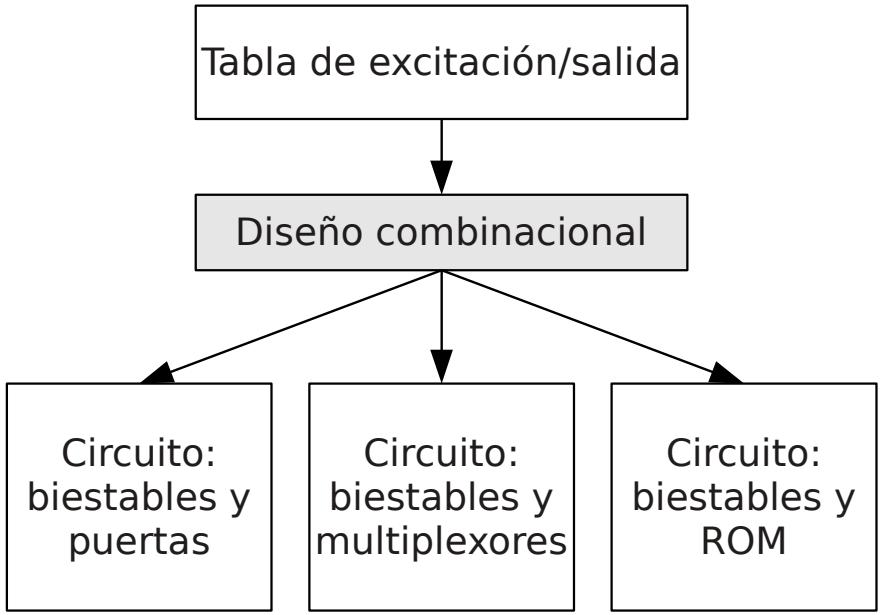
Tabla de transición de estados/salida ⟺ Tabla de excitación/salida

q_1q_2 \ x	0	1
00	00,0	01,0
01	11,0	00,0
11	10,0	01,0
10	00,0	01,1

Q,z
D,z

Diseño de la parte combinacional





- La tabla de excitación/salida es una especificación de la parte combinacional.
- La implementación se realiza mediante cualquiera de las técnicas de diseño de C.C.
  - ⇨ Dos niveles de puertas
  - ⇨ Subsistemas: multiplexores, decodificadores, etc.
  - ⇨ Etc.

Parte combinacional. Ejemplo

q_1q_2 \ x	0	1
00	0x,0x,0	0x,1x,0
01	1x,x0,0	0x,0x,0
11	x0,0x,0	0x,x0,0
10	0x,0x,0	0x,1x,1

J_1K_1,J_2K_2,z

q_1q_2 \ x	0	1
00	0	0
01	1	0
11	x	0
10	0	0

J_1

q_1q_2 \ x	0	1
00	x	x
01	x	x
11	0	x
10	x	x

K_1

q_1q_2 \ x	0	1
00	0	0
01	0	0
11	0	0
10	0	1

z

q_1q_2 \ x	0	1
00	0	1
01	x	0
11	0	x
10	0	1

J_2

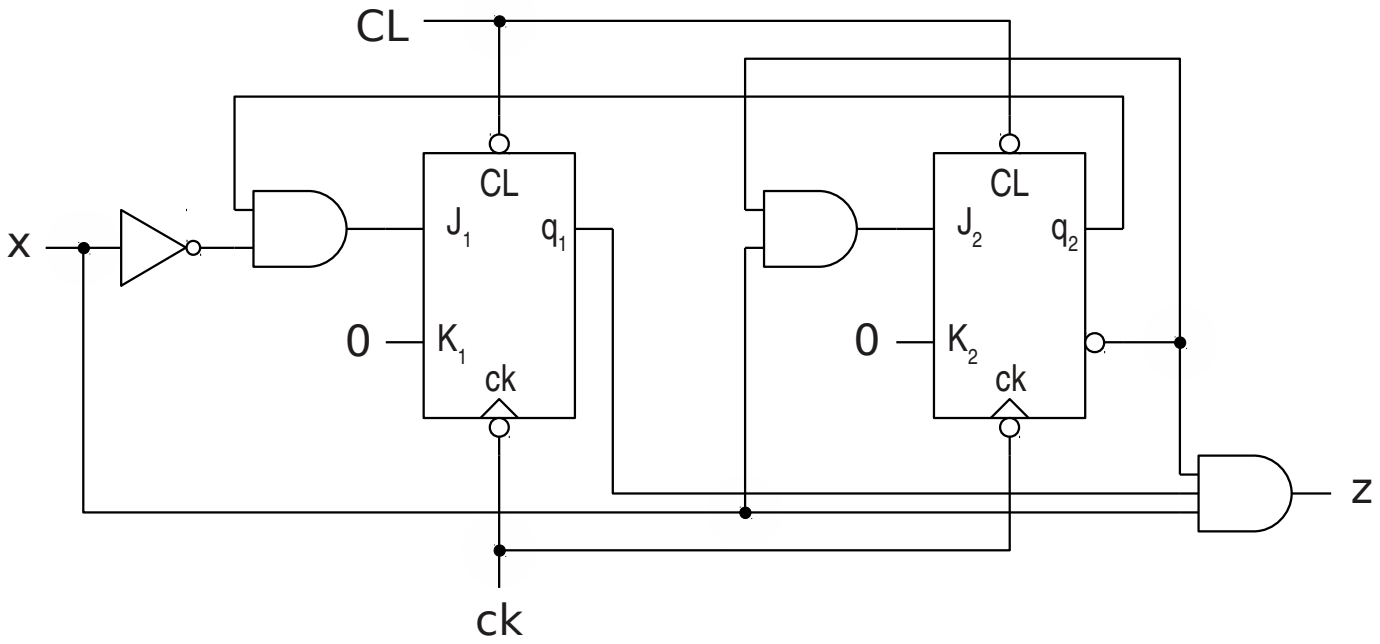
q_1q_2 \ x	0	1
00	x	x
01	0	x
11	x	0
10	x	x

K_2

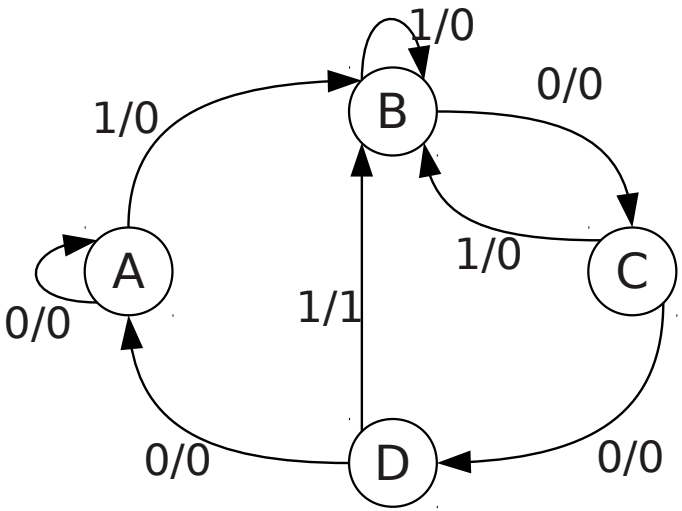
$$J_1 = \bar{x}q_2$$
$$K_1 = 0$$
$$J_2 = x\bar{q}_2$$
$$K_2 = 0$$
$$z = xq_1\bar{q}_2$$

Circuito. Ejemplo

$J_1 = \bar{x}q_2$
$K_1 = 0$
$J_2 = x\bar{q}_2$
$K_2 = 0$
$z = xq_1\bar{q}_2$



Ejemplo. Resumen



S	x=0	x=1
A	A,0	B,0
B	C,0	A,0
C	D,0	B,0
D	A,0	B,1

Q,z

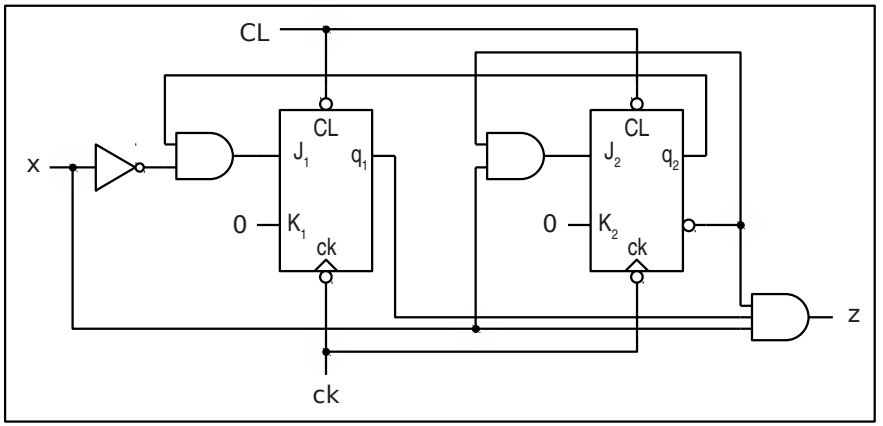
q_1q_2	x=0	x=1
00	00,0	01,0
01	11,0	00,0
11	10,0	01,0
10	00,0	01,1

Q,z

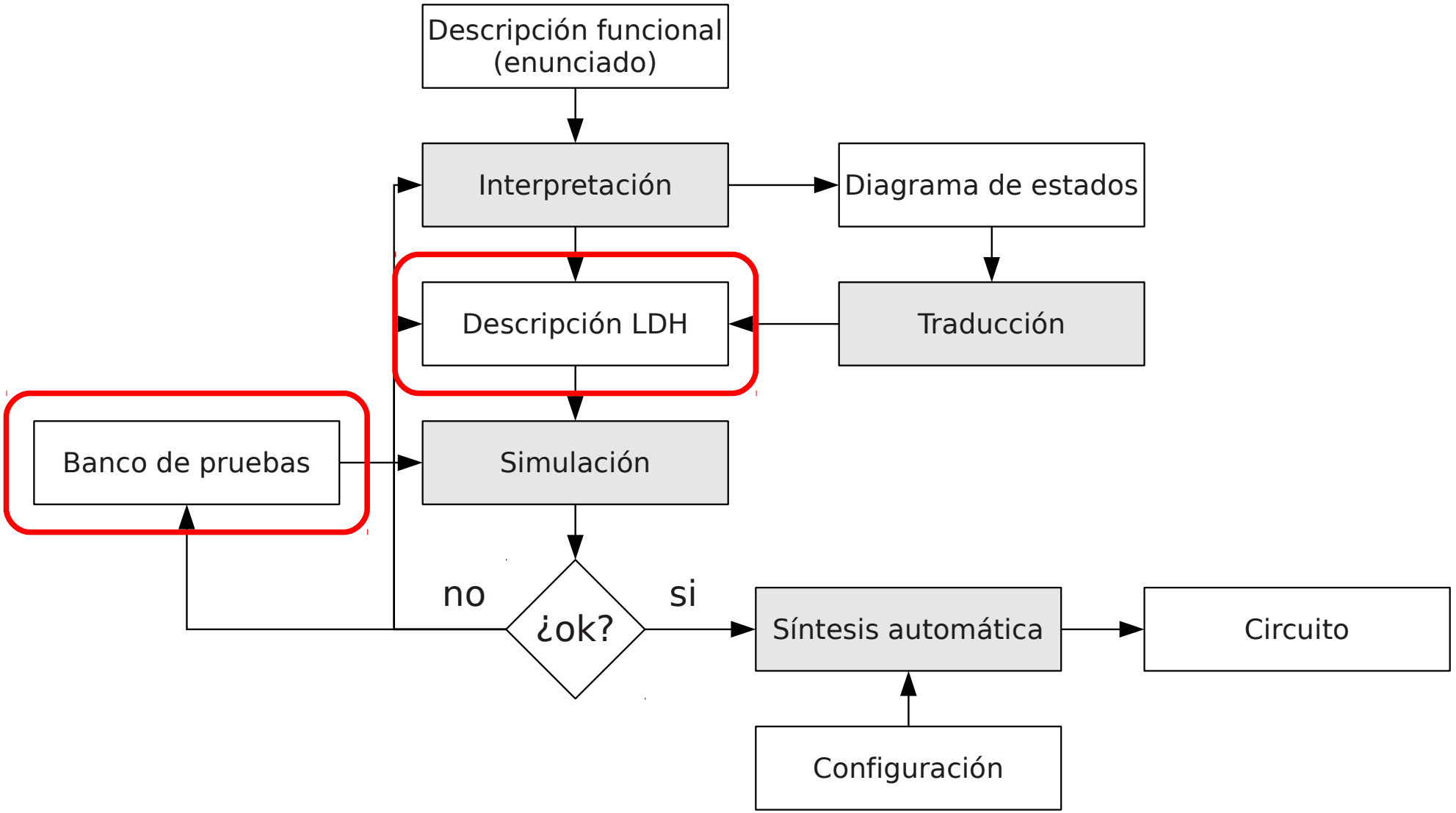
q_1q_2	x=0	x=1
00	0x,0x,0	0x,1x,0
01	1x,x0,0	0x,0x,0
11	x0,0x,0	0x,x0,0
10	0x,0x,0	0x,1x,1

J_1K_1,J_2K_2,z

$$J_1 = \bar{x}q_2$$
$$K_1 = 0$$
$$J_2 = x\bar{q}_2$$
$$K_2 = 0$$
$$z = xq_1\bar{q}_2$$



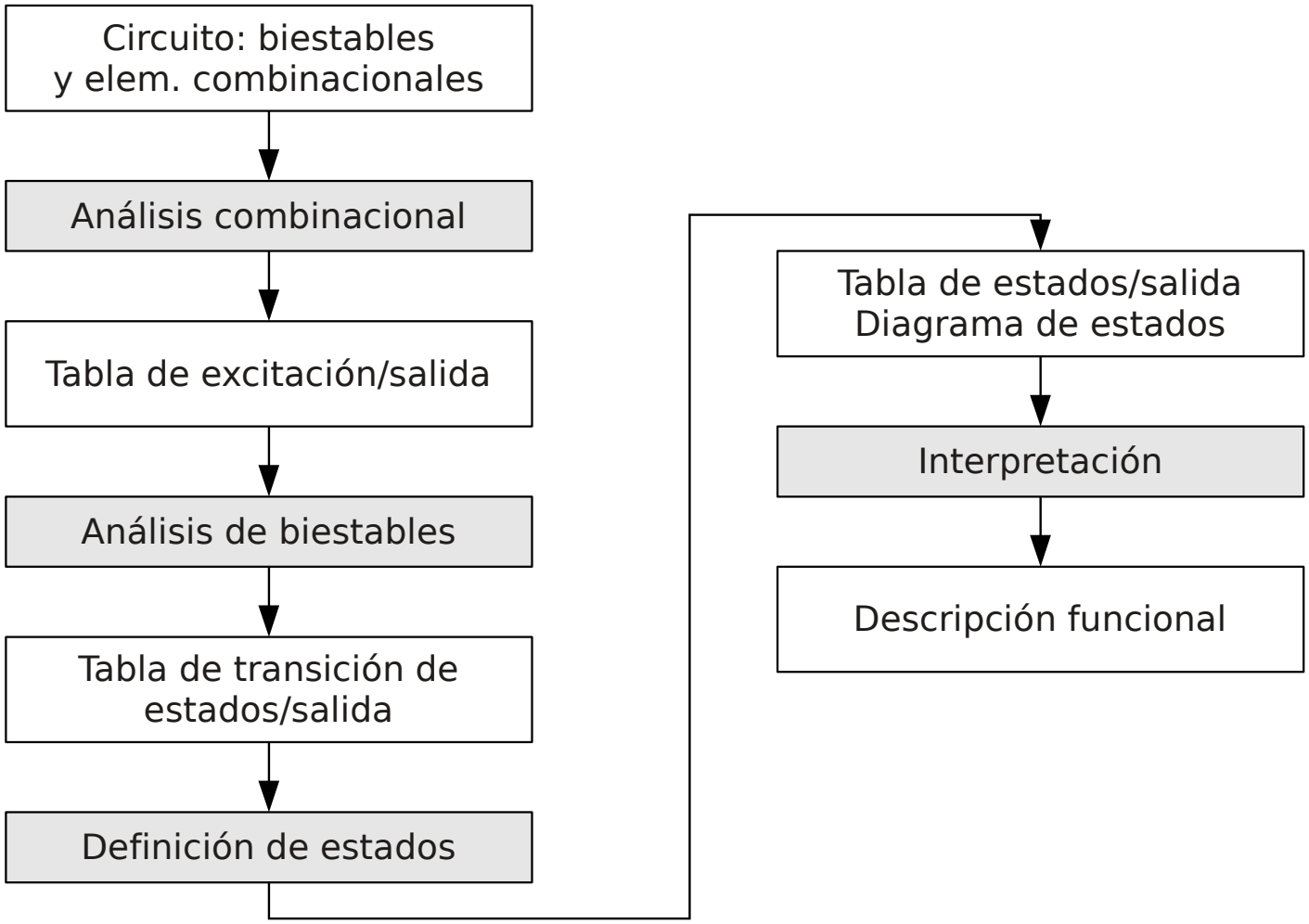
Procedimiento con herramientas de diseño



Análisis de CSS

- Introducción
- Biestables
- Máquinas de estados finitos y circuitos secuenciales síncronos (CSS)
- Diseño de CSS
- Análisis de CSS
  - ⇨ Análisis formal
  - ⇨ Análisis temporal

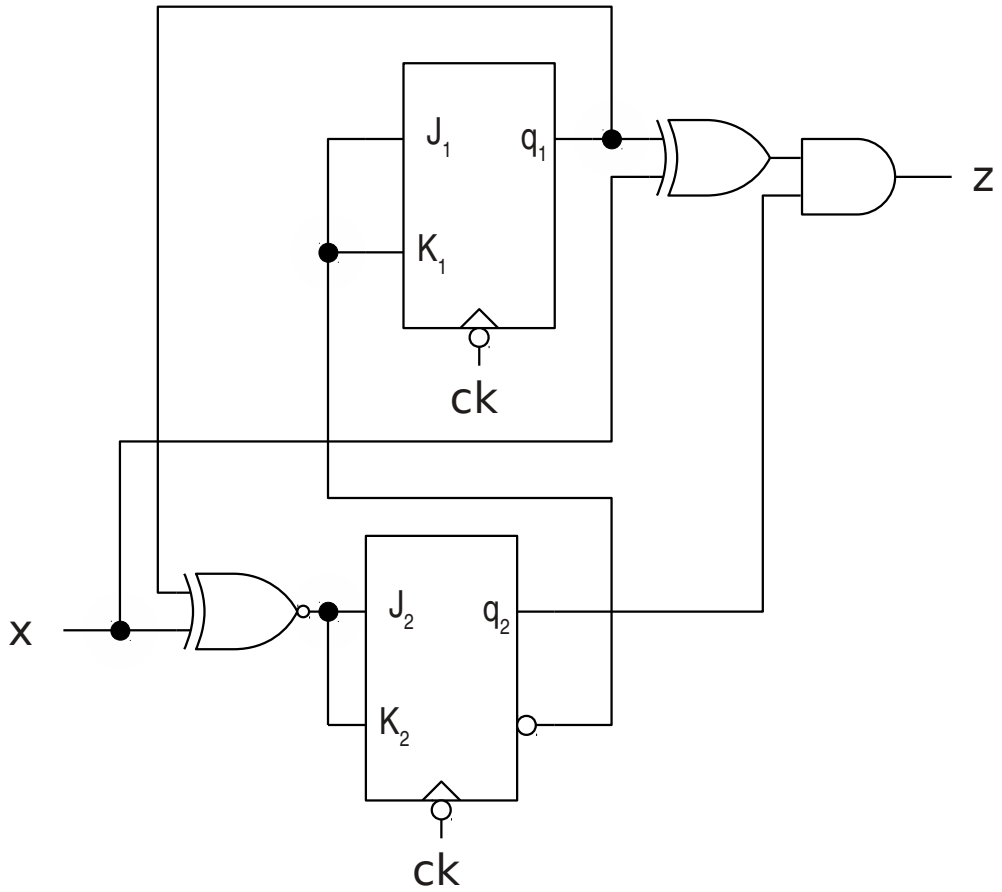
Análisis formal



Análisis formal

- Proceso inverso a la síntesis
- Objetivo:
  - ⇨ Partiendo del circuito construido (esquema del circuito), obtener el diagrama de estados de la máquina que implementa e interpretar su operación/utilidad.
- El proceso hasta obtener el diagrama de estados es sistemático.
- La interpretación no es sistemática
  - ⇨ Experiencia
  - ⇨ Información adicional
  - ⇨ Etc.

Análisis formal. Ejemplo



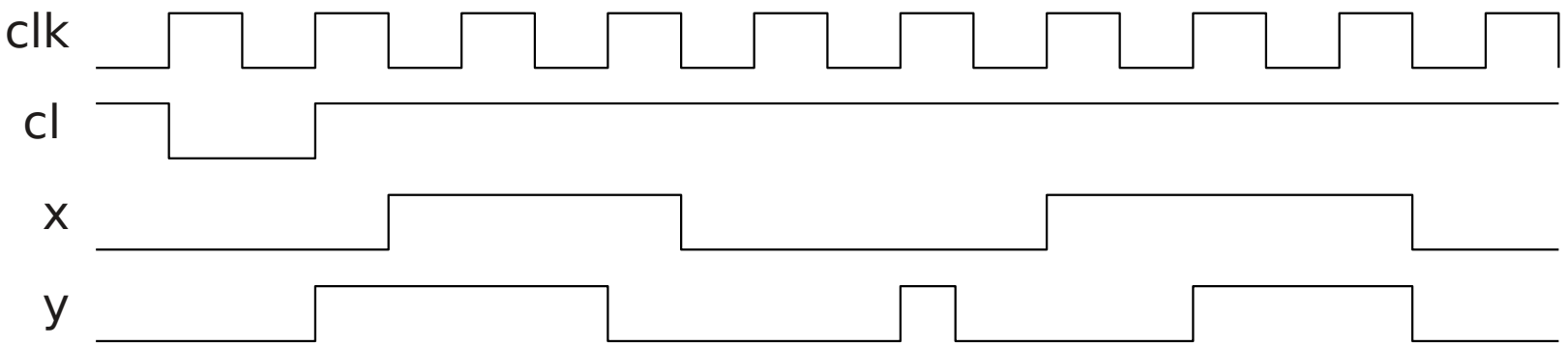
Análisis temporal

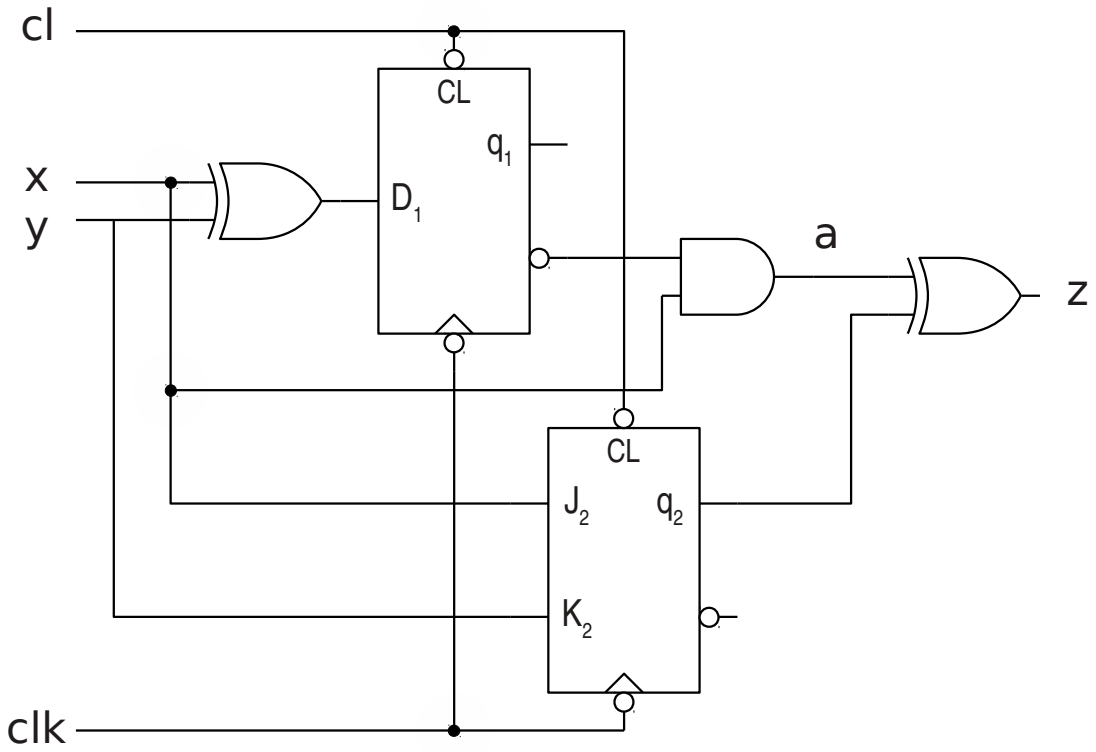
- Objetivo
  - ⇨ Dado un circuito diseñado (biestables, puertas, etc.), obtener el cronograma de las señales de salida para unas señales de entrada dadas.
- Consideraciones
  - ⇨ Es posible analizar circuitos con biestables aunque no sean CSS.
  - ⇨ Si se trata de un CSS, el análisis temporal debe corresponder con la máquina de estados que implementa.

Análisis temporal

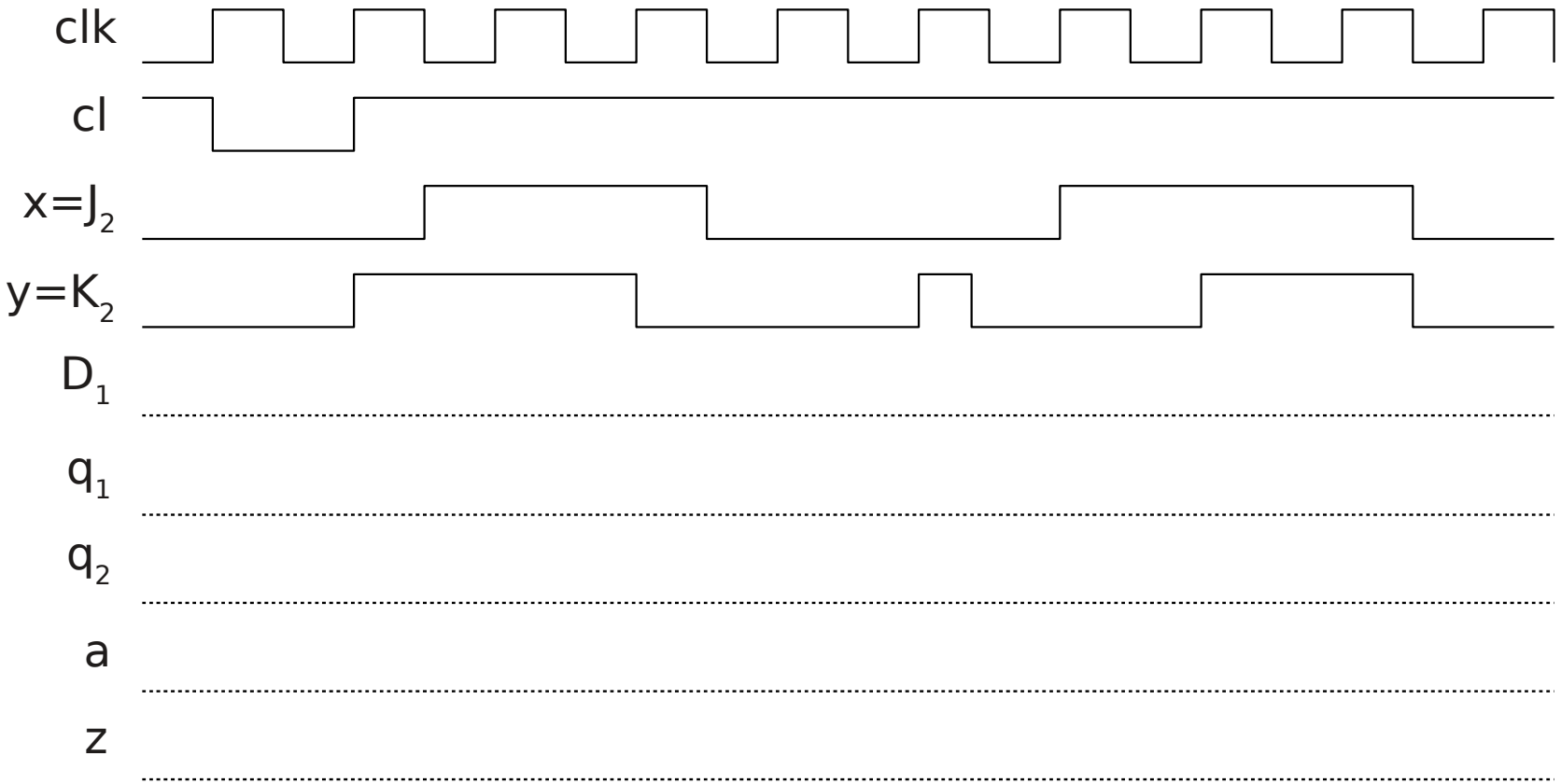
- Procedimiento similar al de circuitos combinacionales
  - ⇨ Parte combinacional: idéntica
  - ⇨ Biestables (por flanco): observando el flanco activo del reloj y calculando la salida (nuevo estado) a partir de la tabla de estados del biestable
  - ⇨ La salida cambia con el retraso definido desde el cambio en el reloj hasta el cambio en el estado (tck-q)

Análisis temporal. Ejemplo





Análisis temporal. Ejemplo



$D_1 = x \oplus y$

$J_2 = x$; $K_2 = y$

$a = \overline{q_1}\, x$

$z = a \oplus q_2$