

Tecnología de Computadores  
Práctica 8  
Sistemas secuenciales:  
Biestables - Registros

Universidad Miguel Hernández <sup>1</sup>

27 de febrero de 2016

<sup>1</sup>Copyright (c) 2016 P. Pablo Garrido Abenza. Todos los derechos reservados.



## Resumen

Este documento describe la práctica número 8 de la asignatura Tecnología de Computadores de 2º del Grado en Ingeniería Informática en Tecnologías de la Información de la Universidad Miguel Hernández. Este material docente ha sido desarrollado enteramente, a menos que se indique lo contrario, por el profesor P. Pablo Garrido Abenza.

# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Objetivos . . . . .	3
1.2. Software necesario . . . . .	3
<b>2. Descripción</b>	<b>4</b>
2.1. Biestables . . . . .	4
2.2. Registro de almacenamiento . . . . .	6
2.3. Registro de desplazamiento . . . . .	7
2.4. Registro carga paralelo asíncrona y serie . . . . .	8
<b>3. Entrega de la práctica</b>	<b>10</b>
<b>Glosario de Acrónimos</b>	<b>11</b>

# Capítulo 1

## Introducción

### 1.1. Objetivos

Tras completar esta práctica el alumno deberá ser capaz de:

- Buscar y manejar la hoja de características (*datasheet*) de algún circuito integrado (IC).
- Conocer el funcionamiento de los distintos biestables utilizados para el diseño de circuitos secuenciales.
- Diseñar y comprobar circuitos secuenciales a partir de biestables y otros componentes combinacionales.
- Comprender el concepto de registro (de almacenamiento o de desplazamiento, con carga serie o paralelo).

### 1.2. Software necesario

El software necesario para la realización de las siguientes prácticas es *Simulín* v5.61 o posterior, el cual se encuentra ya instalado en los ordenadores del aula de informática. También puede se puede instalar en cualquier otro ordenador personal, ya que este software es de libre distribución. Está disponible para Windows, Linux y Mac OS X, y se puede descargar desde el material de la asignatura, o también, desde el siguiente enlace.

## Capítulo 2

# Descripción

Antes de comenzar con el diseño de sistemas secuenciales, comprobaremos el funcionamiento de los distintos biestables. Después, implementaremos un registro de almacenamiento. También implementaremos un registro que permita tanto la carga en paralelo (almacenamiento) como en serie (desplazamiento).

### 2.1. Biestables

Los **biestables** son circuitos lógicos capaces de permanecer en uno de entre dos estados estables, aún después de desaparecer la causa que provocó el paso al estado alcanzado. Son, pues, capaces de almacenar una información binaria (1 bit). Los biestables más comunes son los del tipo S-R, J-K, D, y T, cuyas tablas de transición se muestran en la Figura 2.1. En este apartado se pide simplemente comprobar el buen funcionamiento de los 4 tipos de biestables. Se trata de hacer un circuito con un único biestable, con las entradas y salidas necesarias, incluyendo para probar las entradas asíncronas prioritarias **Clear** y **Preset**. En cuanto al mecanismo de disparo, utilizaremos siempre por flanco de bajada o de subida. La tabla 2.1 enumera los archivos a realizar.

#	Descripción	Archivo
1	Biestable S-R	biestable_sr.sim
2	Biestable J-K	biestable_jk.sim
3	Biestable D	biestable_d.sim
4	Biestable T	biestable_t.sim

Cuadro 2.1: Archivos a implementar

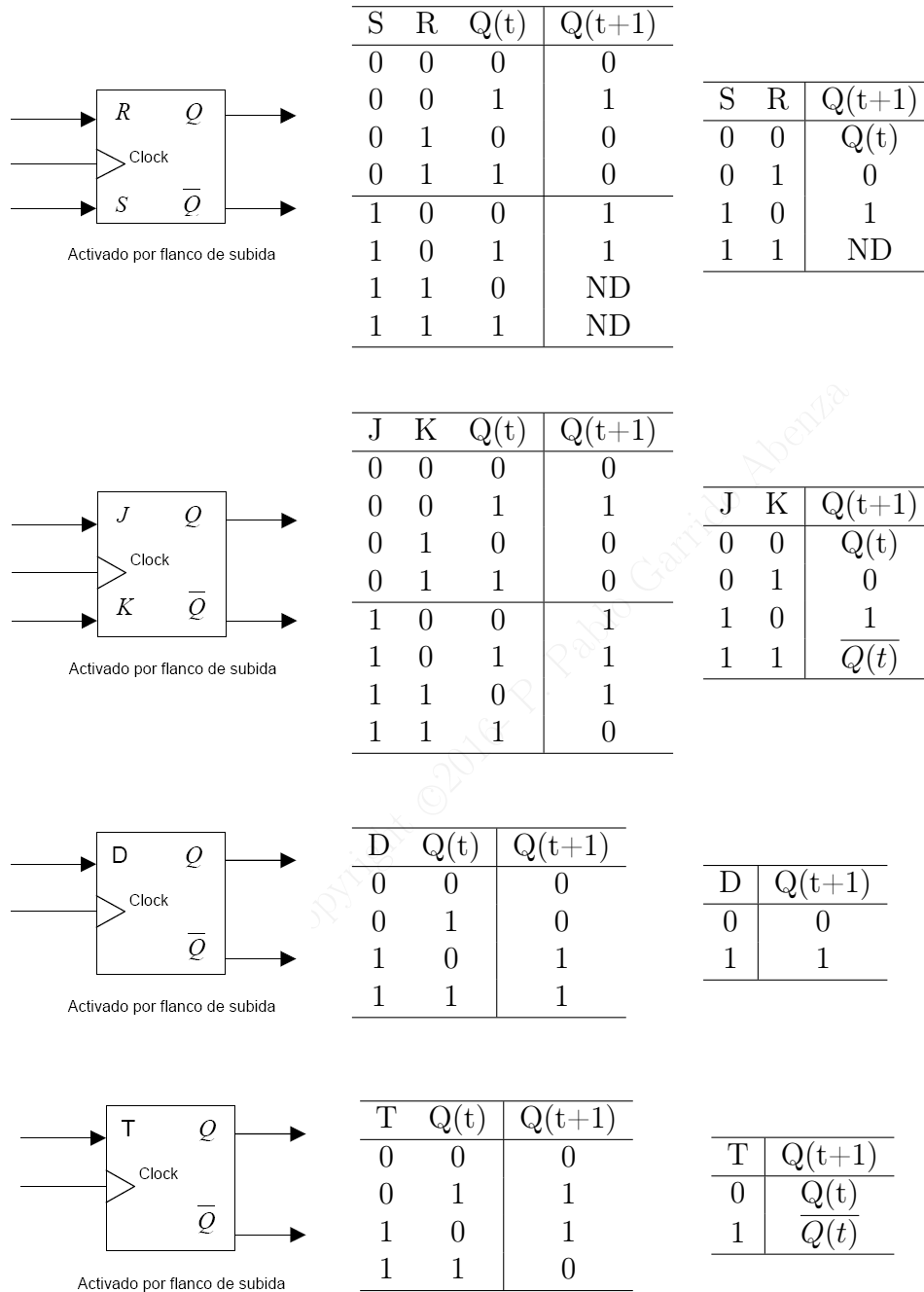


Figura 2.1: Biestables: esquemas de bloque y tablas de transición

## 2.2. Registro de almacenamiento

Diseñar e implementar un registro de almacenamiento de 4-bit utilizando **biestables de tipo D** activados por **flanco de bajada**, y sincronizado por una señal de reloj. Se trataría de emular el IC 74194, pero simplificado, de tal manera que no hay entradas de operación puesto que la única operación que realiza es la de carga en paralelo (no la de desplazamientos a la izquierda o derecha). Es decir, cada pulso de reloj, el registro cargará en paralelo el valor que tenga en las entradas.

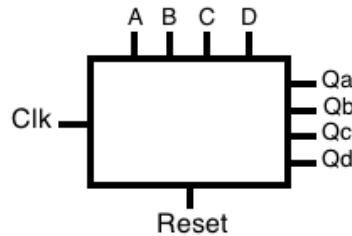


Figura 2.2: Registro de almacenamiento

Como se puede observar en la figura 2.2, lo que sí tendrá será una señal **Reset** para la puesta a 0 del registro cuando valga 1. Puesto que esa señal se conectará a la entrada **CLR'** de los biestables y dicha entrada se activa a nivel bajo, la entrada **Reset** del registro habrá que negarla (archivo: `rgtroalmac.sim`).

## 2.3. Registro de desplazamiento

En este apartado se pide implementar un registro de 4-bit de desplazamiento, esto es, de carga y salida serie. según el símbolo de bloque mostrado en la figura 2.3. En este caso se utilizarán **biestables S-R** activados por **flanco de subida** (archivo: `rgtrodespl.sim`).

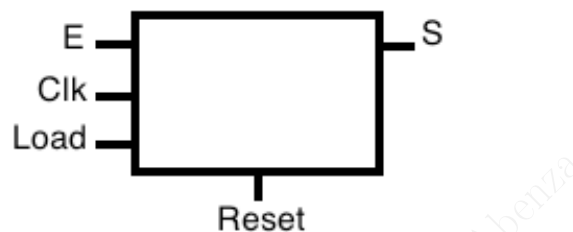


Figura 2.3: Registro de desplazamiento

Como se puede observar, tendrá una única entrada de datos **E** y una única salida de datos **S**. Cada pulso de reloj provocará que se cargue el valor de **E** en el primer biestable, desplazándose el valor anterior al siguiente biestable, y así sucesivamente, quedando el valor del último biestable en la salida **S**.

El circuito tendrá una entrada adicional **Reset** tal que cuando se active (valor 1), se inicialice el registro a 0; para ello se utilizará la entrada asíncrona **CLR'** de los biestables, la cual actúa a nivel bajo.

La señal de reloj **Clk** (en combinación con **Load**, ver abajo) se conectará a todos los biestables, los cuales, como se ha dicho antes, son activados por **flanco de subida**. Puede configurarse la velocidad de funcionamiento del reloj para proceder a la simulación, aunque también puede cambiarse su valor manualmente como cualquier otro módulo de entrada.

Por último, habrá también una señal **Load**, que permita controlar el momento en que el usuario quiere cargar el registro, ya que de lo contrario, el registro siempre desplazará en cada flanco de subida de reloj. Cuando **Load=1**, el registro funcionará como se ha descrito, desplazando un bit en cada pulso de reloj; cuando **Load=0** el registro (los biestables) ignorará la señal de reloj, por lo que habrá que añadir la lógica necesaria para combinar **Clk** con **Load**. En otras palabras, **Load** actúa como una señal de inhibición del reloj.



**NOTA:** Aunque no es necesario, con objeto de facilitar el análisis del funcionamiento con el cronograma pueden incluirse módulos de salida para mostrar la salida **Q** de cada biestable.



## 2.4. Registro carga paralelo asíncrona y serie

Para terminar la práctica realizaremos un registro de 4-bit que permita tanto la **carga asíncrona** en paralelo (almacenamiento) como carga **síncrona** en serie (desplazamiento). Como se puede ver en el símbolo de bloque mostrado en la figura 2.4, el registro dispone de unas entradas y salidas parecidas al caso anterior, añadiendo las 4 entradas en paralelo, y sustituyendo la señal Load por una llamada Despl-Carga' (para abreviar la llamaremos CD). Interiormente, utilizaremos **biestables D** activados por **flanco de subida**, con señales asíncronas Preset' y Reset' (archivo: rgtroalmacdespl.sim).

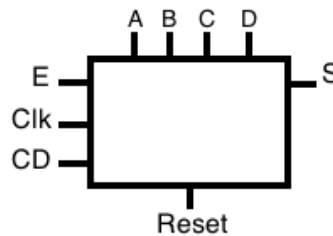


Figura 2.4: Registro de almacenamiento y desplazamiento

La entrada E se utilizará para introducir un nuevo valor cuando se haga un **desplazamiento** (CD=1), modificando también la salida S del mismo modo que antes. Sin embargo, cuando la señal CD=0, esa entrada E se ignorará; lo que se hará en ese caso será una **carga** del registro con los 4 valores de las entradas A-B-C-D de forma asíncrona, es decir, sin tener en cuenta la señal de reloj Clk.

La forma de proceder para realizar la **carga asíncrona en paralelo** es actuando sobre las entradas asíncronas Preset' y Clear' de los biestables, pero sólo cuando CD=0. Hay que conseguir mediante la lógica necesaria, que cuando una entrada A-B-C-D del registro sea 0 se active el Clear=0 de su biestable correspondiente, y cuando sea 1, se active el Preset=0, teniendo en cuenta que ambas entradas se activan a nivel bajo. Asimismo, hay que tener en cuenta también la señal RESET, que pone todos los biestables a 0 cuando RESET=1.

Para averiguar la función lógica del circuito combinacional que hay que colocar en las entradas Preset' y Clear' de los distintos biestables, podemos escribir una tabla de verdad con las entradas que intervienen: la señal RESET, la señal CD, y una de las entradas A-B-C-D del registro; puesto que las funciones a calcular Preset' y Clear' serán las mismas para todos los

biestables, a excepción del nombre de la entrada de datos, a dicha entrada la podemos llamar **X** en la tabla, para hacerlo de forma genérica. Finalmente Luego podemos simplificar las dos funciones mediante mapas de Karnaugh.

RESET	CD	X	Clear'	Preset'
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

### Para pensar:



- ¿Qué le ocurre a una puerta lógica AND si alguna patilla de entrada se deja sin conectar? ¿Sucedre esto mismo en la realidad? ¿Cuál es la recomendación para evitar problemas? ¿Qué le ocurre a un biestable suponiendo si se deja la patilla Clear 'al aire' (sin conectar)?.
- En el apartado del registro de almacenamiento se han utilizado **biestables D**, ¿qué modificaciones habría que realizar para utilizar **biestables S-R**?
- Suponiendo que sólo tuviésemos biestables activados por **flanco de bajada** y quisiéramos implementar un registro activado por **flanco de subida**, ¿se te ocurre alguna solución?.

## Capítulo 3

### Entrega de la práctica

La forma de entrega de esta práctica se realizará de la siguiente forma:

1. Comprimir en un único archivo todos los circuitos \*.sim implementados. Para ello se puede utilizar la opción 'Copia de seguridad' de Simulín, o programas como WinZip o 7z, admitiéndose los formatos: .zip, .7z, o .rar.
2. Subir el archivo comprimido a la tarea correspondiente antes del plazo fijado.

Los archivos a entregar para esta práctica se resumen en la tabla 3.

#	Descripción	Archivo
1	Biestable S-R	biestable_sr.sim
2	Biestable J-K	biestable_jk.sim
3	Biestable D	biestable_d.sim
4	Biestable T	biestable_t.sim
5	Registro almacenamiento	rgtroalmac.sim
6	Registro de desplazamiento	rgtrodespl.sim
7	Registro almac./desplaz.	rgtroalmacdespl.sim

Cuadro 3.1: Archivos para los circuitos a implementar



**RECUERDA:** las prácticas serán corregidas por un programa informático, por lo que se insiste en que los archivos de los circuitos tengan el nombre que se especifica en las tablas, respetando incluso mayúsculas/minúsculas; el nombre del archivo comprimido que los contiene no importa.

# Glosario de Acrónimos

Siglas	Significado
BCD	Binary-Coded Decimal
IC	Integrated Circuit
LED	Light-emitting diode
LSB	Least Significant Bit
MSB	Most Significant Bit