



**Universidad
Europea**

LAUREATE INTERNATIONAL UNIVERSITIES

SISTEMAS SECUENCIALES



© Todos los derechos de propiedad intelectual de esta obra pertenecen en exclusiva a la Universidad Europea de Madrid, S.L.U. Queda terminantemente prohibida la reproducción, puesta a disposición del público y en general cualquier otra forma de explotación de toda o parte de la misma.

La utilización no autorizada de esta obra, así como los perjuicios ocasionados en los derechos de propiedad intelectual e industrial de la Universidad Europea de Madrid, S.L.U., darán lugar al ejercicio de las acciones que legalmente le correspondan y, en su caso, a las responsabilidades que de dicho ejercicio se deriven.

Índice

Presentación	4
El biestable	5
Sincronismo y asincronismo	6
El estado	7
La retroalimentación	8
Biestable R-S síncrono por nivel alto	9
Otros tipos de biestables	11
El registro	15
El registro universal	17
Resumen	18

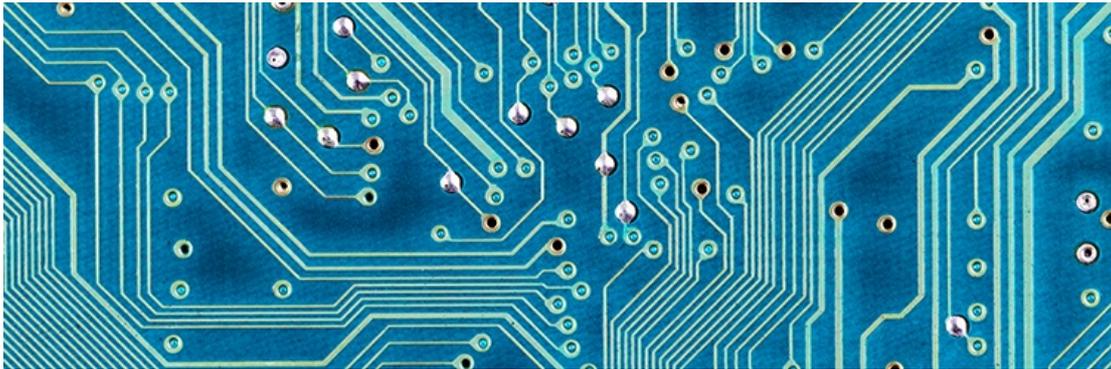
Presentación

A continuación, se van a presentar los **sistemas secuenciales**, para lo que debemos entender en qué consisten los sistemas **síncronos** y los **asíncronos**.

En los sistemas **combinacionales**, las salidas solo dependían de las entradas y el elemento que utilizaban para que funcionara, eran las **puertas lógicas**. En cambio, en los **sistemas secuenciales**, se va a ver que las salidas ya **no solo dependen de las entradas**, sino que pueden depender de situaciones anteriores, por lo que uno de los elementos necesarios para que puedan funcionar dichos sistemas es el concepto de **memoria**.

Los **objetivos** que se pretenden alcanzar en este tema son los siguientes:

- Introducir el concepto de **sistema secuencial**.
- Detallar en qué consisten los **sistemas síncronos y asíncronos** de los sistemas secuenciales.
- Analizar las **salidas de los sistemas secuenciales**.



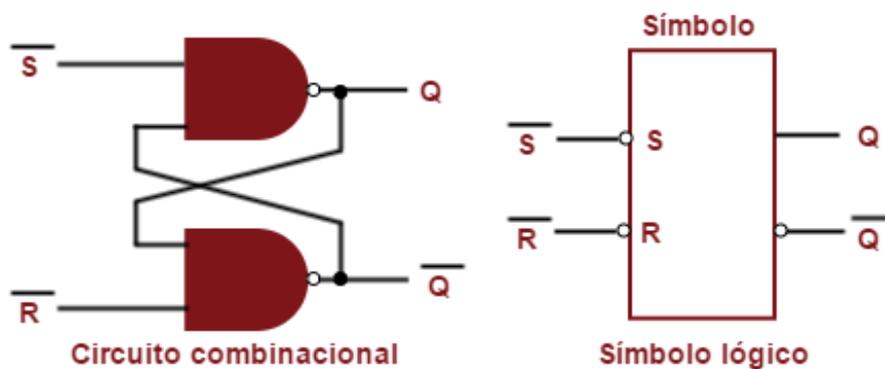
El biestable

Tal como hemos introducido, los sistemas secuenciales son aquellos que dependerán de las entradas y de las situaciones anteriores. Para recordar esas situaciones anteriores, necesitaremos un **sistema de almacenamiento temporal**, que denominaremos **memoria**.

En los sistemas secuenciales básicos, el elemento de memoria que utilizaremos para almacenar un bit es el **biestable**.

El **biestable** es el dispositivo encargado de **almacenar un único bit**, mientras que el dispositivo encargado de almacenar grupos de bits es el **registro**.

Un biestable es un **circuito combinacional simple**. Hay varios tipos de circuitos biestables, uno de los más simples es el R-S, cuya construcción se realiza con solo dos puertas **NAND**. A continuación, se presenta un **ejemplo** gráfico de un circuito biestable y su representación lógica:



En un sistema secuencial, se usan estos biestables para **dotar de memoria al sistema**. Por tanto, en este tipo de circuitos es necesario **implementar biestables** en su construcción.

Sincronismo y asincronismo

Hay que tener en cuenta la influencia del **factor tiempo** en los sistemas secuenciales, para diferenciarlos en sistemas síncronos o asíncronos, dependiendo de cuándo tengamos en cuenta las señales.

Además, existen diversas técnicas para la representación de los sistemas secuenciales en función de la variable tiempo, tales como los **diagramas de transición de estados** o las **tablas de verdad**.

Un sistema secuencial [pasa de un estado a otro](#) dependiendo de las entradas a lo largo del tiempo. Por otro lado, los sistemas secuenciales síncronos solo son capaces de alimentarse de las entradas en unos determinados instantes de tiempo.

Pasa de un estado a otro

En los sistemas **síncronos**, el cambio de estado solo se produce en unos estados de tiempo que vienen definidos por una **señal externa al sistema** (señal de reloj).

Un sistema secuencial **asíncrono** no depende del tiempo directamente, sino que, según se **producen las entradas**, se van modificando los estados.

Cuando un sistema no se encuentra sincronizado, por mucho que cambiaran las entradas, los estados del sistema no se verían modificados, mientras que si se encontraran sincronizados, estarían pendientes de las entradas para que, cuando el instante de tiempo se lo permita, se modifique el estado del sistema.

Los sistemas secuenciales **síncronos** son los que predominarán a lo largo del tema, por ser **más sencillos**, aunque los sistemas **asíncronos** son los más **comunes** en la vida real, pudiendo transformarse en síncronos en cualquier momento.

El **sincronismo del reloj** en los biestables puede ser:

Por nivel	El biestable está dispuesto a cambiar de estado durante todo el tiempo que el reloj está en un determinado valor (por ejemplo mientras está a un 1 lógico).
Por flanco	El biestable solo está dispuesto a cambiar de estado, en el momento en el que el reloj cambia de estado . Es decir, en el momento que pasa de nivel 0 a 1 lógico (flanco de subida) o viceversa (flanco de bajada).

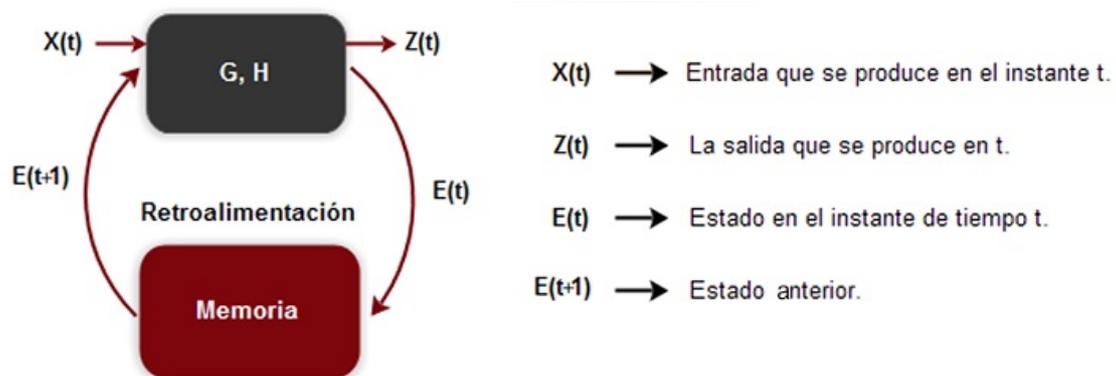
El estado

En los sistemas secuenciales la salida a la que podemos denominar Z , en un determinado instante de tiempo T_i , depende de la entrada X en ese mismo instante de tiempo T_i , además de todos los instantes de tiempo anteriores.

Por esto mismo, nuestros sistemas necesitan el uso de la memoria para almacenar la situación en la que están, a esto se le denomina **estado**.

A continuación, se presenta un **gráfico** representativo del estado:

- $Z(t) = G \cdot (E(t), X(t))$ (donde entendemos que G es la función de salida.
- $E(t) = H \cdot (E(t-1), X(t-1))$ (donde entendemos que H es la función de transición.



Los sistemas secuenciales son finitos, por lo que poseen un almacenamiento de memoria finito. Se puede concluir que posee una serie de **estados finitos**.

La retroalimentación

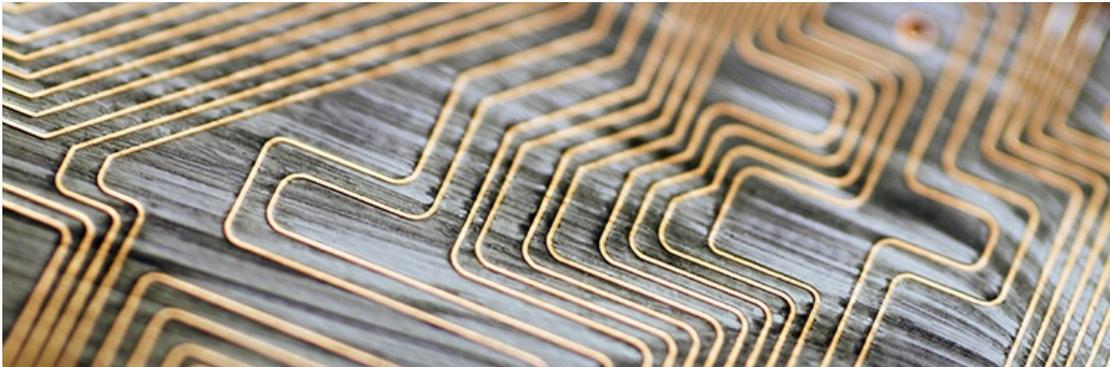
Los sistemas secuenciales deben **disponer de memoria** para almacenar los diferentes estados por los que van atravesando. El contenido de estos elementos de memoria **puede cambiar** a lo largo del tiempo.

Los estados vienen determinados por el **contenido** de los elementos de memoria en cada instante.

En los **sistemas secuenciales** es bastante frecuente encontrar una señal encargada de **inicializar** los elementos de memoria o estados con un **valor predeterminado**. Esta señal de inicialización suele denominarse **Reset**.

La **señal de inicialización** determina el estado del sistema en el momento inicial y por lo general, cambia los **valores** de las memorias a cero.

La salida de un sistema secuencial viene determinada por la entrada y por un estado anterior del sistema. Por lo que se puede deducir que el estado actual en el que se encuentra el sistema, junto con la entrada, determinará el estado en el siguiente instante de tiempo, a lo que denominaremos **realimentación**.



Biestado R-S síncrono por nivel alto

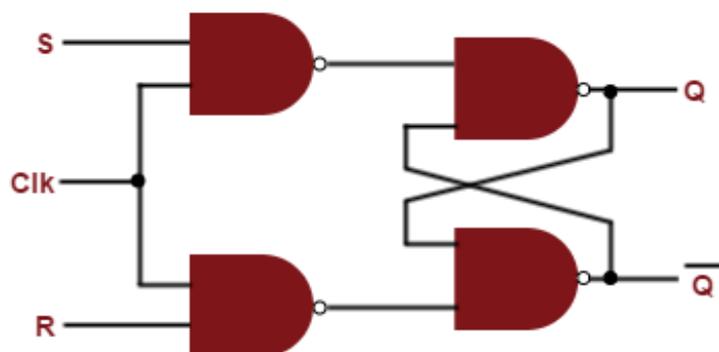
Se le llama R-S, por las siglas de sus señales de excitación: S=Set y R=Reset que provocan que el biestado almacene un 1 o un 0 correspondientemente a la activación de estas señales. Cuando al biestado no se le realiza ni un Set ni un Reset, manteniendo sus entradas a 0 lógico, el biestado R-S mantiene su valor estable. La combinación Set y Reset simultáneamente está prohibida, siendo el comportamiento del biestado en este caso indeterminado.

Cuando el biestado es **síncrono por nivel alto**, también se necesita la señal **clk=Clock** que va a determinar el momento en el que el biestado lee las entradas y actúa en consecuencia. Si el clk está a nivel bajo, el biestado no hace caso de las entradas.

Clk	S(t)	R(t)	Q(t+1)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	U
0	X	X	Q(t)

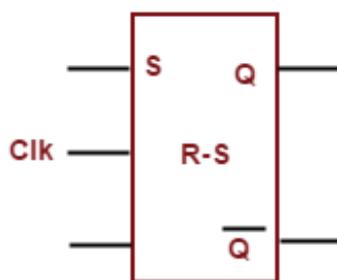
Tabla de verdad de funcionamiento del biestado.

Nota: Q(t) representa el **valor almacenado** actualmente en el biestado, Q(t+1) representa el **estado que alcanzaría** el biestado con esas entradas.



Diseño de la implementación mediante puertas lógicas del biestado

A continuación, se presenta el esquemático que se usa para hacer **diseños de circuitos**:



Otros tipos de biestables

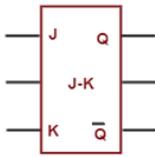
A continuación, presentamos otros tipos de biestables:

Biestable J-K síncrono por nivel (alto)

Modificación del biestable R-S, cuya principal mejora, es que la **combinación de las dos señales de entrada a 1**, prohibida en el R-S, en el J-K provoca la alternancia del valor guardado en el biestable.

Por lo demás, la señal J funciona como un Set y K como un Reset.

Esquemático



Implementación

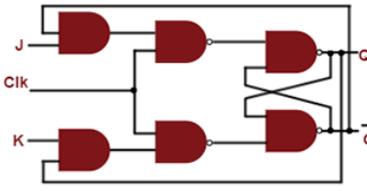


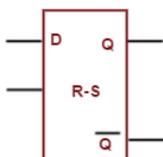
Tabla de funcionamiento

Clk	J(t)	K(t)	Q(t+1)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}(t)$
0	X	X	Q(t)

Biestable D (Delay) síncrono por nivel

A diferencia de los anteriores, solo tiene una **señal de entrada**. Si esta señal está a 1, el valor guardado en 1 y si está a 0, el valor guardado es 0. No tiene combinaciones de entradas que signifiquen el “no hacer nada” como los casos de los biestables anteriores.

Esquemático



Implementación

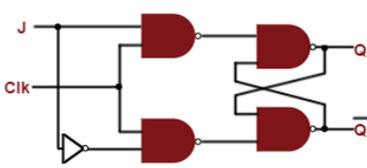


Tabla de funcionamiento

Clk	D(t)	Q(t+1)
1	0	0
1	1	1
0	X	Q(t)

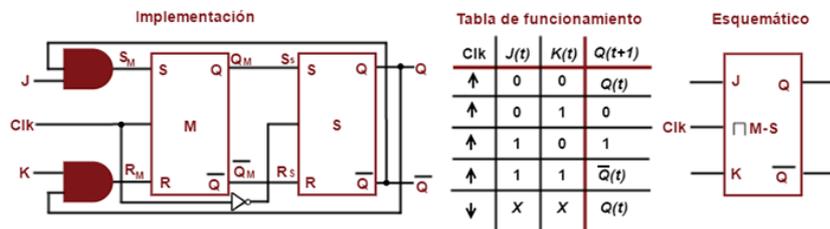
© Universidad Europea de Madrid. Todos los derechos reservados. 11

Biastable J-K maestro/esclavo

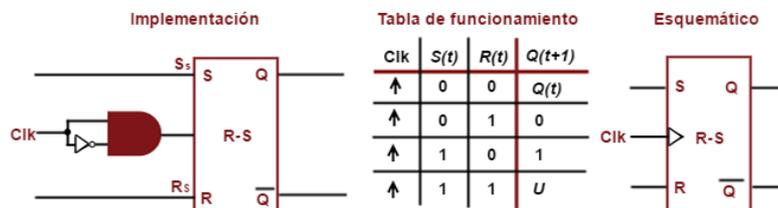
Aproximación a hacer un biastable que el **tiempo de excitación sea muy reducido**, no siendo todo el tiempo en el que el clk permanece a 1.

Esto se consigue gracias a la puerta NOT que hay en su implementación. Esta puerta hace que durante medio ciclo de clk , solo la primera parte del circuito responde a la señales de entrada, mientras que durante la segunda mitad del ciclo de reloj solo responde a las entradas. Por tanto el circuito **solo es excitable** cuando estos **dos tiempos se solapan**. Y este tiempo de solapamiento lo proporciona en el retraso que mete esta puerta NOT.

Es decir, en el instante en el que el reloj pasa de 0 a 1, la primera mitad entraría en el ciclo alto de reloj, lo que pasaría a responder a las entradas, por el contrario la segunda mitad, al estar el clk negado, pasaría a una fase de clk nivel bajo, lo que dejaría de responder a las entradas. Pero si tenemos en cuenta que la puerta NOT no es instantánea, sino que tiene un cierto retraso de milisegundos, sería durante este tiempo en el que las dos mitades funcionarían simultáneamente, lo que provocaría la actualización del biastable



Biastable R-S síncrono por flanco (flip-flop)



<p>Biestable J-K síncrono por flanco (flip-flop)</p>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Implementación</p> </div> <div style="text-align: center;"> <p>Tabla de funcionamiento</p> <table border="1"> <thead> <tr> <th>Clk</th> <th>Clk</th> <th>S(t)</th> <th>R(t)</th> <th>Q(t+1)</th> </tr> </thead> <tbody> <tr> <td>↑</td> <td>↓</td> <td>0</td> <td>0</td> <td>Q(t)</td> </tr> <tr> <td>↑</td> <td>↓</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>↑</td> <td>↓</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>↑</td> <td>↓</td> <td>1</td> <td>1</td> <td>$\bar{Q}(t)$</td> </tr> </tbody> </table> </div> <div style="text-align: center;"> <p>Esquemático</p> </div> </div>	Clk	Clk	S(t)	R(t)	Q(t+1)	↑	↓	0	0	Q(t)	↑	↓	0	1	0	↑	↓	1	0	1	↑	↓	1	1	$\bar{Q}(t)$
Clk	Clk	S(t)	R(t)	Q(t+1)																						
↑	↓	0	0	Q(t)																						
↑	↓	0	1	0																						
↑	↓	1	0	1																						
↑	↓	1	1	$\bar{Q}(t)$																						
<p>Biestable D síncrono por flanco (flip-flop)</p>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Implementación</p> </div> <div style="text-align: center;"> <p>Tabla de funcionamiento</p> <table border="1"> <thead> <tr> <th>Clk</th> <th>R(t)</th> <th>Q(t+1)</th> </tr> </thead> <tbody> <tr> <td>↑</td> <td>0</td> <td>0</td> </tr> <tr> <td>↑</td> <td>1</td> <td>1</td> </tr> </tbody> </table> </div> <div style="text-align: center;"> <p>Esquemático</p> </div> </div>	Clk	R(t)	Q(t+1)	↑	0	0	↑	1	1																
Clk	R(t)	Q(t+1)																								
↑	0	0																								
↑	1	1																								
<p>Biestable T (Toggle) síncrono por flanco (flip-flop)</p>	<p>Este nuevo biestable, funciona con una sola entrada, al igual que el biestable D, solo que este si tiene una 1 en su entrada, el biestable cambia su valor, por el contrario si tiene un 0, el biestable mantiene inalterable su valor.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Implementación</p> </div> <div style="text-align: center;"> <p>Tabla de funcionamiento</p> <table border="1"> <thead> <tr> <th>Clk</th> <th>Q(t)</th> <th>Q(t+1)</th> </tr> </thead> <tbody> <tr> <td>↑</td> <td>0</td> <td>1</td> </tr> <tr> <td>↑</td> <td>1</td> <td>1</td> </tr> </tbody> </table> </div> <div style="text-align: center;"> <p>Esquemático</p> </div> </div>	Clk	Q(t)	Q(t+1)	↑	0	1	↑	1	1																
Clk	Q(t)	Q(t+1)																								
↑	0	1																								
↑	1	1																								

El registro

Un **registro** es una cadena de biestables “D” encapsulados juntos, de modo que permiten **almacenar una palabra** (cada biestable almacena un bit). En los registros, las señales de control **son comunes** a todos los biestables. El reloj es también común a todos ellos, de modo que **funcionan en bloque**. Para almacenar datos, la señal de reloj coincide con la de escritura.

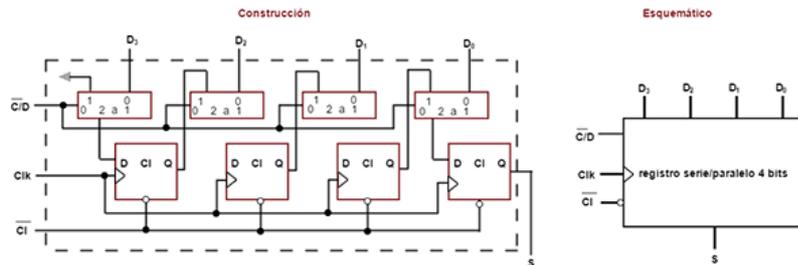
Los **tipos básicos** de registros son los siguientes:

<p>Registro paralelo/paralelo</p>	<p>La carga de los registros es simultánea al instante de flanco de reloj (clk ↑) y la lectura del valor almacenado también es simultánea.</p> <p>La señal extra CL=Clear permite inicializar el registro a valores preestablecidos. Esta es una señal asíncrona, teniendo efecto en cualquier momento, indistintamente del estado del clk.</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="518 974 1045 1198"> <p>Construcción</p> </div> <div data-bbox="1077 974 1316 1198"> <p>Esquemático</p> </div> </div>
<p>Registro serie/paralelo</p>	<p>Se carga un bit en cada pulso de reloj, empezando por el bit menos significativo, este bit se va desplazando a los bits más significativos (propagación) a cada clk, dejando hueco para la entrada de un nuevo bit. La lectura es en paralelo, todos a la vez.</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="534 1489 1045 1713"> <p>Construcción</p> </div> <div data-bbox="1077 1489 1316 1713"> <p>Esquemático</p> </div> </div>

Registro paralelo/serie

Se cargan todos los bits de una vez de **manera paralela**, pero su lectura es **secuencial**, leyendo un bit a cada clk de reloj.

Hay que fijarse en los componentes que ponen “2 a 1”, estos son **multiplexores de dos entradas**, que según una señal de control C/D periten la Carga (C/D=0) o la lectura del valor almacenado (C/D=1).

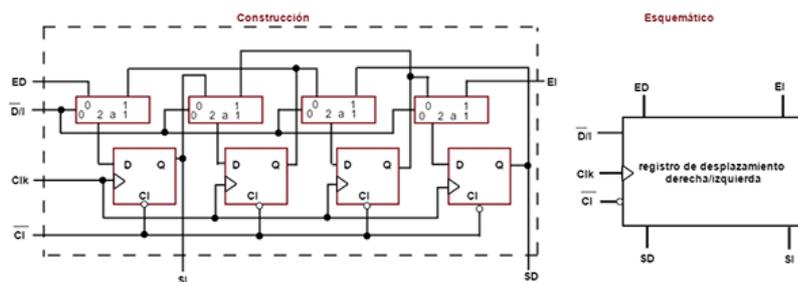


Registro serie/serie o de desplazamiento derecha/izquierda

Tanto el almacenaje de información, como su lectura son en **serie**, a cada pulso de reloj. Una **característica importante** de este circuito es que permite la **carga y lectura en cualquier dirección**.

Para ello, hay una señal de control, D/I que selecciona en qué sentido se hace la carga o la lectura.

Las señales ED y EI son las señales de lectura/escritura Derecha e Izquierda respectivamente. Pudiendo funcionar bidireccionalmente dependiendo la dirección de desplazamiento activa.

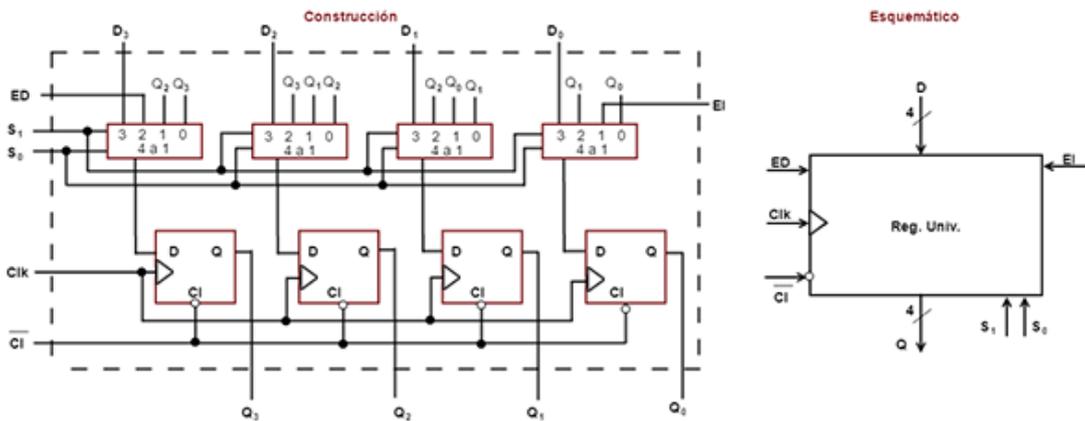


El registro universal

Es el registro que combina el funcionamiento de los **cuatro modelos** anteriores. Para ello tiene **dos bits de control**, que especifican el funcionamiento:

S_1	S_0	Función
0	0	Inhibición
0	1	Desplazamiento izquierda
1	0	Desplazamiento derecha
1	1	Carga paralela

Todo el comportamiento de su funcionamiento, reside en los **multiplexores 4a1** que redireccionan las entradas y salidas de cada biestable:





Resumen

Se ha realizado una aproximación de lo que es un **sistema secuencial** y al principal componente que este necesita: el estado, **sincronismo**, y el **biestable**.

Se han analizado el funcionamiento de los **cuatro** tipos básicos de biestables:

- R-S.
- J-K.
- D (Delay).
- T (Toggle).

Se han agrupado biestables para generar **registros** y se han estudiado los diferentes métodos de **carga** y **lectura**:

- Paralelo – paralelo.
- Paralelo – serie.
- Serie – paralelo.
- Serie – serie.

¡Enhorabuena! Has finalizado con éxito