



Fundamentos de los computadores

Tema 5. SISTEMAS SECUENCIALES (Primera parte)

- Primera parte
 - Conocer los sistemas secuenciales básicos más importantes
 - Registros, Banco de registros, Memoria, Contadores
 - Aprender a representar formalmente un sistema secuencial síncrono sencillo a partir de una especificación
 - Analizar el comportamiento de un sistema secuencial
 - Para determinar las salidas del mismo
- Segunda parte
 - Conocer la metodología de diseño de los sistemas secuenciales síncronos
 - Para obtener un circuito a partir de una especificación

- [1] Wakerly, John F. “Digital Design: Principles and Practices”, 4th ed., Prentice Hall, 2006
 - Secciones 7.3 – 7.5
- [2] Katz, Randy H. y Borriello, Gaetano “Contemporary Logic Design”, 2nd ed., Prentice Hall, 2005
 - Secciones 6.3, 7.1 – 7.3, 8.2
- [3] Sahuquillo, Julio y otros “Introducción a los computadores”
 - Capítulo 6

- Introducción a los sistemas secuenciales (SS)
 - Clases de sistemas secuenciales
 - Sistemas secuenciales asíncronos (SSA)
 - Sistemas secuenciales **síncronos** (SSS)
- Sistemas secuenciales básicos
 - Registros de almacenamiento
 - Banco de Registros, Memoria
 - Registros de desplazamiento
 - Contadores

- Tipos de SSS
 - Autómatas de Mealy
 - Autómatas de **Moore**
- Representación de un autómata de Moore
 - Diagrama de estados
 - Tabla de estados
- Análisis de sistemas secuenciales
 - Cronograma
 - Tabla de estados, diagrama de estados

- Sistema secuencial (SS)
 - Nombre genérico dado a toda agrupación de biestables
 - Clasificación
 - Sistema secuencial asíncrono (SSA): No todos los biestables se activan en el mismo instante (distintos relojes o flancos)
 - Diseño complejo y utilidad reducida (respecto de los SSS)
 - Sistema secuencial síncrono (SSS) : Todos los biestables se activan en el mismo instante (mismo reloj y flanco)
 - También se conocen como
 - » **Máquina de estados finitos** (“Finite State Machine” – FSM)
 - » **Autómata** (de estados finitos)
 - Centraremos nuestra atención en los SSS y no en los SSA

- Definiciones relacionadas con los SS
 - Variables de estado
 - Cada uno de los biestables que componen el sistema
 - Estado
 - De [1]: “El *estado* de un circuito secuencial es una colección de *variables de estado* cuyos valores en cualquier instante de tiempo contienen toda la información acerca del pasado necesaria para determinar el comportamiento futuro del circuito”
 - Estado actual
 - El valor de las variables de estado en el momento actual
 - Estado siguiente
 - El valor que tomarán las variables de estado tras el próximo flanco activo del reloj

- Existen sistemas secuenciales tan importantes y de uso tan común que se les ha dado nombre
 - Registro de almacenamiento
 - Almacén de un dato de N bits
 - Banco de Registros, Memoria
 - Agrupación de varios registros de almacenamiento
 - Misma funcionalidad básica. Difieren fundamentalmente en aspectos de capacidad, velocidad de funcionamiento y diseño

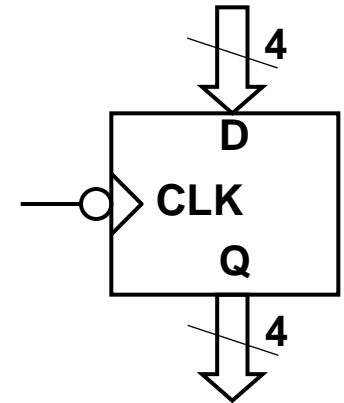
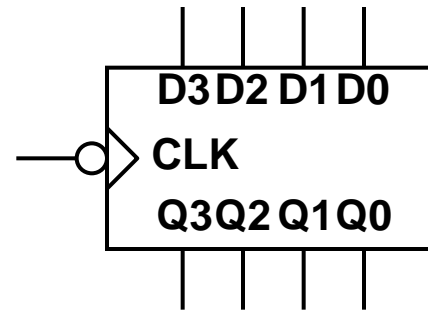
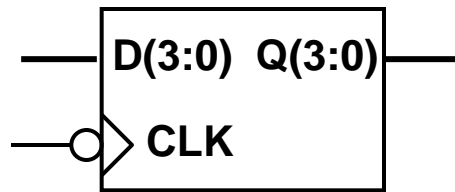
- Existen sistemas secuenciales tan importantes y de uso tan común que se les ha dado nombre (cont.)
 - Registro de desplazamiento
 - Almacén de un dato de N bits en los se necesitan N ciclos de reloj para que la información entre (escritura) y/o salga (lectura)
 - Contador
 - Circuito que cambia de valor de forma autónoma en cada ciclo de reloj siguiendo una secuencia de valores predeterminada
 - Generalmente la cuenta es binaria (ascendente o descendente)

- Agrupación síncrona de biestables D por flanco
 - Tantos biestables D como bits queramos almacenar
 - Cada biestable D almacena un bit
 - Las entradas/salidas de datos del registro coinciden con las de los biestables D
 - Un único reloj (sistema síncrono) interconecta las entradas de reloj de todos los biestables
 - Todos los biestables deben ser activos en el mismo flanco de reloj

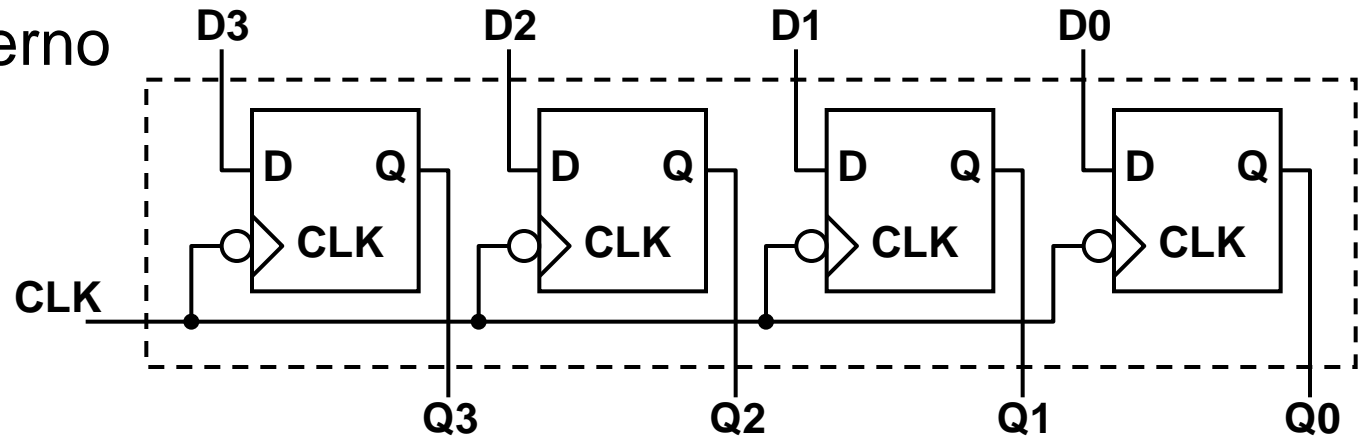
(ii)

- Ejemplo: Registro de almacenamiento de 4 bits activo por flanco de bajada

- Símbolos lógicos posibles



- Esquema interno

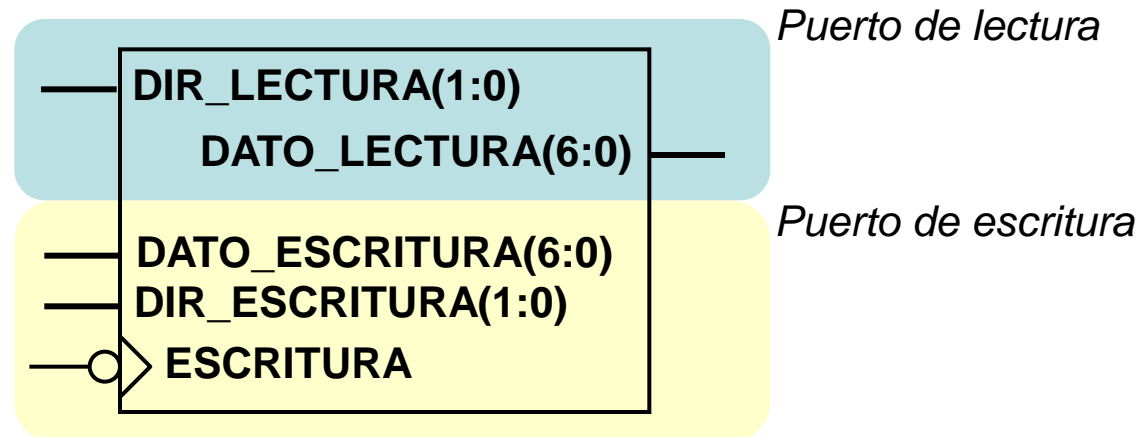


(iii)

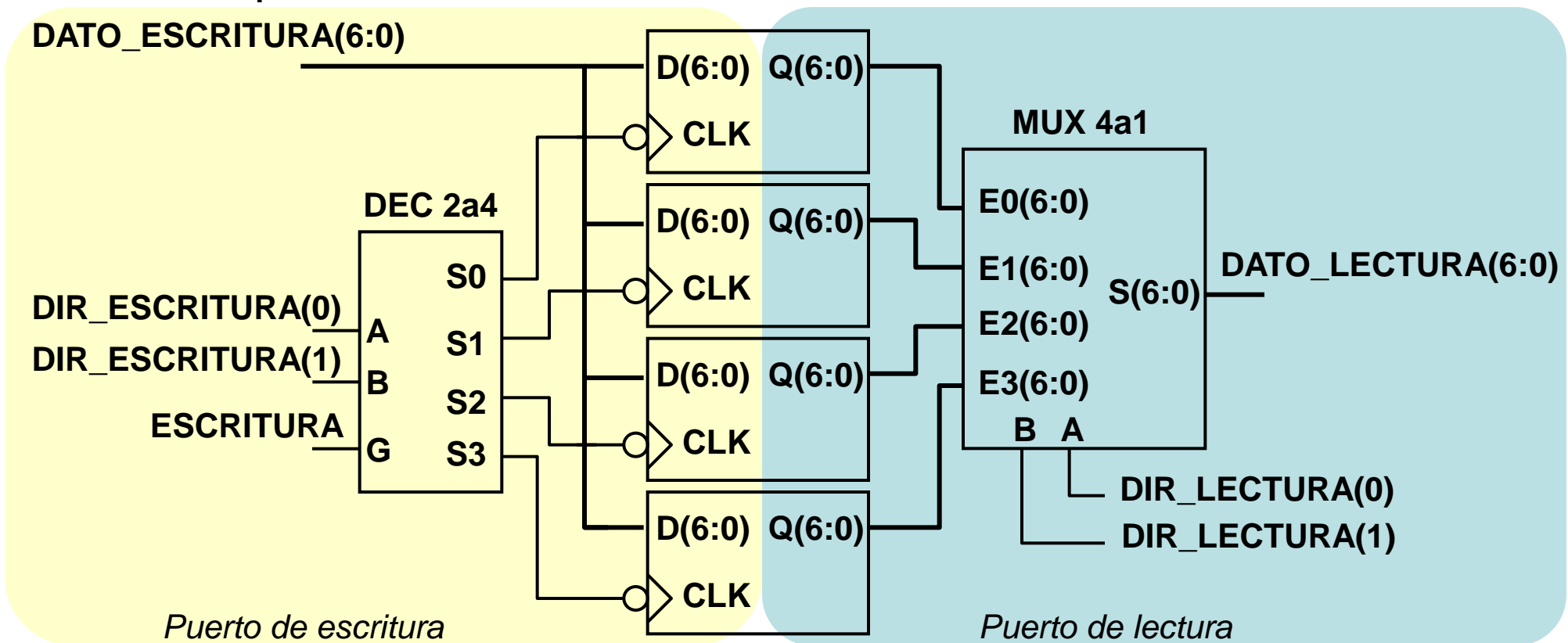
- Operación de escritura
 - Operación destructiva
 - El dato (previamente) almacenado desaparece y es sobre-escrito con el dato de la operación de escritura
 - Operativa
 - 1) Establecer el valor de las entradas D de los biestables
 - 2) Forzar un flanco en la señal de reloj
- Operación de lectura
 - Operación no destructiva
 - El dato almacenado permanece inalterado
 - Operativa
 - Examinar las salidas Q de los biestables

- Agrupación de registros
 - Permite la escritura de un dato sobre un registro
 - Hay que seleccionar qué registro queremos escribir
 - Esta información es la **dirección de escritura**
 - Internamente, un decodificador selecciona qué registro trabaja
 - Permite la lectura de, al menos, un dato almacenado
 - Hay que seleccionar qué registro queremos leer
 - Esta información es la **dirección de lectura**
 - Internamente, un multiplexor selecciona qué dato se obtiene
 - Para permitir la lectura simultánea de dos o más datos es necesario disponer de tantas entradas de dirección de lectura y multiplexores internos como datos se deseen leer

- Ejemplo: Banco de registros de 4 registros de 7 bits con un puerto (vía de acceso) de escritura y otro de lectura
 - 4 registros => 2 bits de dirección
 - Datos de entrada/salida de 7 bits
 - Símbolo lógico



- Ejemplo (cont):
 - Esquema interno



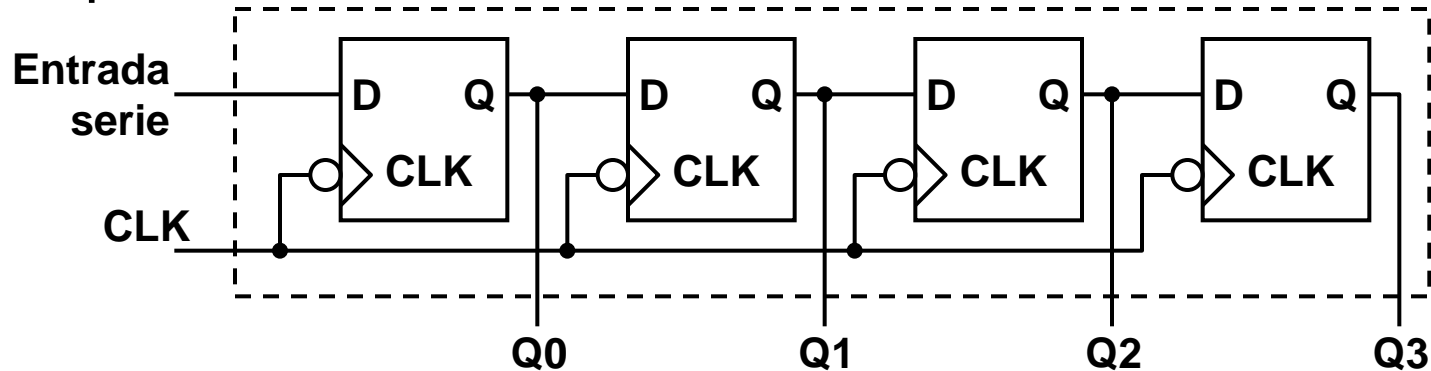
- Operación de escritura
 - Operativa
 - 1) Establecer el valor del dato de escritura
 - 2) Establecer el valor de la dirección de escritura
 - 3) Forzar un flanco en la señal de escritura
 - Habilitando y deshabilitando el decodificador
- Operación de lectura
 - Operativa
 - 1) Establecer el valor de la dirección de lectura
 - 2) Examinar el valor del dato de lectura

- Misma funcionalidad que un banco de registros, pero
 - Mucha más capacidad (Kbytes, Mbytes, Gbytes, ...)
 - Mucho más lenta
 - Tecnología diferente
- Una única operación (lectura/escritura) en un momento dado
 - Una única entrada de dirección y otra de dato
 - Líneas de órdenes de lectura y escritura
 - Para que la memoria sepa qué hacer en cada momento
 - Leer, escribir o nada (si no se activa ni lectura ni escritura)

- Agrupación síncrona de biestables D por flanco
 - Tantos biestables D como bits queremos almacenar
 - La información necesita varios ciclos de reloj para entrar (escritura) o salir (lectura)
 - Se conocen como entrada serie y salida **serie**, respectivamente
 - Cuando todos los bits entran o salen en el mismo ciclo de reloj se dice que el registro tiene entrada o salida **paralelo**
 - Estructura con entrada serie
 - El primer biestable conecta su entrada con la única entrada de datos
 - El resto, cada entrada con la salida del anterior
 - Estructura con salida serie
 - Sólo es accesible la salida del último biestable

- Entrada serie, salida paralelo

- Esquema interno

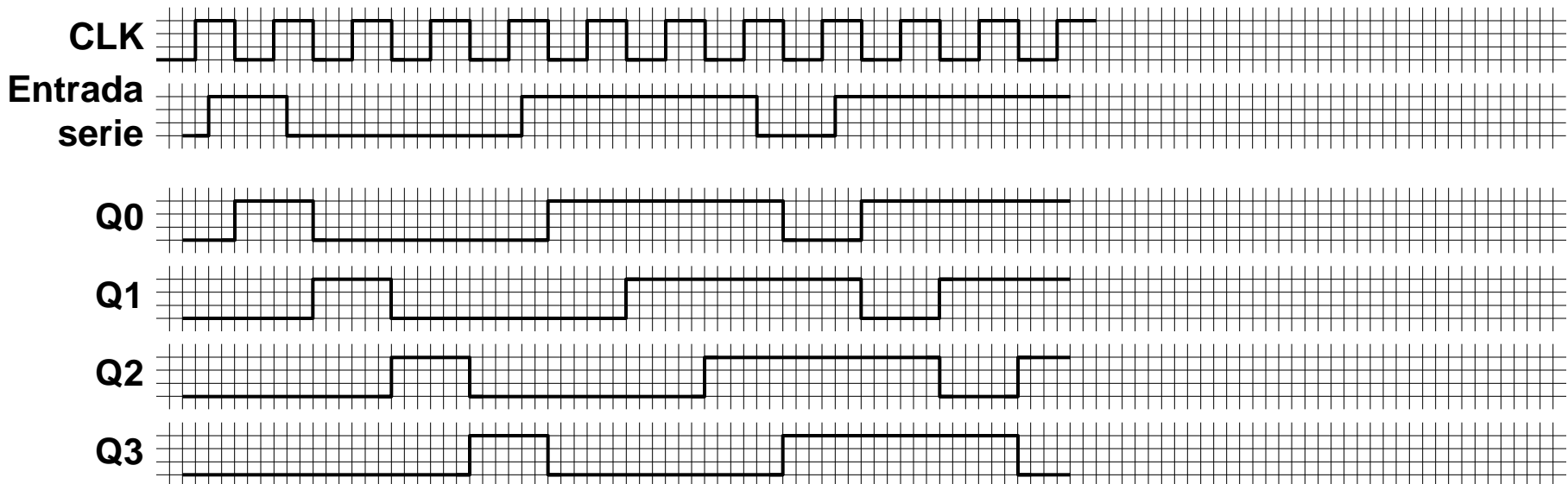


- Funcionamiento

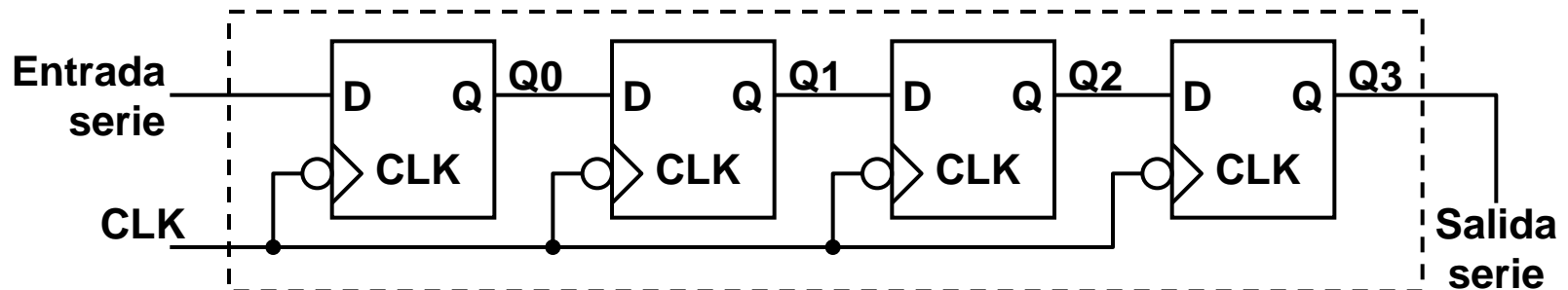
- Un bit (el valor de la entrada serie) entra por la izquierda y desplaza la información almacenada una posición hacia la derecha
 - El bit almacenado en el extremo derecho se pierde

(iii)

- Entrada serie, salida paralelo (cont.)
 - Ejemplo de funcionamiento
 - Estado inicial $Q_3Q_2Q_1Q_0 = 0000$
 - La entrada serie toma los valores indicados en el cronograma (como ejemplo de secuencia de valores)



- Entrada serie, salida serie
 - Mismo circuito interno que con salida paralelo, excepto que la única salida disponible es la del último biestable



- Entrada paralelo, salida serie
 - Necesitamos una entrada adicional (que podemos llamar, por ejemplo, LOAD) para determinar si el sistema debe cargar el dato de entrada (LOAD=1) o desplazar (LOAD=0)

Sist. Básicos: Reg. desplazamiento (v) FCO

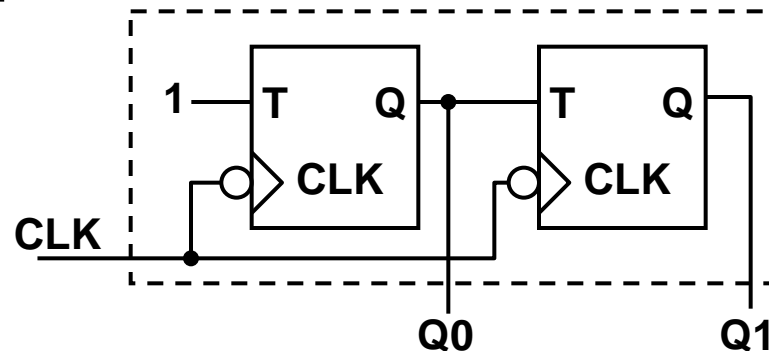
- Registros de desplazamiento (resumen)
 - Un registro de desplazamiento puede desplazar
 - A izquierdas o a derechas
 - No hay unanimidad en lo que significa “a izquierdas” o “a derechas”
 - Mejor indicar el sentido explícitamente
 - Incluso puede desplazar en ambos sentidos
 - No simultáneamente, a veces a izquierdas y a veces a derechas
 - Con una entrada de control adicional que indique el sentido
 - En este caso puede disponer de
 - » *Una única entrada de datos serie*
 - » *Una entrada de datos serie para cada uno de los sentidos de desplazamiento (entrada serie por la izquierda, entrada serie por la derecha)*

- Registros de desplazamiento (resumen, cont.)
 - La entrada de datos puede ser
 - Serie: Cuando la entrada de datos es de un bit por ciclo de reloj
 - Paralelo: Cuando todo el registro se escribe en un único ciclo de reloj
 - La salida puede ser
 - Serie: Cuando para observar el valor almacenado en el registro necesitamos tantos ciclos de reloj como biestables
 - Paralelo: Cuando podemos observar el valor de todos los biestables en un único ciclo de reloj

- Contador
 - Circuito que cambia de valor de forma autónoma en cada ciclo de reloj siguiendo una secuencia, generalmente una cuenta binaria (ascendente o descendente)
 - La cuenta puede ser ascendente o descendente
 - La cuenta también puede ser ascendente/descendente (reversible)
 - No simultáneamente, a veces ascendente y a veces descendente
 - Una entrada de control adicional indica el sentido

- Contadores síncronos

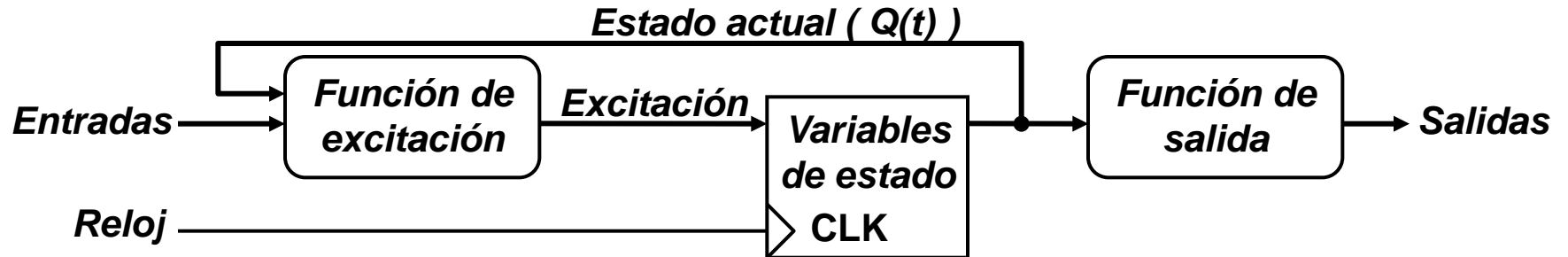
- Suelen estar contruidos con biestables T o J-K (con $J=K$)
 - Porque el circuito resultante utiliza menos puertas lógicas (aunque los biestables sean más complejos)
- Ejemplo: Contador síncrono ascendente de 2 bits
 - Salida 0, 1, 2, 3, 0, ...



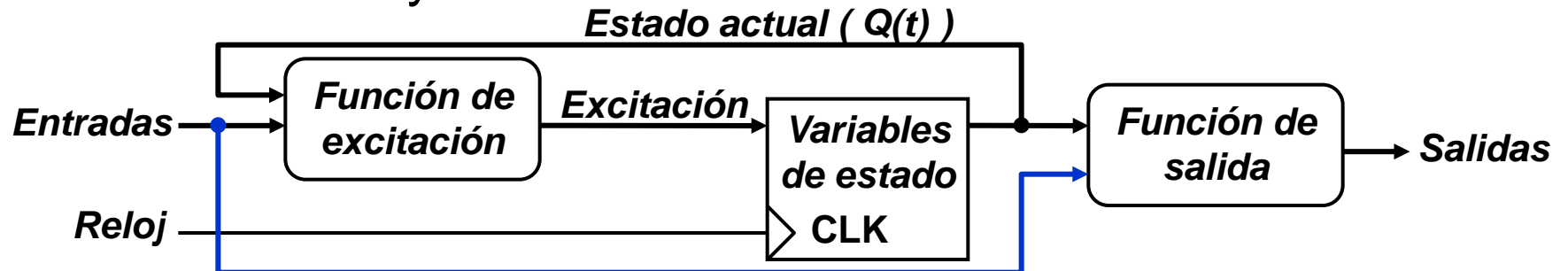
- El mismo sistema construido con biestables D da como resultado un circuito más complejo

- Contadores: Clasificación por el tipo de cuenta
 - Binarios
 - Hacen todas las cuentas posibles entre 0 y $2^{\text{variables de estado}-1}$
 - De módulo N
 - Hacen N cuentas distintas, siendo $N < 2^{\text{variables de estado}}$
 - Suelen incluir la cuenta 0...00, contando entonces de 0 a N-1
 - Ejemplo
 - Un contador de décadas es un contador de módulo 10 que cuenta de 0 a 9

- Existen dos estructuras posibles de SSS
 - **Autómata de Moore**: Las salidas del SSS dependen exclusivamente del estado actual



- **Autómata de Mealy**: Las salidas del SSS dependen del estado actual y de las entradas del sistema

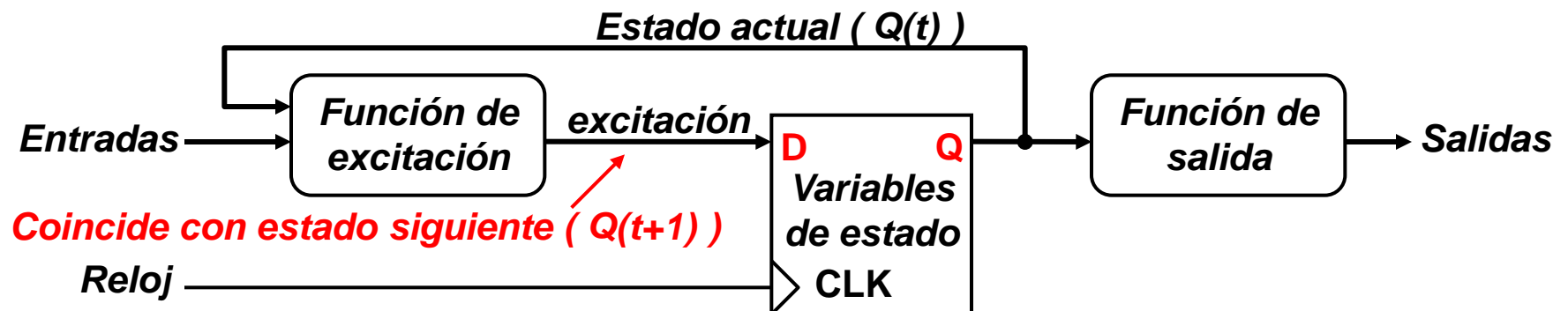


- Elementos distinguibles en los autómatas
 - Variables de estado
 - Generalmente se utilizan biestables D para almacenar el estado
 - Podrían ser D, J-K o T (siempre por flanco)
 - Funciones combinacionales
 - De excitación o transición, que determina el próximo estado
 - De salida, que determina las salidas del sistema

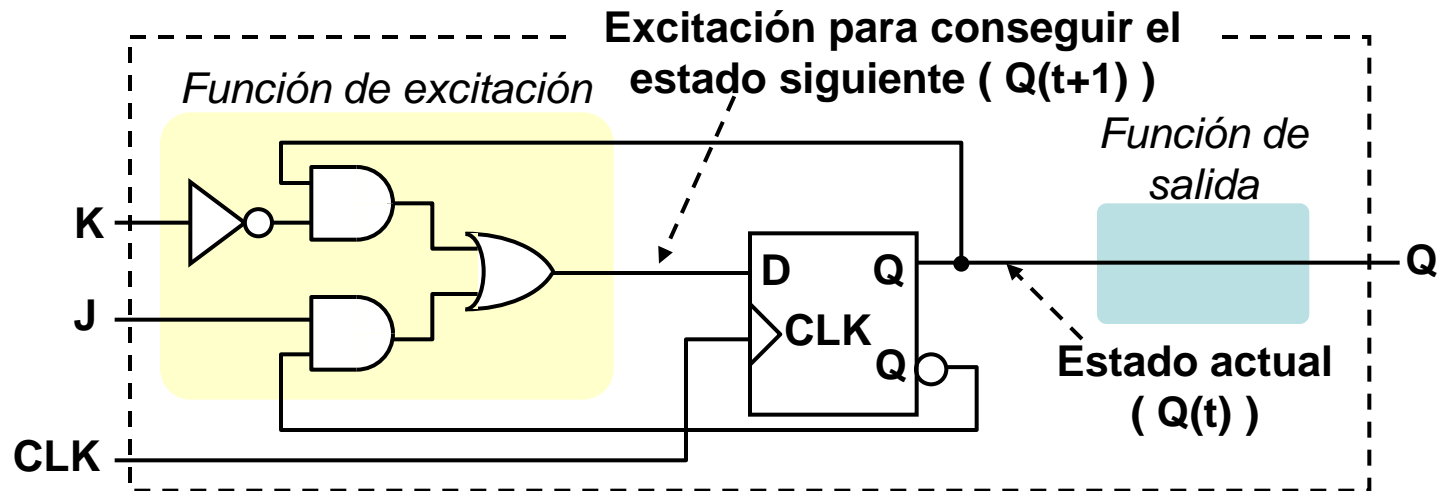
- Autómatas: Moore vs Mealy
 - Los circuitos resultantes de resolver un mismo problema como autómata de Moore o Mealy son radicalmente distintos
 - No coincide el número de variables de estado necesarias
 - No coinciden las funciones de excitación y de salida
 - Generalmente
 - Con un autómata de Mealy se reduce el número de variables de estado
 - Un autómata de Moore es más fácil de diseñar y depurar

- Tipos de biestables: D vs J-K vs T
 - Dado un tipo de autómata (Moore o Mealy) la función de excitación cambia dependiendo del tipo de biestable utilizado
 - La función de excitación da respuesta a la siguiente pregunta
 - “¿Cuál debe ser el valor de las entradas de los biestables (D , J y K o T , dependiendo del tipo empleado) para que el valor de las variables de estado cambie al estado siguiente cuando llegue el próximo flanco activo del reloj?”
 - Evidentemente, la respuesta no puede ser la misma si los biestables utilizados son D , J - K o T (o mezclados)
 - Los biestables D permiten calcular la función de excitación de forma muy sencilla
 - El valor de las entradas D para estos biestables coincide con el estado siguiente

- Centraremos nuestra atención en el diseño de autómatas de Moore utilizando biestables D
 - Por el escaso tiempo disponible,
 - Porque los autómatas de Moore son más sencillos de diseñar y depurar que los de Mealy, y
 - Porque la función de excitación es más sencilla cuando empleamos biestables D (D = estado siguiente)



- Ejemplo de autómata de Moore
 - Ya conocemos uno de los autómatas de Moore (utilizando biestables de tipo D) más sencillos: el biestable J-K
 - Utiliza un único biestable de tipo D (una sola variable de estado)
 - Dos entradas externas: J y K
 - La salida del sistema coincide con el estado

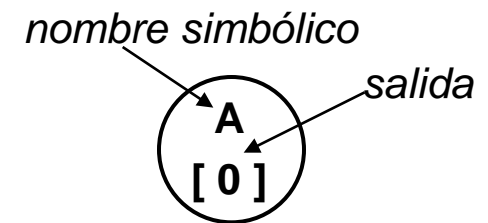


- Representación de un autómata de Moore
 - Formalización (sin ambigüedad) del comportamiento
 - Abstracción que nos permite razonar sobre el SSS
 - Para determinar la salida del sistema, si hemos obtenido la representación a partir de un circuito y queremos analizarlo
 - Para entender el problema a resolver y formalizar todos los casos de funcionamiento, si hemos obtenido la representación a partir de una descripción en lenguaje natural y queremos diseñar el circuito
 - Dos representaciones equivalentes (que contienen la misma información)
 - Diagrama de estados
 - Tabla de estados

- Representación de un autómata de Moore (cont.)
 - Diagrama de estados
 - Representación gráfica abstracta
 - Permite comprender el funcionamiento del autómata
 - Suele ser el primer paso en el diseño de un sistema y el último en el análisis de un circuito
 - Tabla de estados y transiciones
 - Representación tabular
 - Su obtención a partir del diagrama de estados es sencilla
 - También se puede obtener directamente a partir de la descripción del sistema
 - Permite aplicar la metodología de diseño de forma sistemática

- Diagrama de estados de autómatas de Moore
 - Representación gráfica del comportamiento de un SSS
 - Grafo dirigido con nodos y arcos entre estados
 - Cada nodo representa un estado del sistema
 - *Recordatorio: un estado es un valor almacenado, una combinación de valores de las variables de estado*
 - Cada arco representa una transición
 - Una transición es un cambio de estado
 - El reloj está implícito en el diseño
 - La transición entre estados se produce con el flanco activo del reloj

- Diagrama de estados de aut. de Moore (cont.)
 - Cada estado lleva asociado
 - Nombre del estado. Puede ser numérico o simbólico
 - Su utilidad es diferenciar un estado de otro en el grafo
 - Cuando sepamos el estado (el valor de las variables de estado), podemos utilizar dicho valor numérico como “nombre”
 - Cuando no lo sepamos o no lo queramos utilizar, usaremos un nombre simbólico elegido arbitrariamente
 - Salida. El valor de la salida del sistema para ese estado
 - Uno o más valores entre corchetes
 - Los valores de salida y estado (cuando este es numérico) pueden coincidir o no, depende del sistema

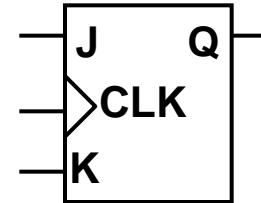


- Diagrama de estados de aut. de Moore (cont.)
 - Cada transición lleva asociada
 - La combinación de valores de las entradas que hace que el sistema realice la transición
 - En forma de vectores (dejando claro el orden de las entradas)
 - En forma de ecuaciones booleanas
 - De cada nodo deben salir transiciones como para cubrir todos los posibles casos de los valores de las entradas
 - En otro caso el sistema no estaría completamente especificado

- Ejemplo de diagrama de estados: Biestable J-K

- Interfaz (entradas y salidas)

- Entradas J y K (y reloj)
- Salida Q (que coincide con el valor del estado)



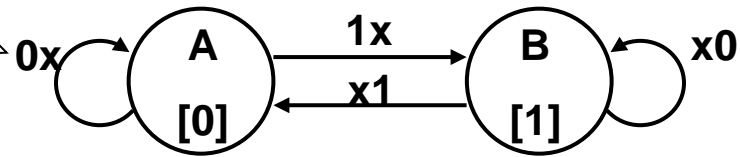
- Estados (sabemos que hay una única variable de estado)

- Dos estados posibles, uno cuando el estado vale 0 y el otro cuando el estado vale 1
- Podemos emplear los valores de estado para diferenciar los estados o utilizar nombres simbólicos elegidos arbitrariamente
 - Ejemplo: A para el estado en el que la salida es 0 y B para el otro

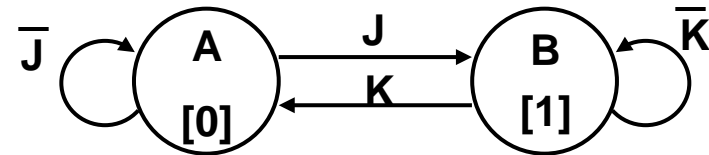
(v)

- Ejemplo de diagrama de estados (cont.): Biestable J-K
 - Diagrama de estados utilizando vectores en el orden J,K

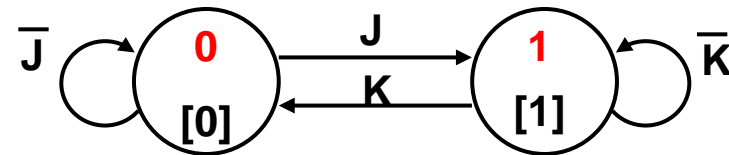
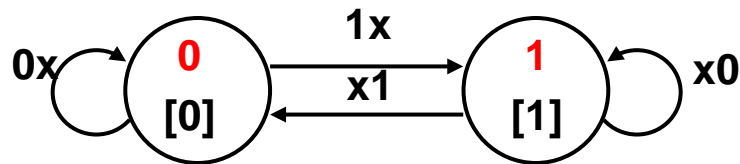
Si $Q(t)=0$ y $J=0$, entonces $Q(t+1) = 0$ independientemente de K :
 Si $K=0$, entonces $Q(t+1)=Q(t)=0$, y
 si $K=1$ entonces $Q(t+1) = 0$.



- Diagrama de estados utilizando ecuaciones booleanas



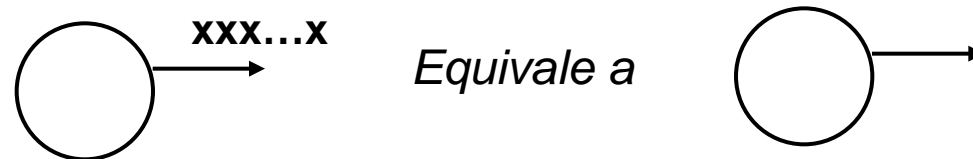
- Ahora utilizando los valores de estado como “nombres”



- Diagramas de estados: convenios habituales
 - Eliminar la ecuación/vector asociada a una transición cuando dicha ecuación/combinación de valores siempre se cumple
 - Ejemplo de ecuación booleana que es siempre cierta: 1



- Vector equivalente: xxx...x (realizar la transición en cualquier caso)



- Tabla de estados de autómatas de Moore
 - Representación tabular del comportamiento
 - Cada fila representa un estado del sistema
 - Los estados pueden ser simbólicos o numéricos, como en el diagrama de estados
 - Columnas (tres clases)
 - Estado actual, estado siguiente y salida(s)
 - Filas. En cada fila se recoge
 - El valor del estado actual
 - Los valores de estado siguiente (uno por cada combinación posible de valores de las entradas partiendo del estado actual)
 - Los valores de la salida para el estado actual

- Tabla de estados de aut. de Moore (cont.)
 - Esqueleto de tabla de estados de un sistema cualquiera

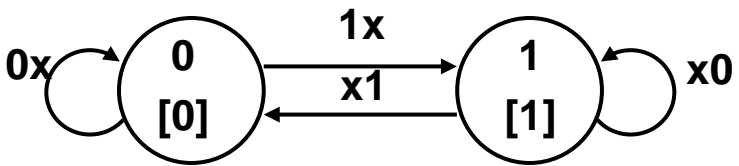
Estado actual	Estado siguiente					Salidas
	Combinaciones de entradas					
	000	001	...	110	111	
0 0 0						
0 0 1						
...						
1 1 0						
1 1 1						

*Los valores de estado siguiente, $Q(t+1)$, dependen del estado actual $Q(t)$ y de los valores de las entradas.
Por tanto, a partir de un estado cualquiera hay tantos valores de estado siguiente como combinaciones de valores de entrada*

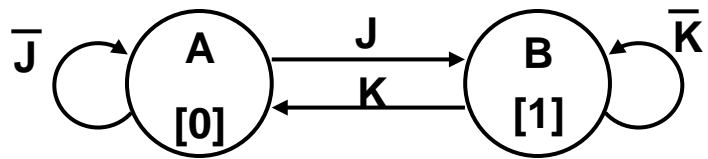
*Los valores de la salida sólo dependen del estado actual $Q(t)$.
Por tanto, en un estado cualquiera sólo hay un valor de salida*

Representación: Tabla de estados (iii) FCO

- A partir del diagrama de estados, obtener la tabla de estados es sencillo
 - Ejemplo: Biestable J-K



Q(t)	Q(t+1)				Salida
	Entradas J,K				
	00	01	10	11	
0	0	0	1	1	0
1	1	0	1	0	1



Q(t)	Q(t+1)				Salida
	Entradas J,K				
	00	01	10	11	
A	A	A	B	B	0
B	B	A	B	A	1

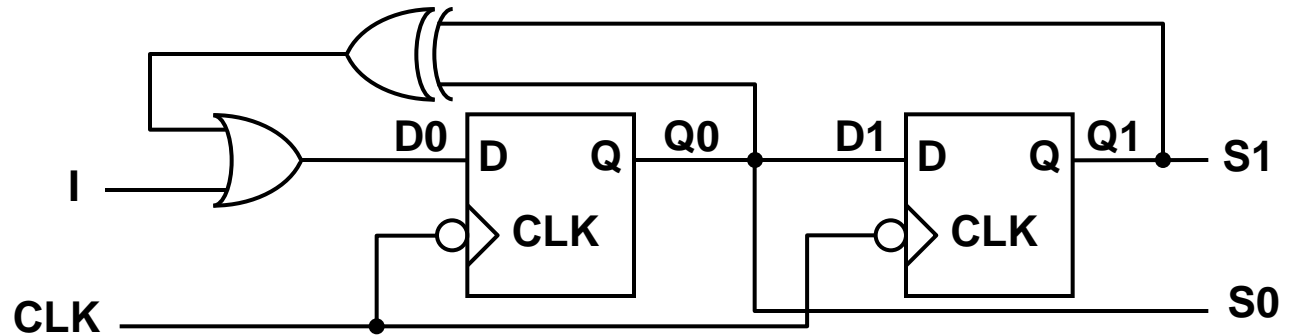
- Análisis de sistemas secuenciales
 - Consiste en obtener la salida del sistema a partir del circuito
 - Métodos
 - Para sistemas secuenciales (SSS o SSA): Cronograma
 - Obtiene la salida del sistema para una secuencia de entradas particular
 - Para SSS: Tabla de estados o Diagrama de estados
 - Obtiene la salida del sistema para cualquier secuencia de entradas

- El análisis de un SS termina
 - Cuando la secuencia de valores en las salidas se repite
 - Si la secuencia de entradas también se repite
- porque
 - El número de estados es finito
 - El próximo estado depende del estado actual y de los valores de las entradas
 - Si ambos se repiten, la salida se repetirá también

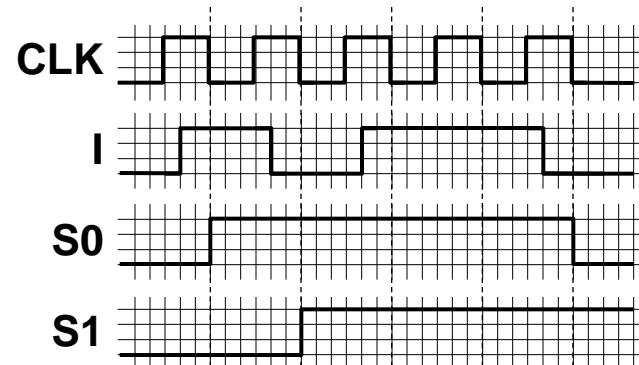
- Análisis por cronograma
 - Diagrama temporal
 - Incorpora todas las entradas y salidas del sistema
 - Puede ser de utilidad para simplificar el análisis añadir también señales internas del circuito
 - Necesitamos
 - El circuito
 - El estado inicial del sistema (si no se puede deducir por las entradas)
 - Una secuencia de entradas
 - Obtenemos
 - La secuencia de salidas del sistema para la secuencia de entradas concreta que hemos empleado en el análisis

- Análisis por cronograma
 - Para realizar el análisis debemos recurrir a las tablas de verdad
 - De cada biestable del circuito
 - Para cada una de las combinaciones de entrada que tenga a lo largo del tiempo
 - Debemos aplicar las combinaciones de entrada una por una en el tiempo
 - Porque el nuevo valor (de un biestable) puede influir en el comportamiento futuro (del mismo biestable o de otro)

- Ejemplo
 - Analizar



- Valor inicial $Q1Q0 = 00_2$
- Secuencia de entrada $I = 1 - 0 - 1 - 1 - 0$
- Cronograma

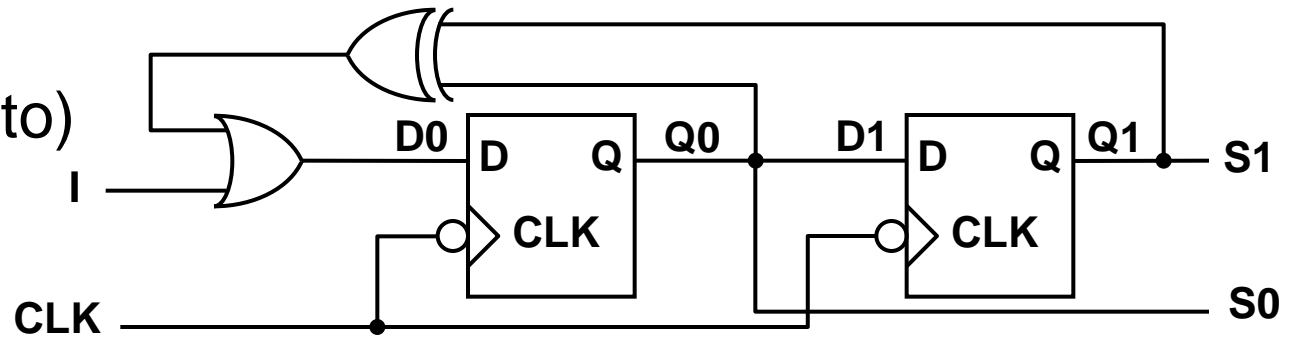


(S1S0 en decimal) Salida 0 1 3 3 3 2

- Tabla de estados
 - Extraída a partir del análisis del circuito
 - Primero se obtiene la tabla de verdad de la función de excitación
 - A partir de ésta se obtienen los valores de estado siguiente
- Diagrama de estados
 - A partir de la tabla de estados, obtener el diagrama de estados es sencillo
 - Los valores numéricos del estado sirven como “nombres”
- Análisis por tabla de estados/diagrama de estados
 - Con cualquiera de los dos

- Ejemplo

- Analizar
(el mismo circuito)



- Ecuaciones y tabla de verdad de la función de excitación

$$D1 = Q0$$

$$D0 = I + Q1 \oplus Q0$$

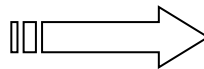
Q1	Q0	I	D1	D0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

- Ejemplo (cont.)

- Tabla de estados

- Los valores D1 y D0 pasan a ser Q1(t+1) y Q0(t+1) respectivamente
 - Las combinaciones de entrada pasan a columnas independientes
 - Añadimos la salida para cada estado

Q1	Q0	I	D1	D0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1



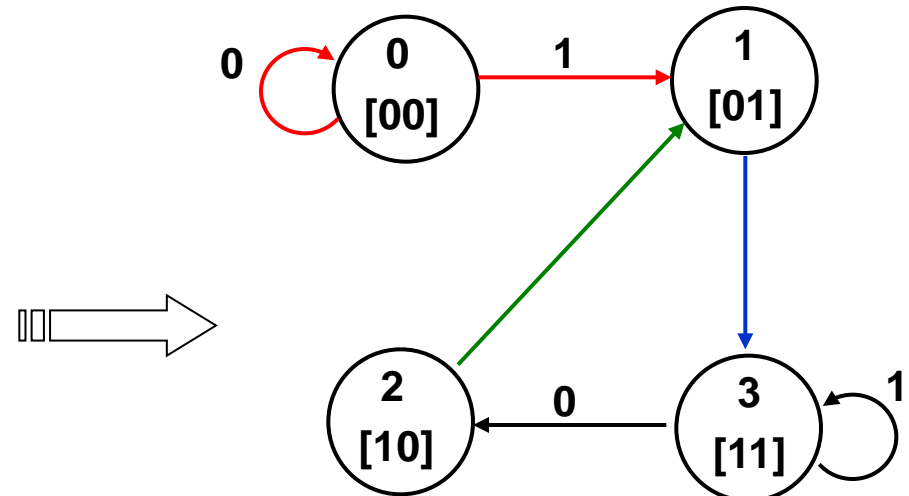
		Q1(t+1) Q0(t+1)			
		Entrada I			
Q1(t)	Q0(t)	I=0	I=1	S1	S0
0	0	00	01	0	0
0	1	11	11	0	1
1	0	01	01	1	0
1	1	10	11	1	1

- Ejemplo (cont.)

- Diagrama de estados

- Los valores de estado siguiente para cada combinación de entrada son las transiciones en el diagrama de estados

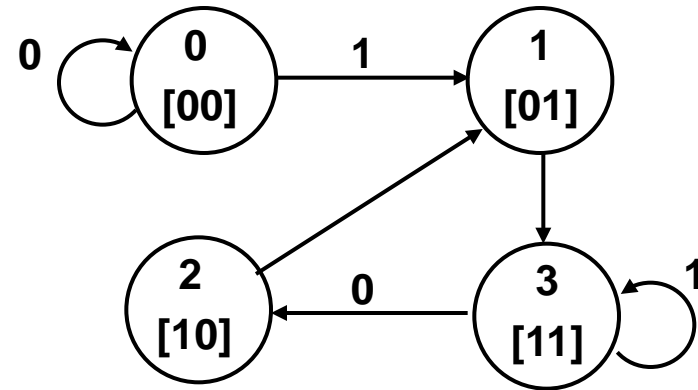
		Q1(t+1) Q0(t+1)			
		Entrada I			
Q1(t)	Q0(t)	I=0	I=1	S1	S0
0	0	0 0	0 1	0	0
0	1	1 1	1 1	0	1
1	0	0 1	0 1	1	0
1	1	1 0	1 1	1	1



- Ejemplo (cont.)

- Con la tabla de estados o con el diagrama de estados resulta sencillo calcular que ...

Q1(t) Q0(t)		Q1(t+1) Q0(t+1)		S1 S0	
		l=0	l=1		
0	0	0 0	0 1	0	0
0	1	1 1	1 1	0	1
1	0	0 1	0 1	1	0
1	1	1 0	1 1	1	1



- Con estado inicial 00, y una secuencia de entradas 1 – 0 – 1 – 1 – 0, la salida del sistema es ... **00 – 01 – 11 – 11 – 10**
- Con estado inicial 10, y una secuencia de entradas 0 – 0 – 1 – 0 – 0, la salida del sistema es ... **10 – 01 – 11 – 11 – 10 – 01**



Fundamentos de los computadores

Tema 5. SISTEMAS SECUENCIALES (Primera parte)