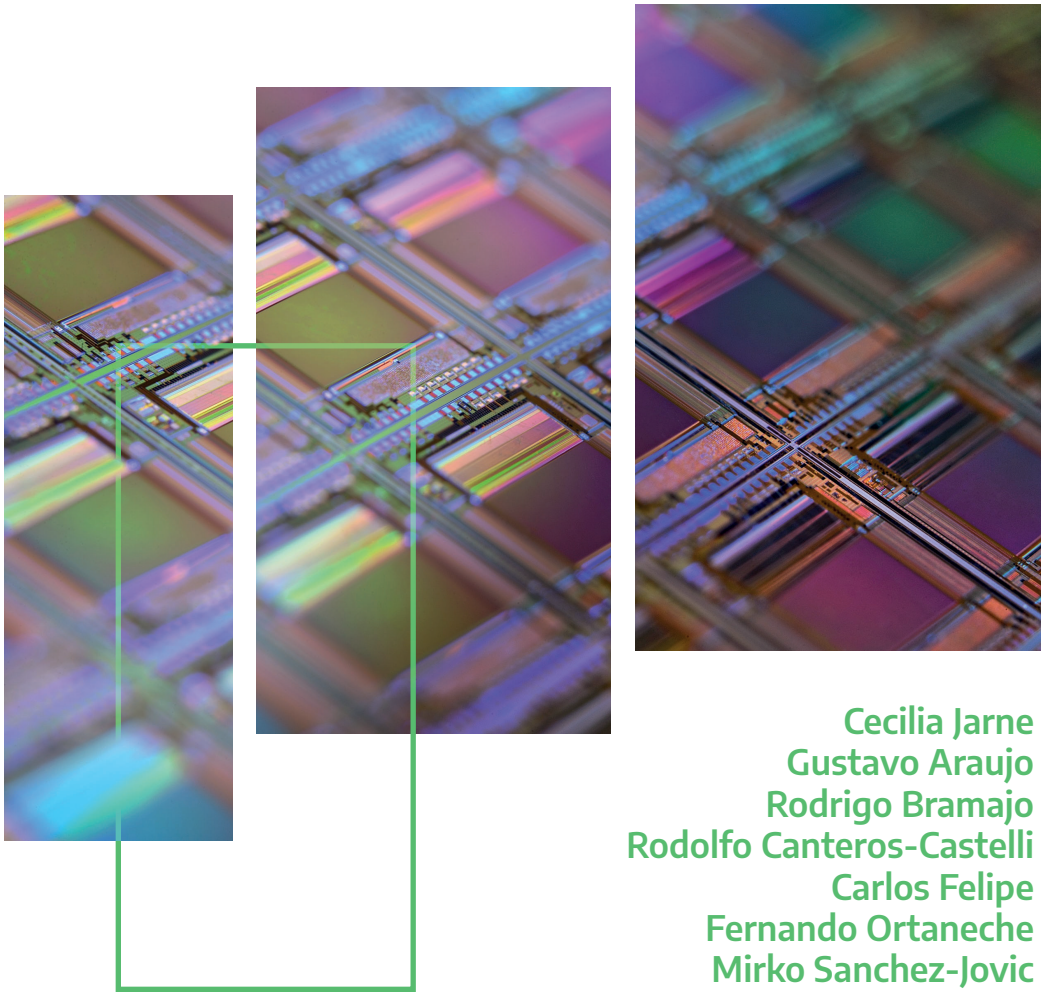
FUNDAMENTOS DE LOS CIRCUITOS SECUENCIALES



Cecilia Jarne
Gustavo Araujo
Rodrigo Bramajo
Rodolfo Canteros-Castelli
Carlos Felipe
Fernando Ortaneche
Mirko Sanchez-Jovic

la educación nuestra bandera

inet | Instituto Nacional de Educación Tecnológica

Ministerio de Educación
Argentina

Colección: Electrónica

FUNDAMENTOS DE LOS CIRCUITOSSECUENCIALES

Autores

Cecilia Jarne
Gustavo Araujo
Rodrigo Bramajo
Rodolfo Canteros-Castelli
Carlos Felipe
Fernando Ortaneche
Mirko Sanchez-Jovic

Breve descripción de la ilustración de la cubierta
Placa de Silicio con integrados. Cada rectángulo es un chip con transistores microscópicos y circuitos. Fotografía obtenida con macro por Laura Ockel.

AUTORIDADES

Presidente
Dr. Alberto Ángel Fernández

Vicepresidenta
Dra. Cristina Fernández de Kirchner

Jefe de Gabinete de Ministros
Dr. Juan Luis Manzur

Ministro de Educación
Lic. Jaime Perczyk

Unidad Gabinete de Asesores
Prof. Daniel José Pico

Director Ejecutivo del Instituto Nacional
de Educación Tecnológica
Prof. Gerardo Marchesini

Fundamentos de los circuitos secuenciales / Cecilia Jarne... [et al.] ; coordinación general de María Jazmin Belossi ; dirigido por Gerardo Marchesini. - 1a ed. - Ciudad Autónoma de Buenos Aires : Ministerio de Educación de la Nación. Instituto Nacional de Educación Tecnológica, 2022.
Libro digital, PDF

Archivo Digital: descarga
ISBN 978-950-00-1656-8

1. Circuitos Electrónicos. 2. Circuitos Integrados Digitales. I. Jarne, Cecilia. II. Belossi, María Jazmin, coord. III. Marchesini, Gerardo, dir.

CDD 621.31

“Dedicado a la memoria
de nuestro querido Carlos Felipe”

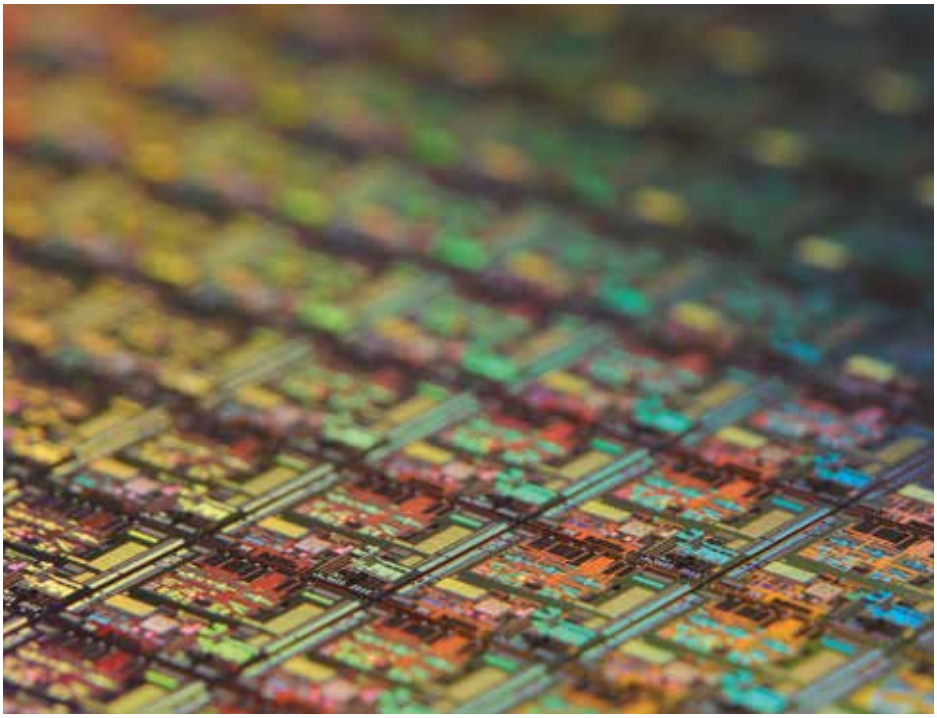
AGRADECIMIENTOS

Los autores agradecemos principalmente a Diego Golombek, por ayudarnos a concretar este proyecto. Gracias por la iniciativa para impulsar la publicación de este texto a él y a Celeste Guagliano. También agradecemos mucho a todas las autoridades del INET.

Un especial agradecimiento es para Haydee Noceti por editar el libro con tanta dedicación y cuidado que permitieron producir un material de calidad que sea útil para un público amplio, pero sin perder rigor. Toda su experiencia enriqueció profundamente este proyecto de un modo que no imaginábamos fuera posible.

Finalmente queríamos agradecer a Roberto Rivas, Daniel Bertolotto y Gabriel Lio por sus valiosas contribuciones. También a Pablo Mereles, quien vino al rescate cuando el hardware falló durante las etapas finales de edición del libro.

FUNDAMENTOS DE LOS CIRCUITOS SECUENCIALES



Cecilia Jarne
Gustavo Araujo
Rodrigo Bramajo
Rodolfo Canteros-Castelli
Carlos Felipe
Fernando Ortaneche
Mirko Sanchez-Jovic

CRÉDITOS

Ilustraciones
Gabriel Lio. Dibujos de líneas.

Esquemas eléctricos y diagramas
Roberto Rivas; Fernando Ortaneche, Gustavo Araujo, Rodolfo Canteros, Cecilia Jarne. Todas las imágenes de líneas y fondo blanco y diagramas y/o esquemas que aparecen.

Fotografías
Todas las fotografías fueron extraídas de bancos de imágenes abiertos: Creative common Licence.
Los autores de las fotografías con créditos son: Laura Ockel (cubierta), Nicolas Thomas, Harrison Broadbent, Liam Briese, Marília Castelli, Franck V., Blaz Erzetic, Florian Olivo, Davi Diório, Mendes Robin Glauser, Luis Perelo, Ranjat, M. Kritsada Jaiyen, Jorge Guillen, Tiki Shabudin, Ingo Jakubke, Emilian Robert Vicol, Arnold Reinhold y Patrick Martin.

Coordinación Académica y Edición:
Prof. Ing. Haydeé Noceti

Diseño Gráfico:
DG Carolina Macedra

LOS AUTORES
Docentes de la EEST N° 2 “Paula Albarracín de Sarmiento”

CECILIA JARNE
Técnica en Electrónica. Dra. de la Facultad de Ciencias Exactas de la UNLP - Área Física. Actualmente trabaja como Profesora en la Universidad Nacional de Quilmes, en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento” y es Investigadora Asistente del CONICET.

GUSTAVO ARAUJO
Técnico Superior en Electrónica. Profesor en Disciplinas Industriales (especialidad Electrónica). Lic. en Educación UNQ. Trabaja actualmente en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento”, E.E.T. N° 1 “Ing. Otto Krause” en CABA y el Instituto Madero (Ciudad Madero).

RODRIGO HERNÁN BRAMAJO
Técnico en Electrónica. Profesor de Educación Técnico Profesional en concurrencia con título de base. Trabaja actualmente como docente de las materias montajes de proyectos electrónico y Jefe de área en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento”. Apasionado por el Montañismo y la Escalada.

RODOLFO A. CANTEROS CASTELLI
Técnico en Electrónica. Profesor de las asignaturas Aplicaciones de Electrónica Digital I y Aplicaciones de Electrónica Analógica I. Trabaja actualmente en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento” y como personal de apoyo para la investigación en CONICET.

CARLOS ALBERTO FELIPE
Técnico en Electrónica. Maestro de Enseñanza Primaria. Profesor de Electrónica Analógica y Digital. Trabaja en la E.E.S.T. N° 5 “Salvador Debenedetti” de la Ciudad de Avellaneda “El Paláa” y la E.E.S.T. N°2 “Paula Albarracín de Sarmiento”. Difusor de temas de Electrónica en las redes. Guionista, productor y animador radial.

LUIS FERNANDO ORTANECHE
Técnico en Electrónica. Ingeniero en Electrónica de la UTN. Trabaja actualmente en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento”. Simplemente un docente que comparte el saber con sus alumnos sin limitaciones.

MIRKO SÁNCHEZ JOVIC
Técnico en Electrónica. Profesor de la asignatura Lenguajes Electrónicos. Trabaja actualmente en la E.E.S.T. N° 2 “Paula Albarracín de Sarmiento”. Desarrollador Python. Entusiasta de los sistemas embebidos, el teatro y la aviación.

INDICE GENERAL

INTRODUCCIÓN

1. Sobre este libro

Este libro surge como resultado de la colaboración de varios profesores con 15 años de docencia en electrónica, más específicamente en el área de la electrónica digital.

Su origen es a partir de la recopilación de las clases teóricas, los trabajos de laboratorio y distintos ejemplos utilizados en la enseñanza de las técnicas digitales.

Queremos que este libro sirva como bibliografía básica para el estudio de la electrónica digital y que permita el prototipado rápido y la implementación de distintos tipos de circuitos.

El propósito es sentar las bases del conocimiento necesario para futuros técnicos y técnicas, pero también tender un puente para quien comienza a cursar las materias básicas de la carrera de ingeniería electrónica y para quienes les interesa la temática aquí desarrollada. También esperamos que sirva de consulta para docentes de disciplinas técnicas y que lo pueden usar para orientar y armar sus clases.

El principal objetivo del libro es explicar el funcionamiento de los circuitos secuenciales y dominar el conjunto de aplicaciones más comunes que constituyen las partes fundamentales de los sistemas digitales complejos.

En cada capítulo se incluyen propuestas de práctica de laboratorio que son fáciles de implementar y se pueden realizar con los instrumentos que están disponibles en los laboratorios de electrónica de Escuelas Técnicas, Universidades y Centros de Formación, así como también con los componentes electrónicos comerciales más comunes que actualmente se pueden conseguir.

Asimismo, dichas propuestas se pueden implementar como simulaciones computacionales con cualquier software didáctico de código abierto o comercial, aunque recomendamos fuertemente el uso de software de código abierto.

Parte de este material fue recopilado originalmente por Roberto Rivas, el asistente del pañol de electrónica de la Escuela Técnica N° 2 “Paula Albarracín de Sarmiento” de la provincia de Buenos Aires, conocida popularmente como “El Chaparral”. Él tomó los apuntes de nuestras clases de años, y las de Daniel Bertolotto para empezar el proceso de digitalización de dichos apuntes. A ellos infinitas gracias por dar el puntapié inicial.

El resto del libro se fue armando a partir de las contribuciones de nuestras clases, la revisión de nueva bibliografía y la elección de ejemplos claros. Los autores contribuimos de manera equitativa en la escritura, revisión del manuscrito, y principalmente en el desarrollo de las prácticas.

Queremos agradecer a quienes colaboraron en la realización de este libro que terminó de cristalizarse en un contexto muy particular del mundo, en plena pandemia del Covid-19.

Pretendemos que sea una herramienta en español que acerque los contenidos a los y las estudiantes y que siente las bases para trabajar en el laboratorio cuando todo mejore.

Por esto, quisiéramos dedicar este texto a quienes fueron, son y serán **nuestros alumnos y nuestras alumnas de electrónica del Chaparral**.

Este libro es para ustedes en especial, pero también lo es para los y las estudiantes de electrónica del país y de habla hispana en general.

Nuestro deseo es que los circuitos que armen en las prácticas, los alumnos y las alumnas, funcionen bien la segunda vez y no la primera, porque un buen técnico/a o ingeniero/a debe prepararse no solo para el diseño de circuitos, sino para la solución de los problemas cuando estos no funcionan.

2. ¿Para qué sirven los circuitos secuenciales y los Flip Flops?

Un tour imaginario a través de un edificio

Recorramos con nuestra imaginación un edificio, puede ser donde vivimos o cualquier otro.

¿Por qué este recorrido imaginario? ¿Qué relación tiene un edificio con los circuitos secuenciales o con la electrónica en general?

En el recorrido ficticio observemos a nuestro alrededor.

La persona que nos guía nos invita a pasar. Cuando cruzamos el umbral de la puerta, lo primero que vemos es que la apertura se realiza con una llave magnética. La cerradura está conectada a un circuito que habilita la puerta cuando sensa el código de la llave. Allí nos encontramos con el primer sistema digital. Tendremos dos estados posibles “abierto” y “cerrado” para la puerta y los códigos que habilitan la puerta deberán estar guardados o memorizados en el circuito de control. Algunas puertas poseen una alarma que indica si la puerta está abierta demasiado tiempo, ¡otro sistema digital!

El almacenamiento de los códigos habilitados y la posibilidad de agregar nuevos códigos se relaciona con los circuitos secuenciales como veremos en los **capítulos 1 y 2**, cuando nos ocupemos de los Flip Flop y de los registros.

En nuestro recorrido, no reparamos antes en la iluminación del pasillo en el que estamos hasta que, de pronto las luces se apagan. La mayoría de los edificios tienen sistemas automáticos para el control de la iluminación por tiempo. También estos dispositivos pueden estar hechos con sensores de movimiento, de modo que cuando permanecemos quietos se apagan las luces. Otra forma es con un pulsador y algún biestable junto con un contador para brindar algunos segundos de iluminación. Exploraremos en los **capítulos 3, 4 y 5**, circuitos que nos permitan pensar estos diseños.

Para tener conocimientos sobre los componentes físicos que usaremos en los circuitos que mencionamos, contamos con el **capítulo 6**, mientras que para comprender lo necesario para el manejo de señales digitales tendremos el **capítulo 7**.

Seguimos con nuestro recorrido y llegamos a la puerta del ascensor. El ascensor puede parecer un dispositivo simple, sin embargo, involucra una gran cantidad de circuitos y secuencias para su funcionamiento. El diseño del control de un ascensor se puede realizar a través de la programación de microcontroladores o sistemas programables como los que introduciremos en el **capítulo 8**.

Si continuamos prestando atención a nuestro alrededor podemos observar que, también existen otros sistemas para controlar: la caldera del edificio, para controlar la temperatura del agua, y el aire acondicionado (si el edificio tuviera un sistema centralizado), y los sistemas de seguridad, en algunos casos. En cada capítulo presentaremos conocimientos que nos permitirán diseñar estos y otros sistemas que están a nuestro alrededor.

Dejemos por ahora nuestro tour para continuar con la presentación de los saberes que contiene el presente libro.

En el capítulo 2 incursionaremos en los Flip Flops o biestables. Estos circuitos juegan un papel fundamental al diseñar elementos de memoria y circuitos sincrónicos. Los Flip Flops son los elementos básicos de la electrónica digital secuencial que nos permiten construir los principales circuitos que componen






Dispositivos electrónicos

los dispositivos electrónicos que nos rodean. Nos referimos a dispositivos que van desde un simple reloj hasta un sistema de cómputo cualquiera (celulares, computadoras, etc.).

Por un lado, la demanda de dispositivos eficientes y más rápidos es la principal preocupación en el área relacionada con estos dispositivos. Por otro lado, estos circuitos también son utilizados en algunos sistemas de control.

Los usos de los Flip Flops, en general incluyen sincronizar (generadores de señales de reloj), contar pulsos, memorizar secuencias y muchos otros. La lectura de las páginas de este libro nos permitirá comprender los fundamentos que, luego nos guiarán al diseño de todas las distintas aplicaciones.

Comenzamos con las definiciones básicas para posteriormente analizar en profundidad los Flip-Flops.

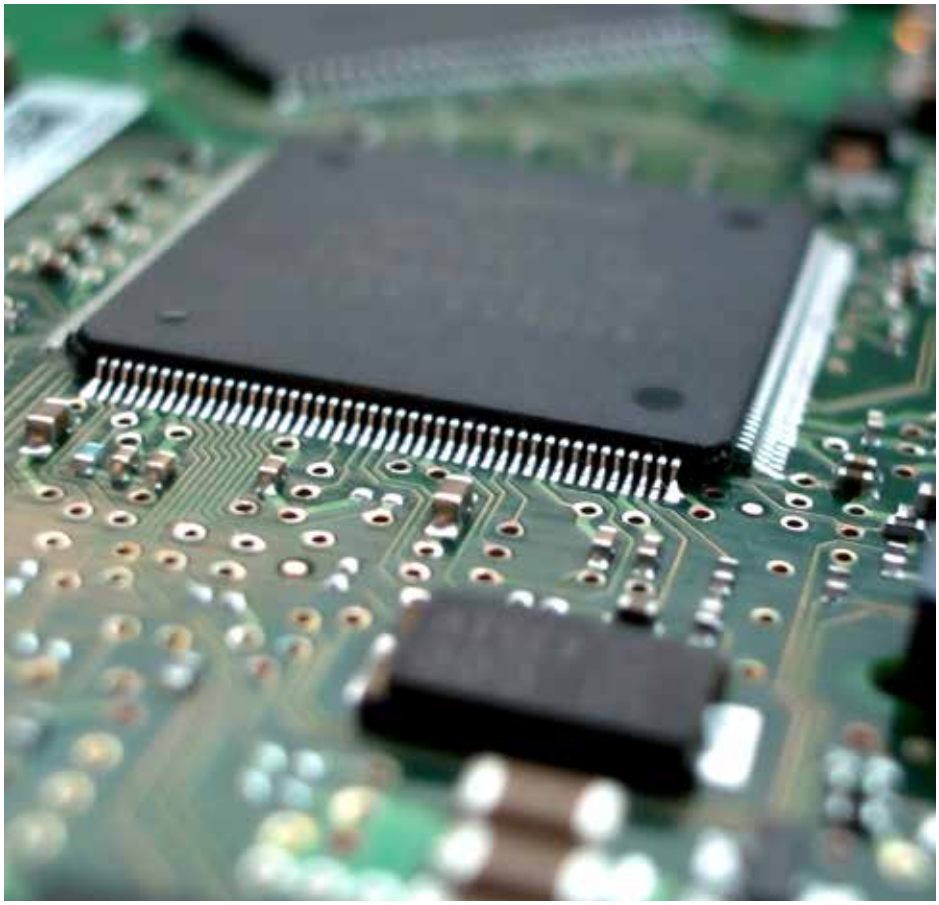
Los capítulos posteriores los dedicaremos a explorar cada tipo de circuito básico que podemos construir con ellos.

A partir de la comprensión de las partes de los sistemas descriptos en los distintos capítulos y el proceso de digitalización de señales, el lector será capaz de aprovechar de manera integral las prestaciones que ofrecen los sistemas programables como los descritos en el capítulo 8.

Con un sistema programable podemos construir cualquier aplicación que necesitemos. Algunos ejemplos serán mostrados a lo largo de los capítulos, pero sus usos son tan vastos como las necesidades que surjan y la imaginación para resolverlas.

Al final de cada capítulo se plantean actividades tales como: trabajos prácticos de laboratorio y ejercicios con el propósito que puedan adquirir el manejo de todos los dispositivos que podrán usar en sus proyectos.

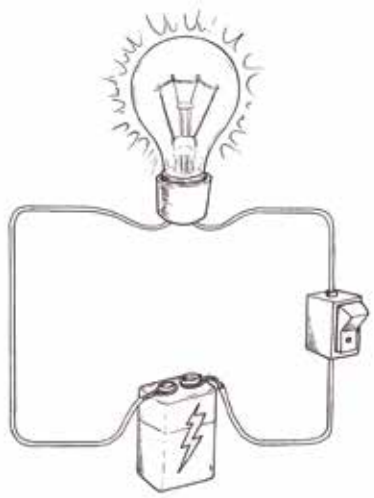
Capítulo 01
NOCIONES PARA COMENZAR



1.1. Señales y circuitos digitales

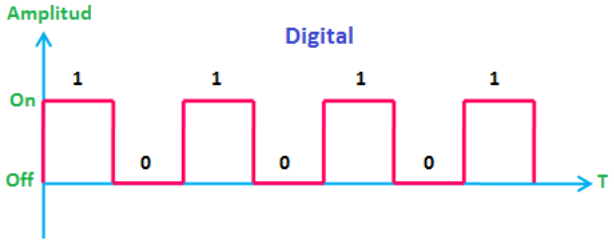
Comenzamos por una definición importante, ya que todo el libro tratará del procesamiento de señales digitales. Se define como señal digital aquella cuyas magnitudes se representan mediante valores discretos que varían entre dos estados. Por ejemplo, supongamos que tenemos un circuito, como el que se muestra en la Figura 1.1, con una lámpara, una batería y un interruptor o llave. El interruptor de luz solo puede tomar dos valores o estados: abierto o cerrado, lo cual indica que la lámpara está encendida o apagada.

Figura 1.1 *Circuito con llave y lámpara*



Los circuitos digitales son aquellos que permiten procesar las señales digitales. Los sistemas digitales, como por ejemplo la computadora, usan lógica de dos estados: la corriente circula o no por los componentes electrónicos del sistema. Estos dos estados son en realidad dos niveles de tensión, alto, H (High) y otro bajo, L (Low). Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria o booleana. Si el nivel alto se representa por 1 y el bajo por 0, como en la Figura 1.2, se habla de lógica positiva y en caso contrario de lógica negativa.

Figura 1.2 *Señal digital*



Un sistema digital es cualquier dispositivo que esté destinado a la generación, transmisión, procesamiento o almacenamiento de señales digitales. Para el análisis y la síntesis de sistemas digitales binarios se utiliza como herramienta el álgebra de Boole. Por lo que recomendamos tener nociones básicas para poder manejar los contenidos del presente texto.

A grandes rasgos los sistemas digitales pueden clasificarse como:

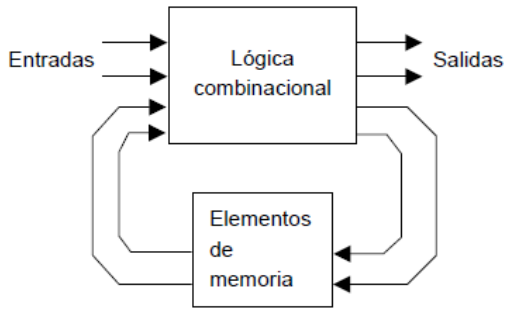
Sistemas digitales combinacionales

En estos circuitos las salidas solo dependen del estado de sus entradas en un momento dado. Por lo tanto, no necesitan módulos de memoria, ya que las salidas no dependen de los estados previos de las entradas.

Sistemas digitales secuenciales

Sus salidas dependen del estado de sus entradas en estados previos. Esta clase de sistemas necesitan elementos de memoria que recojan la información de la historia pasada del sistema.

Figura 1.3 *Esquema general de un sistema secuencial*



Algunos ejemplos de circuitos combinacionales son los circuitos codificadores, decodificadores y los multiplexores. Ejemplos de circuitos secuenciales veremos ampliamente a lo largo de cada capítulo de este libro.

En la Figura 1.3 podemos ver un esquema para comparar dichos sistemas.

Para la implementación de los circuitos digitales, se utilizan compuertas lógicas (AND, OR y NOT). Estas compuertas siguen el comportamiento las tres funciones básicas del álgebra de Boole (ver Apéndice A).

El **transistor** es el dispositivo electrónico que dio lugar a los circuitos integrados y demás elementos que aparecen a lo largo de este libro. Surgió en los Laboratorios Bell de la AT&T. Posteriormente se utilizó como reemplazo de la válvula (o tubo) de vacío, originando una revolución en el campo de la electrónica y en especial de la electrónica digital. La patente del transistor de juntura (o unión), es de 1951, le fue concedida a William Shockley.

1.2. Unidades de información

El procesamiento de las señales digitales de alto nivel de abstracción es realizado por una computadora. Existen varios niveles de complejidad y abstracción entre el procesamiento de señales digitales a partir de compuertas y el procesamiento de alto nivel que realiza una computadora.

Pero dado que hoy en día muchos dispositivos son en definitiva computadoras (teléfonos celulares, tablets), definimos por computadora al dispositivo electrónico digital programable que se encarga de procesar y/o manipular información digital.

Las unidades de información procesadas por sistemas digitales, desde los más simples a los más complejos son:

- **el bit**: es la unidad básica de información. Existen dos estados (1: on, encendido; 0: off, apagado),
- **el byte**: es el conjunto de 8 bits que pueden ser manipulados en conjunto (IBM 1964),
- **la palabra** (o word en inglés): es el número de bits adyacentes que pueden ser manipulados simultáneamente por una determinada arquitectura (8, 16, 32, 64).

1.3. Algunas definiciones importantes

En el estudio de los sistemas digitales es importante definir los siguientes conceptos que aparecerán a lo largo de éste y otros textos sobre el tema.

Dispositivo Programable: es aquel dispositivo que puede actuar sin intervención humana, según una secuencia de instrucciones.

Procesamiento digital: puede referirse al aritmético o lógico a través de un circuito.

Manipulación: puede tratarse de almacenamiento o entrada/salida de datos digitales.

Por otro la secuencia o conjunto de instrucciones se la denomina **programa**.

Recordando la definición de algoritmo (conjunto de reglas que permite resolver un problema en un número finito de pasos), podemos decir que un programa es un algoritmo, aunque no necesariamente sea verdadera la afirmación inversa. Cuando un algoritmo puede implementarse como un programa de computadora decimos que es factible o que el problema es computable.

En este punto se puede hacer una clasificación de los sistemas programables del siguiente modo. Llamamos computadora reprogramable a aquella tal que se le puede cambiar el programa para que resuelva diferentes problemas (por ejemplo, esta definición abarca una PC o notebook ya que se puede transformar en un procesador de textos o en una calculadora, según el programa que se utilice, o en un sistema completo de edición de video).

Por otro lado, existen los **Embedded System** (sistemas dedicados o embebidos): el programa está fijo (por ejemplo, un horno a microondas).

Al conjunto de dispositivos electrónicos que integra una computadora se los denomina **hardware**. A los programas que se ejecutan en la computadora se los denomina **software**.

El estudio de estos sistemas programables se divide en **Organización** y **Arquitectura de las Computadoras**.

La **organización de computadoras** estudia los detalles constructivos de la computadora (cómo funciona), esto está directamente relacionada con el hardware.

Por otro lado, la **arquitectura de computadoras** estudia la estructura y el comportamiento de la computadora, del modo que la ve un programador. En este caso está directamente relacionado con el software. Ambas disciplinas están muy vinculadas entre sí y el límite entre ambas a veces es bastante difuso.

Sin importar su tamaño las computadoras realizan sus tareas según tres pasos fundamentales:

- aceptar una entrada (input),
- procesar según un programa,
- producir una salida (output).

En particular para poder realizar todas estas tareas de manera adecuada, con la perspectiva de un técnico o de un ingeniero, es necesario comprender las

distintas partes del sistema. Nos enfocaremos en uno de los componentes clave para poder construir estos sistemas: los **Circuitos Secuenciales**.

Utilizaremos además dichos circuitos para producir aplicaciones y circuitos a lo largo del libro que nos servirán de base y de punto de partida para una mejor comprensión más adelante cuando nos dediquemos a comprender los sistemas programables descriptos en el Capítulo 8.

Los **Circuitos Secuenciales** se caracterizan por que el estado de la salida depende de la combinación aplicada a las entradas y además del estado previo de la propia salida. Dicho estado está memorizado en el circuito. Esto implica que se requiere memoria y que una misma combinación de entradas usada en distinto momento, puede originar un resultado distinto a la salida.

1.4. Sobre la electrónica y el "Espanglish"

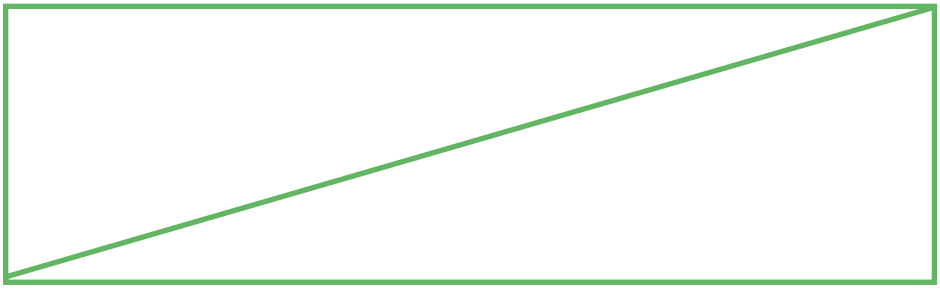
Como sucede en distintas disciplinas existe un lenguaje y una jerga en particular utilizada en electrónica. Muchos de los términos y palabras que usamos están en inglés. Hemos realizado un esfuerzo por traducir ciertos conceptos lo mejor posible, sin embargo, de la práctica cotidiana del lenguaje surgen términos que permanecen y se usan en inglés, tal como *Flip Flop*, *SET*, *enable* u otros términos.

Por otro lado, los manuales de dispositivos en general están disponibles en inglés, por lo cual alentamos a que se sientan cómodos utilizando los términos que encontrarán cuando busquen las características de los dispositivos.

De todas formas, cada vez que introduzcamos alguno de estos conceptos serán explicados y en determinados casos traduciremos al español algunas palabras sajonas, en otros casos usaremos las palabras en inglés. Por ejemplo, el uso de la palabra "googlear" como verbo es una práctica muy difundida. A veces se utiliza ese mismo mecanismo con algunas palabras referidas a los dispositivos.

Capítulo 2
FLIP FLOPS

Lo que cambió ayer
Tendrá que cambiar mañana
Así como cambio yo
En esta tierra lejana
(Julio Numhauser)

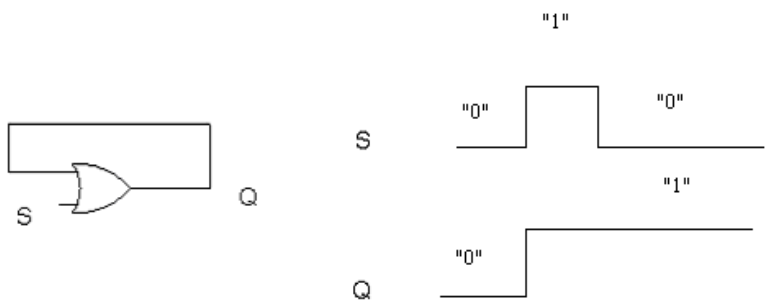


Se define como **Flip Flop** al circuito secuencial más simple que podemos construir que conmuta o cambia entre dos estados. Constituye una celda básica de memoria capaz de guardar 1 bit de información.

El Flip Flop es un circuito lógico biestable, es decir posee dos estados estables, denominados SET (1 o activación) y RESET (0 o desactivación), en los cuales se puede mantener indefinidamente. Esto es lo que permite el almacenamiento de un bit. Los Flip Flop se implementan con compuertas lógicas y son los bloques básicos de construcción de contadores, registros y otros circuitos de control secuencial que veremos a lo largo de los siguientes capítulos. También se emplean en ciertos tipos de memorias.

La forma más simple de hacer una memoria de un bit es utilizando una compuerta OR realimentada como se observa en la a la izquierda de la Figura 2.1. El estado alto, o “1”, aparece en la salida, gracias a la realimentación. Dicho valor se autosostiene, aunque desaparezca el estado “1” que originalmente fuera aplicado a la entrada llamada S (SET), tal y como se ve en el diagrama temporal a la derecha de la Figura. Sin embargo, el circuito de la Figura 2.1 presenta el siguiente problema: una vez grabado no puede borrarse.

Figura 2.1 *Compuerta OR realimentada*



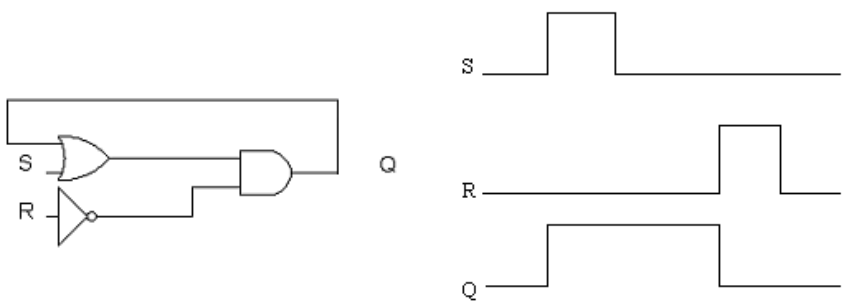
Al establecer un nivel alto “1” en la entrada S, la salida pasa a estar en estado alto “1”. Si el valor de la entrada se establece ahora en “0”, la salida continua aun en estado alto indefinidamente.

2.1. Flip Flop tipo SR

Para poder borrar el estado alto obtenido en el ejemplo anterior (Fig. 2.1), podríamos hacer una modificación como la que se muestra en la Figura 2.2, donde agregamos una compuerta AND y una entrada extra que llamamos RESET. Dicha entrada llega a la compuerta AND a través de un inversor. Esa entrada extra permite borrar el nivel alto sostenido por la compuerta OR al enviar un nivel alto en la entrada de borrado llamada RESET.

Ahora, el circuito de la Figura 2.2 cumple con los requisitos de poder grabar y borrar la información, ya que tenemos un modo de almacenar un bit, pero también una forma de borrarlo.

Figura 2.2 Circuito con compuertas para un Flip Flop SR con su correspondiente diagrama de Nivel



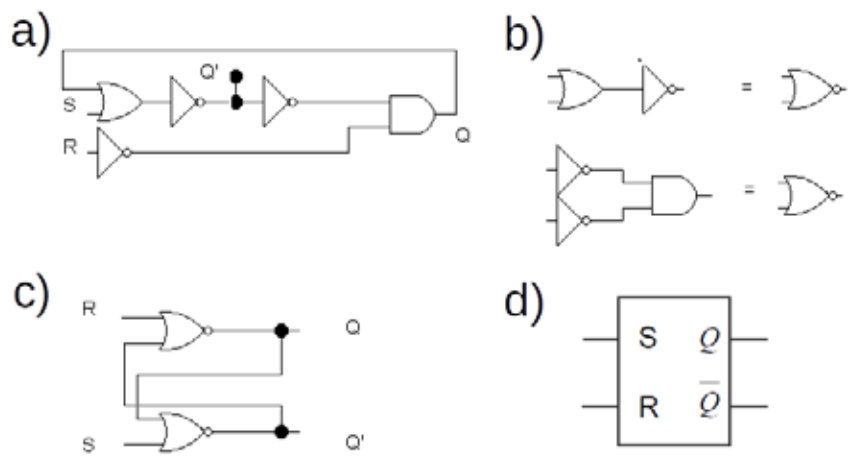
Podemos avanzar un paso más buscando otro circuito equivalente que utilice compuertas de un solo tipo. Este último objetivo es deseable, sobre todo a nivel implementación (cuando lo queremos armar). Este paso es necesario por el siguiente motivo: cada tipo distinto de compuerta que se agregue en un circuito sería un nuevo circuito integrado que hay que comprar y colocar en el circuito. Tener un único tipo de compuerta simplifica el armado del circuito, muchas veces reduciendo el número de integrados y compuertas necesarias para el armado.

Para poder conseguirlo generalmente se aplica la técnica del álgebra de Boole normalmente conocida como reducción por compuertas NAND o NOR. Este procedimiento se observa en la Figura 2.3.

A este circuito obtenido se lo llama Flip Flop con NOR o también biestable SR. El nombre de la entrada S viene de la palabra SET, o grabar en memoria. El de la entrada R viene de RESET, retornar a “0” o borrar la memoria. El punto llamado Q’ sobre los circuitos a) y c) de la Figura 2.3, se caracteriza por presentar siempre el estado opuesto al de Q salvo que se haga un uso indebido aplicando simultáneamente orden de grabar y borrar. Esta situación debe ser evitada y se la llama estado prohibido. En la Tabla 2.1 lo representaremos con una X.

La tabla de verdad podemos obtenerla con la ayuda de los diagramas de tiempo, como el que se muestra en la Figura 2.4, e inspeccionando las distintas configuraciones de entrada sobre el circuito de los paneles.

Figura 2.3 a) Circuito para compuertas para un Flip Flop SR. b) Equivalencia entre compuertas a partir del álgebra de Boole. c) Reducción por compuertas NOR. d) Símbolo para el Flip Flop SR.



En el circuito a) de la Figura 2.3, se ve gráficamente aplicado el razonamiento para transformar el circuito (agregar de manera teórica dos inversores no cambia la función implementada).

En la Tabla 2.1, Qn y Qn+1 representan siempre a la misma salida Q de los circuitos a), c) y d) en la Figura 2.3. Respectivamente representan el estado actual (n) y en el estado futuro (n+1) que dicha salida adopta.

Para obtener la tabla reducida se debe observar la Tabla de verdad, y resumir los estados en términos de Q, de modo que quede la Tabla 2.2. La ecuación característica se obtiene aplicando la técnica de mapas de Karnaugh (ver Apéndice B).

$$Qn + 1 = \overline{S} + R \cdot Qn \quad \textbf{(2.1)}$$

Tabla 2.1 *Tabla de verdad completa que surge del análisis del Flip Flop SR*

S	R	Qn	Qn+1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Figura 2.4 Diagrama temporal para los estados de entrada de Flip Flop SR

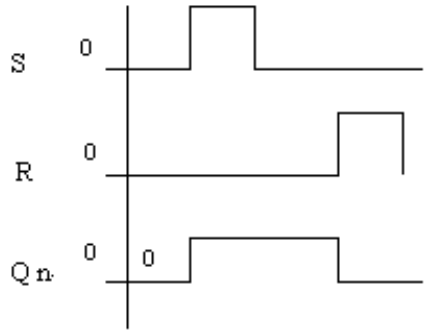


Tabla 2.2 *Tabla de verdad resumida del Flip Flop SR*

S	R	Qn+1	Significado
0	0	Qn	Permanece igual
0	1	0	borra o “Resetea”
1	0	1	Graba un bit o “Setea”
1	1	X	Estado prohibido

El primer flip-flop electrónico fue inventado en 1918 por los físicos británicos William Eccles y F. W. Jordan. Inicialmente se llamó circuito de disparo Eccles-Jordan y consistía en dos elementos activos (tubos de vacío). Este circuito se usó en la computadora británica de descifrado de códigos Colossus de 1943. Tales circuitos y sus versiones transistorizadas eran comunes en las computadoras incluso después de la introducción de circuitos integrados, aunque los flip-flop hechos de compuertas lógicas también son comunes y se utilizan ahora.

2.2. Flip Flops sincrónicos y asincrónicos

En los llamados Flip Flops asincrónicos la salida cambia de estado en forma inmediata al aplicar las señales de comando en las entradas S o R. Funcionan como hemos descrito antes.

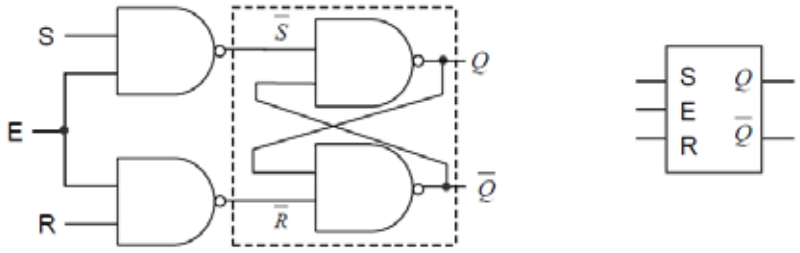
En los Flip Flops sincrónicos, al aplicar señales a S o R la salida no cambia inmediatamente, sino que se quedan esperando la llegada de una señal de habilitación llamada clock o reloj. Recién cuando la señal de clock lo habilita, se produce el cambio. S y R se convierten en entradas preparatorias en este caso, ya que permiten preparar al Flip Flop para el cambio que ocurrirá cuando llegue el nivel alto de la señal de clock. Este funcionamiento es muy útil en circuitos que combinan Flip Flops, ya que permite sincronizar la conmutación de todos en el mismo momento. La señal de clock es normalmente una señal tipo rectangular y se obtiene de un generador de señal, adecuado. Más adelante en el Capítulo 5 se describe un método posible para obtener funciones o señales del tipo reloj.

2.2.1. Flip Flop sincrónico por nivel

Generalmente resulta de utilidad poder controlar el funcionamiento del Flip Flop o latch de manera que las entradas se puedan activar en unos instantes determinados. El diagrama y el símbolo lógico con entrada de habilitación se muestra en la Figura 2.5. Se controla el estado al que va a cambiar el latch cuando se aplica un “1” en la entrada de habilitación (E, enable). También a esta entrada la llamaremos a veces “clock o ck”. El latch no cambiará de estado hasta que la entrada E esté en nivel alto. Esta tercera entrada (E) permite habilitar o inhibir las acciones del resto de entradas.

En estos circuitos, como el que se muestra a la izquierda en la Figura 2.5, la salida se puede conmutar cuando el clock está en nivel alto. En este caso, se obtiene adaptando al SR con NAND, partiendo del circuito de la Figura 2.3, que habíamos construido en la sección anterior.

Figura 2.5 *Circuito y símbolo para Flip Flop SR sincrónico por nivel*



Cuando la señal de habilitación E esté a nivel bajo, las señales $\overline{S}$ y $\overline{R}$ estarán a nivel alto sin importar el valor de las entradas R y S. Esto coloca al latch en su modo de memoria, evitando que la salida cambie de estado. Cuando se activa la entrada de habilitación, las señales R y S se invierten y se aplican al circuito indicado por la línea punteada. El circuito actúa como un latch SR con entrada a nivel alto.

S y R solo deben ser activados en el momento en que el clock está en estado bajo para evitar fallos en la sincronización. El ruido que puede aparecer en S o en R provoca conmutaciones erróneas si coincide con el estado alto de clock. Este tipo de comportamiento de los circuitos motiva una nueva modificación para el circuito que hemos visto, que veremos a continuación.

2.2.2. Flip Flop sincrónico por flanco

Se llama flanco de una señal al instante en que esta conmuta de nivel “0” a nivel “1” (flanco de subida) o al momento en que hace la conmutación opuesta (flanco de bajada) como se observa en la Figura 2.6.

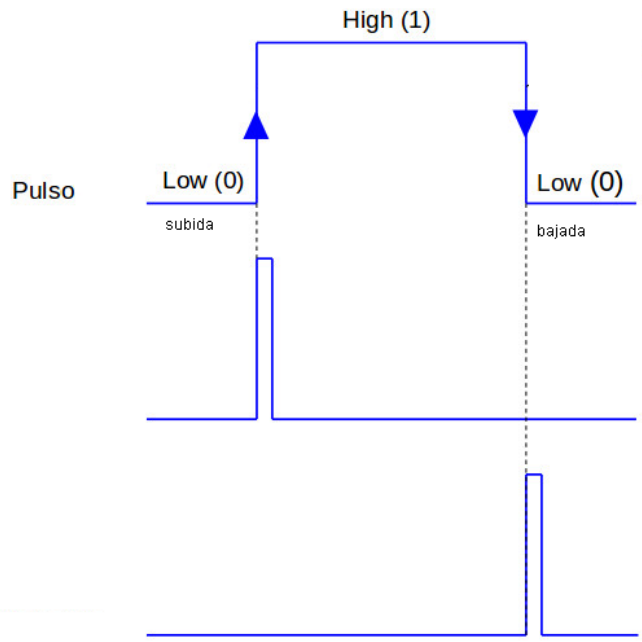
En los paneles inferiores se identifican dichos flancos con pulsos más cortos.

El Flip Flop sincrónico por flanco sólo podrá conmutar su salida en el momento en el que ocurren los flancos de la señal de clock y no cuando el clock está en nivel alto, como el que estudiamos antes.

Este funcionamiento nos permitirá activar en cualquier momento las entradas preparatorias S o R sin riesgo de ocasionar conmutaciones fuera de sincronismo. Además, resulta una mayor inmunidad al ruido ya que este solo ocasionará errores si se produce en el instante del flanco del clock.

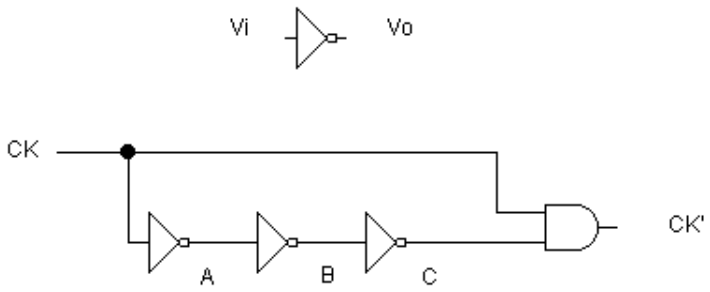
Se pueden hacer Flip Flop por flanco de subida o por flanco de bajada y sus símbolos son generalmente una flecha hacia arriba y otra hacia abajo respectivamente dibujadas sobre el borde del nivel o también los símbolos | > o bien |◦ > colocados en la entrada de clock sobre el circuito.

Figura 2.6 *Flancos de subida y bajada de un pulso digital*



Para obtener un Flip Flop sincrónico por flancos se agrega el circuito de la Figura 2.7 en la entrada del clock del SR sincrónico por nivel de la Figura 2.5.

Figura 2.7 *Modificación para obtener un Flip Flop sincrónico por flanco*



La acumulación de tiempos de demora producida por las compuertas NOT permite obtener breves instantes de tiempo en los que la señal coincide para dar a la salida un nivel alto.

De este modo, el flanco ascendente de la señal sucede como resultado del breve instante de tiempo en el que ambas entradas se encuentran en nivel alto, como consecuencia de la acumulación de tiempos de demora. Esto produce a la salida un intervalo muy corto en el que dará como resultado "1". Así en su salida se obtiene un pulso muy corto que a los fines prácticos se puede considerar que coincide con el flanco del clock. Este pulso breve es el que se usa para habilitar la conmutación del Flip Flop. Queda el flanco ascendente entonces indicado como en la Figura 2.8.

Para obtener un circuito activado por flancos de bajada se pondrá una NOR en vez de AND. Para dichos circuitos, el símbolo utilizado será como el que se muestra de la Figura 2.9. En la Figura 2.10 se muestra un ejemplo de las formas de onda en las entradas y en la salida de un Flip Flop disparado por flanco ascendente, obtenidas a partir de la Tabla 2.3.

Figura 2.8 *Representación del flanco ascendente de una señal rectangular que se utiliza como señal de clock*

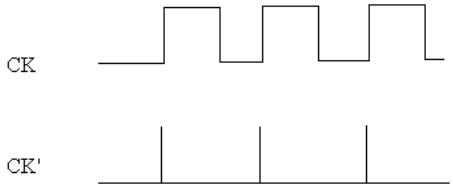


Figura 2.9 *Símbolo del Flip Flop sincrónico por flanco ascendente (a) y descendente (b)*

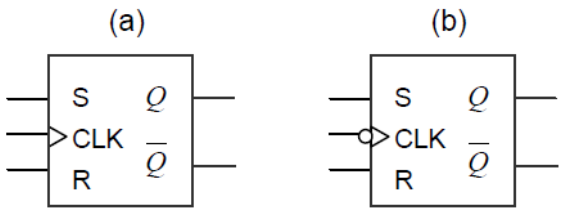
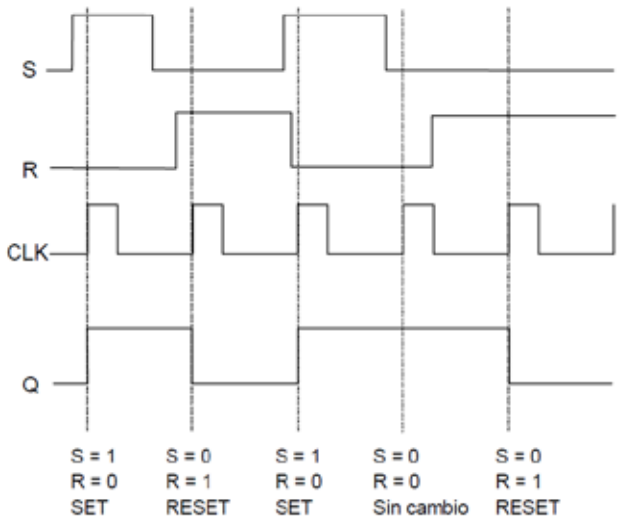


Tabla 2.3 *Tabla de verdad del Flip Flop SR sincrónico por Flanco ascendente*

S	R	CLK	n	Qn	Q
0	0	↑	n – 1	Qn–1	Q
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	0	0	Ambiguo
X	X	X	n – 1	Qn–1	Q

Figura 2.10 *Ejemplo de formas de onda en las entradas y en la salida de un Flip Flop SR disparado por flanco ascendente*



2.2.3. Flip Flop tipo D

Se obtiene partiendo de un circuito SR sincrónico por flancos o nivel agregándole un inversor en la entrada R. Luego unimos las entradas resultantes en lo que da origen a una única entrada preparatoria D. El circuito para hacer este Flip Flop se observa en la Figura 2.11. Por simple inspección del circuito podemos construir la tabla de verdad asociada.

Tabla 2.4 *Tabla de verdad del análisis del Flip Flop D*

D	Qn+1
0	0
1	1

Por simple inspección también se puede observar que la ecuación característica de este Flip Flop es:

$$Qn + 1 = D \quad \textbf{(2.2)}$$

Figura 2.11 *Circuito para el Flip Flop tipo D activado por nivel y su respectivo símbolo*

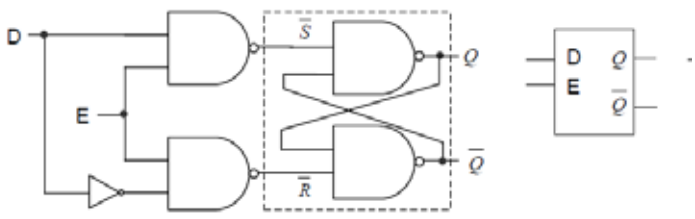
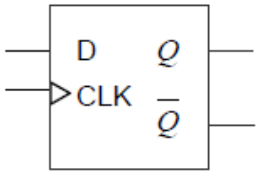


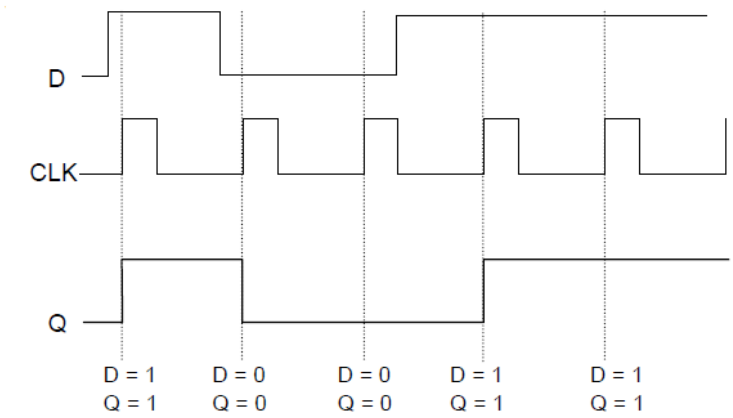
Figura 2.12 *Símbolo del Flip Flop D activado por flanco*



En el caso de que sea activado por flanco, el símbolo cambia levemente por el mostrado en la Figura 2.12. Veamos en la Figura 2.13 un diagrama temporal con un ejemplo de activación de la entrada.

Este Flip Flop lo utilizaremos a lo largo de los capítulos 3 y 4 para diseñar, como aplicaciones, registros y contadores.

Figura 2.13 *Ejemplo de formas de onda en las entradas y en la salida de un Flip Flop D disparado por flanco ascendente*



Por ahora veremos una aplicación simple que luego será clave para entender los diseños de dichos circuitos. Analicemos el funcionamiento del circuito de la Figura 2.14. Al estar realimentada la entrada con la salida invertida, cada vez que la señal de clock habilite el cambio de estado, la salida copiará el estado que aparezca en la entrada. De modo que, si inicialmente el estado era bajo, el estado de la salida invertida o negada será "1", y la salida intentará copiar dicho valor. Como la activación es por flanco ascendente, el estado de la salida solo cambiará recién cuando ocurra dicho flanco. Así la salida presentará una señal rectangular cuya frecuencia vale la mitad de la señal de clock.

Es decir que:

$$F_Q = \frac{F_{ck}}{2} \quad \textbf{(2.3)}$$

Donde FQ es la frecuencia de la señal de salida y Fck la frecuencia de la señal de clock. En términos del periodo la ecuación se escribe como:

$$T_Q = 2T_{ck} \quad \textbf{(2.4)}$$

Al circuito de la Figura 2.14 se lo conoce como "divisor de frecuencia". La señal de salida de este circuito comparada con la señal de clock puede observarse en la Figura 2.15.

Figura 2.14 Divisor de frecuencia a partir del Flip Flop tipo D

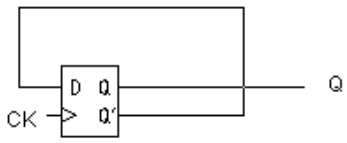
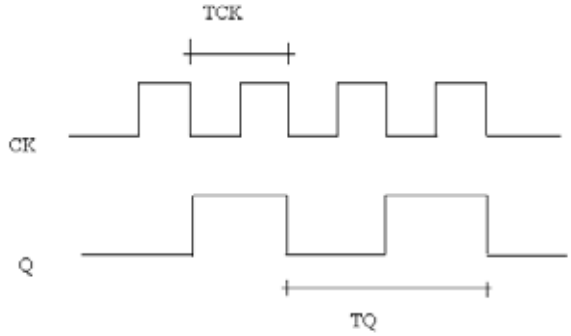


Figura 2.15 Diagrama de nivel para el circuito de la Figura 2.14



2.2.4. Flip Flop Tipo JK

El Flip Flop JK se comporta como el SR a excepción de que resuelve el problema de tener una salida indeterminada cuando las entradas se encuentran activas a la vez. La entrada J es el equivalente a la entrada S de un Flip Flop SR y la entrada K, el equivalente a la entrada R. En este dispositivo cuando las dos entradas se colocan a nivel alto la salida cambia al estado opuesto al que se encontraba. A este modo de funcionamiento se le traduce a veces como modo de basculación.

Con el agregado de dos compuertas AND en las entradas de un Flip Flop SR sincrónico por flancos, se evita el estado prohibido que se produce al querer dar las señales de grabar y de borrar simultáneamente. J y K cumplen funciones equivalentes a las de S y R respectivamente, excepto para el estado prohibido. El circuito para obtener el Flip Flop JK se puede observar en la Figura 2.16 y la tabla de verdad corresponde a la Tabla 2.5.

A partir de la tabla de verdad se puede obtener la ecuación que representa la función del Flip Flop utilizando el método del mapa de Karnaugh. Dicho mapa se observa en la Figura 2.17.

A partir del análisis de dicho mapa se obtiene la ecuación que representa la función del Flip Flop JK. Veamos en la Figura 2.18 un ejemplo de activación de JK y el correspondiente cambio en la salida a través del diagrama temporal.

$$Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n \quad \textbf{(2.5)}$$

Figura 2.16 *Flip Flop tipo JK y su respectivo símbolo*

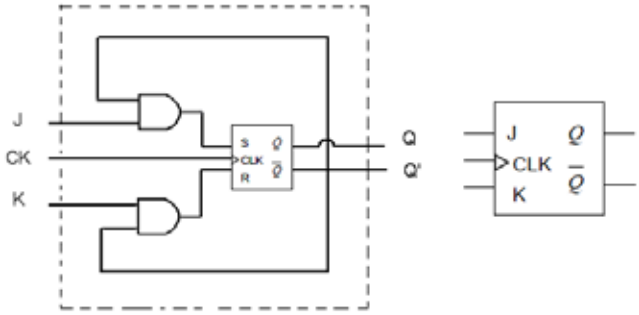


Tabla 2.5 *Tabla de verdad del Flip Flop JK*

J	K	Qn	Qn+1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figura 2.17 *Mapa de Karnaugh para el Flip Flop tipo JK*

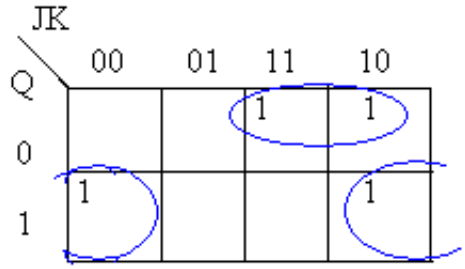
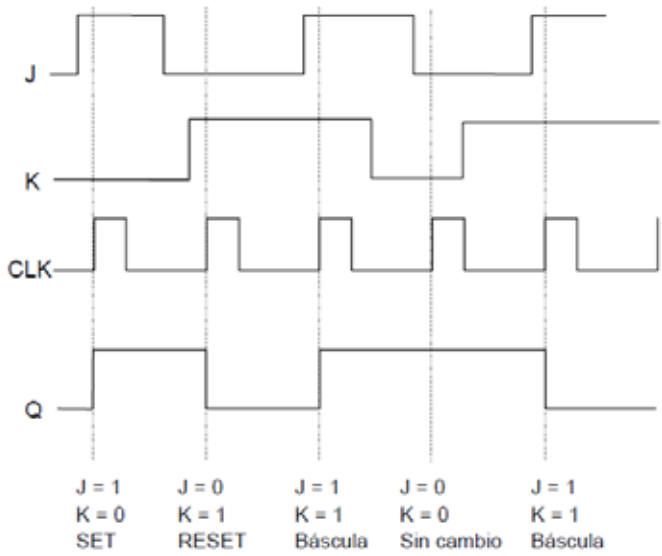


Figura 2.18 *Ejemplo de formas de onda en las entradas y en la salida de un Flip Flop JK disparado por flanco ascendente*



También podemos resumir el funcionamiento de este Flip Flop con la Tabla 2.6 resumida, para el caso de un Flip Flop JK activado por flanco ascendente.

Tabla 2.6 *Tabla de verdad del Flip Flop JK sincrónico por Flanco ascendente*

J	K	CLK	n	Qn	Q
0	0	↑	n – 1	Qn–1	Qn+1
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	0	0	Qn
X	X	X	n – 1	Qn–1	Qn

2.2.5. Flip Flop Tipo T

Este Flip Flop se diseña a partir de un JK al que se le unen sus entradas para formar una única entrada que llamaremos T, como en la Figura 2.19. Convencionalmente no existe como circuito integrado y siempre se lo construye a partir del JK. El símbolo que representa a este Flip Flop se muestra en la Figura 2.20.

Si en la ecuación 2.5 hacemos J = K, las únicas opciones posibles son que ambas entradas tomen el mismo valor. O bien ambas valen cero o bien ambas valen 1. Este Flip Flop presenta el siguiente comportamiento: cuando el valor de la entrada T es bajo (T = 0), la salida permanece en el mismo estado en el que se encontraba en un tiempo anterior. Si la entrada del Flip Flop es un nivel alto (T = 1) la salida cambiará al estado opuesto al que tenía.

Así reemplazando en la ecuación 2.5, J = K = T, se obtiene:

$$Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n \quad \textbf{(2.6)}$$

De modo que podemos resumir en una tabla como la 2.7 el funcionamiento de este Flip Flop.

Tabla 2.7 *Tabla de verdad del análisis del Flip Flop T*

T	Qn+1
0	Qn
1	$\overline{Qn}$

Figura 2.19 *Flip Flop tipo T partiendo del Flip Flop JK*

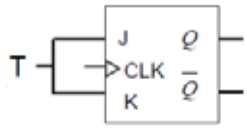
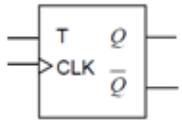


Figura 2.20 *Símbolo para el Flip Flop T*

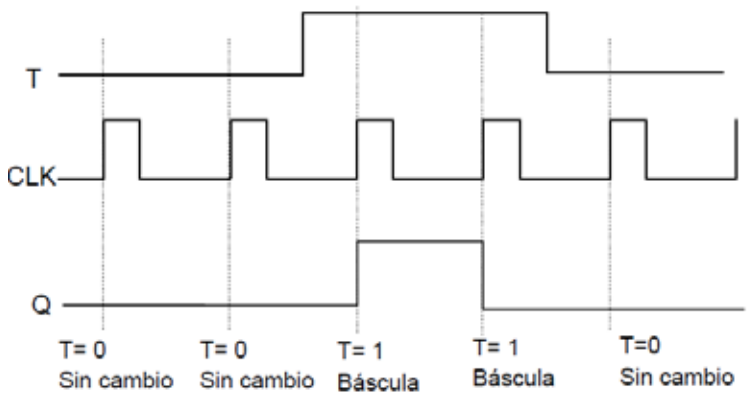


Del mismo modo que se hizo para el Flip Flop tipo D, se puede implementar con el T un circuito divisor de frecuencia. En el caso del T, para lograrlo es necesario conectar la entrada del Flip Flop a un nivel alto constantemente, de modo que se vea forzado siempre a cambiar de estado.

En la Figura 2.21, se puede observar el comportamiento para una determinada serie temporal en la entrada T.

Con estos cuatro Flip Flop que hemos descripto, sus tablas y ecuaciones tenemos el set funciones que utilizaremos a lo largo del libro y estamos en condiciones para diseñar los distintos circuitos que son fundamentales para el desarrollo de cualquier sistema digital secuencial. En la siguiente sección describiremos un set de entradas de control extra que pueden presentar cualquiera de los Flip Flop que hemos analizado.

Figura 2.21 *Comportamiento del Flip Flop T*



2.2.6. Entradas de fuerza de los Flip Flop

Las entradas de los diversos Flip Flops, es decir, R, S, J, K, D y T, sólo tienen efecto en el momento de una transición apropiada de la señal de reloj (CLK). Por tanto, nos referimos a estas entradas de control como síncronas, pues su funcionamiento está sincronizado con la entrada de reloj.

En muchas aplicaciones resulta útil poner o forzar la salida de un Flip Flop a “0” o a “1” en cualquier momento, independientemente del reloj. Por tanto, algunos dispositivos tienen entradas adicionales para efectuar estas funciones. Estas reciben el nombre de entradas asíncronas pues no están controladas por el estado del reloj.

Entre ellas se encuentran la entrada PRESET (PRE), que cuando se active colocará la salida a nivel alto (Q=“1”) y la entrada CLEAR (CLR) que cuando se active llevará a la salida a nivel bajo (Q= “0”). Como ocurre con el resto de las entradas estas pueden ser activas a nivel alto o a nivel bajo. Las entradas PRESET y CLEAR pueden anular las otras entradas al circuito. Es necesario asegurarse de que ambas entradas asíncronas no están activas simultáneamente.

El objetivo de estas entradas es poder manejar al Flip Flop prevaleciendo en jerarquía por sobre las entradas preparatorias y el clock. Esto lo hacen en forma asincrónica. Dichas entradas pueden ser activados o controladas con estado alto o “1” o con estado bajo o “0”. Activar PRESET consiste en forzar el estado

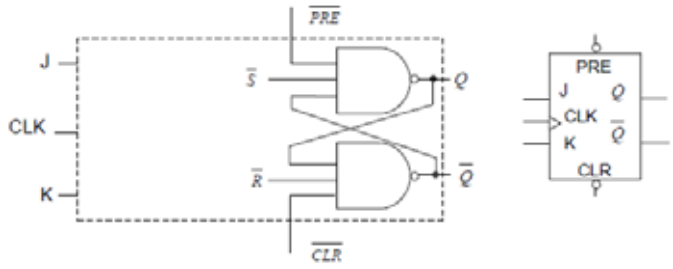
del Flip Flop para que valga “1” el nivel de la salida y al activar CLEAR forzará el estado de la salida del Flip Flop a que tome estado “0”.

Para el caso de las entradas de fuerza activadas por “1” (lado izquierdo de la Figura 2.22), la correspondiente tabla es la 2.8.

Tabla 2.8 *Entradas de fuerzas activadas por “1”*

Preset	Clear	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	x

Figura 2.22 *Entradas asíncronas o de “Fuerza” en un Flip Flop JK*



Para el caso de las entradas de fuerza activadas por “0” (lado derecho de la Figura 2.22), el funcionamiento está descripto por la tabla 2.9 .

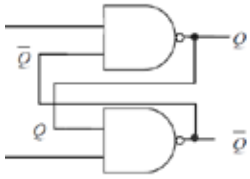
Tabla 2.9 *Valores de la salida en función de los estados de Preset y Clear.*

Preset	Clear	Qn+1
0	0	x
0	1	1
1	0	0
1	1	Qn

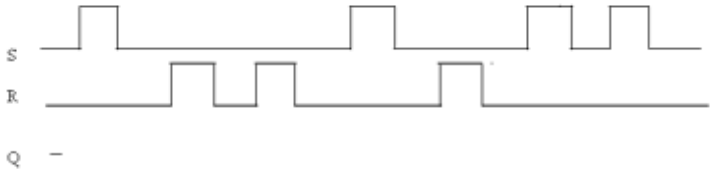
2.3. Ejercicios Propuestos

A continuación, se propone la resolución de los siguientes ejercicios:

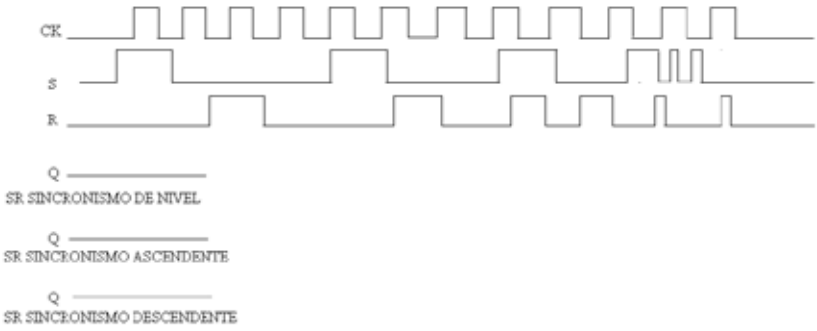
2.3.1. Considerar y analizar el comportamiento del circuito de la siguiente figura. Construir su tabla de verdad y explicar su funcionamiento. Comparar con el Flip Flop SR.



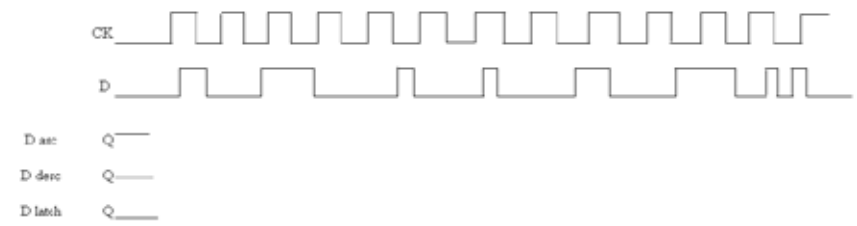
2.3.2. Considerar que el siguiente diagrama de nivel corresponde a un Flip Flop SR asincrónico y completar el estado correspondiente de la evolución a la salida dependiendo de los valores de S y R.



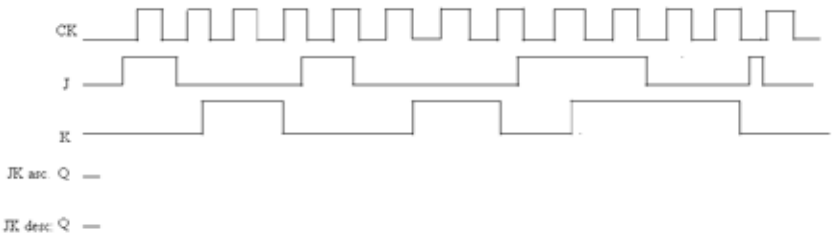
2.3.3. Considerar ahora el siguiente diagrama de nivel para un Flip Flop SR sincrónico. Considerar la activación por nivel, por flanco ascendente y por flanco descendente para completar la evolución del estado de la salida según los valores de entrada y de la señal de clock.



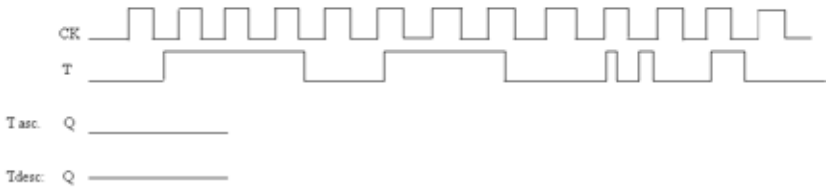
2.3.4. Considerar ahora el siguiente diagrama de nivel para un Flip Flop D sincrónico. Considerar la activación por nivel, por flanco ascendente y por flanco descendente para completar la evolución del estado de la salida según los valores de entrada y de la señal de clock.



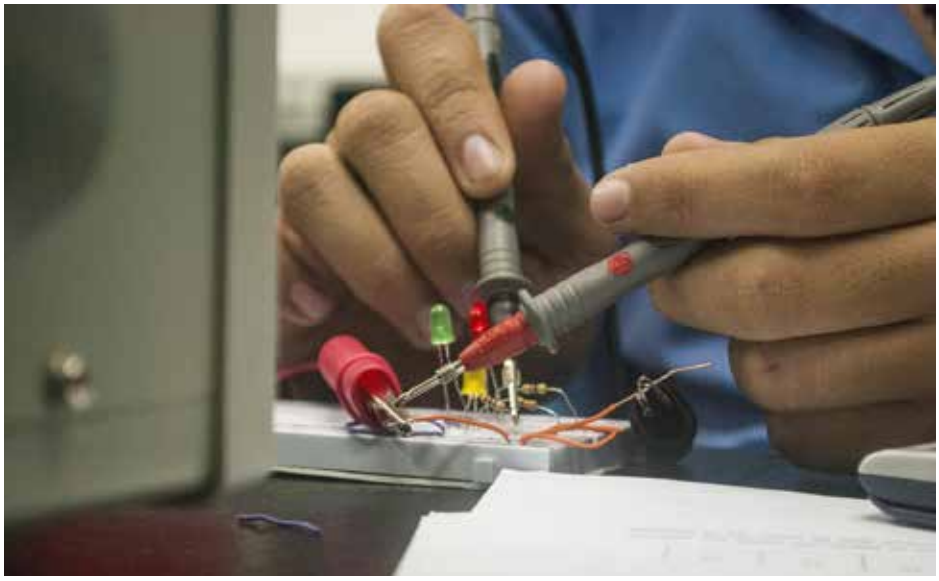
2.3.5. Considerar el siguiente diagrama de nivel para un Flip Flop JK. Considerar la activación por Nivel, por flanco ascendente y por flanco descendente para completar la evolución del estado de la salida según los valores de entrada y de la señal de clock.



2.3.6. Considerar por último el siguiente diagrama de nivel para un Flip Flop T. Considerar la por flanco ascendente y por flanco descendente para completar la evolución del estado de la salida según los valores de entrada y de la señal de clock.



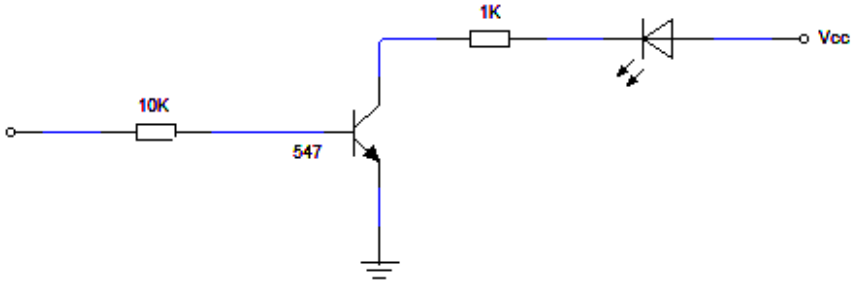
2.3.7. Buscar las características y la distribución de las patas para los siguientes integrados: CMOS 4001; 4011 y 4013. En el caso del presente texto se selecciona la familia CMOS Logic dada la disponibilidad local, pero la práctica puede realizarse con otra familia.



2.4. Práctica de laboratorio o simulaciones propuestas

2.4.1. Armar y probar el circuito de la Figura 2.23 para utilizarlo en la detección del estado de la salida de los circuitos que se estudiarán.

Figura 2.23 *Circuito útil para la detección del estado de la salida de un circuito digital con CMOS*



2.4.2. Armar un Flip Flop SR asincrónico con compuertas NOR. Describa que se observa al aplicar el estado prohibido.

2.4.3. Armar un Flip Flop sincrónico con compuertas NOR. Explique nuevamente para este caso que sucede al aplicar el estado prohibido.

2.4.4. Probar el Flip Flop D contenido en el integrado CMOS 4013 para familiarizarse con el funcionamiento de dicho integrado.

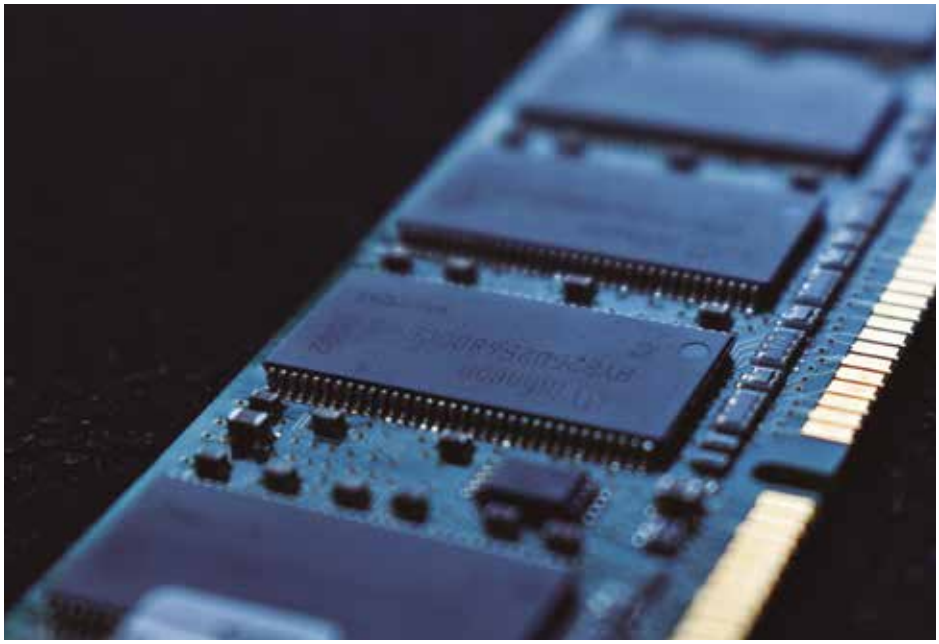
2.4.5. Usando un osciloscopio y un generador de funciones, probar el circuito divisor de frecuencia con Flip Flop tipo D.

2.5. Proyecto: Sistema de alarma simple

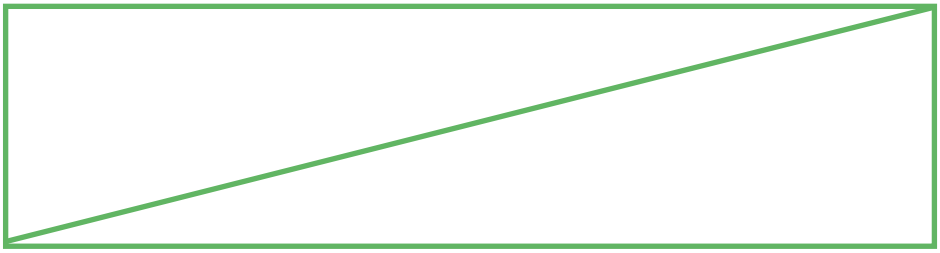
Actividad grupal en el marco de un curso de circuitos secuenciales, posterior a la realización de la práctica de laboratorio y al aprendizaje del uso del manual u hoja de datos.

Siguiendo la idea del recorrido ficticio por el edificio, que se presentó en el ítem 2 de la INTRODUCCIÓN, se propone como ejercicio integrador diseñar, utilizando Flip Flop, así como los contenidos del capítulo, un sistema de alarma simple para la apertura de una puerta o una ventana. La idea es que el lector utilice los contenidos de este capítulo e investigue distintas alternativas para encontrar la propuesta más económica y simple que involucre los dispositivos estudiados.

Capítulo 3
REGISTROS



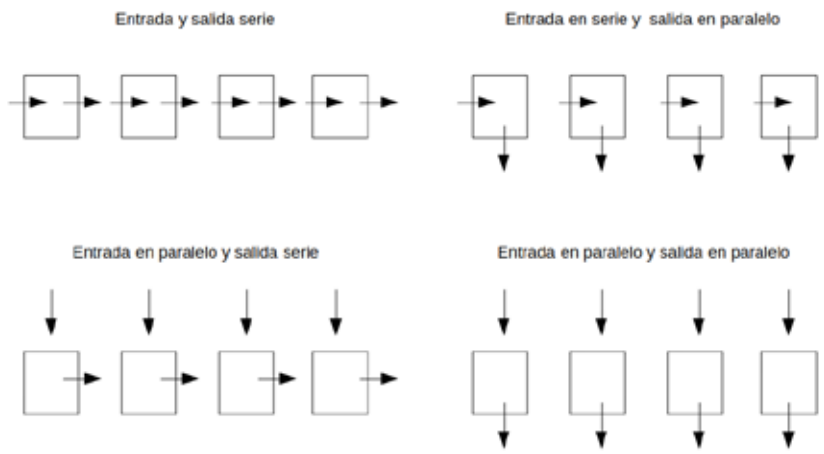
"Todo está guardado en la memoria
sueño de la vida y de la historia.
La memoria despierta para herir
a los cuervos dormidos que no la dejan vivir.
Libre como el viento".
León Gieco, "La memoria".



Los registros son circuitos capaces de memorizar un conjunto de bits. Si almacenan *n* bits, entonces están formados por *n* Flip Flops. Según el modo en que los datos ingresan en el circuito y la forma en que los datos se extraen, se puede obtener diferentes configuraciones básicas. Dichas configuraciones se observan en la Figura 3.1.

La clasificación no es exhaustiva ya que, también hay registros que combinan en un mismo circuito dos o más variantes de los tipos presentados. Los registros son importantes en los sistemas de cómputo ya que, tanto los datos que se procesan en dichos sistemas como la información sobre las operaciones a realizar son almacenados en registros antes de ser procesados.

Figura 3.1 *Clasificación básica de los registros, según el modo en que los datos son ingresados en los Flip Flops*



A excepción del registro con entrada y salida en paralelo de la clasificación anterior, a todos los demás se los llama registros de desplazamiento. Esto se debe a que los datos se tienen que desplazar pasando cada bit de Flip Flop en Flip Flop.

En el presente capítulo estudiaremos distintas opciones para la construcción de estos circuitos.

Normalmente para el diseño de los registros el Flip Flop más utilizado es el tipo D. Según las conexiones entre los biestables, se puede obtener desplazamiento a la izquierda o a la derecha de la información binaria.

Diversos fabricantes presentan circuitos integrados para registros con todas las versiones posibles.

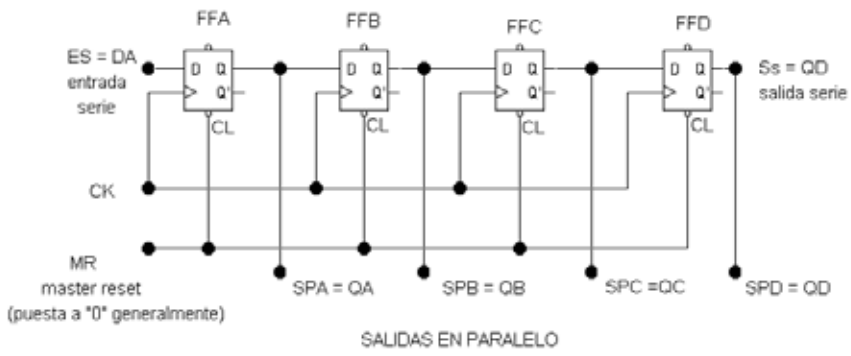
A continuación, veremos ejemplos de cómo construirlos a partir de bloques con Flip Flops.

3.1. Registro de desplazamiento con carga en serie y salida en serie o paralelo

Observemos el circuito de la Figura 3.2. En este caso, el circuito es un registro con entrada en serie y salida en serie o paralelo para almacenar una palabra de 4 bits. Al recibir la señal de clock, el circuito comenzará a desplazar de izquierda a derecha el conjunto de bits que se hubieran ingresado en la entrada serie.

En éste y el resto de los registros, la entrada llamada MR o máster reset permite borrar los bits almacenados a través de un pulso en las entradas de fuerza CLEAR.

Figura 3.2 *Registro de desplazamiento con carga de datos en serie y salida en serie o en paralelo para 4 bits*



Consideremos que deseamos ingresar al registro el dato 1001 para ejemplificar el funcionamiento del circuito.

La Figura 3.3 muestra el diagrama de nivel habiendo ingresado el dato por la entrada serie. En cada ciclo de clock, ocurre qué salida del Flip Flop D copia el

dato que fue ingresado en la entrada. Dado que los Flip Flop están conectados en cascada, es necesario que sucedan varios ciclos de clock (cada ciclo correspondiente a un desplazamiento) para que todos los bits que fueron ingresados en la entrada salgan por la salida en serie.

Se puede construir una tabla de verdad para describir este proceso. Dicha tabla se corresponde en la Figura 3.2.

Si utilizamos la primera columna para contar los ciclos transcurridos de la señal de clock, la segunda columna para indicar el estado del "RESET maestro", la tercera columna para indicar la entrada en serie y las últimas cuatro columnas para indicar los estados de las salidas.

Figura 3.3 *Diagrama temporal para el registro de la Figura 3.2 ejemplificando con el dato 1001*

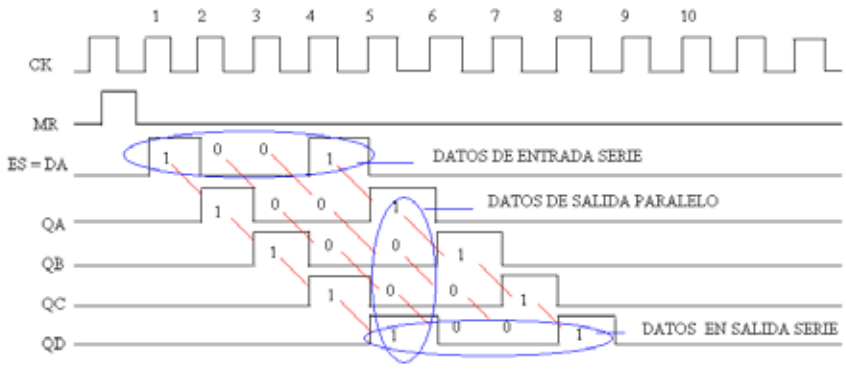


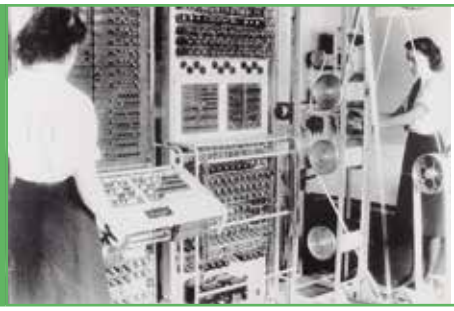
Tabla 3.1 *Tabla de verdad para el registro de la Figura 3.2.*

CK	MR	ES	QA	QB	QC	QD
X	1	X	0	0	0	0
X	0	1	0	0	0	0
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	1	0	0	1	0
4	0	X	1	0	0	1
5	0	X	X	1	0	0
6	0	X	X	X	1	0
7	0	X	X	X	X	1
8	0	X	X	X	X	X

DATOS DE ENTRADA EN SERIE

DATOS DE SALIDA EN PARALELO

DATOS DE SALIDA EN SERIE

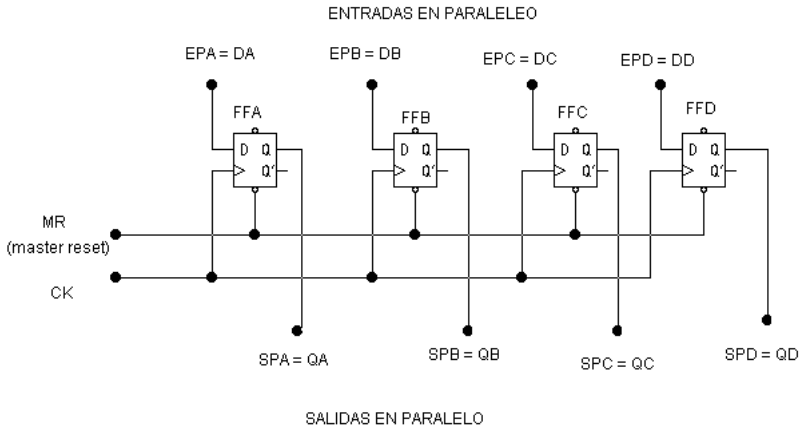


Uno de los primeros ejemplos conocidos de un registro de desplazamiento fue el que se utilizó en Mark 2 Colossus, la segunda versión de Colossus, también fue una máquina de descifrado de códigos construida en 1944.

3.2. Registro con carga en paralelo sincrónica y descarga en paralelo

Para este tipo de circuitos los datos son ingresados por las entradas, de modo paralelo de cada Flip Flop. Cuando el nivel o el flanco del clock lo habilite, los datos ingresados en las entradas aparecerán en la salida. En el caso del circuito que se muestra en la Figura 3.5, éste es activado por el flanco ascendente de la señal de clock. El ejemplo que se muestra en dicha Figura corresponde también a 4 bits como el ejemplo anterior.

Figura 3.5 *Registro con carga en paralelo sincrónica y descarga paralela con Flip Flop tipo D*



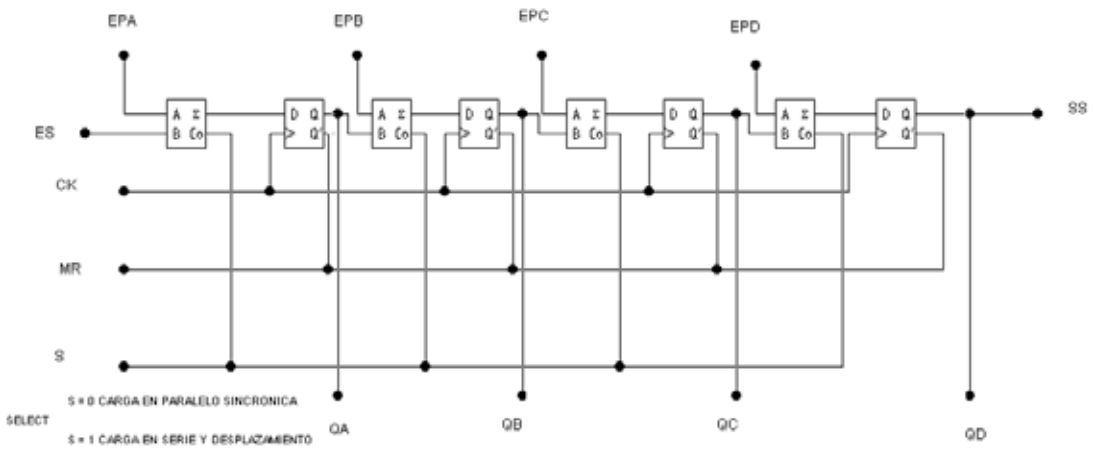
3.3. Registro de desplazamiento con carga y descarga en serie o en paralelo

Este circuito, que se observa en la Figura 3.6, se presenta como una opción que permite combinar un circuito con entradas en serie o en paralelo según sea necesario, controlando esta opción a través de una entrada de control.

Este control lo realizamos a partir de circuitos llamados multiplexores, en este caso de dos canales. El funcionamiento básico consiste en elegir mediante un control de selección, *S* en el circuito, alguno de los canales de entrada para presentar dicha información a la salida del multiplexor. En general estos circuitos se pueden construir a partir de compuertas AND, o se pueden utilizar directamente integrados que tienen implementada dicha función.

Al conectar todos los controles de selección de cada multiplexor juntos en una entrada única de control, obtenemos los dos métodos de carga de información en el circuito según el canal que elegiremos a través del bit de control.

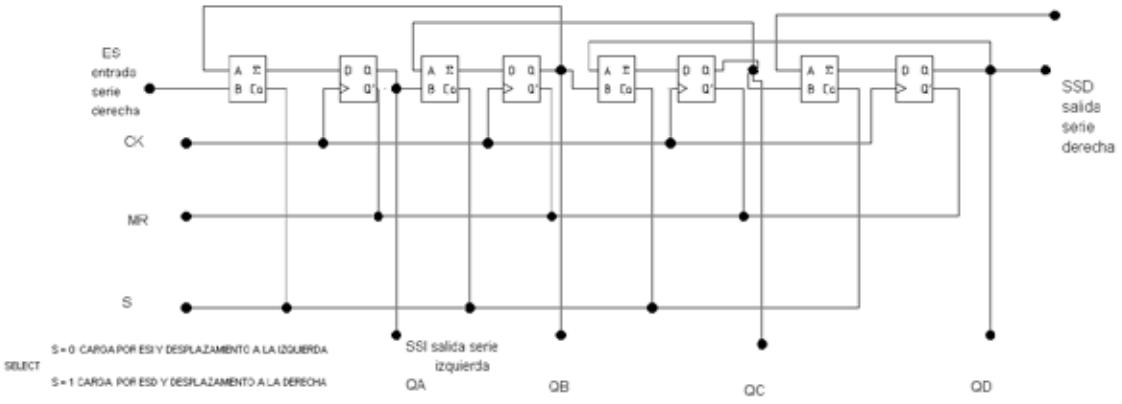
Figura 3.6 *Circuito con Flip Flop tipo D para el registro de desplazamiento con carga y descarga en serie o en paralelo sincrónico*



3.4. Registro de desplazamiento bidireccional

Este circuito permite combinar la opción de que la información circule de izquierda a derecha o de derecha a izquierda implementando el mismo método de los multiplexores que utilizamos en el circuito de la Figura 3.6. El ejemplo de la Figura 3.7 corresponde a un circuito de 4 bits.

Figura 3.7 *Registro bidireccional. Utilizando la entrada de control podemos configurar la dirección de desplazamiento de los bits*



3.5. Registro de desplazamiento con carga en paralelo asincrónica

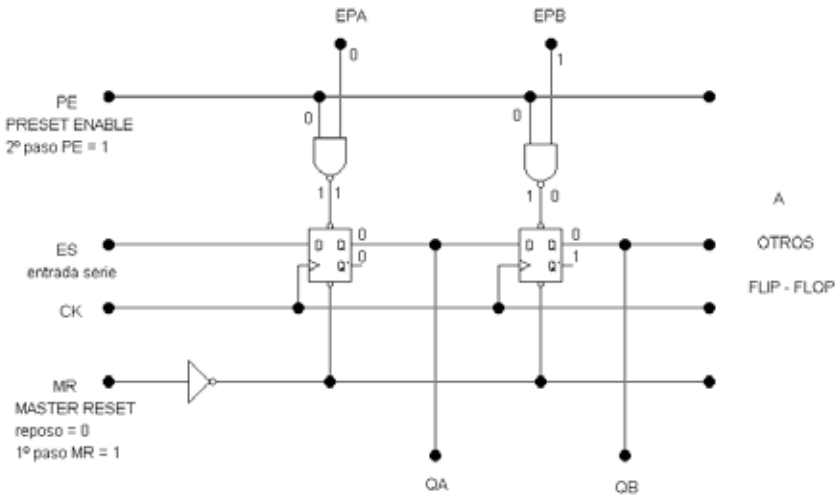
Otra manera de ingresar datos a un circuito digital consiste en la carga asincrónica. Dicho método utiliza las entradas de fuerza de los Flip Flop para realizar la carga de los datos. La carga asincrónica se realiza mediante las entradas de fuerza y se hace principalmente a través de dos métodos.

El primer método consiste en la carga en dos pasos. El circuito que implementa este método se puede observar en la Figura 3.8. En el primer paso se aplica un pulso de “RESET Maestro” para garantizar que todos los Flip Flop tomen estado “0”. En el segundo paso se aplica el dato a las entradas paralelo y luego se aplica un pulso a la entrada “PRESET enable”, que actuará como orden de carga haciendo que se activen los preset de los Flip Flop en cuya entrada paralela

"Ep" se haya aplicado el estado "1". Esto equivale a decir que en el primer paso se graban todos los "0" y en el segundo todos los "1".

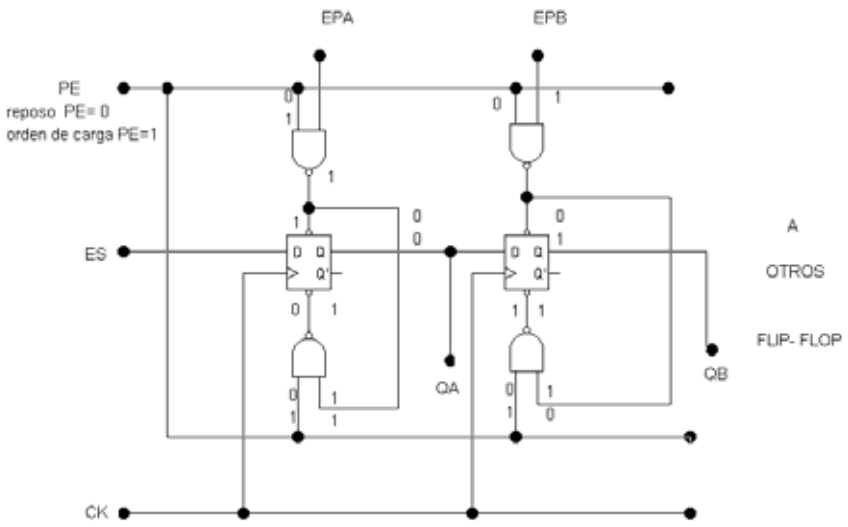
Si las entradas de "Reset maestro" y de "Preset enable" se mantienen en estado de reposo (cero), se puede efectuar la carga en serie y el desplazamiento.

Figura 3.8 *Circuito ilustrando la carga de datos con el método en dos pasos en un registro*



El segundo método consiste en la carga en un paso. El circuito para realizar la carga con este método se muestra en la Figura 3.9. En este caso la entrada "Preset Enable" también controla a las entradas "clear" de los Flip Flops, de modo que al aplicar la orden de carga se graban simultáneamente los estados "0" y los "1". Dicha orden activa a los "clear" de los Flip Flop en cuya entrada paralela hay cero, y a los "preset" de los Flip Flops en cuya entrada paralela hay "1". Dejando la entrada "Preset enable" en reposo se puede hacer carga serie y desplazamiento de la información binaria.

Figura 3.9 *Carga de datos con el método en un paso en un registro*

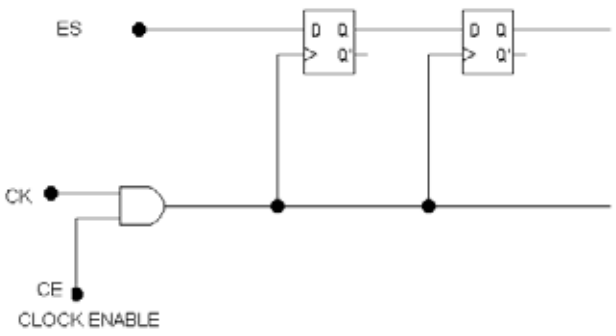


3.6. Mantenimiento de los datos cargados en un registro

Los registros mantienen la información binaria en el circuito hasta que la señal de clock lo haga variar su estado. A veces es deseable conservar por más tiempo la información binaria en el circuito. Por ejemplo, si uno desea hacer alguna operación en la que se requiera esperar el resultado de alguna otra. Para mantener los datos en el registro un tiempo arbitrario existen dos métodos básicos que se pueden aplicar a cualquiera de los registros vistos en este capítulo.

El primero que mostraremos consiste en armar un control como el que se puede observar en la Figura 3.10. Este método se basa en generar una entrada de control que llamaremos “clock enable” o habilitación del clock. Esto lo conseguimos mediante una AND en la entrada del clock. Al aplicar 0 a esta entrada, la señal de clock no puede llegar a los Flip Flop y estos mantienen los datos un tiempo arbitrario hasta que decidamos cambiar el estado de la entrada de habilitación de clock a nivel alto.

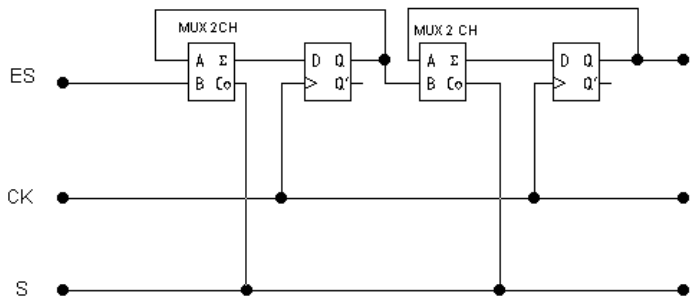
Figura 3.10 *Ejemplo de un circuito para el mantenimiento de datos de un registro*



Cuando el valor de "Clock enable" es 1, los datos circulan por el registro al ritmo que indique el clock. Si en cambio vale cero, el valor permanece retenido.

El segundo método que mostraremos consiste en alimentar la salida de cada Flip Flop hacia su propia entrada para que al recibir la señal de clock se recargue el mismo dato preexistente. Esto se puede realizar con ayuda de multiplexores que permiten seleccionar dicha realimentación o establecer la conexión normal para la carga y el desplazamiento. El ejemplo de cómo implementar dicho método se muestra en la Figura 3.11. Cada canal del multiplexor se utiliza o bien para obtener y desplazar el dato del Flip Flop anterior, o bien para realimentar el estado, dependiendo del nivel que se coloque en la entrada de control.

Figura 3.11 *Ejemplo utilizando los multiplexores del mantenimiento de datos en un registro*



Si en la entrada de control “Select” o “S” se establece un nivel bajo, los datos son mantenidos en el registro por la realimentación, si en cambio se establece un valor alto, se selecciona el segundo canal del multiplexor y los datos circulan a través del registro.

3.7. Registro universal

En este caso veremos un ejemplo de cómo combinar varias de las funciones que vimos a lo largo de este capítulo en un solo circuito.

Llamamos registro universal a aquel que combina en un solo circuito varios de los modos de funcionamiento que hemos visto. La selección del modo de funcionamiento se realiza desde las entradas de selección de los multiplexores que se utilizan para realizar los cambios de conexión que se requieren.

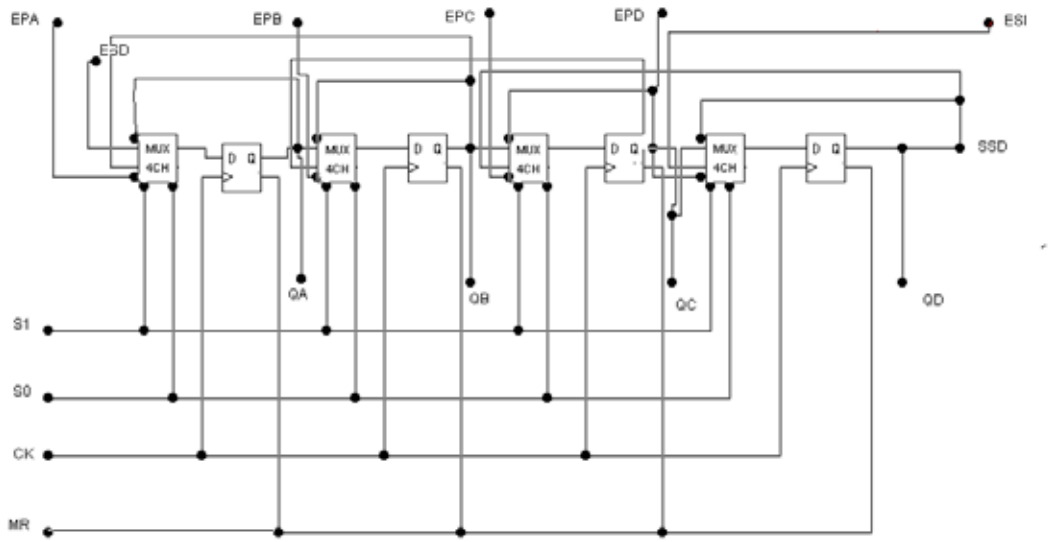
Veamos a continuación un ejemplo de un registro universal de 4 bits con los modos de funcionamiento que se indican en la Tabla 3.2

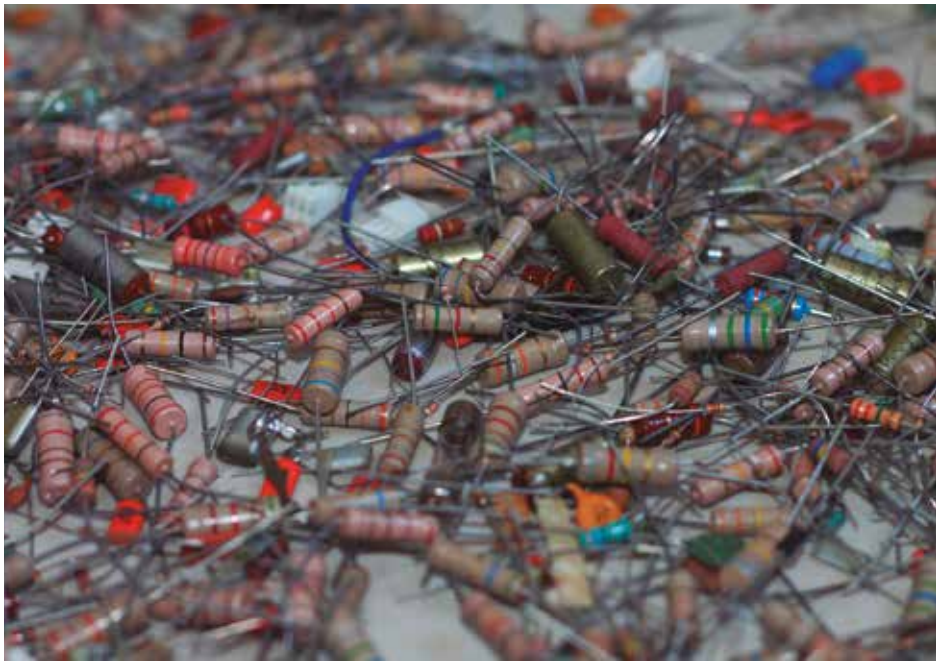
Tabla 3.2 *Tabla de funcionamiento para el registro Universal de la Figura 3.12*

S_0	S_1	Modos de funcionamiento
0	0	Mantenimiento de datos
0	1	Carga por la entrada en serie y desplazamiento a derecha
1	0	Carga por la entrada en serie y desplazamiento a izquierda
1	1	Carga en paralelo sincrónica

De este modo, utilizando Flip Flop tipo D combinados con multiplexores de 4 canales se puede obtener el registro que combina todas las funciones que hemos estudiado.

Figura 3.12 *Registro "Universal" cuyas opciones de funcionamiento están descriptas en la Tabla 3.2*





3.8. Práctica de laboratorio o simulaciones propuestas

Esta práctica puede realizarse en el laboratorio de electrónica o implementarse a través de simulaciones de circuitos con algún software adecuado. Dado que los registros no se utilizan directamente para el diseño de aplicaciones sino como parte de los sistemas de digitales más complejos, dejaremos de lado en este capítulo nuestro ejemplo del edificio que luego retomaremos y propondremos ejercicios que permitan entender como un registro realiza la carga, descarga y manejo de los bits en memoria.

Les solicitamos que realicen la siguiente actividad:

1) Buscar y catalogar cinco registros en el manual de circuitos integrados CMOS.

Identificar el tipo de registro, cantidad de entradas y salidas, cantidad de bits que almacenan y características propias de cada tipo.

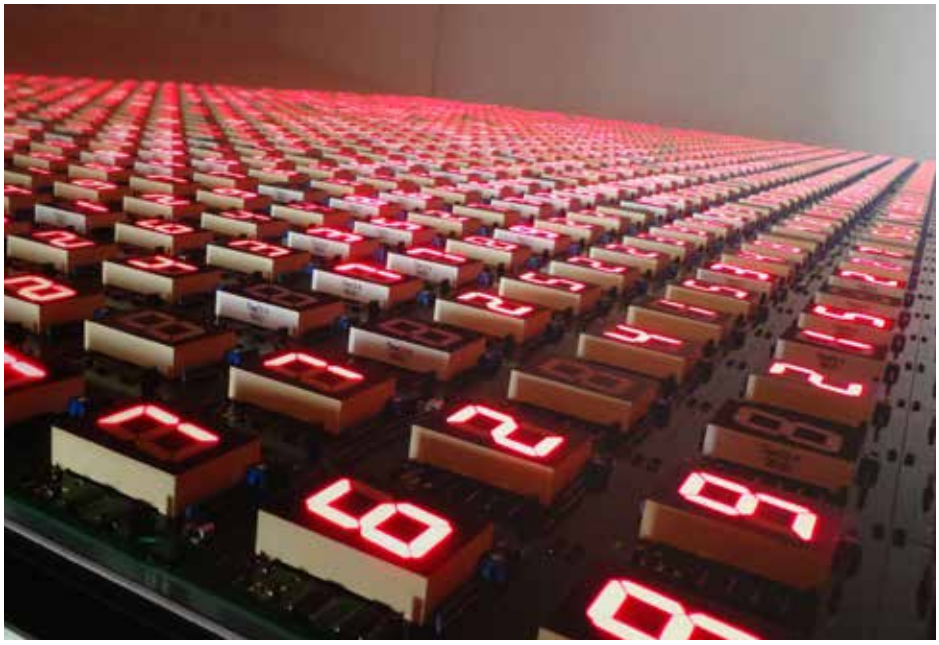
2) Diseñar un registro de desplazamiento de 4 bits de capacidad con carga serie y descarga serie y paralelo. Colocar en las cuatro salidas del registro un circuito detector para observen el estado de las salidas en función de los bits colocados en la entrada.

Utilizar frecuencias adecuadas que permitan dicha observación a simple vista.

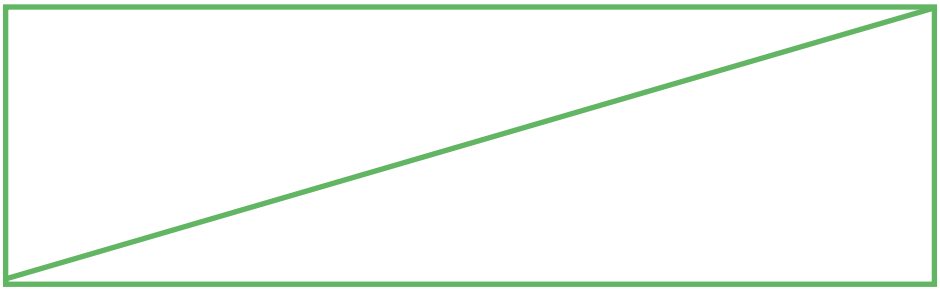
3) Modificar el circuito del punto para que sea carga y descarga en paralelo.

4) Modificar el circuito del punto 3 para que cargue a través de las entradas de fuerza con alguno de los métodos estudiados.

Capítulo 4
CIRCUITOS CONTADORES



"Marcando el tiempo de los momentos que conforman un día aburrido.
Malgastar y malgastar las horas de manera improvisada.
Patear el suelo en tu ciudad natal.
Esperando que alguien o algo te muestre el camino".
Pink Floyd, "Time".



Los circuitos contadores son circuitos secuenciales constituidos por Flip Flop cuya función es contar los pulsos que se aplican por la entrada de cuenta que normalmente es clock.

Antes de comenzar a estudiar implementaciones de distintas versiones de contadores, es necesario definir algunos conceptos.

Comencemos por el **módulo**. Se define el **módulo** de un contador como la cantidad de estados distintos que puede tomar la cuenta. Si llamamos *N* a la cantidad de Flip Flops, el **módulo** de un contador resulta:

$$M \leq 2^N \quad \textbf{(4.1)}$$

Es decir que si un contador va desde 0 hasta 7 contará 8 estados distintos, es decir $M = 8$.

Dependiendo de que la señal de clock llegue a todos los Flip Flops simultáneamente o que cada Flip Flop tenga señal de clock distinta de los demás, el circuito puede ser respectivamente sincrónico o asincrónico. Estos últimos presentan la ventaja de ser más sencillos y económicos, sin embargo, presentan ciertas limitaciones que los hacen inadecuados para determinadas aplicaciones en los que habrá que usar contadores sincrónicos.

Según la forma en que se presenta el resultado de la cuenta en la salida, los contadores se pueden clasificar en distintos tipos, como pueden ser:

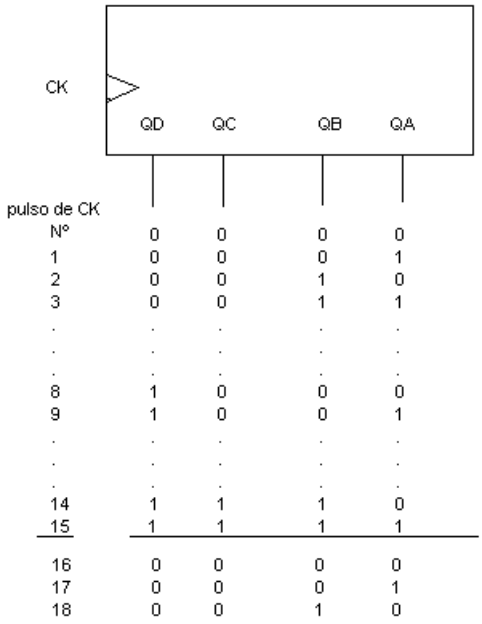
- Contador Binario.
- Contador BCD, donde las salidas presentan el resultado en código BCD.
- Contador decimal o por décadas con salidas decodificadas.
- Otros tipos.

En el caso del contador binario, se lo llama así porque la salida sigue la cuenta binaria natural. Un esquema general para este tipo de circuitos se presenta en la Figura 4.1, donde a medida que avanzan los pulsos de reloj, la cuenta de estos a la salida del circuito se incrementa en binario.

En el caso del contador BCD, las salidas presentan los resultados de la cuenta en código BCD, que significa poner en binario con cuatro bits los dígitos decimales, un esquema se muestra en la Figura 4.2. El tercer ejemplo que ilustramos en la Figura 4.3 es un contador decimal o por décadas con salidas

decodificadas. En este caso, las salidas presentan los códigos con un nivel alto correspondiente a la salida numerada que corresponde al estado de la cuenta.

Figura 4.1 *Contador Binario: la salida sigue la cuenta binaria natural*



Las salidas del contador propiamente dicho van a un decodificador que nos da una presentación final en la que hay una cantidad de patas de salida igual al módulo y se muestra el estado de la cuenta mediante una única entrada que toma estado “1”.

Los anteriores ejemplos son los tipos más usados, pero existen otros como el contador Jonson, Aiquen, el BCD exceso de 3, el Grey, etc. El Flip Flop más práctico para diseñar contadores es el T o bien el JK. Pero también se podrían utilizar el SR y el D.

Analizaremos en la siguiente sección los circuitos contadores asincrónicos.

Figura 4.2 *Contador BCD: las salidas presentan en código BCD*

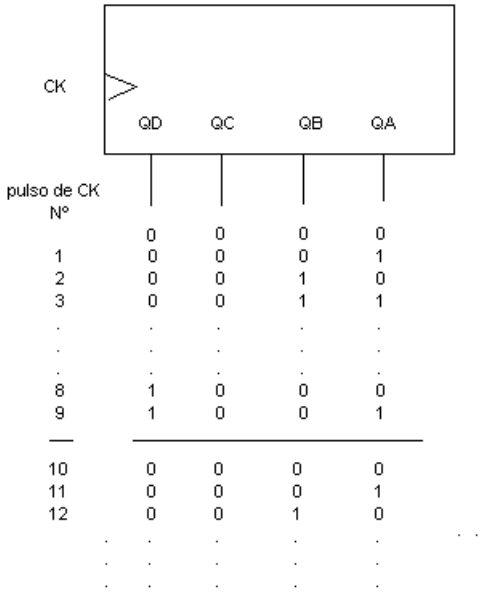
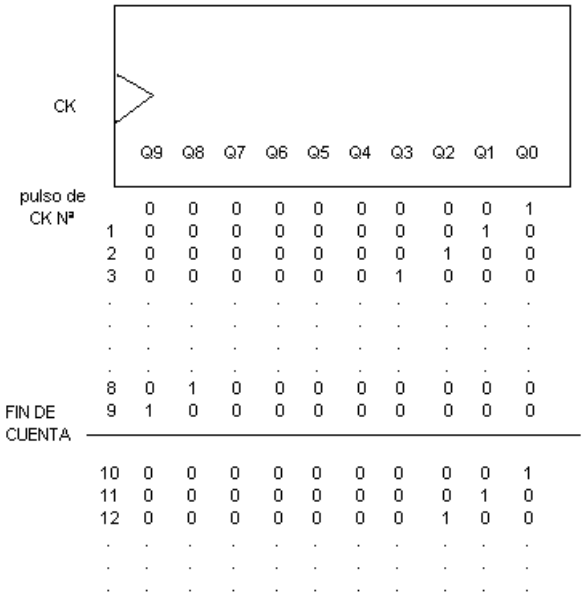


Figura 4.3 *Contador decimal o por décadas con salidas decodificadas*



Existen diversos circuitos integrados que se pueden adquirir que son directamente circuitos contadores. A continuación, veremos de un modo pedagógico, cómo se pueden construir a partir de Flip Flops los contadores.

Mucho antes de que la electrónica se volviera común, los dispositivos mecánicos se usaban para contar eventos. Estos se conocen como dispositivos de conteo o contadores mecánicos. Por lo general, consisten en una serie de discos montados en un eje, con los dígitos del cero al nueve marcados en su borde. El disco más a la derecha se mueve un incremento con cada evento. Cada disco, excepto el de la izquierda, tiene una protuberancia que, después de completar una revolución, mueve el siguiente disco al incremento de la izquierda. Dichos contadores se utilizaron como odómetros (dispositivos para medir distancias) para bicicletas y automóviles y en grabadoras. También fueron, usados en dispensadores de combustible, en maquinaria de producción, así como en otros tipos de maquinaria.

4.1. Contadores asincrónicos

Este tipo de circuitos se caracterizan por que la señal de clock de entrada solo llega al primer Flip Flop del circuito, y todos los demás Flip Flop reciben la señal de clock provenientes desde la salida del Flip Flop que lo precede.

Veamos por ejemplo algunos circuitos.

4.1.1. Contador asincrónico binario de M = 8

En el ejemplo que se presenta en la Figura 4.4 se muestra un contador asincrónico ascendente de *M = 8* con Flip Flop tipo T. La activación del circuito se produce por flanco descendente. Consideremos inicialmente que el circuito está en cero. Al ocurrir el primer flanco descendente de la señal de clock, el estado de la salida del primer Flip Flop cambiará. Cada flanco descendente de cada salida o *Q* controlará la conmutación del Flip Flop siguiente. La evolución de cada una de las salidas de

los Flip Flop se pueden observar en la Figura 4.5. Combinando los estados de cada salida al tiempo *t* se obtiene la cuenta binaria correspondiente a cada estado.

Notemos que cada bloque del circuito funciona como un divisor de frecuencia de la señal de clock hecho con Flip Flop T como el que estudiamos en la Sección 2.2.3. De manera que, disparando adecuadamente los Flip Flop con el flanco correspondiente, podemos combinar los estados para tener la cuenta ascendente.

La línea azul muestra el momento donde la cuenta se reinicia.

Figura 4.4 *Contador asincrónico ascendente con M = 8*

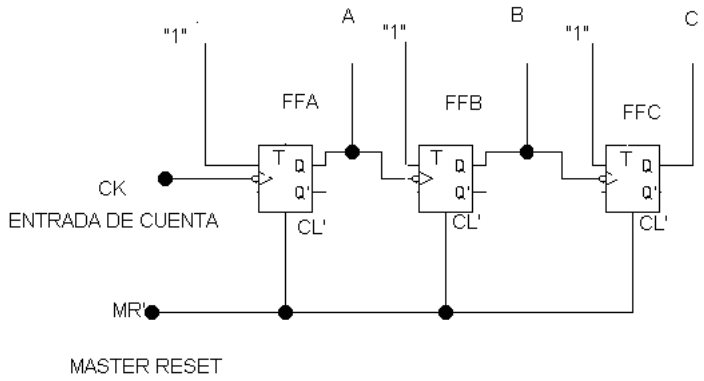
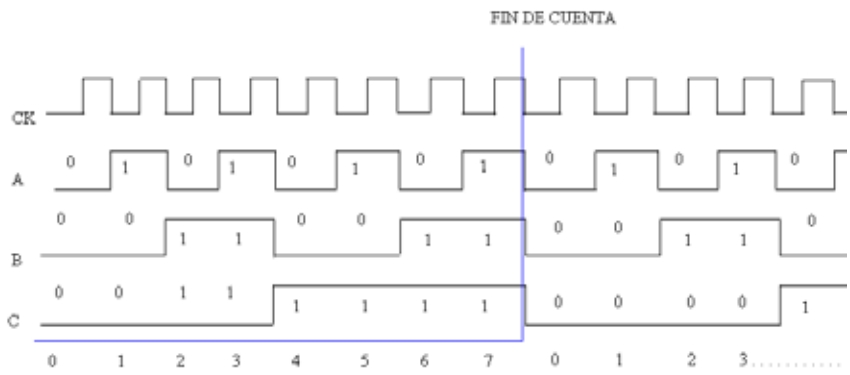


Figura 4.5 *Diagrama temporal para el contador asincrónico ascendente con M = 8*



4.1.2. Contador asincrónico descendente

En el ejemplo que se presenta en la Figura 4.6 se muestra un contador asincrónico descendente con *M = 8*. La activación del circuito se produce por flanco descendente. Consideremos inicialmente que el circuito está en cero. Al ocurrir el primer flanco descendente de la señal de clock, el estado de la salida del primer Flip Flop cambiará. En este caso la diferencia que observamos es que la es la salida negada de cada Flip Flop la que está conectada al siguiente. Así mismo cada flanco descendente de cada salida o $\bar{Q}$ controlará la conmutación del Flip Flop siguiente. La evolución de cada una de las salidas de los Flip Flop se pueden observar en la Figura 4.7. Combinando los estados de cada salida al tiempo *t* se obtiene la cuenta binaria correspondiente a cada estado, en este caso la cuenta descendente.

Figura 4.6 *Contador asincrónico descendente con M = 8*

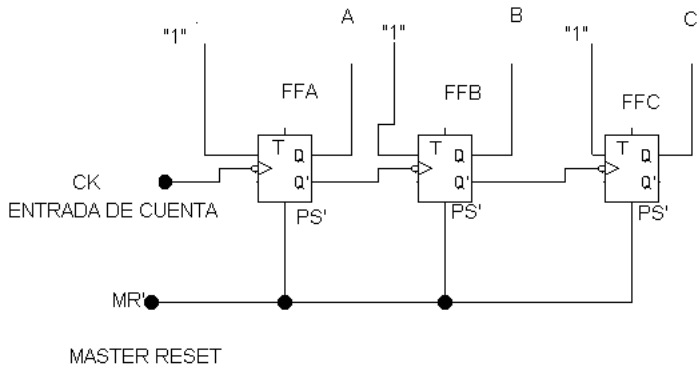
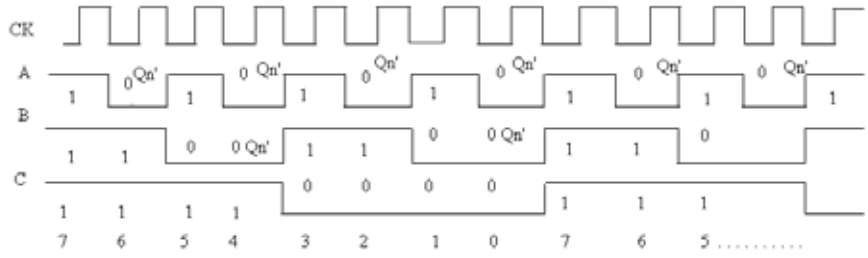


Figura 4.7 *Diagrama temporal para el contador asincrónico descendente con M = 8*



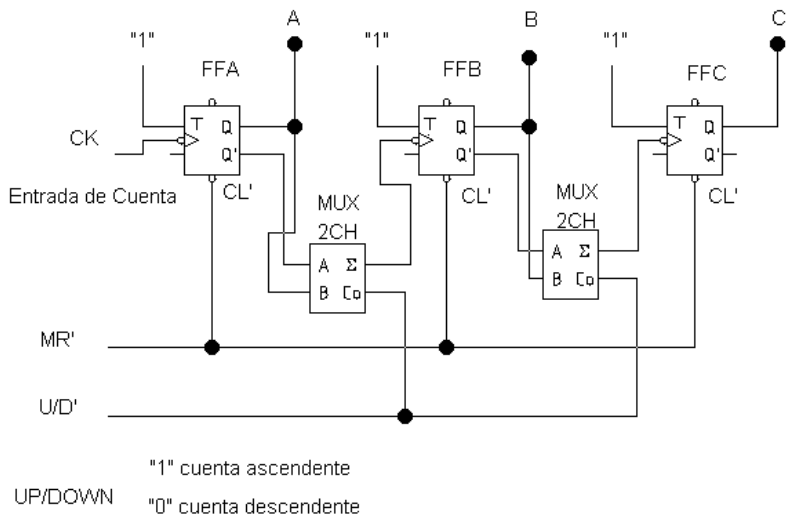
A continuación, veremos cómo nuevamente es posible combinar las funciones de dos circuitos contadores mediante el uso de multiplexores.

4.1.3. Contador asincrónico binario ascendente o descendente (up/down)

Ejemplo para M = 8

Con la ayuda de los multiplexores conectados como se observa en el circuito de la Figura 4.8, este circuito puede contar de manera ascendente o descendente dependiendo del nivel que se coloque en la entrada de control. La evolución del estado de la cuenta será como en los dos circuitos que vimos en las secciones anteriores dependiendo del estado de la entrada de control.

Figura 4.8 *Contador asincrónico binario ascendente/descendente*

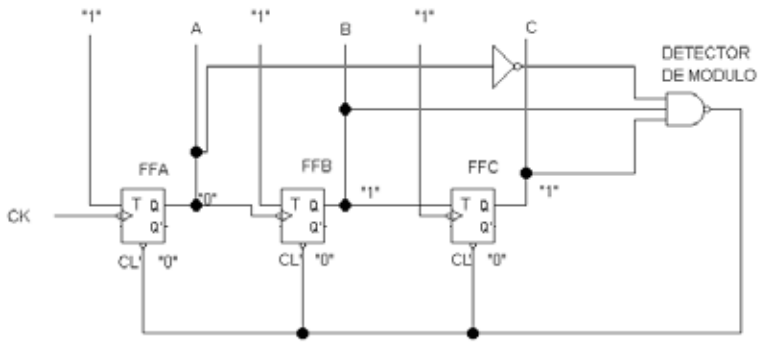


4.2. Contadores con módulo menor que 2^n

Este tipo de contadores se obtiene agregando un circuito que detecte el estado de la cuenta que corresponde al módulo deseado y en ese momento actúe sobre la entrada de fuerza CLEAR de cada Flip Flop aplicándole un pulso de “Reset Maestro” para provocar que el contador se reinicie anticipadamente.

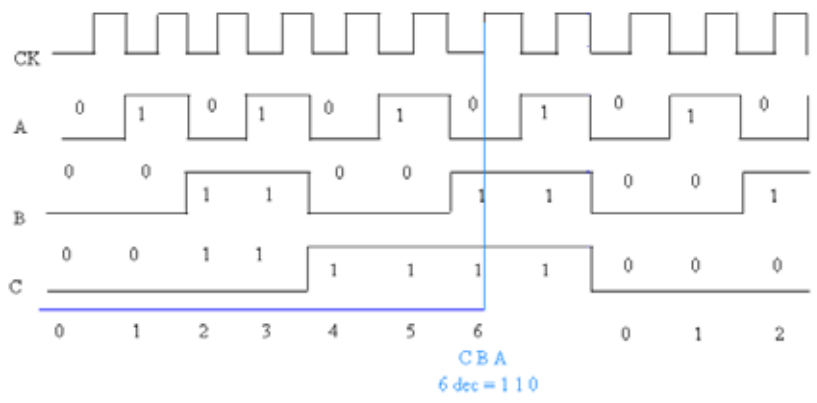
Veamos a continuación un ejemplo para un contador binario de módulo 6 asincrónico en la Figura 4.9.

Figura 4.9 *Contador con módulo M = 6*



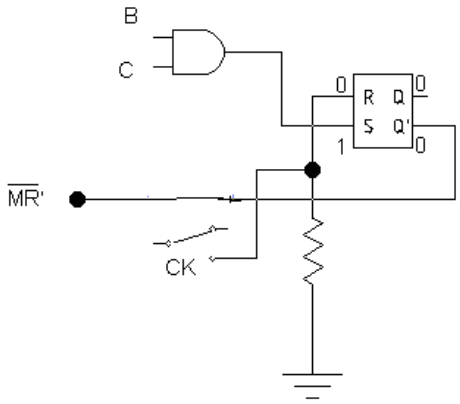
En este caso particular se puede agregar una compuerta NAND de 2 entradas, conectando en cada Flip Flop B y C (en vez de usar una NAND de tres entradas y un inversor).

Figura 4.10 *Diagrama temporal para el contador de módulo 6 en la Figura 4.9*



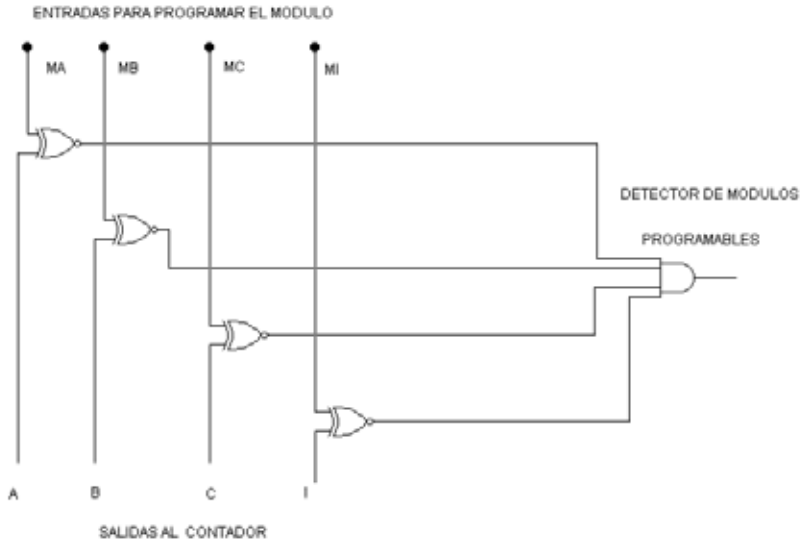
Si en el lugar del detector de módulo se incorpora el circuito de la Figura 4.11 eso permitirá que al completarse la cuenta el contador quede detenido en “0” hasta que provoquemos el disparo a través del control de la llave.

Figura 4.11 *Detector de módulo programable*



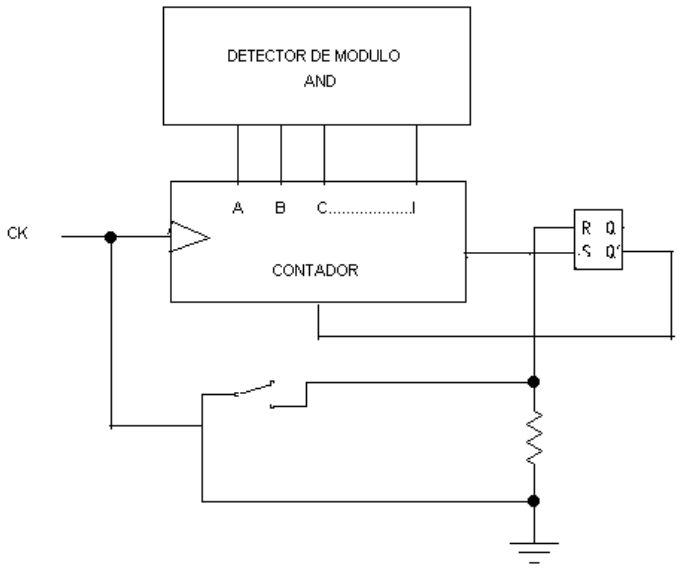
Esta idea se puede generalizar para cualquier módulo menor que 2^n, de modo que se puede obtener el diagrama presentado en la Figura 4.12.

Figura 4.12 *Generalizamos el detector de módulo a través de compuertas*



Basándose en la misma idea se puede utilizar un detector de módulo programable que es un circuito que detectara el momento en que las salidas del contador se igualen con el número binario aplicado a las entradas de programación. Un ejemplo se muestra en la Figura 4.13.

Figura 4.13 *Generalizamos el detector de módulo programable con un bloque*



4.3. Condiciones reales de los contadores asincrónicos

Las señales digitales no se propagan en los circuitos de manera instantánea. Desde que reciben la señal de habilitación en el clock hasta que cambia de estado la salida de cada Flip Flop, las señales demoran un tiempo de propagación (tp) y lo mismo sucede desde la entrada de un Flip Flop hacia el siguiente clock en la cadena. Por este motivo pueden surgir dos problemas:

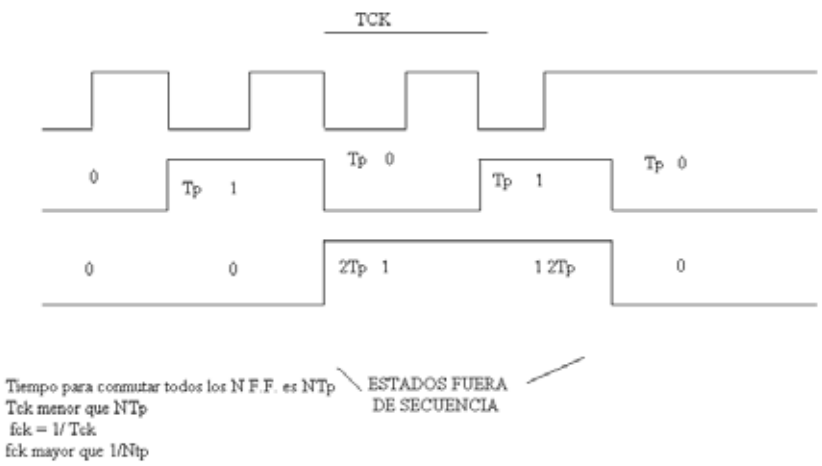
- La frecuencia máxima de clock queda limitada.
- Aparecen estados intermedios erróneos fuera de secuencia.

Se pueden apreciar estas dos cuestiones en el diagrama de tiempo ampliado de la Figura 4.14.

En muchas aplicaciones estos efectos son despreciables, pero hay otras aplicaciones, en general de alta frecuencia que por estos efectos no permiten usar el contador asincrónico. Todo dependerá de la frecuencia de trabajo de la señal de

clock que necesitemos utilizar. Veamos por ejemplo lo que sucede para un contador de M=4 en la Figura 4.14 el diagrama de nivel con los efectos descriptos.

Figura 4.14 *Diagrama de nivel con los estados fuera de sincronismo*



4.4. Contadores Sincrónicos

Estos circuitos se caracterizan por que la señal de clock de la entrada llega a todos los Flip Flops al mismo tiempo y provoca que todos conmuten simultáneamente. Recordemos que en los circuitos asincrónicos la conmutación era escalonada. Este tipo de circuitos se puede aplicar para señales de más alta frecuencia ya que aquí se evitan los estados fuera de secuencia y la acumulación de tiempos de demora.

Como todos los pulsos de clock de entrada son recibidos por todos los Flip Flops se debe determinar mediante un procedimiento los circuitos adecuados para excitar a las entradas preparatorias de manera que cada Flip Flop comúnmente solo en el momento oportuno y no en todos niveles o flancos de cada pulso de clock.

Veremos a continuación un método de diseño que es válido para cualquier circuito secuencial sincrónico ya sea un contador o un generador de secuencia pseudo aleatorio, aplicación que describiremos luego.

Para aplicar el método de diseño, primero es necesario conocer la secuencia deseada que se puede representar como una tabla de verdad o como un gráfico de

estados. Con esta información se confecciona una tabla de verdad en la que debe figurar el estado futuro que le corresponde a cada estado de la salida del contador y además la excitación que requieren las entradas preparatorias para cada uno de estos cambios de estado. Luego por simple inspección de la tabla o por algún método de diseño como el mapa de Karnaugh (ver Apéndice B) se obtiene las funciones y los correspondientes circuitos para excitar las entradas preparatorias. Para entender este procedimiento vamos a implementar estos pasos a través de un ejemplo.

4.4.1. Ejemplo 1. Contador sincrónico binario de módulo 16

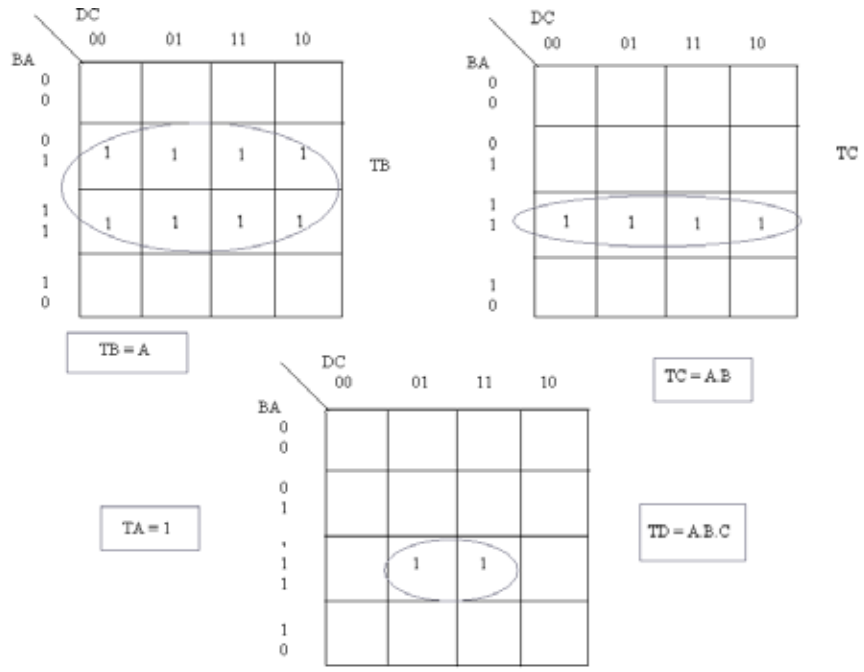
Supongamos que deseamos obtener la cuenta de manera ascendente considerando la posibilidad de contar 16 estados distintos. En primer lugar, del lado izquierdo de la tabla colocaremos todas las combinaciones posibles (2^N) que se pueden obtener con N bits o Flip Flops. Dichos estados de la salida los llamaremos estados Q_x actuales, donde X corresponde a cada Flip Flop. A continuación, colocaremos los estados Q^*_X que llamaremos estados futuros. Finalmente, en las últimas 4 columnas pondremos el estado que debería tener la entrada T_X para cada Flip Flop X si deseásemos obtener el cambio que nos lleva del estado actual Q_X al futuro Q^*_X.

Tabla 4.1 *Tabla de verdad para obtener el funcionamiento del contador sincrónico ascendente*

Q_D	Q_C	Q_B	Q_A	Q^*_D	Q^*_C	Q^*_B	Q^*_A	T_D	T_C	T_B	T_A
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	1	0	1	0	0	0	1	1
1	0	1	0	1	0	1	1	0	0	0	1
1	0	1	1	1	1	0	0	0	1	1	1
1	1	0	0	1	1	0	1	0	0	0	1
1	1	0	1	1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	1	0	0	0	1
1	1	1	1	0	0	0	0	1	1	1	1

De este modo con primero se puede observar por simple inspección que la entrada T_A debe conectarse al nivel alto. Los restantes resultados de cada una de las columnas de T los colocaremos en un mapa de Karnaugh elaborado con los estados actuales Q_X. Dichos mapas se observan en la Figura 4.15.

Figura 4.15 *Mapas de Karnaugh para poder obtener las funciones correspondientes para las salidas de los Flip Flops*



Utilizando los resultados de estos mapas podemos diseñar el circuito. En este caso tenemos dos posibles opciones para construir estos circuitos: circuito con transporte en serie (Figura 4.16) y circuito con transporte en paralelo (Figura 4.17).

El circuito con transporte en paralelo es el contador más rápido que hay ya que produce acumulación de tiempo de demora e independientemente de la cantidad de etapas del circuito tarda solamente el tiempo de un Flip Flop más el de una compuerta desde que recibe el clock hasta que está preparado para recibir el próximo clock. Tiene como desventaja que usa compuertas de distinta cantidad de entradas y que la cantidad de etapas está limitada por la cantidad de cargas que pueda soportar la salida de la primera etapa.

Figura 4.16 *Circuito con transporte en serie*

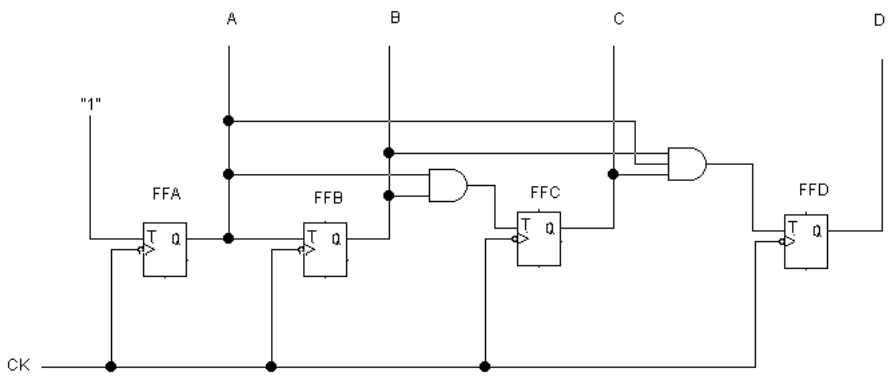
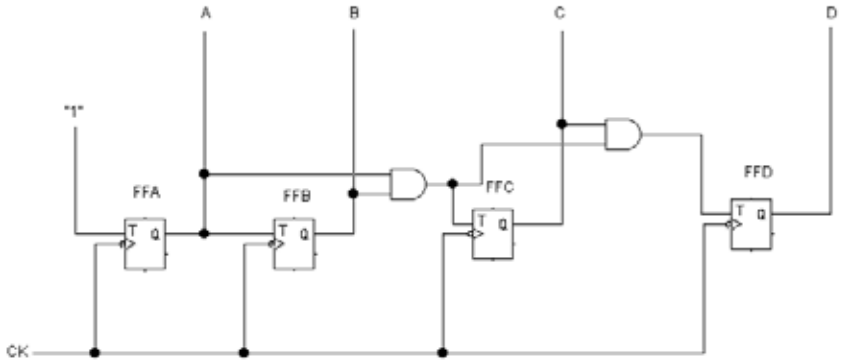


Figura 4.17 *Circuito con transporte en paralelo*



El circuito con transporte en serie, si bien desde que se le aplica el clock tarda el tiempo de un Flip Flop en dar el nuevo estado (esto es igual que en el circuito anterior), a partir de aquí se produce la acumulación de los tiempos de las *n* compuertas encadenadas (*n* cantidad de Flip Flops). Por este motivo el valor de frecuencia máxima de clock es relativamente bajo y parecido al de los circuitos asincrónicos. Tiene la ventaja de usar solo compuertas de dos entradas y de no tener límite para la cantidad de etapas que se pueden usar dado que, independientemente esta cantidad la primera salida solo soporta dos cargas.

4.4.2. Ejemplo 2. Diseño de un generador pseudoaleatorio

Llamamos generador pseudoaleatorio a un circuito que permite obtener una secuencia desordenada. Se llama pseudoaleatorio porque para a quien desconoce dicha secuencia puede parecerle que los estados son obtenidos al azar, sin embargo, no es cierto, pseudo significa falso. A continuación, aplicamos el mismo método para diseñar un generador pseudoaleatorio que siga la secuencia dada en el siguiente diagrama de estados de la Figura 4.18. En este caso utilizaremos el Flip Flop JK. Consideremos la secuencia de la Figura y representemos la tabla de verdad asignando al lado de cada estado de la secuencia el estado siguiente y redundancias "X" a aquellos estados que no aparezcan en la secuencia. En este caso tendremos que evaluar los cambios que deben ocurrir en ambas entradas J y K de cada uno de los siguientes Flip Flops.

Figura 4.18 *Secuencia pseudoaleatoria*

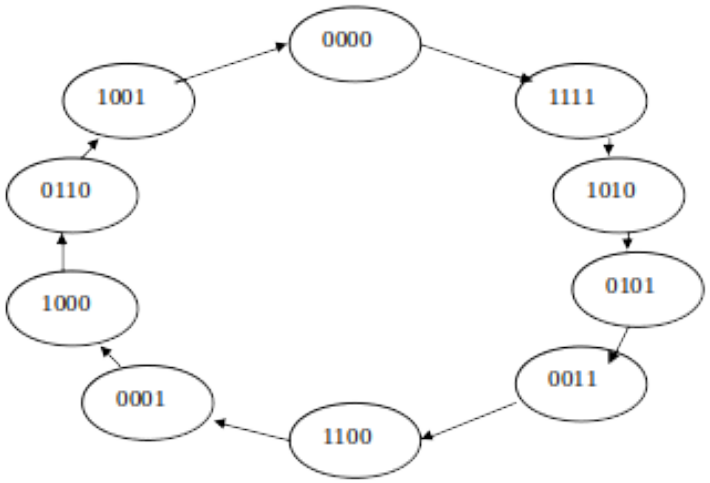


Tabla 4.2 *Tabla para construir un generador pseudoaleatorio para la secuencia ejemplificada en la Figura 4.18*

Q_D	Q_C	Q_B	Q_A	Q^*_D	Q^*_C	Q^*_B	Q^*_A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	1	1	1	1	1	x	1	x	1	x	1	x
0	0	0	1	1	0	0	0	1	x	0	x	0	x	x	1
0	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x
0	0	1	1	1	1	0	0	1	x	1	x	x	1	x	1
0	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x
0	1	0	1	0	0	1	1	0	x	x	1	1	x	x	0
0	1	1	0	1	0	0	1	1	x	x	1	x	1	1	x
0	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x
1	0	0	0	0	1	1	0	x	1	1	x	1	x	0	x
1	0	0	1	0	0	0	0	x	1	0	x	0	x	x	1
1	0	1	0	0	1	0	1	x	1	1	x	x	1	1	x
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	0	0	0	1	x	1	x	1	0	x	1	x
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	1	0	1	0	1	x	x	1	1	x	x	1

Una vez construida la tabla de verdad, procedemos a utilizar la técnica de Karnaugh sobre cada entrada como se realizó en el ejemplo anterior. Se deja para el lector como ejercicio, la construcción de los mapas y el correspondiente circuito.

4.5. Ejercicios propuestos

Utilizando el método aprendido para los circuitos sincrónicos, diseñen los siguientes circuitos.

1) Contador BCD ascendente usando JK.

2) Contador BCD descendente usando T.

3) Generador pseudoaleatorio usando JK utilizando la inversa de la secuencia del ejemplo estudiado.

4) Elijan alguna secuencia de 3 bits y diseñen un generador pseudoaleatorio utilizando el Flip Flop tipo T.

4.6. Práctica de laboratorio o simulaciones propuestas

1) Diseñar un contador asincrónico ascendente módulo 16 utilizando el Flip Flop tipo D (integrado CMOS 4013).

Aclaraciones:
a) Tener en cuenta que se activa por flanco ascendente.
b) Tener en cuenta cómo armar el divisor de frecuencia con el Flip Flop tipo D.
c) Las salidas de cada Flip Flop van a circuitos detectores.

2) Modificar el circuito anterior para que la cuenta sea descendente.

3) Modificar el circuito del punto 1 para que corte la cuenta en los estados de (0 a 9).

4) Al circuito del punto 3 agregarle utilizando el integrado 4511 un display de siete segmentos para que muestre los dígitos del 0 al 9.

5) Dar la distribución de patas y características del 4511. Expliquen la función de las patas LT; BI; LA.

6) Buscar cinco circuitos contadores en el manual CMOS y describirlos.

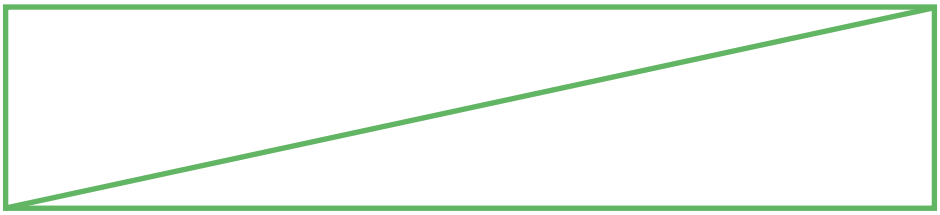
4.7. Proyecto: Cerradura electrónica con código

Actividad grupal en el marco de un curso de circuitos secuenciales, posterior a la realización de la práctica de laboratorio y al aprendizaje del uso del manual u hoja de datos. Se aconseja que esta actividad sea guiada por el docente.

Siguiendo la idea del recorrido ficticio por el edificio, que se presentó en el ítem 2 de la INTRODUCCIÓN, se propone como ejercicio diseñar utilizando un contador una cerradura electrónica. Se propone por ejemplo la realización de un circuito utilizando por ejemplo el contador de CMOS 4017, o bien otro contador binario.

Capítulo 5
CIRCUITOS BIESTABLES Y OSCILADORES

"Para decidir, para continuar,
para recalcar y considerar.
Solo me hace falta que estés aquí
con tus ojos claros".
Víctor Heredia



Los osciladores son circuitos que permiten generar señales que varían en forma periódica o cuasi periódica. En el caso de los circuitos digitales, dichas señales suelen ser señales rectangulares o cuadradas. En el presente capítulo nos enfocaremos en explicar los fundamentos que permiten obtener dichas señales a través del uso de circuitos biestables y en particular aprovechando la característica de realimentación. En particular estas señales las hemos utilizado como señales de clock para los circuitos presentados en los capítulos previos. Son las señales de sincronización o señales a partir de las cuales los circuitos tomarán decisiones de activación.

5.1. Osciladores con compuertas: fundamentos

5.1.1. Ecuaciones para la carga y descarga de un capacitor

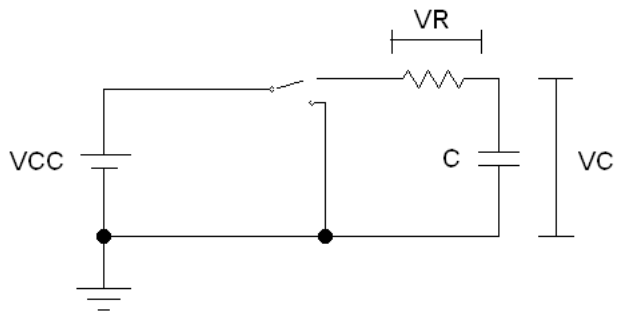
La mayoría de los osciladores con compuertas se implementan bajo el principio de relajación. Este principio se basa en un comparador con histéresis cuya tensión proviene de un capacitor que a su vez se carga y se descarga controlado por la salida del comparador. Se realizan las conexiones de modo tal, que la tensión del capacitor, que será otra pieza fundamental de estos sistemas, siempre irá variando en el sentido adecuado para provocar la conmutación del comparador. La tensión de salida del comparador tiene forma rectangular y la tensión sobre el capacitor forma de diente de sierra, suponiendo que se lo cargue a través de resistencia.

Para comprender mejor el funcionamiento de estos circuitos, repasemos el concepto de carga y descarga de un capacitor que luego será fundamental para el diseño de los osciladores.

Imagen 5.1 Capacitores Electrolíticos

En el circuito de la Figura 5.1, tenemos un capacitor en serie con una fuente, una resistencia y una llave.

Figura 5.1 *Circuito para estudiar el proceso de carga y descarga del capacitor utilizando una llave manual*



Al colocar la llave en la posición superior el capacitor comenzará a cargarse. Llamamos VC a la caída de potencial sobre el capacitor y VR sobre la resistencia. Este proceso ocurrirá tendiendo asintóticamente al valor de tensión de la fuente de alimentación. Escribimos las ecuaciones relacionadas con este proceso. Planteamos la ecuación de mallas para la llave colocada en la posición superior (ecuación 5.1*)*.

$$V_R = V_{cc} - V_C \quad \textbf{(5.1)}$$

V cc es la tensión correspondiente a la fuente de alimentación.

A continuación, consideramos la ecuación 5.1. y escribimos en la ecuación 5.2. cada caída de potencial en término de la corriente *i* y la carga *Q*, sabiendo que $V_C = \frac{Q}{C}$

$$i \,.\, R = V_{cc} - \frac{Q}{C} \quad \textbf{(5.2)}$$

Hasta aquí pudieron seguir el texto sin necesidad de tener conocimientos sobre el Análisis Matemático. Para las ecuaciones que siguen sí se requieren al menos tener conocimientos fundamentales. Si no, basta decir con que la corriente *i(t)* estará dada por la ecuación 5.7.

Recordamos que, por definición la corriente es la derivada de la carga respecto del tiempo. Esto implica que si reemplazamos la expresión de la intensidad $i = \frac{dQ}{dt}$ en la ecuación 5.2 se obtiene:

$$\frac{dQ}{dt} . R = Vcc - \frac{Q}{C} \quad \textbf{(5.3)}$$

La ecuación 5.3 es una ecuación diferencial de variables separables.

Para obtener la ecuación que describe la evolución de la carga en función del tiempo, en la ecuación 5.3. agrupamos los términos de Q en uno de los miembros de la ecuación, e integramos respecto del tiempo entre 0 y un tiempo arbitrario *t*, donde para ese tiempo el valor de la carga será *q*.

$$\int_0^q \frac{dQ}{Vcc - \frac{Q}{C}} . R = \int_0^t dt \quad \textbf{(5.4)}$$

Por conveniencia para nuestro cálculo, en la ecuación 5.4 dividimos ambos miembros por *RC*.

$$\int_0^q \frac{dQ}{C . Vcc - \frac{Q}{C}} = \frac{1}{RC} \int_0^t dt \quad \textbf{(5.5)}$$

Integrando el miembro izquierdo de la igualdad $\int_0^q \frac{dQ}{C . Vcc - \frac{Q}{C}}$ obtenemos $\ln (C . Vcc - \frac{Q}{C}) \Big|_0^q$

Integrando en el miembro derecho de la igualdad el factor $\int_0^t dt$ obtenemos $\frac{1}{RC} \Big|_0^q$

Ahora despejamos la carga *Q* del capacitor para obtener la ecuación 5.6, considerando la carga como una función del tiempo.

$$Q(t) = Vcc . C (1 - e^{-\frac{t}{RC}}) \quad \textbf{(5.6)}$$

Para obtener la corriente en función del tiempo es necesario derivar la ecuación 5.6.

De este modo obtenemos la ecuación 5.7.

$$i(t) = \frac{Vcc}{R} e^{-\frac{t}{RC}} \quad \textbf{(5.7)}$$

La corriente en el proceso de carga del capacitor está dada por la ecuación 5.7 y se muestra en la Figura 5.2 a la izquierda del valor VF. También se puede expresar en términos del potencial si se divide la ecuación por *R*.

Analizamos ahora la malla que se forma al colocar la llave en la posición inferior. En esta posición se producirá la descarga del capacitor. La ecuación de malla en este caso será como se muestra en la ecuación 5.8.

$$0 = i \,.\, R + \frac{Q}{C} \quad \textbf{(5.8)}$$

Reemplazando a la corriente por $i = \frac{dQ}{dt}$, la ecuación 5.8 se puede escribir como de visualiza en la ecuación 5.9.

$$0 = \frac{dQ}{dt} \,.\, R + \frac{Q}{C} \quad \textbf{(5.9)}$$

Despejamos la ecuación 5.9 para poder aplicar separación de variables y multiplicamos ambos miembros por $\frac{1}{RC}$ para obtener la ecuación 5.10, donde integramos miembro a miembro. Pero esta vez, consideremos que inicialmente (al tiempo $t = 0$) el capacitor está cargado con un valor q y que al final estará completamente descargado:

$$-\int_0^t dt \frac{1}{RC} = \int_q^0 \frac{dQ}{Q} \quad \textbf{(5.10)}$$

De este modo la descarga del capacitor queda determinada por la función:

$$Q(t) = q \,.\, e^{-\frac{t}{RC}} \quad \textbf{(5.11)}$$

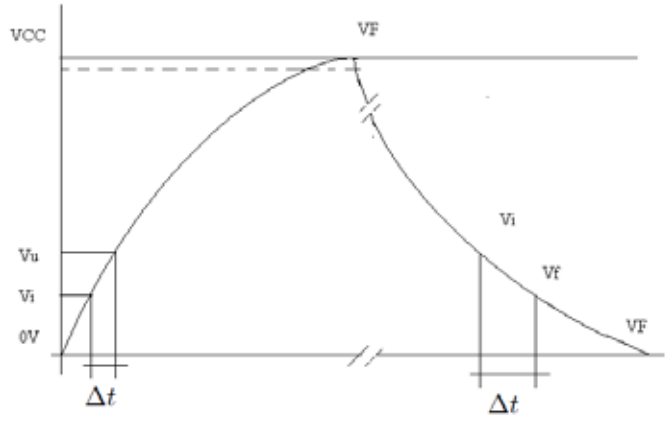
O bien podemos escribirla en términos de la corriente, que consiste en derivar la ecuación 5.11, así tendremos:

$$i(t) = Vcc \,.\, R\, e^{-\frac{t}{RC}} \quad \textbf{(5.12)}$$

De modo que la Figura 5.2 del lado derecho del valor VF muestra el correspondiente proceso de descarga.

Del mismo modo que antes, también se puede expresar en términos del potencial, si se divide la ecuación por *R*.

Figura 5.2 *Diagrama temporal de la tensión en el capacitor en función del tiempo para la carga y la descarga del circuito mostrado en la Figura 5.1*



5.1.2. Ecuación general para el diseño de un oscilador

Conocidos ciertos valores de tensión asociados a momentos particulares de la carga y a la descarga, y partir de las ecuaciones 5.7 y 5.12 que hemos visto, es posible despejar los valores para ciertos tiempos característicos *t**. Dichos tiempos característicos tendrán que ver en nuestro caso con tiempos de diseño en los cuales nuestros circuitos osciladores presentarán la salida en estado alto o en estado bajo.

Invirtiendo dichas ecuaciones para despejar el tiempo, entendido como un intervalo que se desee calcular asociado a la carga o a la descarga Δt, se puede obtener las ecuaciones que permiten diseñar circuitos osciladores. De este modo escribimos la ecuación más general que utilizaremos a lo largo de este capítulo para poder realizar el diseño de los circuitos osciladores.

$$\Delta t = R \,.\, C \,.\, ln \frac{V_f - V_i}{V_f - V_U} \quad \textbf{(5.13)}$$

En la ecuación 5.13, la tensión V_i significa el valor de tensión en el instante inicial en el que comenzamos a contar el intervalo de tiempo Δt. Podría ser al principio de la carga, por ejemplo, en cuyo caso valdría cero o en otro instante de tiempo. El valor de tensión V_f significa el valor de tensión que tendría al final de un proceso de carga o descarga completo, es decir, 0 o *Vcc* dependiendo del

caso. El valor de tensión llamado V_U, o tensión umbral, es el valor de tensión que alcanzará realmente el capacitor al momento de producirse la conmutación del estado de nuestro circuito biestable. Observamos en cada caso cuánto vale dicha tensión umbral para cada diseño particular.

De este modo, para diseñar el circuito tal que satisfaga cierto intervalo de tiempo de duración para obtener el estado alto de la salida debemos considerar dicho intervalo, que denominamos t_A como se muestra en la ecuación 5.14.

$$t_A = R \,.\, C \,.\, ln \frac{V_{cc} - V_i}{V_{cc} - V_U} \quad \textbf{(5.14)}$$

Donde del mismo modo, para el tiempo característico de descarga la ecuación será:

$$t_B = R \,.\, C \,.\, ln \frac{0 - V_i}{0 - V_U} \quad \textbf{(5.15)}$$

De este modo V_U y V_i se corresponden con los valores de conmutación.

5.2. Oscilador con compuertas con histéresis

En los gráficos de la Figura 5.3 se observan los símbolos para dos compuertas inversoras: una estándar y otra con histéresis.

En este contexto llamamos histéresis al hecho que la conmutación entre el

Figura 5.3 *Símbolos para una compuerta NOT con y sin histéresis, izquierda y derecha respectivamente*

estado alto y bajo se produce en un nivel de tensión distinto según el estado previo de la salida. En una compuerta común, el valor de conmutación es en general la mitad del valor de tensión de alimentación, es decir $\frac{Vcc}{2}$. En una compuerta con histéresis, el valor de conmutación desde el estado bajo de la entrada hacia el alto corresponde con un valor llamado *VT−* y el valor de conmutación del estado alto hacia el bajo se producirá en *VT+*.

Para el caso de la compuerta NOT con histéresis, esto quiere decir que la salida conmutará hacia el estado bajo (desde el estado alto) cuando la entrada tenga un valor por encima de $VT+$ y que la salida conmutará hacia el estado alto (desde el estado bajo) cuando la tensión descienda por debajo de $VT-$, pasando por alto el valor de $VT+$ y generando el fenómeno de ventana de histéresis.

Comparamos la curva o función de transferencia de una compuerta común con una compuerta con histéresis en la Figura 5.4. En esta figura llamamos VDD a la tensión de alimentación de la fuente. A veces puede aparecer escrita de ese modo, en particular en los manuales de componentes CMOS. Deriva del inglés “Voltage Drain Drain”, en lugar del tradicional Vcc que utilizamos anteriormente que también deriva del inglés “Voltage Common Collector”. Pueden utilizarse ambos nombres para llamar a la tensión de alimentación.

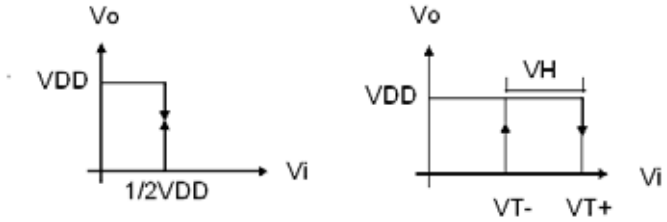
Aprovechamos el diseño de la compuerta NOT con histéresis para diseñar un circuito oscilador. Dicho circuito se muestra en la Figura 5.5. El comportamiento del estado de la salida y la tensión sobre el capacitor vienen dados por las curvas que se muestran en la Figura 5.6 izquierda y derecha respectivamente.

De este modo, los tiempos característicos de nuestro oscilador van a estar dados por las ecuaciones 5.14 y 5.15 donde sustituimos el valor de V_i y V_U.

$$T_A = R_A \cdot C \cdot ln \frac{V_{cc} - VT-}{V_{cc} - VT+} \quad \textbf{(5.16)}$$

$$T_B = R_B \cdot C \cdot ln \frac{0 - VT+}{0 - VT-} \quad \textbf{(5.17)}$$

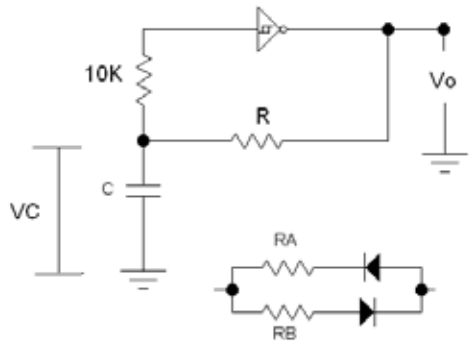
Figura 5.4 *Función de transferencia. Gráfico de la evolución del estado de una compuerta NOT con y sin histéresis como las observadas en la Figura 5.3*



El periodo será $T = T_A + T_B$, la frecuencia $f = \frac{1}{T}$ y la amplitud de la señal de salida será aproximadamente igual al valor de tensión de alimentación, como ocurre siempre con las compuertas lógicas.

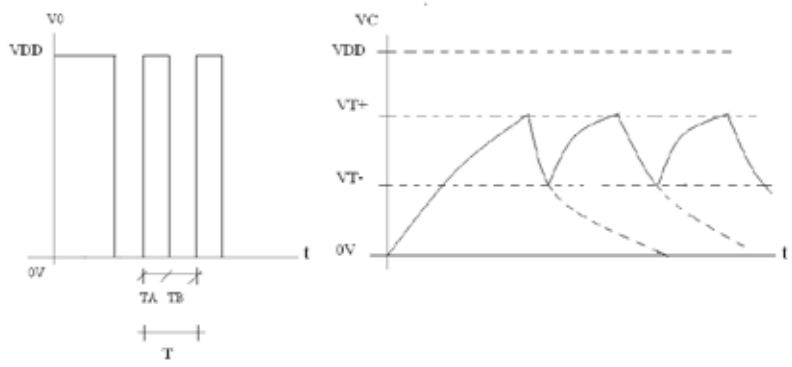
Notemos en la Figura 5.6 que el primer ciclo del oscilador es más largo en duración, esto se debe a que inicialmente el capacitor está completamente descargado y la carga no se produce desde la tensión $VT-$, sino desde cero.

Figura 5.5 *Circuito para un oscilador con compuerta con histéresis*



Si queremos valores distintos de duración para el estado bajo y el estado alto, debemos sustituir R por R_A y R_B en paralelo con sus respectivos diodos en serie para la carga y la descarga.

Figura 5.6 *Diagrama temporal para circuito para un oscilador con compuerta con histéresis. VO en función del tiempo y VC en función del tiempo*



5.3. El integrado 555

El 555 es un circuito integrado utilizado en una gran variedad de aplicaciones como temporizador, para generación de pulso y oscilador. El 555 se puede usar para proporcionar retrasos de tiempo y para diversas aplicaciones que requieran control temporal.

Fue introducido en 1972 por la empresa Signetics y todavía se usa ampliamente debido a su bajo precio, facilidad de uso y estabilidad. Por este motivo le dedicamos esta sección. Actualmente es fabricado por muchas compañías en las tecnologías CMOS bipolares y de baja potencia originales.

La Figura 5.7 muestra el diagrama en bloques del dispositivo. Entre el voltaje de alimentación positivo *Vcc* y la tierra llamada *GND* hay un divisor de voltaje que consta de tres resistencias idénticas. Dichas resistencias crean dos voltajes de referencia: $Vs = \frac{2}{3} \cdot Vcc$ y $\frac{Vs}{2} = \frac{1}{3} \cdot Vcc$

La segunda referencia está conectada al pin llamado “Control”. Las tres resistencias tienen el mismo valor, 5 *k*Ω para temporizadores bipolares, 100 *k*Ω (o valores de resistencia más altos) para temporizadores CMOS.

La entrada negativa del comparador está conectada al divisor de voltaje de referencia más alto de $\frac{Vs}{2} = \frac{1}{3} \cdot Vcc$, y el pin “Control”. Mientras que la entrada positiva del comparador está conectada al pin “Reposición”.

La entrada positiva del segundo comparador está conectada al divisor de voltaje de referencia inferior de $\frac{1}{3} \cdot Vcc$, y la entrada negativa del comparador está conectada al pin “Disparo”.

Un Flip Flop SR almacena el estado del temporizador y es controlado por los dos comparadores. El pin “Reset” anula las otras dos entradas, por lo tanto, el Flip Flop (y todo el temporizador) se pueden restablecer en cualquier momento.

La salida del Flip Flop es seguida por una etapa de salida con controladores push-pull (P.P.) que pueden cargar el pin “Salida” con hasta 200 *mA* (varía según el dispositivo). La salida del Flip Flop activa un transistor que conecta el pin de “Descarga” a tierra.

Si no se conectan resistencias, ni se aplica tensión a la pata 5, las tensiones de

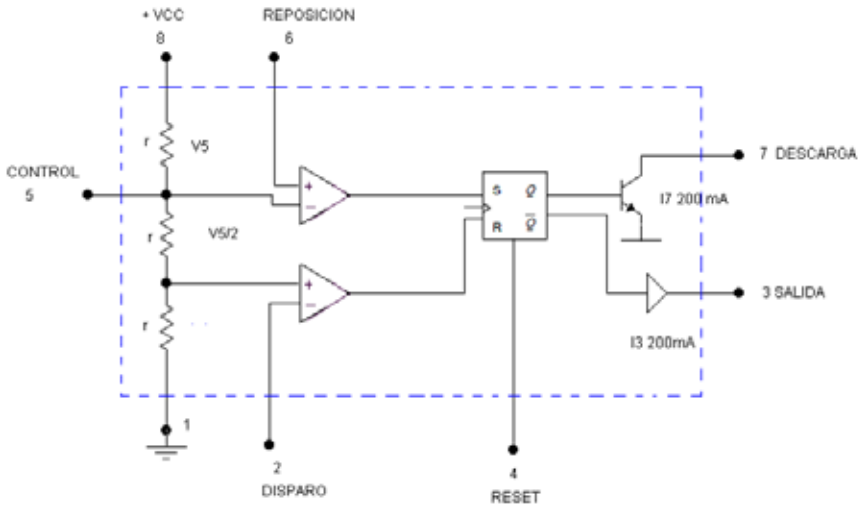
conmutación *Vs*, equivalentes a las que llamamos V_{T+} y V_{T-} en el oscilador de compuerta con histéresis, valdrán: $Vs = \frac{2}{3} \cdot Vcc$ y $\frac{Vs}{2} = \frac{1}{3} \cdot Vcc$

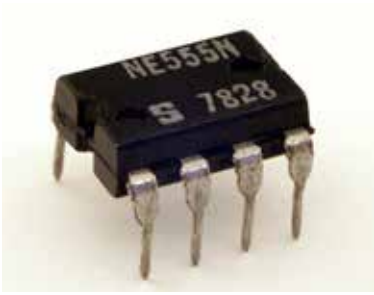
Sin embargo, si conectamos elementos que cambien la distribución de tensión en la pata de control dichos valores de referencia deben volver a calcularse. La tabla 5.1 resume el funcionamiento del circuito según los estados de las entradas de comparación.

Tabla 5.1 *Tabla de Funcionamiento para el integrado 555 en función de los estados de V_2 y V_6.*

V_6	V_2	R	S	$Qn+1$	RESULTADO
$<V_5$	$<V_5/2$	0	1	1	Disparo
$<V_5$	$>V_5/2$	0	0	Q_n	Mantenimiento
$>V_5$	$<V_5/2$	1	1	X	Prohibido
$>V_5$	$>V_5/2$	1	0	0	Reset

Figura 5.7 *Diagrama interno para el integrado 555*





El integrado 555 fue diseñado en 1971 por Hans Camenzind quien trabajaba bajo contrato con Signetics. Él diseñó un oscilador para que la frecuencia no dependiera del voltaje o la temperatura de la fuente de alimentación. El primer diseño para el 555 fue revisado a mediados de 1971. Camenzind tuvo la idea de usar una resistencia directa en lugar de una fuente de corriente constante y descubrió que funcionaba bien. El cambio disminuyó los 9 pines requeridos a 8, por lo que el integrado podría entrar perfectamente en un encapsulado de 8 pines en lugar de uno de 14 pines. La copia de 9 pines ya había sido lanzada por otra compañía fundada por un ingeniero que asistió a la primera revisión y se retiró de Signetics. Esa firma retiró su versión poco después del lanzamiento del 555. El temporizador 555 fue fabricado por 12 compañías en 1972 y se convirtió en el producto más vendido. Se ha dicho que el 555 obtuvo su nombre de las tres resistencias de 5 kΩ dentro del integrado. Hans Camenzind ha declarado que el número de pieza era arbitrario, por lo tanto, fue simplemente una coincidencia.

El integrado 555 presenta en general distintos modos de funcionamiento:

1) Modo astable: el 555 puede funcionar como un oscilador electrónico. Los diferentes usos incluyen LED y luces intermitentes, generación de pulso, relojes lógicos, generación de tono, alarmas de seguridad, modulación de posición de pulso, etc.

También es posible usar el 555 como un Conversor Analógico Digital o ADC simple, convirtiendo un valor analógico en una longitud de pulso. Estos dispositivos ADC se describen en el capítulo 7. Esta aplicación se puede hacer, por ejemplo, seleccionando un termistor como resistencia de temporización permite el uso del 555 en un sensor de temperatura y el período del pulso de salida está determinado por la temperatura. El uso de un circuito basado en microprocesador puede convertir el período de pulso a temperatura, linealizarlo e incluso proporcionar medios de calibración.

2) Modo monoestable: en este modo, el 555 funciona como un generador de impulsos “one-shot”. Las aplicaciones incluyen temporizadores, detección de pulso faltante, interruptores sin rebote, interruptores táctiles, divisor de frecuencia, medición de capacitancia, modulación de ancho de pulso (PWM), etc.

3) Modo biestable (Flip Flop): el 555 funciona como un Flip Flop SR. Los usos incluyen interruptores bloqueados sin rebote.

4) Modo Schmitt Trigger (inversor): el 555 funciona como una puerta de inversor schmitt trigger que convierte una entrada ruidosa en una salida digital limpia.

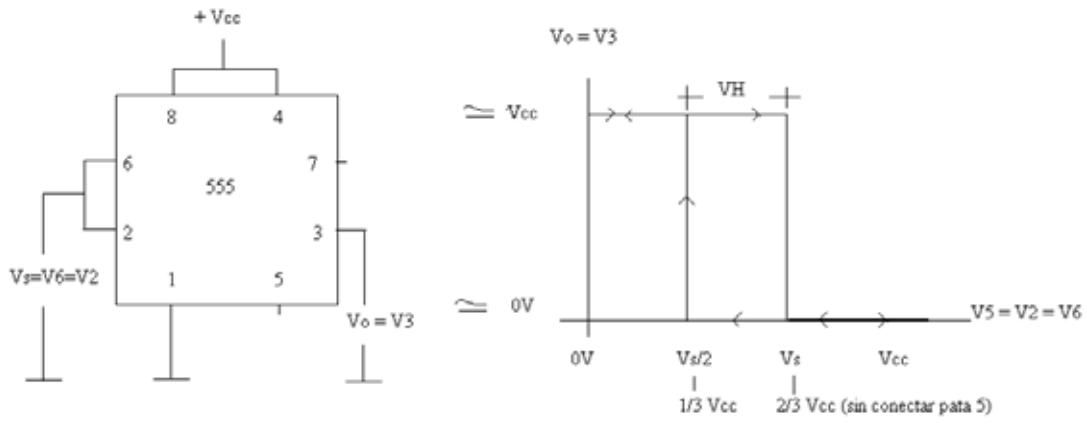
5.4. Algunos ejemplos de aplicación

Existe un sin fin de circuitos que se pueden armar con el integrado 555. Aquí presentaremos tres ejemplos que se pueden implementar de manera muy sencilla en el laboratorio, entre los cuales está por supuesto el oscilador que es el circuito que motivó este capítulo.

5.4.1. Comparador con histéresis

Este circuito, presentado a la izquierda de la Figura 5.8 permite comparar el valor de la señal de entrada con dos valores de referencia dados por la ventana de histéresis, mostrada a la derecha de la Figura 5.8. De esta manera que el circuito conmutará su salida de (positivo a cero) en el valor V_{T+}, y conmutará su salida (de cero a positivo) en el valor V_{T-}.

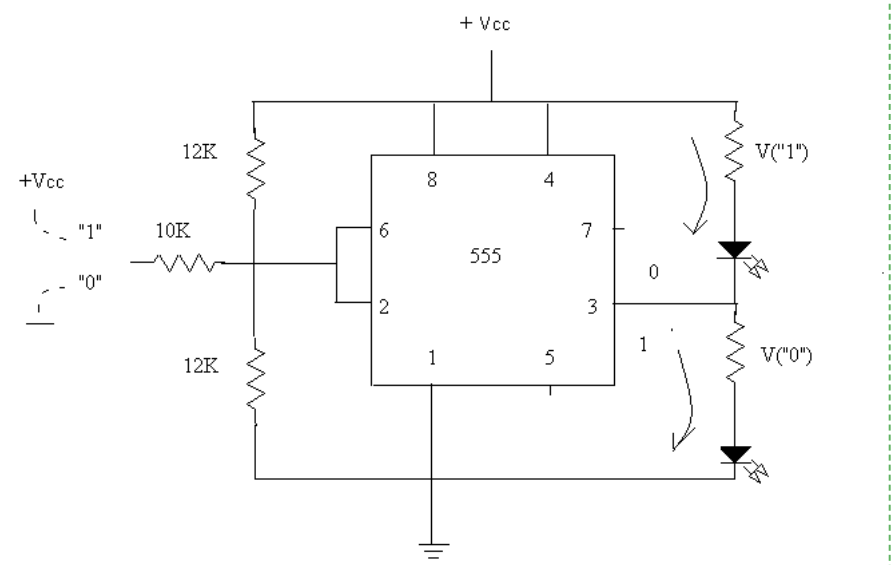
Figura 5.8 *Circuito y diagrama de la tensión de salida en función de la tensión de entrada para el comparador con histéresis*



5.4.2. Punta lógica

Este circuito, mostrado en la Figura 5.9, permite medir el nivel lógico de un circuito digital. Si la tensión está por encima de cierto valor, indicará un nivel alto, y si está por debajo de dicho valor, indicará un nivel bajo a la salida.

Figura 5.9 *Punta lógica implementada con el integrado 555*



5.4.3. Generador de pulsos rectangulares

Este es un circuito que entrega una señal rectangular de amplitud, aproximadamente igual a la tensión de alimentación ($V_0 \sim Vcc$), donde la frecuencia y ciclo de actividad dependerán de la configuración del capacitor y de las resistencias que constituyen la red de carga y descarga de dicho circuito, que se muestra en la Figura 5.10 y el estado de la salida en la Figura 5.11.

El ciclo de actividad se define como la relación entre el tiempo que el oscilador se encuentra en estado alto, respecto a la duración del ciclo total. Es decir:

$$\delta = \frac{T_A}{T} \quad \textbf{(5.18)}$$

Donde además $T = T_A + T_B$ como se ve en la Figura 5.11.

Si se desea diseñar un circuito como este, sabiendo la frecuencia y el ciclo de actividad, se puede adoptar un valor adecuado de capacitor, y obtener el valor necesario de la resistencia despejado de las ecuaciones 5.16 y 5.16, T_A o T_B según corresponda para el valor de carga o descarga.

Figura 5.10 *Circuito oscilador con el integrado 555*

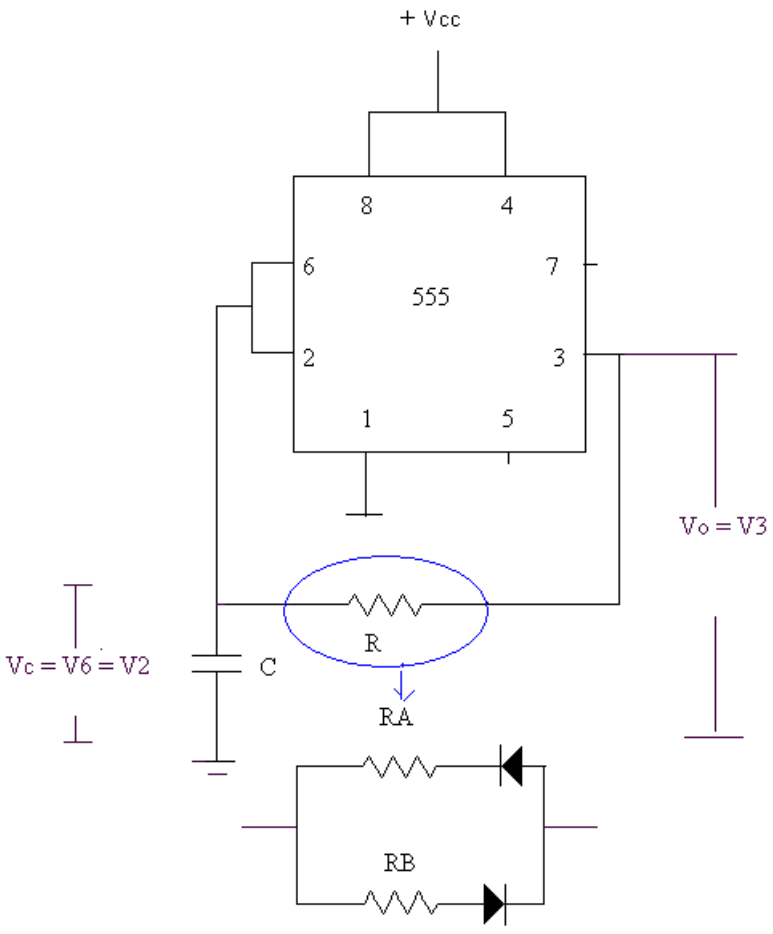
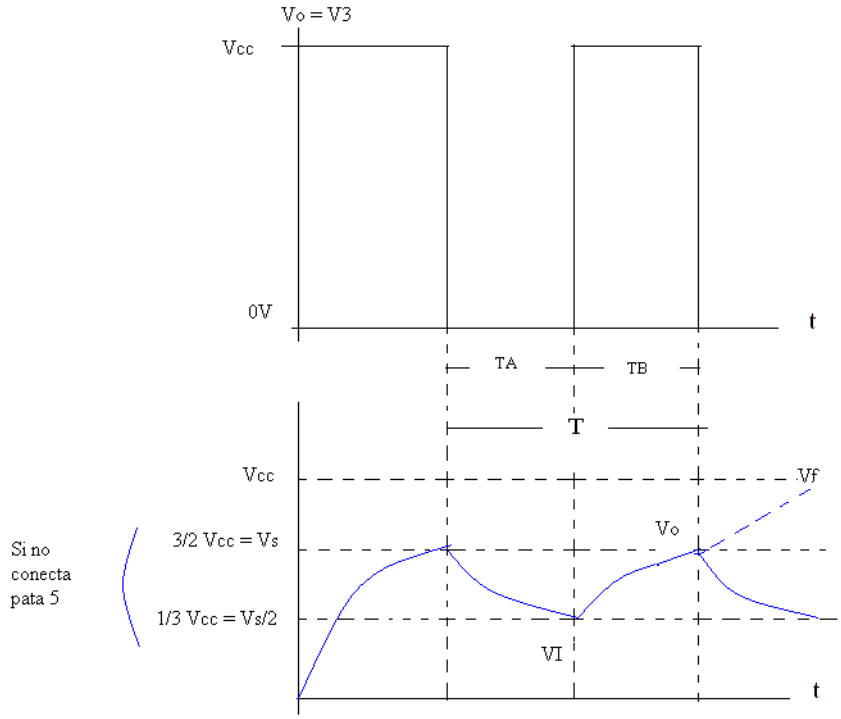


Figura 5.11 *Diagrama temporal para la salida y el estado del capacitor en función del tiempo para el oscilador con integrado 555*



5.5. Ejercicios propuestos

1) Diseñar, calcular y simular los siguientes osciladores. Adoptar el capacitor de 0,10μf en cada caso:

a) F = 10 kHz; δ = 1/3; V_o = 10 V

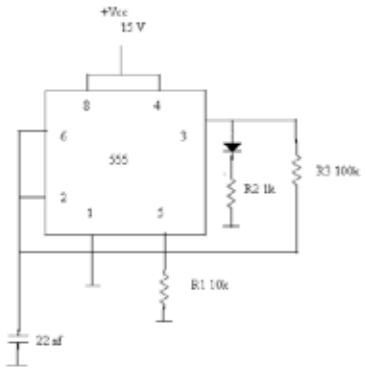
b) F = 1 kHz; δ = 1/2; V_o = 12 V

c) F = 1 kHz; δ = 2/3; V_o = 10 V

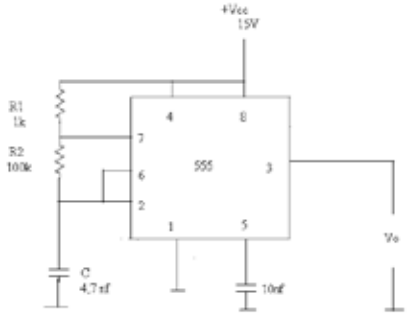
2) Repetir los cálculos ahora suponiendo que el capacitor es de 10μf ¿Que observa? ¿Por qué?

3) Calcular para los siguientes circuitos los valores de T_A,T_B, T, δ y obtenga los gráficos de V_O y de la tensión sobre el capacitor en función del tiempo, así como en la Figura 5.11.

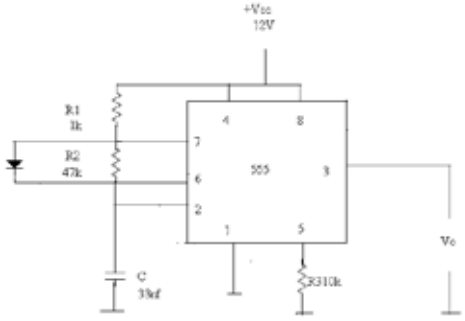
a)



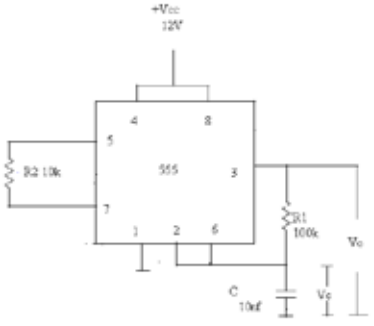
b)

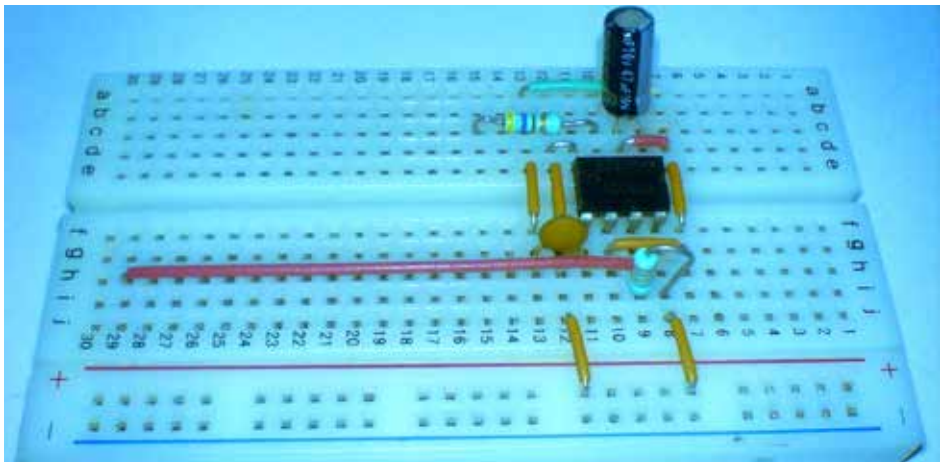


c)



d)





5.6. Práctica de laboratorio o simulaciones propuestas

1) Para el integrado 555, busque y describa la distribución de patas y especificaciones en el manual.

2) Armar en el protoboard el comparador con histéresis. Usar el osciloscopio para obtener V_0 vs V_i (la curva de transferencia).

3) Armar y probar el circuito de la punta lógica.

4) Diseñar (desde el cálculo hasta la prueba en el protoboard) los siguientes osciladores:

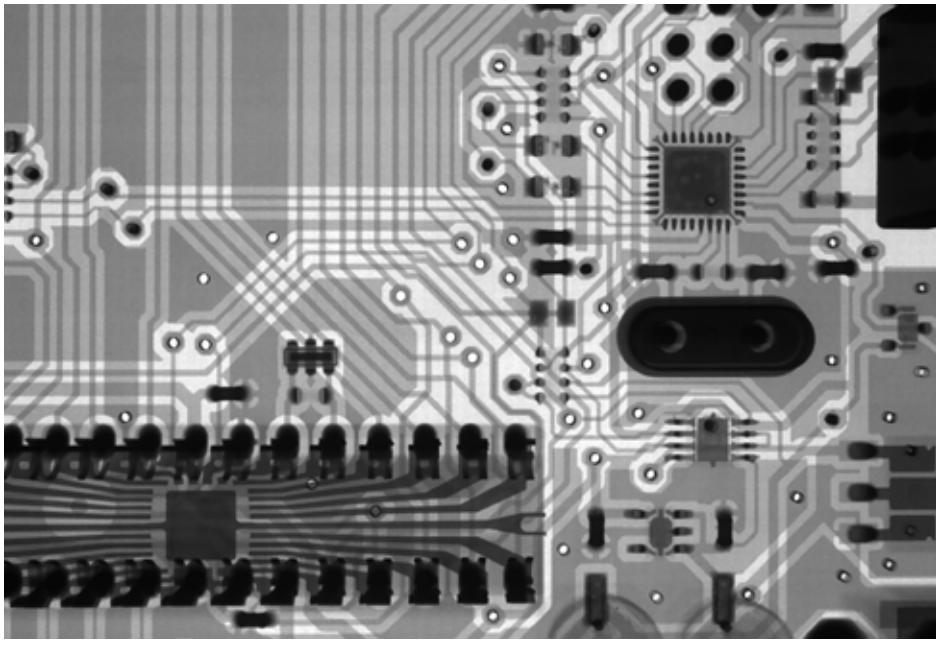
a) F = 10 kHz; δ = 1/3; V_o = 10 V (adoptar el capacitor que tenga disponible)
b) F = 1 kHz; δ = 1/2; V_o = 12 V
c) F = 1 kHz; δ = 2/3; V_o = 10 V

5.7. Proyecto: Temporizador para luz de pasillo

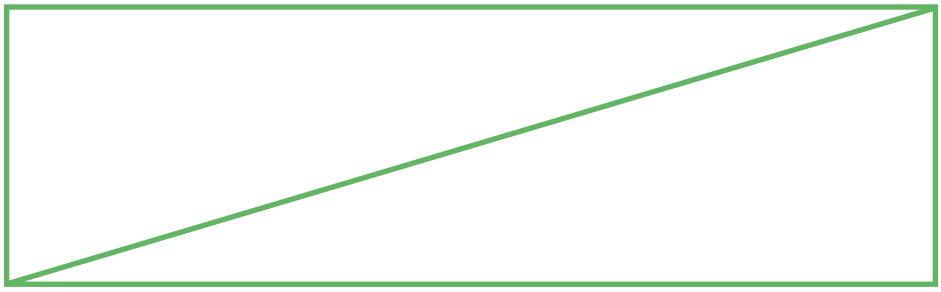
Actividad grupal en el marco de un curso de circuitos secuenciales, posterior a la realización de la práctica de laboratorio y al aprendizaje del uso del manual u hoja de datos. Se aconseja que esta actividad sea guiada por el docente.

Siguiendo la idea del recorrido ficticio por el edificio, que se presentó en el ítem 2 de la INTRODUCCIÓN, se propone como ejercicio el diseño de un temporizador para iluminar un pasillo durante 55 segundos. Se deberán calcular los valores de resistencias adecuados, adoptar un capacitor y controlar el disparo del biestable.

Capítulo 6
FAMILIAS DE TECNOLOGÍAS LÓGICAS



"We are family, I got all my sisters with me...".
"Somos una familia, tengo todas mis hermanas aquí conmigo...".
Sister Sledge



Una familia lógica puede referirse a uno de dos conceptos relacionados. Una familia lógica de dispositivos de circuitos integrados digitales es un grupo de compuertas lógicas electrónicas construidas utilizando uno de varios diseños diferentes, generalmente con niveles lógicos compatibles y características de suministro de energía dentro de una familia. Muchas familias lógicas se produjeron como componentes individuales, cada una con una o unas pocas funciones lógicas básicas relacionadas, que podrían usarse como "bloques de construcción" para crear sistemas o como el llamado "conexión" para interconectar circuitos integrados más complejos.

Una familia lógica también puede referirse a un conjunto de técnicas utilizadas para implementar la lógica dentro de los circuitos integrados, como procesadores centrales, memorias u otras funciones complejas. Algunas de estas familias lógicas utilizan técnicas estáticas para minimizar la complejidad del diseño. Otras familias lógicas, usan técnicas dinámicas sincronizadas para minimizar el tamaño, el consumo de energía y los tiempos de propagación.

En este texto abrazaremos el primer concepto que es el que nos permitirá ampliar nuestras capacidades prácticas en el laboratorio. La familia lógica más común en los dispositivos semiconductores modernos es la lógica de metal-óxido-semiconductor (MOS), debido al bajo consumo de energía, los pequeños tamaños de transistores y la alta densidad de transistores, y es la que a lo largo del libro hemos sugerido para su uso.

Los circuitos integrados digitales están diseñados utilizando cualquiera de los dispositivos bipolares o MOS o una combinación de ambos. Las familias lógicas que pertenecen al primer tipo se denominan familias bipolares, esto incluye lógica de diodo (DL), lógica de acoplamiento emitido (ECL), lógica de transistor de resistencia (RTL), lógica de transistor de diodo (DTL), lógica de transistor de transistor (TTL).

Los miembros de otra familia lógica, es decir, la familia MOS son PMOS, familia NMOS, familia CMOS. Ahora, la familia lógica Bi-MOS es la que utiliza dispositivos bipolares y MOS.

Consideremos, por ejemplo, las familias CMOS y TTL, ya que son las más comunes. Estos agrupamientos están divididos en subfamilias:

1) Subfamilias TTL: la familia TTL consta de varias subfamilias como TTL estándar, TTL de baja potencia, TTL de alta potencia, TTL de Schottky de baja potencia,TTL de Schottky, TTL de Schottky avanzado de baja potencia, TTL de

Schottky avanzado y TTL rápido. Los circuitos integrados que pertenecen a la familia TTL se designan de la siguiente manera: 74 o 54 para TTL estándar, 74L o 54L para TTL de baja potencia, 74H o 54H para TTL de alta potencia, 74LS o 54LS para TTL schottky de baja potencia, etc.

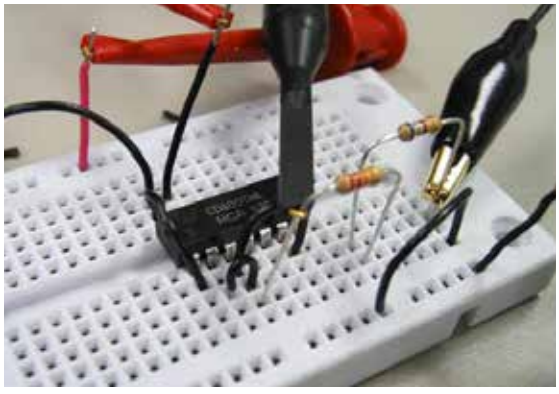
2) Subfamilias CMOS: esta es una familia lógica popular que incluye las familias 4000A, 4000B, 4000UB, 54 / 74C, 54 / 74HC, 54 / 74HCT, 54 / 74AC y 54 /74ACT.

Las subfamilias se dividen en función de la diferencia de voltaje y otros parámetros.

Imagen 6.1 Subfamilia CMOS

Imagen 6.2 Subfamilia TTL

La serie 7400 de circuitos integrados fue una de las familias lógicas más populares de chips lógicos transistor-transistor logic (TTL). En 1964, Texas Instruments presentó a los primeros miembros de su serie de encapsulados de semiconductores de cerámica, los SN5400. En 1966 se introdujo un encapsulado de plástico de bajo costo de la serie SN7400 que rápidamente ganó más del 50 % del mercado de chips lógicos. Eventualmente se convirtió en componentes electrónicos estandarizados. A lo largo de las décadas, muchas generaciones de familias "descendientes" compatibles evolucionaron para incluir soporte para conectar tecnología CMOS de baja potencia.



La serie 4000 de integrados fue presentada como la serie CD4000 COS/MOS en 1968 por RCA, como una alternativa de menor potencia y más versátil a la serie 7400 de chips de lógica transistor-transistor (TTL). Las funciones lógicas se implementaron con la nueva tecnología Complementary Metal-Oxide-Semiconductor (CMOS). Aunque inicialmente se comercializó con el etiquetado "COS/MOS" de RCA (que significaba semiconductor de óxido de metal de simetría complementaria), la terminología CMOS más corta surgió como la preferencia de la industria para referirse a la tecnología. Los primeros chips de la serie fueron diseñados por un grupo dirigido por Albert Medwin.

La adopción más masiva de esta tecnología se vio obstaculizada inicialmente por las velocidades comparativamente más bajas de los diseños en comparación con los diseños basados en TTL. Las limitaciones de velocidad finalmente se superaron con los métodos de fabricación más nuevos, lo que dejó los chips TTL más antiguos para eliminarlos gradualmente. La serie se extendió a fines de los años setenta y ochenta con nuevos modelos que recibieron las designaciones 45xx y 45xxx, pero que los ingenieros aún consideran como parte de la serie 4000. En la década de 1990, algunos fabricantes (por ejemplo, Texas Instruments) portaron la serie 4000 a diseños más nuevos basados en HCMOS para proporcionar mayores velocidades.

6.1. Características

Consideremos ahora algunas de las características que distinguen a las distintas familias, que es necesario conocer para poder tener una lectura comprensiva de los manuales de dispositivos electrónicos. También son esenciales para poder usarlas a la hora de elegir el integrado que más se adecue a nuestro diseño y posibilidades. Veamos la lista de las más relevantes:

1) IOH_{MAX}: es la corriente de salida máxima para el estado alto. Si se aumenta el valor de esta corriente (disminuyendo la resistencia conectada a la salida), la tensión de salida disminuirá.

2) IOL_{MAX}: es la máxima corriente que puede tomar una salida en estado bajo para que se mantenga de tensión adecuado al estado bajo.

3) IIH_{MAX}: es la máxima corriente que toma una entrada al aplicarle estado alto.

4) IIL_{MAX}: es la máxima corriente que se requerirá absorber desde una entrada al aplicarle estado bajo.

5) FAN-OUT: es la cantidad de entradas de compuertas de la misma familia que se pueden conectar a la salida de una compuerta, y depende de la relación entre IOH_{MAX} o IOL_{MAX} y IIH_{MAX} o IIL_{MAX}.

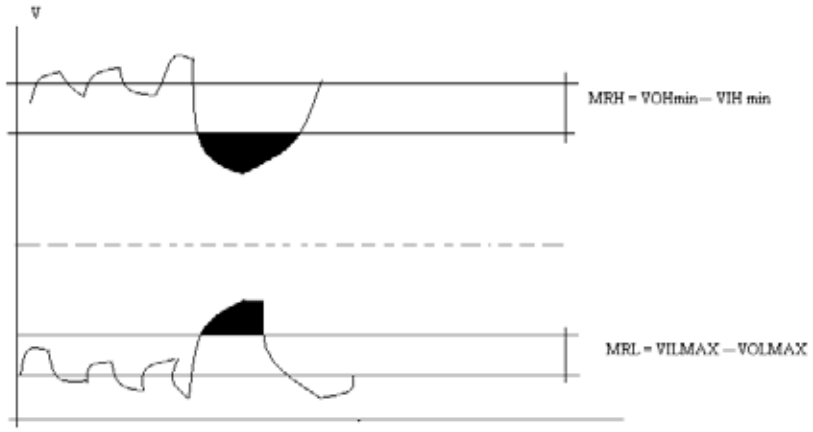
$$FAN-OUT = \frac{IOH_{MAX}}{IIH_{MAX}} = \frac{IOL_{MAX}}{IIL_{MAX}} \quad \textbf{(6.1)}$$

6) **Margen de ruido**: es la diferencia entre la tensión que requiere una entrada y la que le suministra una salida de la misma familia tanto en estado bajo como en estado alto.

La familia CMOS poseen un margen de ruido mayor al de los TTL, por ejemplo.

Un gráfico para explicar este concepto se observa en la Figura 6.1.

Figura 6.1 *Familias lógicas. Definición del margen de ruido*



7) Icc: es la corriente que consume cada compuerta desde la fuente de alimentación.

8) Tp: tiempo de propagación:

Tp_{HL}: tiempo que tarda la salida en conmutar del estado alto al bajo. Se mide desde que la señal de entrada está al 50 % hasta que la salida está al 50 % de su valor.

Tp_{LH}: tiempo que tarda en conmutar la salida del estado bajo al alto.

6.2. Conexión de integrados de distintas clases

Partimos del supuesto que, por las propiedades del diseño que debamos realizar, o bien por el hecho de tener a disposición stock de determinados integrados, surge la necesidad de interconectar integrados de distintas familias tal y como se ve en la parte superior de la Figura 6.2.

Existe en este caso una serie de condiciones que deberán cumplirse para poder hacerlo que indicamos a continuación.

$$V\,OH_{MIN1} > V\,IH_{MIN2}$$
$$V\,OL_{MAX1} < V\,IL_{MAX2}$$
$$IOH_{MAX1} > IIH_{MAX2}$$
$$IOL_{MAX1} > IIL_{MAX2}$$

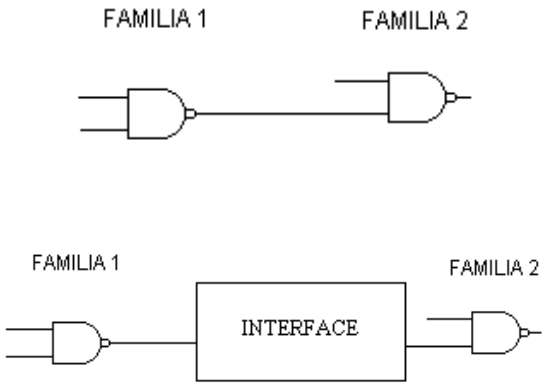
Aquí “1” representa la familia de la cual se está tomando la señal de salida por una segunda familia “2”.

Si se cumplen estas verificaciones se pueden conectar directamente entre si las compuertas de distintas familias. Si alguna de las cuatro relaciones no se cumple, no es posible realizar la conexión directa y si fuera imprescindible se debe utilizar un circuito intermedio denominado interfaz para realizar la adaptación que se requiera.

Dependiendo del caso la interfaz requerida puede ser una simple resistencia de *pull up* o de *pull down*, o una etapa con un transistor adecuadamente polarizado, o alguna compuerta especial para actuar de interfaz cuya entrada es compatible con una familia y su salida es compatible

con otra familia. Por lo general en los apéndices de los manuales figuran los métodos aconsejados para las interfaces con las distintas familias.

Figura 6.2 *Arriba: Interconexión directa de compuertas de distintas familias. Abajo: Interconexión de compuertas a través del uso de interfaz*



6.3. Ejercicios propuestos

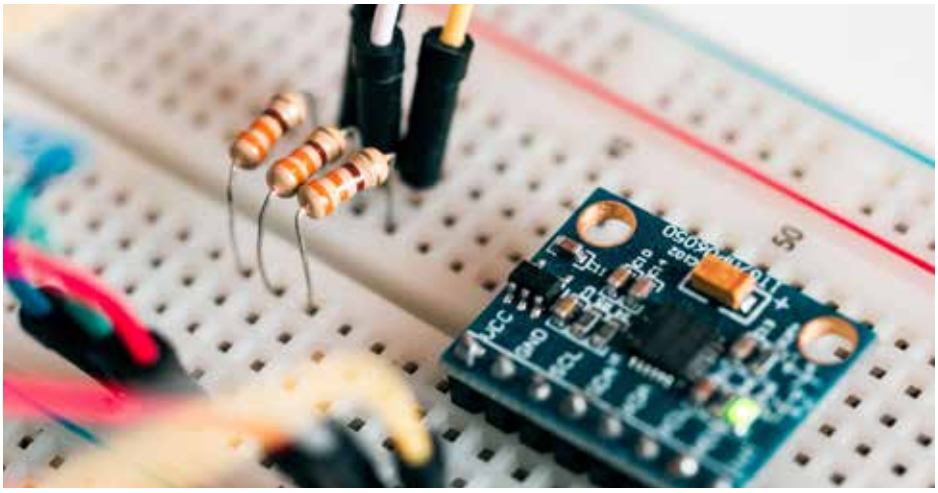
Esta actividad es para realizar con los respectivos manuales de integrados.

1) Estimar los márgenes de ruido para las familias TTL (74XX) y CMOS (40XX).

2) Encontrar los valores para los tiempos de propagación de cada familia, para la la misma funcionalidad de integrados, por ejemplo, compuertas AND y diga cual es la diferencia porcentual en tiempos.

3) Verificar si se pueden interconectar TTL (74XX) y CMOS (40XX) como familia 1 y 2 respectivamente y luego invierta el orden.

4) Calcular el FAN-OUT de ambas familias, TTL (74XX) y CMOS (40XX).

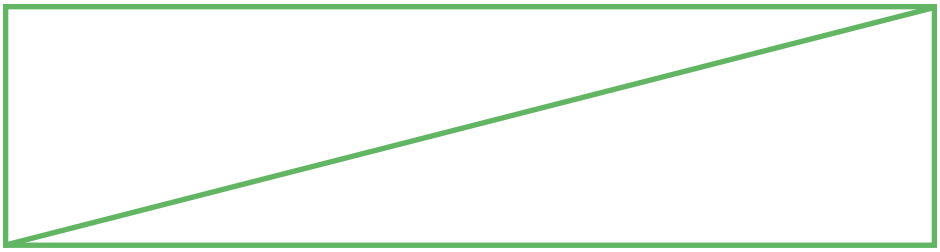


6.4. Práctica de laboratorio o simulaciones propuestas

Proponer dos circuitos de interconexión entre las dos clases de integrados TTL (74XX) y CMOS (40XX), o a partir de los materiales disponibles y discutir cuál es la mejor alternativa y el porqué.

Capítulo 7
DIGITALIZACIÓN DE SEÑALES Y CONVERSIÓN

"¡Qué terrible pesadilla! Unos y ceros por todas partes... ¡Hasta me pareció ver un 2!
Era un sueño, Bender, no existe eso que llamas 2…".
Futurama, pesadilla del robot Bender.



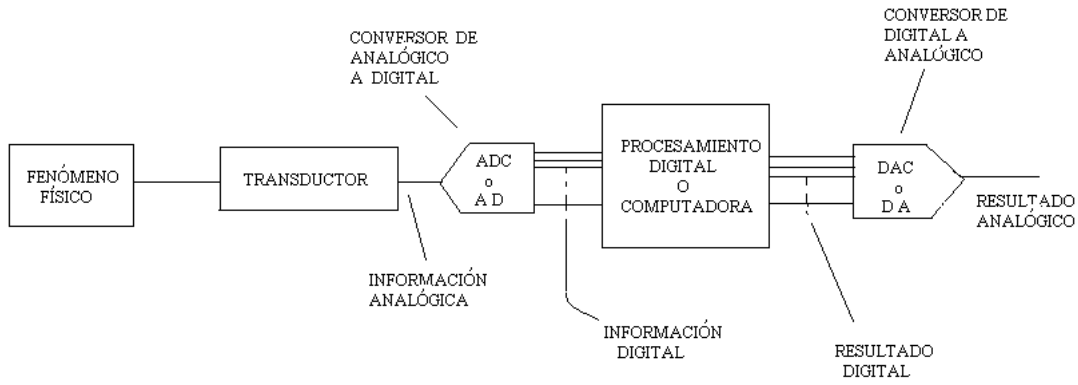
La naturaleza de los fenómenos físicos que se encuentran a nuestro alrededor es de carácter analógico. La información que se obtiene de dichos fenómenos mediante transductores también lo es. Es habitual digitalizar esta información para procesarla en un circuito digital o en una computadora. Los resultados digitales así obtenidos se utilizan directamente, o bien se almacenan o también se los puede volver a convertir a la forma analógica.

A lo largo de este capítulo se presentan detalles del proceso de digitalización de las señales analógicas, así como también el modo de recuperar la información digitalizada para luego, por ejemplo, controlar procesos analógicos. Los conversores digitales analógicos y analógicos digitales son circuitos clave en una gran cantidad de sistemas, no solo en sistemas de control.

La Figura 7.1 muestra el esquema en bloques más general del proceso de digitalización de una señal analógica y recuperación de una señal analógica a partir de una señal digital de dicho proceso.

Nos concentraremos primero, por simplicidad, en la descripción de la etapa de conversión de una digital a una señal analógica.

Figura 7.1 *Diagrama en bloques del proceso de digitalización de una señal analógica y la recuperación de esta a partir de la señal digitalizada*



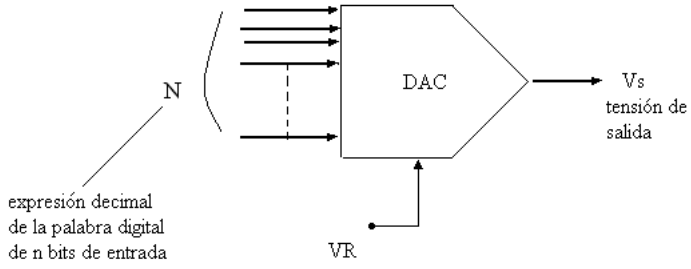
7.1. Conversor digital analógico o DAC

Estos dispositivos llevan a cabo el proceso correspondiente al bloque final de etapas que se muestra en la Figura 7.1. La Figura 7.2 presenta un esquema de esta etapa. Allí se observan N canales de entrada, cada uno correspondiente a un bit que constituye la palabra digital. Un valor o tensión de referencia V_R nos permite realizar la conversión a un rango apropiado de tensiones analógicas con una cierta resolución, mientras que dicha tensión se presenta en el canal de salida a la derecha de la figura.

La expresión decimal correspondiente a la palabra digital de la señal de entrada viene dada por la ecuación:

$$N = b_0 2^0 + b_1 2^1 + ... + b_i 2^i + ... + b_{n+1} 2^{n-1} \quad \textbf{(7.1.)}$$

Figura 7.2 *Diagrama con el bloque para el proceso de un conversor digital-analógico*



Donde los términos b_i representan a los bits del número binario o palabra digital *N* y los términos 2^i representan a las potencias enteras de la base 2.

La función del conversor digital analógico o "DAC" es la de entregar una tensión de salida proporcional al número binario de entrada, esto se manifiesta en la expresión de la ecuación 7.1.

$$\text{Si } Vs = \frac{V_R}{2^n} N \text{ , entonces } N = \frac{2^n}{V_R} Vs \quad \textbf{(7.2)}$$

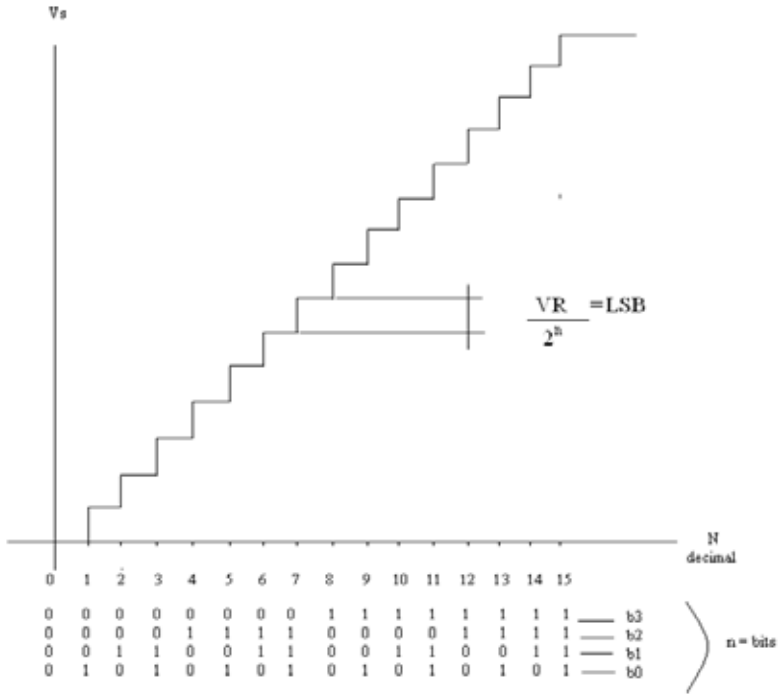
Reemplazamos a *N* en la ecuación 7.1 y pasando V_R y 2^n al segundo miembro obtenemos la ecuación 7.3:

$$Vs = \frac{V_R}{2^n} N \ (b_0\, 2^0 + b_1\, 2^1 + ... + b_i\, 2^i + ... + b_{n+1}\, 2^{n-1}) \quad \textbf{(7.3)}$$

La palabra digital N puede tomar 2^n valores distintos, de manera que $\frac{V_R}{2^n}$ representa a la variación de tensión que produce la variación en una unidad en *N*. Así si *N* vale cero, V_S será cero, si vale 2^n (que es su máximo valor) V_S se igualará a V_R, y para valores intermedios de *N* y V_S tomará valores intermedios de forma proporcional.

Por ejemplo, si graficamos a V_S en función de *N* para n = 4 bits, tendremos un gráfico como el que se muestra en la Figura 7.3.

Figura 7.3 *Tensión de salida y sus valores digitales correspondientes*



Se llama resolución a la cantidad de bits de la palabra digital de entrada, por ejemplo, un DAC de cuatro pines de entrada tiene cuatro bits de resolución.

Si un DAC tiene n bits de resolución la variación más pequeña de la tensión de salida es de $\frac{V_R}{2^n}$ y se producirá al variar en una cuenta (incrementar o decrementar en una unidad al bit menos significativo) de la palabra digital de entrada. Se llama a esto variación de un LSB (LSB: least significant bit). Entonces cuando N varía en un LSB, varía en $\frac{V_R}{2^n}$ de tensión.

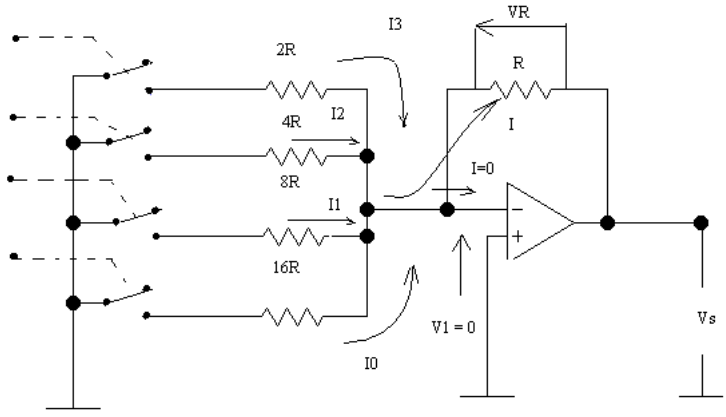
A continuación, presentamos algunos ejemplos de circuitos clásicos que permiten obtener la función de un conversor DAC y se pueden armar o simular sencillamente.

7.1.1. DAC o red de abanico o de corrientes pesadas

Consiste en un circuito constituido por un amplificador operacional puesto en la configuración sumador como se muestra en la Figura 7.4.

Una dificultad para realizar este circuito es encontrar el conjunto de resistencias de valor comercial con la relación de valores adecuados. Además, dependiendo de la ubicación, se debe ser cuidadoso con el tratamiento del error estático del operacional.

Figura 7.4 *Circuito conversor digital-analógico con amplificador operacional, ejemplo para cuatro bits*



Escribamos las ecuaciones que describen el estado de la salida en término de los valores de la palabra digital. Comencemos planteando las leyes de Kirchoff para el nodo correspondiente al pin negativo del amplificador operacional:

$$I = I_0 + I_1 + I_2 + I_3 \quad \textbf{(7.4)}$$

Donde cada corriente expresada en el miembro de la igualdad de la derecha corresponde respectivamente a la corriente que pasa por cada una de las ramas de entrada.

Ahora sabemos que la salida del amplificador dada por: V_S está dada por:

$$V_S = -I \cdot R \quad \textbf{(7.5)}$$

De modo que sustituyendo I obtenemos:

$$V_S = -R \cdot (I_0 + I_1 + 1_2 + I_3) \quad \textbf{(7.6)}$$

Esto implica que la circulación de corriente I_i en cada rama, que denominamos i, dependerá de que la llave esté abierta o cerrada. Si a cada llave le asignamos el valor b_i, donde b_i puede ser 0 o 1, la corriente por cada rama se podrá escribir como:

$$I_i = b_i \frac{V_R}{R_i} \quad \textbf{(7.7)}$$

Donde V_R será la alimentación común o referencia que recibirá cada resistencia.

Si la llave está cerrada y R_i es la resistencia de cada rama, que cumple la relación: 2R, 4R y 8R y 16R. La salida V_S se puede escribir como:

$$Vs = -R\left(b_0 \frac{V_R}{16R} + b_1 \frac{V_R}{8R} + b_2 \frac{V_R}{4R} + b_3 \frac{V_R}{2R}\right) \quad \textbf{(7.8)}$$

en la ecuación (7.8) sacamos factor común $\frac{1}{16R}$

$$V_S = -R \frac{1}{16R} (b_0 V_R 2^0 + b_1 V_R 2^1 + b_2 V_R 2^2 + b_3 V_R 2^3) \quad \textbf{(7.9)}$$

Simplificamos R y expresamos 16= 2^4, obtenemos:

$$V_S = -\frac{1}{2^4}\,(b_0\,V_R\,2^0 + b_1\,V_R\,2^1 + b_2\,V_R\,2^2 + b_3\,V_R\,2^3) \quad \textbf{(7.10)}$$

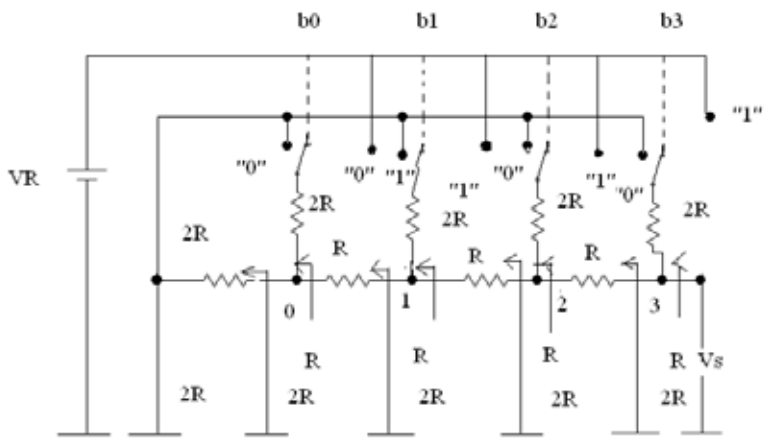
En general, para n ramas, correspondientes a n bits de resolución, quedaría la expresión de la ecuación 7.3.

7.1.2. DAC o red en escala o R-2R

Este circuito, como el que se muestra en la Figura 7.5, posee una configuración muy sencilla y además constituye un ejemplo muy ilustrativo para aprender a calcular un circuito con múltiples divisores de tensión.

El análisis es análogo al que hicimos en el circuito de la sección anterior, de manera que se deja como ejercicio para el lector.

Figura 7.5 *Circuito DAC o red en escala o R 2R*



7.2. Conversor analógico digital o ADC

Ahora nos concentramos en el proceso inverso al que estudiamos en la sección 7.1. Sabemos que las señales analógicas, pueden tomar infinitos valores intermedios, mientras que la señal digital solo una cantidad discreta, y en particular solo dos valores si es binaria.

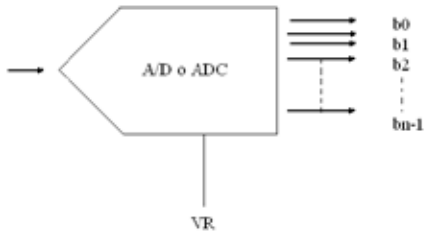
En definitiva, los ADC convierten variables analógicas en palabras digitales, como se muestra en el esquema de la Figura 7.6. En este caso, la tensión de entrada a convertir la llamaremos V_X y la salida será una palabra digital *N*, de n bits de resolución, de manera que:

$$N = 2^n \frac{V_X}{V_R} \quad \textbf{(7.11)}$$

Donde nuevamente V_R es la tensión de referencia del circuito.

Existen varios ejemplos de estos circuitos y los describiremos en la sección siguiente.

Figura 7.6 *Esquema general de un Conversor Analógico Digital*



7.3. Conversores analógico-digitales indirectos

Estos circuitos se caracterizan por tener un conversor digital analógico con un circuito de realimentación. Existen tres tipos principales: conversor tipo contador, conversor contador servo por arrastre y conversor por aproximaciones sucesivas.

Dependiendo del tipo de circuito se puede obtener o mejorar distintas características tales como resolución o velocidad de conversión. A continuación, mostraremos algunos ejemplos de dichos circuitos detallando su funcionamiento y sus características.

Veremos el diagrama en bloque general de cada tipo y daremos algunos circuitos a modo de ejemplo en cada caso.

7.3.1. Conversor contador

La Figura (7.7) muestra el diagrama en bloques para un conversor contador. En estos circuitos al aplicar un pulso de RESET al contador se le induce a éste a que comience en el estado 0 y se inicie la cuenta de los pulsos de clock. Es el contador quién dará la orden para el inicio de conversión.

La salida del contador V_S es convertida nuevamente a la señal analógica por el DAC. Esta señal analógica se compara con una señal de referencia, V_X. Mientras la señal analógica V_S sea menor que V_X, la cuenta continuará, ya que la salida del comparador será el nivel alto “1” y que habilitará el clock. Cuando V_S alcance el valor de V_X y lo supere, el comparador devolverá el nivel bajo “0” inhibiendo el clock del contador, por lo tanto, quedará detenido. En este momento las salidas del contador representan la palabra digital de salida resultante de la conversión.

En un conversor contador el estado se incrementa por pasos discretos, por lo que a lo sumo el estado de la salida puede excederse en una cuenta del bit menos significativo (LSB).

A esta diferencia se la llama **error de cuantificación**.

Para reducir el error de cuantificación a la mitad, se puede modificar el circuito de la figura de manera de sumar una tensión equivalente a medio bit menos significativo dentro del lazo de realimentación.

Algunas características generales de este conversor son las siguientes:

- Permite medir el valor instantáneo de la señal de entrada.
- Es relativamente lento ya que tiene que esperar que el contador efectúe la cuenta.
- Es sensible al ruido que tenga superpuesta la señal de entrada.

Algunos ejemplos de implementación de este circuito se observan en la Figura 7.8 y en la Figura 7.9, donde se utiliza el integrado CMOS 4516 y 4520 respectivamente. En el caso del primer circuito se aprovecha la función de PRESET del 4516 para mantener fijo el estado final de la cuenta (resultado de la conversión) hasta que llegue la nueva orden de conversión. En el caso del segundo circuito se detiene la cuenta cuando la señal de salida alcanza la mitad de la referencia sobre la entrada Enable del 4520, manteniéndose este estado hasta nueva orden de conversión.

Figura 7.7 *Diagrama en bloque para un conversor contador*

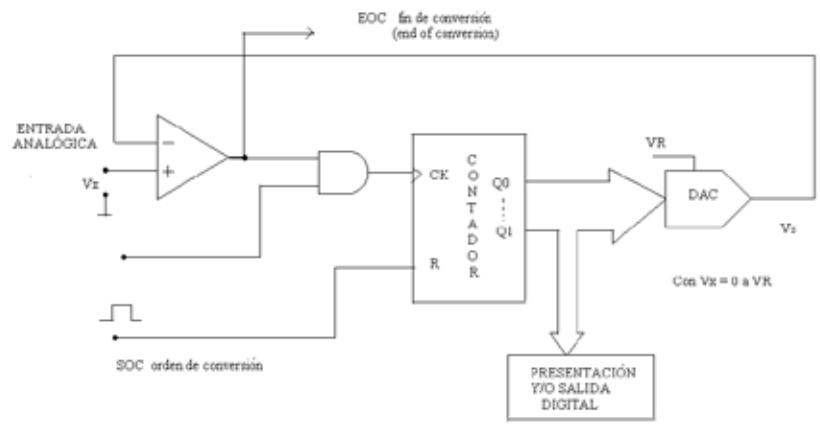


Figura 7.8 *Conversor contador con integrado 4516*

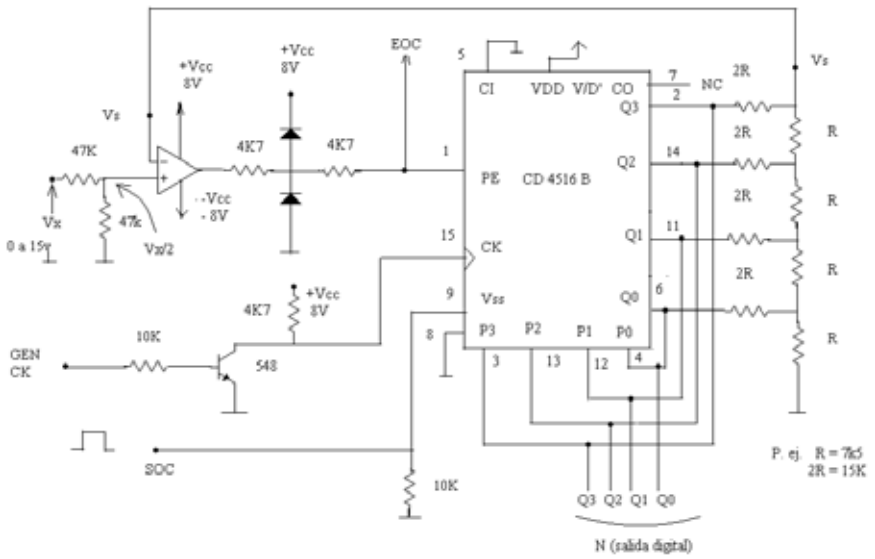
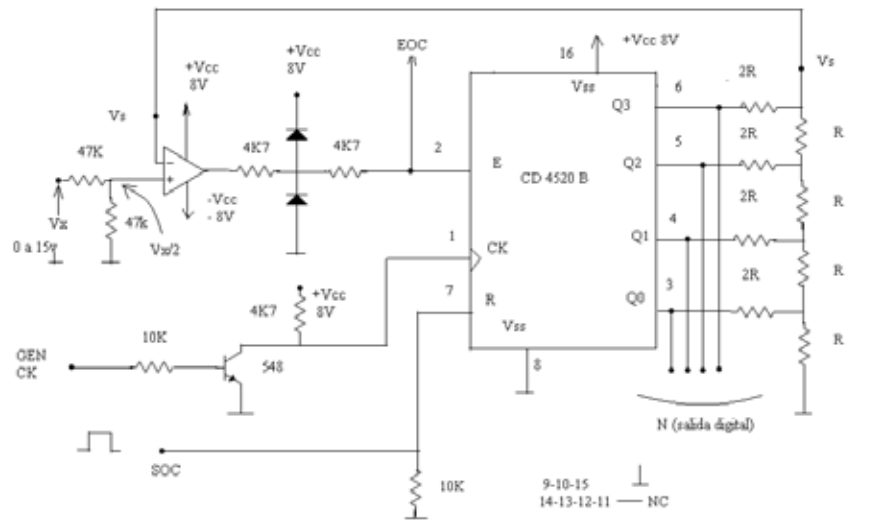


Figura 7.9 *Conversor contador con integrado 4520*



7.3.2. Conversor analógico digital por arrastre (tracking) o servo contador

El contador es controlado para la cuenta ascendente o descendente dependiendo de la salida del comparador. Si inicialmente la tensión de referencia V_X es mayor que la tensión V_S, el contador actúa en sentido ascendente hasta que la tensión V_S sea mayor que la tensión de referencia V_X. En este punto se invierte el sentido de la cuenta. Así la tensión V_S subirá y bajará a través de la referencia buscando la igualdad o por lo menos la mejor aproximación. Este es el mismo principio de los sistemas servo. Un diagrama en bloques para este circuito se observa en la Figura 7.10.

Este contador puede seguir un cambio continuo de la señal de entrada, siempre que la velocidad de cambio de la señal no exceda la velocidad de respuesta del conversor.

Para pequeños cambios de la señal de entrada, este conversor es el más rápido, dado que no necesita volver a cero luego de cada conversión. Las características principales de estos circuitos son las siguientes:

- Es de valor instantáneo por lo tanto es sensible al ruido.
- Es utilizable en servomecanismos.
- Es necesario utilizar en estos circuitos un filtro pasa bajos en la entrada.

En la Figura 7.11 podemos observar un ejemplo de diseño de estos circuitos utilizando el integrado 4516.

Figura 7.10 *Diagrama en bloque del conversor por arrastre*

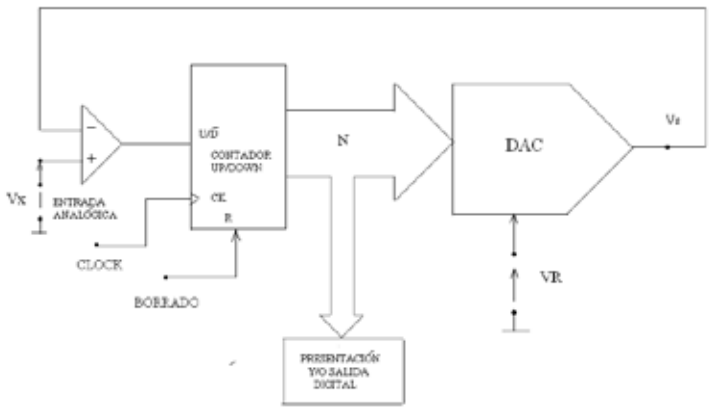
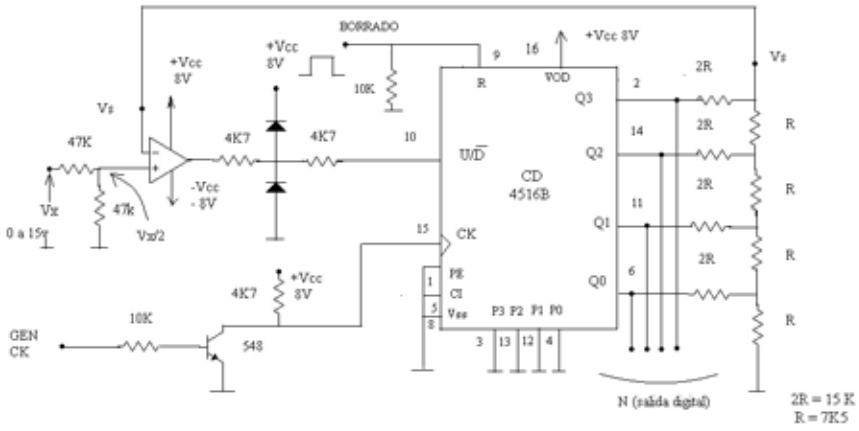


Figura 7.11 *Ejemplo de un circuito conversor por arrastre usando el 4516*



7.3.3. Conversor por aproximaciones sucesivas

Un diagrama en bloques del funcionamiento de estos circuitos se observa en la Figura 7.12. En este circuito, al recibir la orden de conversión, el registro coloca el bit más significativo en 1, como consecuencia de esta situación hay dos alternativas:

- Si la tensión V_S resultó mayor que la tensión de referencia dejará el bit más significativo en 1.
- Si la tensión V_S resultó menor que la tensión de referencia volverá el bit más significativo a cero.

Este proceso se repite en forma consecutiva con todos los bits del registro hasta llegar al menos significativo. Al alcanzar esta instancia, el registro da la orden de fin de conversión y salidas presentar al resultado.

Las características de la conversión por aproximaciones sucesivas son las siguientes:

- El resultado corresponde al valor instantáneo.
- Es sensible al ruido.

- Debe garantizarse que la tensión de referencia se mantenga constante durante el tiempo de conversión, por ejemplo, puede utilizarse un mecanismo de Sample and Hold.
- Es relativamente rápido si se lo compara con los conversores antes estudiados. Los tiempos de conversión típicos son de 100 µS dependiendo de la resolución.

El circuito de la figura 7.13 muestra un conversor por aproximaciones sucesivas que, es un circuito de prueba básico tomado del manual del ADC0804. Los diodos led están conectados para aprovechar la corriente, pero indican el resultado opuesto al de la conversión.

Figura 7.12 *Diagrama en bloque del conversor por aproximaciones sucesivas*

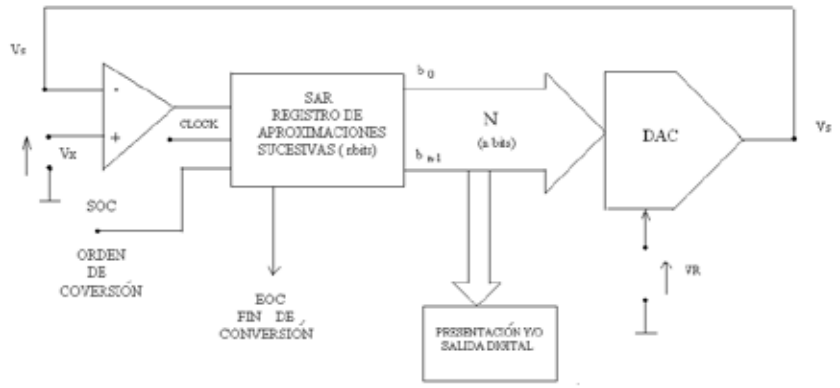
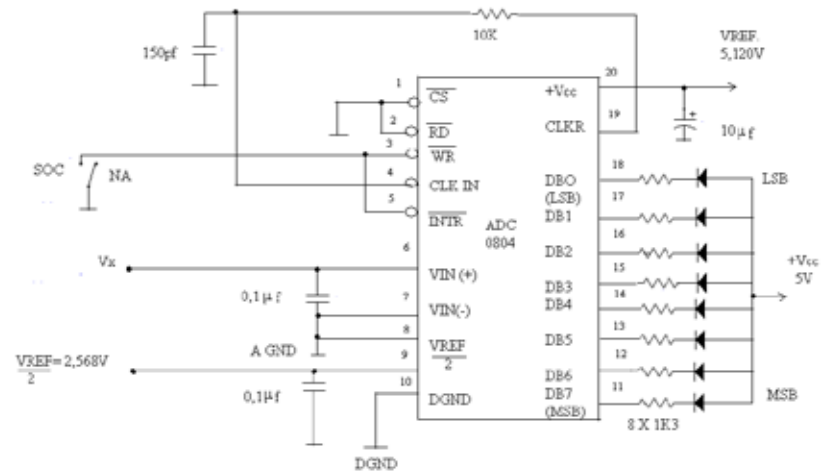


Figura 7.13 *Circuito de un conversor por aproximaciones sucesivas con el integrado ADC0804*



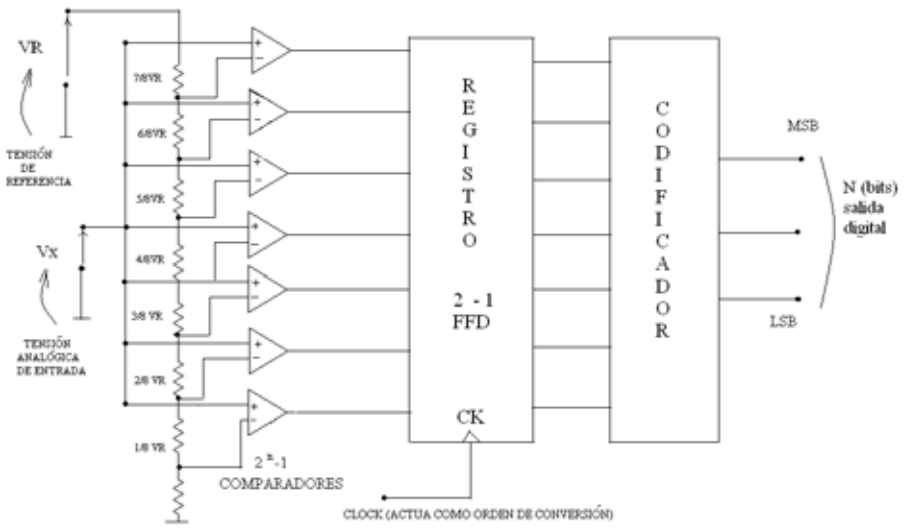
7.3.4. Conversor analógico digital por multi-comparación o conversión paralelo o flash

Para n bits de resolución, la V_x se compara con 2n−1 tensiones de referencia mediante 2n−1 comparadores. La salida de los comparadores es memorizada en un registro. Luego las salidas del registro son enviados al codificador que obtiene la palabra digital de salida *N*. Este circuito se muestra en la Figura 7.14.

El registro no es indispensable en todos los casos ya que las salidas de los comparadores podrían ir directamente al codificador, pero es conveniente cuando V_x varía rápidamente ya que permite "congelar" el estado en el momento del pulso de clock hasta que sea tomada la lectura de las salidas N. Veamos a continuación las características principales del ADC "FLSH":

- Es el conversor más rápido, ya que todo el proceso de conversión tiene lugar simultáneamente y no secuencialmente. Tiempos de conversión típicos de 20 Ns.
- El resultado corresponde al valor instantáneo que tenga V_x en el instante del pulso de clock.
- Es sensible al ruido.
- Una de sus aplicaciones típicas es la digitalización de señales de video.

Figura 7.14 *Diagrama en bloque de un conversor paralelo*



Los conversores flash comerciales aparecieron en instrumentos y módulos de los años 1960 y 1970 y migraron rápidamente a circuitos integrados durante la década de 1980. El ADC flash monolítico de 8 bits se convirtió en un estándar de la industria en aplicaciones de vídeo digital de la década de 1980. Hoy, el convertidor flash se usa en diversas aplicaciones.

7.3.5. Conversor por simple rampa

El diagrama en bloques para un ADC por simple rampa se puede observar en el gráfico de la Figura 7.15. Respecto del funcionamiento, el control del integrador mantiene el capacitor C cortocircuitado para que $V_S = 0$ durante el estado de reposo.

Al dar la orden de conversión, se abre la llave y V_S comienza a realizar una rampa ascendente (para esto es V_R negativa ya que el integrador es inversor). Al mismo tiempo que transcurre la rampa, el contador realiza su cuenta al ritmo de su clock, ya que la SOC lo reseteó. Cuando la rampa V_S se iguala con V_X, el comparador da "0" en la entrada de habilitación del contador.

Así este se detiene y en sus salidas queda el resultado *N*.

En resumen, su funcionamiento se basa en contar pulsos de CLOCK de frecuencia fija, durante un período de tiempo variable que es proporcional a la tensión analógica de entrada.

Respecto a los inconvenientes principales de este circuito son los siguientes:

- Es sensible al ruido.
- Es sensible a la temperatura.

La Figura 7.16 muestra la rampa de carga para V_S.

Figura 7.15 *Diagrama en bloques de un ADC por simple rampa*

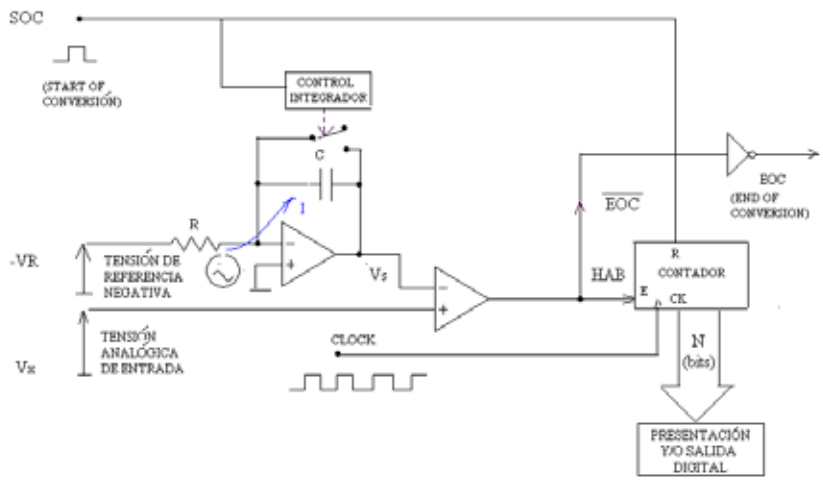
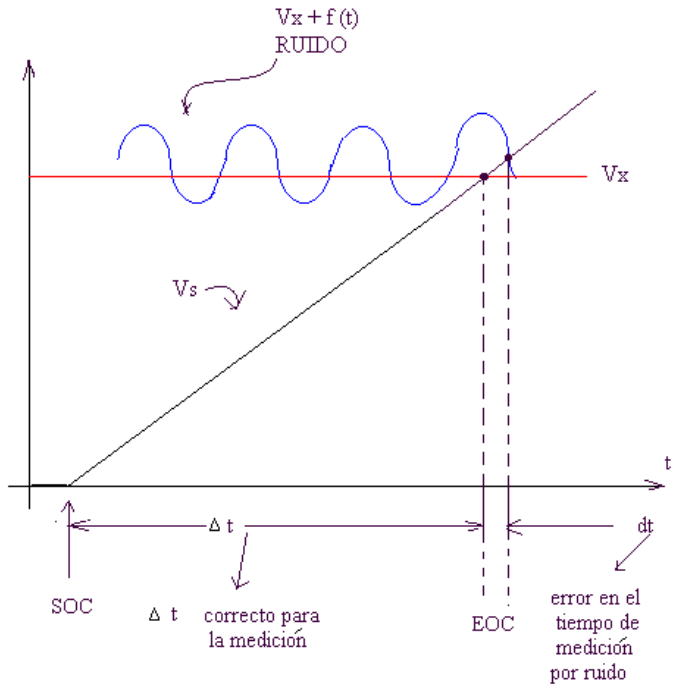


Figura 7.16 *Diagrama de la rampa de V_s y el ruido para el conversor por rampa simple*



7.3.6. Conversor por doble rampa

En el caso de este tipo de circuitos, podemos realizar la descripción de su funcionamiento a través de la siguiente secuencia operativa. La Figura 7.17 muestra el diagrama en bloques al que nos referiremos en dicha secuencia.

- Estado 1: se encuentra en reposo luego de haber terminado una conversión.

  - EOC (fin de conversión) = 1
  - Contador = N y deshabilitado
  - Llave S2 cerrada y C descargado → tensión e = 0
  - Llave S1 en V_x

  Si hay orden de conversión pasa a:

- Estado 2:

  - EOC = 0
  - Se resetea el contador y se habilita (primero N = 0 y listo a contar)
  - Se abre S2
  - Se genera rampa ascendente en e y transcurre un tiempo T de integración.
  - El contador cuenta pulsos de clock.

- CARRY → Cuando aparece CARRY (contador completo la cuenta completa y *N* = 0 nuevamente) se puede pasar al:

- Estado 3:

  - S1 cambia V_R
  - Se genera rampa descendente en e Transcurre Δt, tiempo de conteo.
  - Contador cuenta pulsos de clock e = 0 → Cuando el comparador detecta e = 0, terminó la conversión y se retorna al estado 1. *N* presenta el resultado.

Las características del ADC doble rampa se pueden resumir en las siguientes:

- Dado que el resultado N resulta independiente de R, C, y f_{ck} presenta mucha más exactitud respecto del conversor de simple rampa.
- Los errores por offset del comparador harán que este conmute a voltaje diferente de “0”, pero se compensan ya que la señal e cruza dos veces por “0”.

- Los errores por offset del integrador no son compensados y limitarán la exactitud. El resultado de la conversión corresponde al valor medio de V_x a causa del proceso de integración.
- Una desventaja de este conversor es la lentitud, no permitiendo más de 50 a 100 conversiones por segundo.
- Es el conversor más usado en multímetros digitales.

Figura 7.17 *Diagrama en bloques para el conversor por doble rampa*

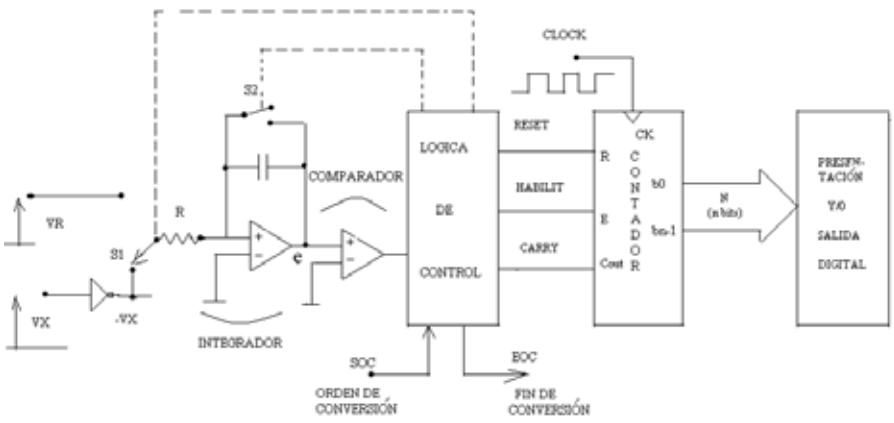
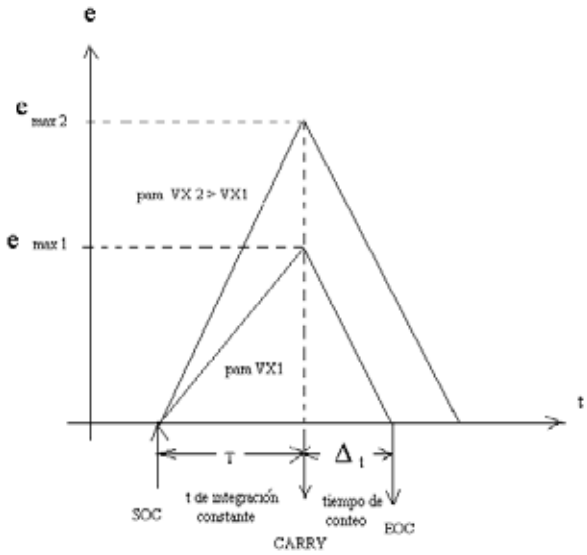


Figura 7.18 *Rampa de carga y descarga del conversor*



7.4. Aspectos generales de la digitalización de señales

La digitalización de señales es el proceso de convertir la información a un formato digital en el que la información se organiza en bits. El resultado es la representación de una señal (generalmente una señal analógica) generando una serie de números que describen un conjunto discreto de puntos o muestras.

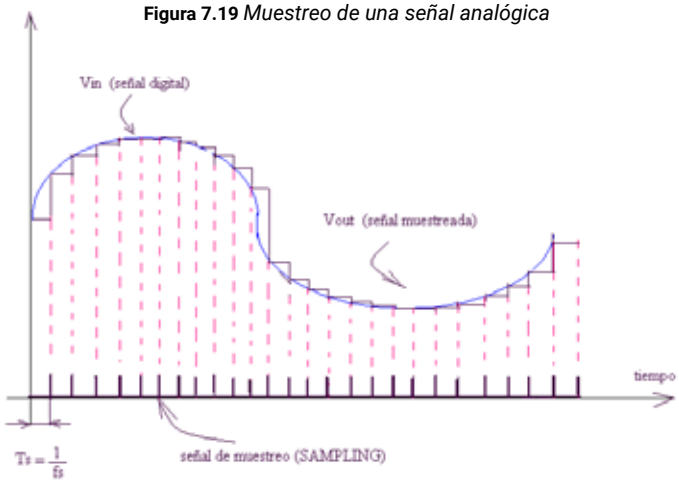
El resultado se llama **representación digital** para la señal. En la práctica los datos digitalizados están en forma de números binarios, pero, estrictamente hablando, la digitalización simplemente significa la conversión de material fuente analógico en un formato numérico.

La digitalización constituye una etapa crucial para el procesamiento, almacenamiento y transmisión de datos. Normalmente este proceso se divide en dos partes conocidas como: **muestreo y cuantificación.**

La cuantificación es el proceso que realiza el conversor analógico digital y que hemos estudiado en la sección 7.2. Es decir que consiste en asignar los valores de la palabra digital de salida que corresponden a los valores analógicos de entrada.

El muestreo es un proceso necesario cuando la señal analógica de entrada varía en forma relativamente rápida respecto del tiempo que tarda el ADC en hacer la conversión. Este proceso consiste en "congelar" el nivel de tensión de entrada durante pequeños intervalos de tiempo durante los cuales el ADC realiza las conversiones teniendo su señal de entrada constante. Este proceso se muestra con un ejemplo en la Figura 7.19.

Figura 7.19 *Muestreo de una señal analógica*

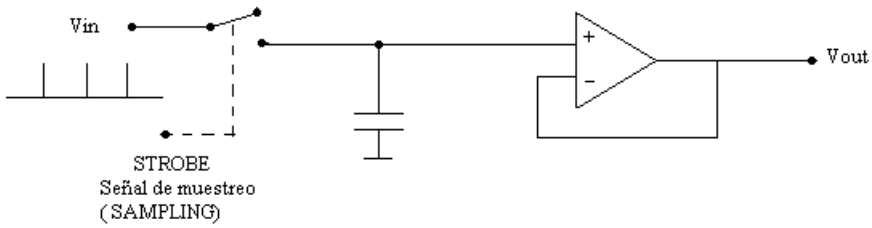


El muestreo se realiza mediante los circuitos llamados sample and hold.

Estos circuitos Sample and Hold (muestreo y retención, o adquisición y mandamiento), son un tipo de memoria analógica que almacenan en un capacitor de muy bajas pérdidas el valor de la tensión de entrada presente en el instante de tomar la muestra. Dicho instante es el momento en que se cierra la llave que permite aplicar la tensión de entrada al capacitor para que este se cargue. La carga del capacitor tiene que ocurrir muy rápidamente para que la muestra sea tomada en un tiempo muy corto.

Este tiempo de adquisición típicamente es de 10 nS a 10 µS, según el tipo de circuito y la aplicación. Esta tensión almacenada en el capacitor es mantenida en la salida durante un tiempo determinado para que pueda ser usada por el circuito siguiente (en nuestro caso el ADC). La forma más común del circuito Sample and Hold consiste el esquema de la Figura 7.20

Figura 7.20 *Procedimiento de Sample and Hold*



Cada pulso de la señal de muestreo (sampling) hace cerrar a la llave (strobe) para que el capacitor se cargue al valor de tensión que tiene V_{in} en ese instante. El amplificador operacional, conectado como seguidor de tensión o buffer, presenta una resistencia de entrada altísima para evitar cargar al capacitor. La tensión de salida V_{out} será igual a la que se cargó en el capacitor y se mantendrá constante hasta el próximo pulso de la señal sampling que producirá un nuevo muestreo renovado a la información anterior.

f_S es la frecuencia de la señal de muestreo (sampling) y su periodo T_S, es el tiempo durante el que queda "congelado" es la salida (V_{out}) al valor que tenía la tensión de entrada (V_{in}) es el instante de tomar la muestra.

7.4.1. Teorema de Nysquist

Este teorema nos da las condiciones necesarias para que se pueda reconstruir la señal muestreada. Para que se pueda reconstituir la señal original a partir de la señal muestreada, se tienen que tomar como mínimo dos muestras por ciclo de la señal de entrada.

$$f_S > 2f_i \quad \textbf{(7.12)}$$

Donde f_S es la frecuencia de muestreo y f_i es la componente en frecuencia más alta de la señal de entrada.

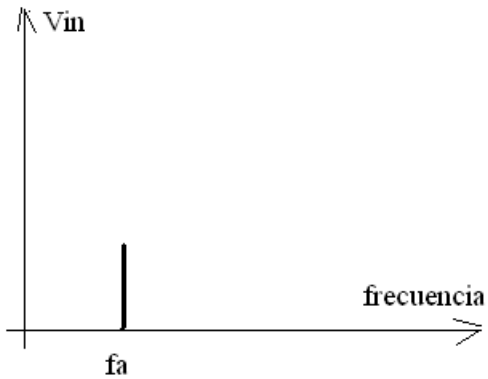
7.4.2. Recuperación de la señal original a partir de la muestreada: filtros suavizador y antialiasing

La señal original puede recuperarse a partir de la señal muestreada mediante un filtro pasa bajos llamado filtro suavizador.

Suponiendo que la señal de entrada sea una senoidal, tendrá un espectro en frecuencia como el de la Figura 7.21.

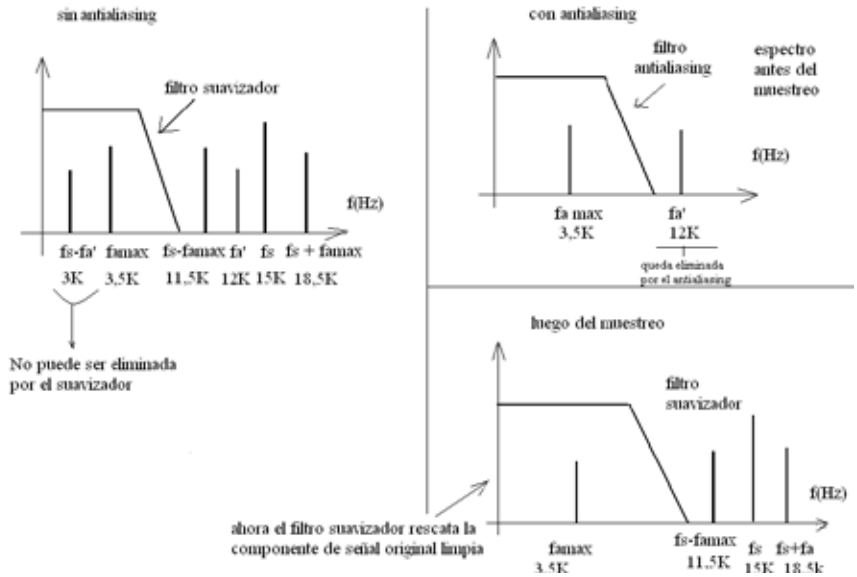
Vemos que, si se cumple con el teorema de Nyquist, la componente f_S - f_a tiene frecuencia mayor que f_a, y así el filtro suavizador permite pasar a la componente de la señal original.

Figura 7.21 *Espectro en frecuencia para una señal senoidal*



El proceso de muestreo genera nuevas componentes. De modo que la señal muestreada tendrá componentes a frecuencias f_a, y f_S las de frecuencia suma $f_S + f_a$ y diferencia $f_S - f_a$, resultando un espectro como se observa en la Figura 7.22.

Figura 7.22 *Espectro generado por la digitalización*

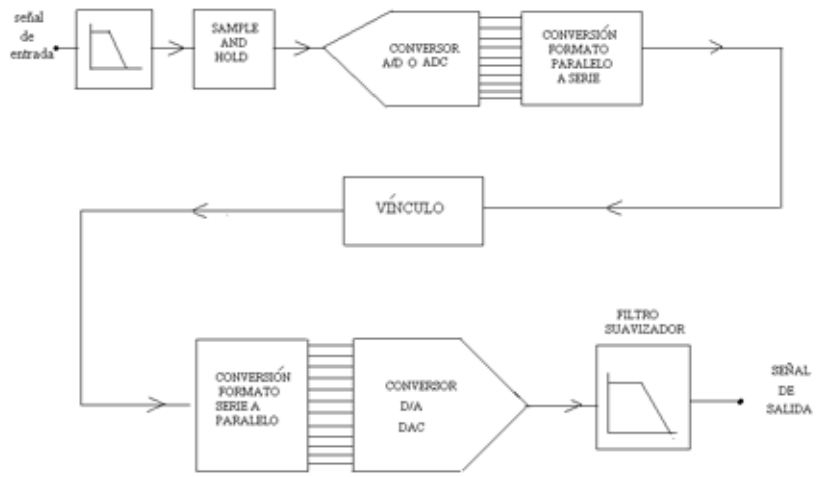


Para que esto funcione del modo descripto, es importante evitar que la señal de entrada tenga componentes de alta frecuencia que no cumplan con la frecuencia de Nyquist. Si hay posibilidad de que existan esas componentes de alta frecuencia, se las tiene que eliminar antes del muestreo. Esto se logra haciendo pasar la señal por un filtro pasa bajos denominado filtro Antialiasing. De lo contrario los componentes de f_S menos dichas frecuencias pueden caer en la zona del espectro de la señal original y no podrán ser eliminadas por el filtro suavizador.

Por ejemplo, la estructura típica de un sistema de comunicaciones digitales puede ser como se muestra en la Figura 7.23.

Es decir que luego del muestreo y antes de la recuperación se hace la conversión A/D, el proceso digital, en este ejemplo una comunicación y luego la conversión D/A.

Figura 7.23 *Estructura típica de un sistema de comunicaciones digitales*



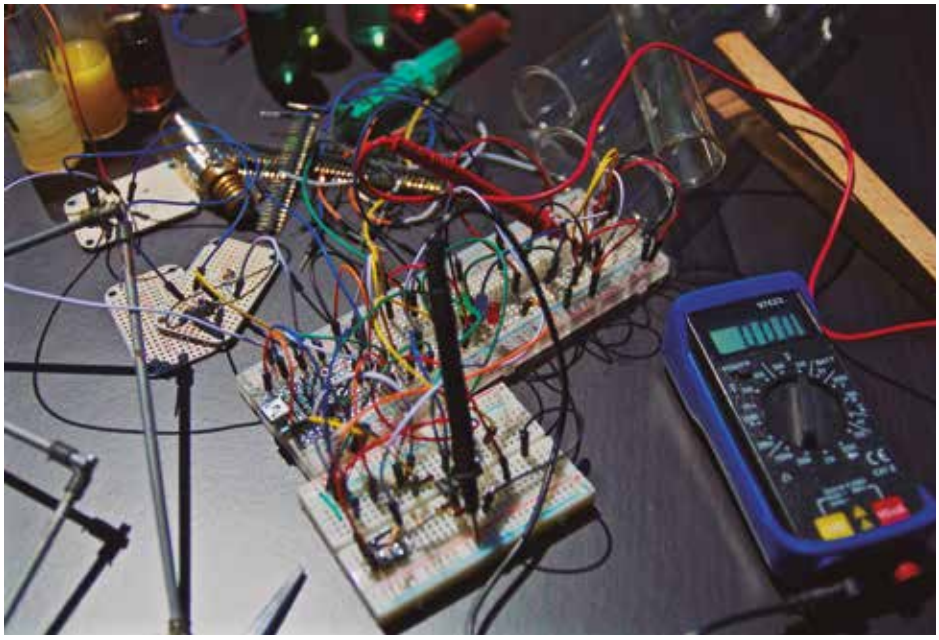
7.5. Ejercicios propuestos

1) Para el conversor DAC en escala de la sección 7.1.2, obtener la ecuación de salida resolviendo el divisor de tensión.

2) Buscar y describir cinco integrados correspondientes a los tipos de conversores indirectos estudiados en el presente capítulo en el manual de integrados CMOS.

3) Comparar la velocidad de conversión y resolución de los integrados elegidos a través de un cuadro comparativo.

4) Buscar ejemplos de sistemas de control que utilicen alguno de los conversores indirectos estudiados.



7.6. Práctica de Laboratorio o simulaciones propuestas

1) Diseñar y armar el circuito del conversor digital analógico de la sección 7.1.1 y obtener la curva de salida experimental de la Figura 7.3 para dicho circuito.

2) Repetir el procedimiento para el circuito 7.1.2

7.7. Proyecto: Control de temperatura digital para el ambiente de un edificio.

Actividad grupal en el marco de un curso de circuitos secuenciales, posterior a la realización de la práctica de laboratorio y al aprendizaje del uso del manual u hoja de datos. Se aconseja que esta actividad sea guiada por el docente.

Una vez más, siguiendo la idea del recorrido ficticio por el edificio, que se presentó en el ítem 2 de la INTRODUCCIÓN, se propone el diseño de un sistema de control para la temperatura de un edificio. Se deberán utilizar los conocimientos integrados de los capítulos del 1 al 7. Debe contar con un display que muestre la temperatura del edificio, temporizador para la activación, y el sistema de control propiamente dicho.

Capítulo 8
INTRODUCCIÓN A LOS SISTEMAS DIGITALES PROGRAMABLES

¿Sueñan los androides con ovejas eléctricas?
Novela de Philip K. Dick, adaptada por Ridley Scott en la película Blade Runner.

Cuando un sistema digital es fabricado y tiene un propósito que luego es determinado por el usuario se lo conoce como **dispositivo lógico programable**. Existen una multiplicidad de sistemas que se ajustan a esta definición.

Podemos encontrar estos dispositivos dentro de multitud de sistemas electrónicos que usamos en nuestra vida diaria, como pueden ser desde un simple timbre hasta un completo robot pasando por juguetes, heladeras, televisores, lavadoras, microondas, impresoras, el sistema de arranque de nuestro vehículo, etc.

Actualmente estos dispositivos son utilizados para diseñar una gran diversidad de circuitos, desde los que se usan para resolver problemas simples hasta sistemas circuitales más complejos.

La máquina analítica fue la primera computadora mecánica de propósito general diseñada por el matemático inglés y pionero de la computación Charles Babbage. Fue descrita por primera vez en 1837. Ada Lovelace fue la primera en reconocer que la máquina tenía aplicaciones más allá del cálculo puro, y publicó el primer algoritmo destinado a ser llevado a cabo por dicha máquina. Como resultado, es ampliamente considerada como la primera en reconocer todo el potencial de las computadoras y uno de las primeras programadoras de computadoras que existió.

Existen una gran oferta de sistemas de propósitos generales que van desde Microcontroladores hasta placas programables multipropósito tales como FPGA, Arduino o Raspberry.

En particular en este capítulo hemos elegido dar información general sobre los microcontroladores y las placas FPGA y además describir las características del sistema Arduino, brindando en el proceso algunos ejemplos de diseño de circuitos.

La elección de Arduino se justifica en su amplia difusión, la disponibilidad del código abierto y su simplicidad para la implementación.

Hemos elegido estos dos dispositivos, sin embargo, la elección que se realice debe hacerse según criterios pedagógicos o de disponibilidad para adquirir los dispositivos.

El presente capítulo no pretende ser exhaustivo simplemente es una introducción breve para el uso de estos sistemas. Existe extensa bibliografía detallando el uso de cada uno de los dispositivos que se ajustan a la definición brindada, aquí nos enfocaremos solo en estos dos ejemplos.

8.1. Microcontroladores

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea determinada Los mismos incluyen en su interior las principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de clock con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia. Por lo general, tendrá la capacidad de mantenerse a la espera de un evento como pulsar un botón o de otra interrupción.

Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Al ser fabricado el microcontrolador, no contiene datos en la memoria ROM. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando este es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

Los microcontroladores más comunes en uso están listados en la siguiente tabla, aunque consideren que la misma puede envejecer con este texto, de modo que es necesario consultar fuentes actualizadas.

Empresa	8 bits	16 bits	32 bits
Atmel	AVR (mega y tiny), 89Sxxxx fam. similar 8051		SAM7 (ARM7TDMI), SAM3 (ARM Cortex-M3), SAM9 (ARM926), AVR32
Freescale (antes Motorola)	68HC05, 68HC08, 68HC11, HCS08	68HC12, 68HCS12, 68HCSX12, 68HC16	683xx, PowerPC, ColdFire
Holtek	HT8		
Intel	MCS-48 (fam.8048) MCS51 (fam. 8051) 8xC251	MCS96, MXS296	x
National Semiconductor	COP8	x	x
Microchip	Fam. 10f2xx Familia 12Cxx, 12Fxx, 16Cxx y 16Fxx 18Cxx y 18Fxx	PIC24F, PIC24H y dsPIC30FXX, dsPIC33F	PIC32
NXP Semiconductors (antes Philips)	80C51	XA	Cortex-M3, Cortex-M0, ARM7, ARM9
Renesas (antes Hitachi, Mitsubishi y NEC)	78K, H8	H8S, 78K0R, R8C, R32C/M32C/M16C	RX, V850, SuperH, SH-Mobile, H8SX
STMicroelectronics	ST 62, ST 7		STM32 (ARM7)
Texas Instruments	TMS370	MSP430	C2000, Cortex-M3 (ARM), TMS570 (ARM)
Zilog	Z8, Z86E02		

Por lo general, los programas de microcontroladores deben caber en la memoria en chip disponible, ya que sería costoso proporcionar un sistema con memoria externa expandible.

Los compiladores y ensambladores se utilizan para convertir códigos de lenguaje de ASSEMBLER y de alto nivel en un código de máquina compacto para almacenar en la memoria del microcontrolador. Dependiendo del dispositivo, la memoria del programa puede ser permanente o borrable. Puede ser una memoria de solo lectura que solo puede programarse en la fábrica, o puede ser una memoria flash de solo lectura. O puede ser una memoria de solo lectura que se puede modificar con un dispositivo programador.

Los microcontroladores generalmente contienen desde varios hasta docenas de pines de entrada y salida que son de propósito general (GPIO). Los pines GPIO son configurables por software para un estado de entrada o de salida. Cuando los pines GPIO se configuran en un estado de entrada, a menudo se usan para leer sensores o señales externas. Configurados para el estado de salida, los pines GPIO pueden controlar dispositivos externos como LED o motores, a menudo indirectamente, a través de la electrónica de alimentación externa.

8.2. FPGA: Field-programmable gate arrays

Se puede usar un dispositivo FPGA para resolver cualquier problema que sea computable. Los FPGA se pueden usar para implementar un microprocesador suave, como Xilinx MicroBlaze o Altera Nios II. Su ventaja consiste en que estas placas son significativamente más rápidas para algunas aplicaciones debido a su naturaleza paralela y su optimización en términos del número de compuertas utilizadas para ciertos procesos.

Los FPGA comenzaron originalmente como competidores de los dispositivos llamados CPLD o Complex Programmable Logic Device. Estos últimos se forman con múltiples bloques lógicos ,que se comunican entre sí utilizando una matriz programable de interconexiones, lo cual hace eficiente el uso del silicio, conduciendo a una mejor eficiencia a menor costo.

A medida que aumentaron su tamaño, capacidades y velocidad, los FPGA asumieron funciones adicionales hasta el punto de que algunos ahora se comercializan como sistemas completos en chips (SoC).

Los FPGA contienen una serie de bloques lógicos programables y una jerarquía de "interconexiones reconfigurables" que permiten que los bloques se conecten juntos, como si fueran muchas compuertas lógicas que se pueden interconectar en diferentes configuraciones. Los bloques lógicos se pueden configurar para realizar funciones combinacionales complejas, o simplemente compuertas lógicas simples como AND y XOR.

En la mayoría de los FPGA, los bloques lógicos también incluyen elementos de memoria, que pueden ser Flip Flop simples o bloques de memoria más completos.

Muchos FPGA se pueden reprogramar para implementar diferentes funciones lógicas, permitiendo una computación reconfigurable flexible tal como se realiza en el software de la computadora.

8.3. Arduino

8.3.1. ¿Qué es Arduino?

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida: activar un motor, encender un LED, publicar algo en línea. Se puede programar

El proyecto Arduino comenzó en 2005 como una herramienta para estudiantes en el Design Institute Ivrea en Ivrea, Italia. El objetivo de proporcionar una manera fácil y de bajo costo para que los principiantes y profesionales creen dispositivos que interactúen con su entorno utilizando sensores y actuadores. El nombre Arduino proviene de un bar en Ivrea, Italia, donde solían reunirse algunos de los fundadores del proyecto. El bar lleva el nombre de Arduino de Ivrea, quien fue el marqués de Ivrea y luego Rey de Italia desde 1002 hasta 1014.

enviando un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en el cableado) y el software Arduino (IDE), basado en el procesamiento.

Arduino ha sido implementado en miles de proyectos, desde objetos cotidianos hasta instrumentos científicos complejos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) trabajan en torno a esta plataforma de hardware abierto. Sus contribuciones se han sumado a una gran cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos.

Este dispositivo nació en el Instituto de Diseño de Interacción Ivrea como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta de placas simples de 8 bits a productos para aplicaciones IoT (internet de las cosas), dispositivos portátiles, impresión 3D y entornos integrados.

Todas las placas Arduino son de hardware abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares.

El software también es de código abierto y está creciendo a través de las contribuciones de los usuarios de todo el mundo. La programación puede hacerse usando lenguaje C++ con librerías que le agregan funcionalidad y facilidad de desarrollo. Es de código libre y que es multiplataforma. Pueden usarse otros entornos de desarrollo también que usan como motor el IDE de Arduino (Eclipse, vscode, etc.).

Existen varios modelos. La placa correspondiente al Arduino UNO se muestra en la Figura 8.1. Detalles se pueden extraer de la página oficial (https://store.arduino.cc/usa/), así como también mucha información que permite aprender y hacer pronto prototipos funcionales.

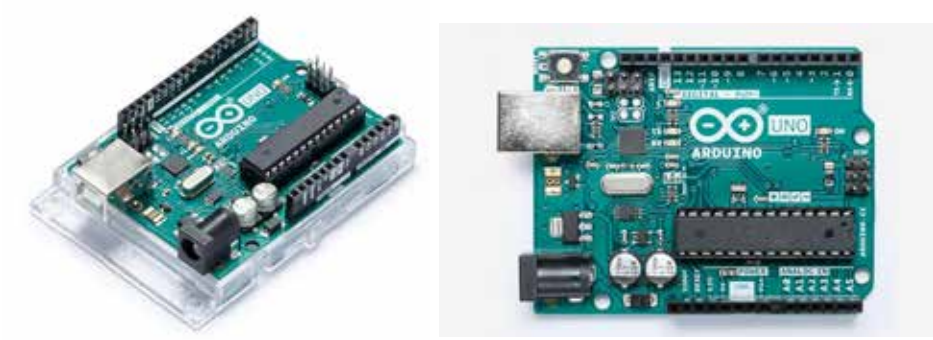


Imagen 8.1 Placa de Arduino UNO vista lateral y superior
Imagen extraída de la página oficial para la venta del dispositivo.

8.3.2. Como instalar el software y comenzar a utilizarlo

El software de Arduino (IDE) le permite escribir programas y subirlos a su "pizarra". En la página del software Arduino hay disponibles dos opciones:

- Si se tiene una conexión a Internet confiable, es conveniente usar el IDE en línea (Arduino Web Editor). Este permitirá guardar los proyectos en la nube, tenerlos disponibles desde cualquier dispositivo y hacer una copia de seguridad. Siempre tendrá la versión más actualizada del IDE sin la necesidad de instalar actualizaciones o bibliotecas generadas por la comunidad.
- Si se prefiere trabajar sin conexión, debe usarse la última versión del IDE de escritorio.

En el caso de usar el código en línea en el editor web de Arduino, para usar el IDE en línea, simplemente hay que seguir las instrucciones en el sitio web. No es necesario instalar nada.

8.3.3. Entorno de desarrollo

El Arduino Integrated Development Environment, o Arduino Software (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos.

8.3.4. Escribir Sketches

Los programas escritos con el software Arduino (IDE) se denominan **Sketches.** Estos bocetos se escriben en el editor de texto y se guardan con la extensión de archivo *.ino*. El editor tiene funciones para cortar, pegar y para buscar y reemplazar texto.

El área de mensajes proporciona comentarios útiles al guardar y exportar. También muestra errores. La consola muestra la salida de texto del software Arduino (IDE), incluidos los mensajes de error completos y otra información. La esquina inferior derecha de la ventana muestra la placa configurada y el puerto serie. Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar Sketches, y abrir el monitor en serie.

El software Arduino (IDE) utiliza el concepto de un cuaderno de bocetos o Sketches: un lugar estándar para almacenar sus programas (o bocetos). Los bocetos en su cuaderno de bocetos se pueden abrir desde el menú "Archivo" > "Cuaderno" de bocetos o desde el botón "Abrir" en la barra de herramientas. La primera vez que ejecute el software Arduino, creará automáticamente un directorio para su cuaderno de bocetos. Puede ver o cambiar la ubicación de la ubicación del cuaderno de bocetos desde el cuadro de diálogo "Preferencias".

Los comandos principales son:

- Verificar: comprueba su código en busca de errores al compilarlo.
- Subir: compila el código y lo carga en la placa configurada. Es importante tener en cuenta que, si está utilizando un programador externo con su placa, puede mantener presionada la tecla "shift" en su computadora cuando use este icono. El texto cambiará a "Cargar usando el programador".
- Nuevo: crea un nuevo sketch.
- Abrir: presenta un menú de todos los bocetos en su cuaderno de sketch. Al hacer clic en uno, se abrirá dentro de la ventana actual sobrescribiendo su contenido.
- Salvar: guarda el sketch.
- Monitor en serie: abre el monitor en serie.

8.3.5. Cargando los Sketches

Antes de cargar su boceto, debe seleccionar los elementos correctos en los menús Herramientas > Tablero y Herramientas > Puerto.

En Mac, el puerto serie es probablemente algo así como:

/dev/tty.usbmodem241 (para Uno o Mega2560 o Leonardo), o
/dev/tty.usbserial – 1B1 (para una placa USB Duemilanove o anterior), o
/dev/tty.U SA19QW 1b1P 1,1 (para una placa serie conectada con un adaptador USB a serie Keyspan).

En Windows, probablemente sea COM1 o COM2 (para puerto serie) o COM4, COM5 o COM7 o superior (para una placa USB); para averiguarlo, busque un dispositivo serie USB en la sección de puertos del Administrador de dispositivos de Windows.

En Linux, debería ser:

/dev/ttyAC M x, /dev/ttyU SBx
o similar. Una vez que haya seleccionado el puerto serie y la placa correctos, presione el botón de carga en la barra de herramientas o seleccione el elemento Cargar del menú Boceto.

Las placas Arduino actuales se reiniciarán automáticamente y comenzarán la carga. Con tableros más antiguos (pre-Diecimila) que carecen de reinicio automático, deberá presionar el botón de reinicio en el tablero justo antes de comenzar la carga. En la mayoría de las placas, verá que los LED RX y TX parpadean a medida que se carga el boceto. El software Arduino (IDE) mostrará un mensaje cuando se complete la carga o mostrará un error.

Cuando carga un sketch, se está utilizando el gestor de arranque Arduino, un pequeño programa que se ha cargado en el microcontrolador de su placa. Esto permite cargar código sin usar ningún hardware adicional. El gestor de arranque está activo durante unos segundos cuando la placa se reinicia; luego comienza el boceto que se cargó más recientemente en el microcontrolador. El gestor de arranque parpadeará el LED incorporado (pin 13) cuando se inicie (es decir, cuando la placa se reinicie).

8.3.6. Librerías o bibliotecas

Ambos términos se suelen utilizar en español de manera indistinta para referirse a conjuntos de funciones de software ya escritas que permiten realizar distintas operaciones. Aunque el término más apropiado sería biblioteca. En el caso de Arduino, las bibliotecas proporcionan una funcionalidad adicional para su uso en bocetos o trabajando con hardware o manipulando datos. Para usar una biblioteca en un sketch, selecciónela en el menú Boceto, Importar biblioteca. Esto insertará una o más declaraciones *#include* en la parte superior del boceto y compilará la biblioteca con su boceto. Debido a que las bibliotecas se cargan en el tablero con su boceto, aumentan la cantidad de espacio que ocupa. Si un boceto ya no necesita una biblioteca, simplemente elimine sus declaraciones #include de la parte superior de su código.

Hay una lista de bibliotecas en la referencia. Algunas bibliotecas están incluidas con el software Arduino. Otros se pueden descargar desde una variedad de fuentes o mediante el Administrador de la Biblioteca. A partir de la versión 1.0.5 del IDE, puede importar una biblioteca desde un archivo ".zip" y usarla en un boceto abierto. Consulte estas instrucciones para instalar una biblioteca de terceros.

8.3.7. Características del hardware de ARDUINO UNO R3

El sistema didáctico se divide en tres subsistemas: un software basado en el entorno de programación Processing/Wiring que se ejecuta en una computadora estándar, una placa ARDUINO UNO R3 que se interconecta entre la computadora y diversos sensores o actuadores que le permiten al hardware interactuar con el medio físico.

Configuración de la placa ARDUINO UNO R3

La plataforma está basada en el entorno de desarrollo Processing y el microcontrolador Atmega 328 de la firma ATMEL. Dicha empresa dispone de 14 pines digitales de entrada/salida (de las cuales seis se pueden utilizar como salidas PWM), un oscilador cerámico de 16 MHz, una interfaz USB y 32 kB de memoria FLASH. Cada pin puede proporcionar o recibir un máximo de 40 mA y tienen una resistencia de 20 – 50 KΩ pull-up, 6 entradas analógicas. Algunos de estos pines tienen funciones especiales (2).

Características técnicas de la placa ARDUINO UNO R3

A continuación, daremos la descripción de las características:

- Microcontrolador Atmega328P.
- Tensión de operación 5V Tensión de entrada(recomendada) 7 – 12 V.
- Tensión de entrada(limite) 6 – 20 V.
- Pines digitales de E/S 14(de los cuales 6 proveen salidas PWM).
- Pines de entrada analógicos 6 Corriente DC por pin E/S 40 mA .
- Corriente DC para pin 3.3 V 50 mA.
- Memoria Flash 32 kB (de los cuales 0,5 kB usados para bootloader) SRAM 2 kB EEPROM 1 kB Frecuencia de reloj 16 MHz.

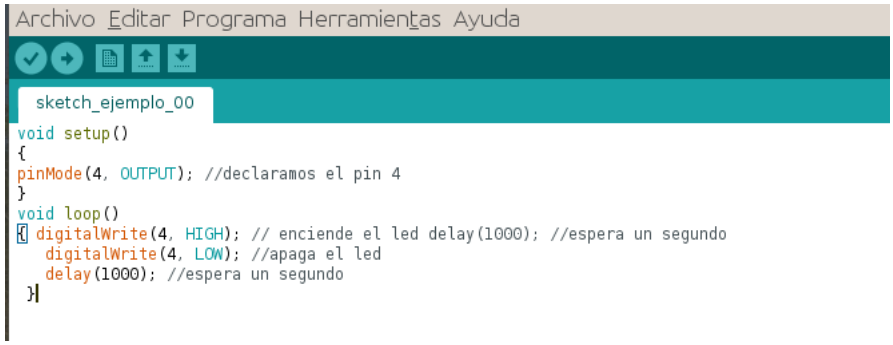
8.3.8. Ejemplos de aplicación y prácticas relacionadas con técnicas digitales

Dentro de la familia Arduino, una de la más conocidas es la UNO R3. Los ejercicios básicos, expuestos en este capítulo están basados en este modelo.

Ejemplos básicos

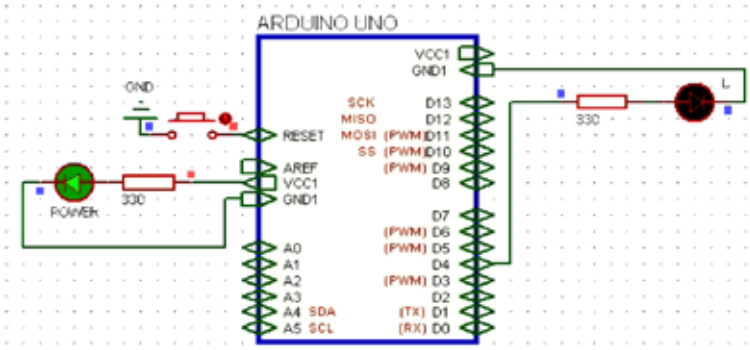
Los siguientes programas nos demuestran los comandos básicos de Arduino y su fácil realización.

- **Encendido y apagado de un LED**
  Este es el programa básico usado simplemente para activar o desactivar algo. En este ejemplo, un LED está conectado al pin 4, y parpadea cada segundo. La resistencia puede omitirse en este pin ya que el ARDUINO tiene una integrada.



El circuito correspondiente se puede observar en la Figura 8.2.

Figura 8.2 *Circuito correspondiente al ejemplo presentado*



Sobre la Salida PWM: La modulación de ancho de pulso (PWM) es una forma de falsificar una salida analógica por la salida pulsante. Esto podría usarse para atenuar e iluminar un LED o posteriormente controlar un servomotor. El siguiente ejemplo ilumina y atenúa lentamente un LED usando bucles for.

- **Ajuste del brillo de un LED**

El circuito correspondiente es el mismo que para el programa anterior, es decir el que se muestra en la Figura 8.2.

Sensor de temperatura

El siguiente código usa un sensor de temperatura LM35 conectado al pin analógico A0 en el cual podemos leer valores de 0 a 1023. Convertiremos estos valores analógicos a digitales y obtendremos la temperatura, si esta es igual o sobre pasar los 25°C el LED se encenderá, de lo contrario el LED permanecerá apagado.

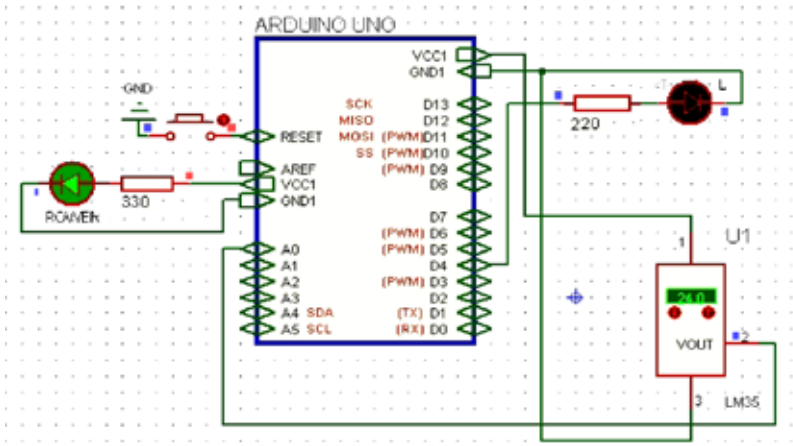
```
Archivo Editar Programa Herramientas Ayuda

sketch_ejemplo_02

const int sensor = 0; // sensor conectado a pin A0
const int ledpin = 4; // pin de salida para el LED
long temperatura; // variable que almacena el dato de la temperatura
long leersensor; // variable que almacena el dato del sensor en milivoltios

void setup()
{
pinMode(ledpin, OUTPUT); //declara ledPin como salida
 }
void loop()
{
%leersensor = (analogRead(sensor)*(5000L)/1023; //lee el sensor multiplica por 5000mv y divide en 1023.
%temperatura= leersensor/10; //se dividen los mv en 10 y obtiene la temperatura.
%if(temperatura 25) // lee temperatura si es mayor o igual a 25 grados
{
digitalWrite(ledpin, HIGH); //enciende ledpin
}
else // si no es mayor o igual a 25
{ digitalWrite(ledpin, LOW); // apaga el led pin
}
}
```

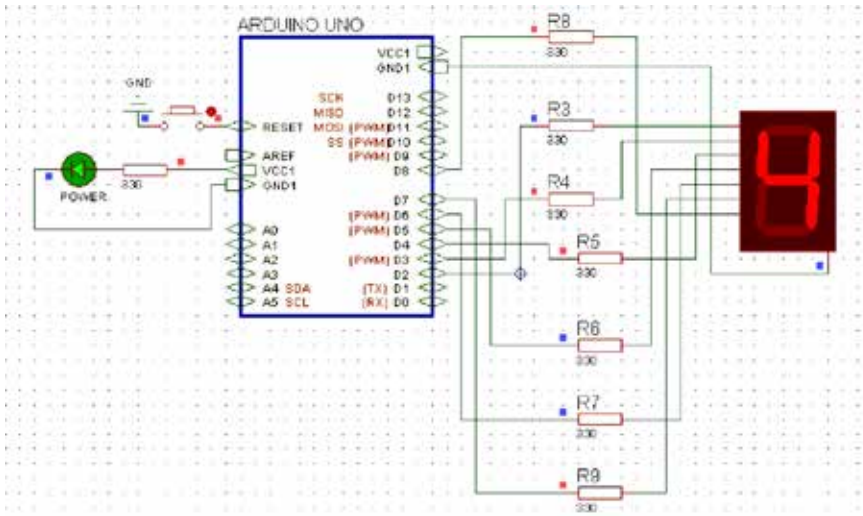
Figura 8.3 *Circuito correspondiente al ejemplo presentado*



- **Contador de pulsos con salida de segmentos**
  Este código es básicamente igual al ejemplo anterior, se diferencia en que este lleva las funciones if y switch asociadas lo que permite incrementar la cuenta por medio de un pulsador. // contador de pulsos // display conectado en los pines 2 a

El circuito correspondiente se puede observar en la Figura 8.4.

Figura 8.4 *Circuito correspondiente al ejemplo presentado*



Esperamos estos ejemplos les resulten de utilidad y como motivación para probar otros ejemplos y también diseños propios, así como también intentar explorar la posibilidad de usar otro tipo de placas, como las mencionadas al principio del capítulo.

8.4. Comentarios finales

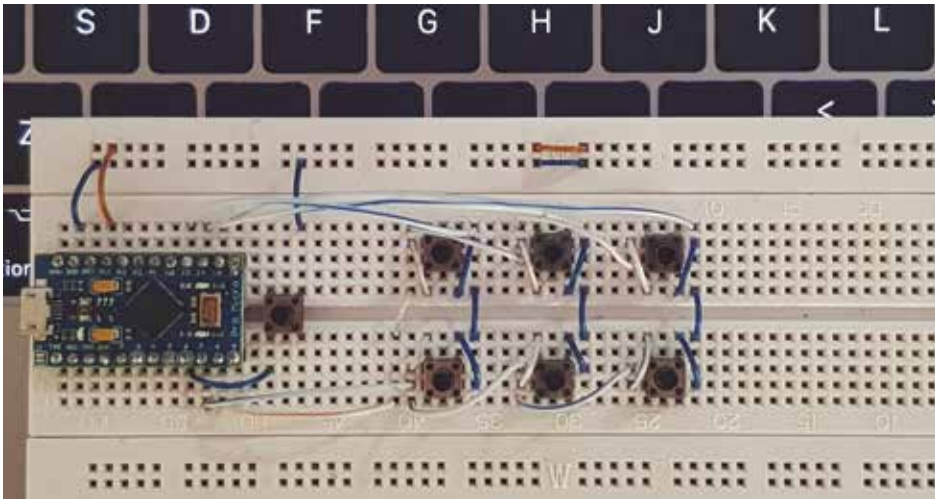
Las placas de Arduino, como las que vimos en este capítulo actualmente pueden ser utilizados con mínimos conocimientos de electrónica. Sin embargo, su uso profesional y para resolver problemas técnicos, industriales y en el laboratorio requieren conocimientos de electrónica digital de los dispositivos que describimos a lo largo de los capítulos de este libro.

También dichos conocimientos son fundamentales para el uso de microcontroladores. Tener este set de conocimientos básicos que ofrecemos a lo largo del libro permite a los técnicos e ingenieros encontrar la solución más apropiada en función de las distintas escalas de complejidad, presupuesto o circunstancias en los diseños.

Lo que distingue especialmente a los profesionales en Argentina es la fortaleza en los conocimientos básicos que permite adaptarse a las distintas situaciones de necesidad o abundancia. Por este motivo no alentamos la hiperespecialización y mostramos una diversidad de dispositivos. Desde un fuerte conocimiento básico, lo suficientemente general, se pueden aprender a utilizar distintos dispositivos, programar en distintos lenguajes adaptarse a las distintas circunstancias.

Esperamos estos contenidos les resulten útiles y puedan implementarlos en las aulas y laboratorios.

8.5. Proyecto Final: Control de un ascensor



Actividad grupal en el marco de un curso de circuitos secuenciales y como aplicación de los sistemas programables. Se aconseja que esta actividad sea guiada por el docente.

Una vez más, siguiendo la idea del recorrido ficticio por el edificio, que se presentó en el ítem 2 de la INTRODUCCIÓN, se propone como ejercicio el diseño de un sistema de control un ascensor de un edificio. Se deberán utilizar los conocimientos integrados de todos los capítulos.

Lineamientos propuestos

Diseñar un ascensor automatizado de 4 plantas. El cual constará de 4 sensores, 1 por cada planta. Tendrá 4 pulsadores para llamar al cubículo del ascensor, 1 por cada planta. El objetivo es que el cubículo llegue al lugar solicitado independientemente de donde se encuentre, y si hacemos otra solicitud mientras se ejecuta la primera, que se mantenga en espera y luego se ejecute finalizada la primera. Colocar un botón de emergencia por medio de cableado externo, de modo tal que se activará cuando se accionen los sensores de todos los pisos (menos plata baja). Este botón hará bajar el cubículo hasta la planta baja y abrir la puerta.

Apéndice A
ÁLGEBRA DE BOOLE Y COMPUERTAS

El álgebra de Boole se construye como un conjunto de reglas y teoremas que se han adaptado para utilizar con variables bivalentes (solo pueden tener dos estados “0” o “1”), los valores de tensión (alto o bajo) que podemos tener en los terminales o cables de un circuito, y la relación que existe entre el uso del álgebra con las operaciones lógicas que realizan las compuertas AND, OR y NOT (o inversor).

El uso del álgebra de Boole nos servirá de herramienta para analizar, simplificar y diseñar circuitos digitales combinacionales. Primero debemos definir los términos y las operaciones que utilizaremos.

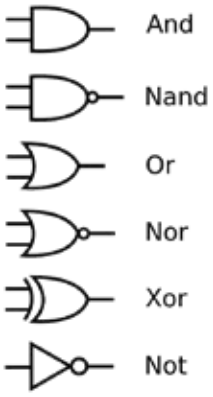
Definimos el término **variable** como un símbolo que utilizamos para representar a una magnitud lógica (por convención utilizamos letras en imprenta y mayúscula). Definimos al complemento como el inverso de una variable y se indica mediante una barra encima de la letra que representa a dicha variable. Por ejemplo, el complemento de la variable A es $\bar{A}$.

Por otro lado, una **función lógica**, es una relación que me permitirá determinar el valor de una variable llamada dependiente, dado el valor de otras variables llamadas independientes (relacionadas estas por funciones lógicas OR, AND y NOT).

Representamos con símbolos cada una de las distintas funciones lógicas, como se observan en la Figura A.1.

Las funciones básicas son AND, OR y NOT, todas las demás se pueden construir a partir combinaciones de éstas.

Figura A.1 *Símbolos con todas las compuertas representando las diferentes funciones lógicas*



A.1. Tabla de verdad

Definimos a una tabla de verdad, como aquella tabla que, para un determinado juego de valores de las variables independientes, da el correspondiente valor de la variable dependiente.

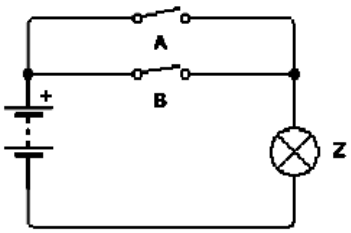
En dichas tablas, B y A representan a las variables independientes y Z representa a la variable dependiente. A continuación, se definen las operaciones básicas del álgebra de Boole a partir de las tablas de verdad y se representa la función con un circuito con llaves.

A.2. Compuertas

A.2.1. Función OR

Definimos entonces que la compuerta OR será un sumador booleano. En este obtendremos un resultado igual a 1, cuando las variables de entrada presenten un 1, en uno o más de sus términos y será igual a 0 el resultado cuando el valor de las variables de entrada se encuentre en cero. Representamos lo anterior utilizando un circuito con llaves como el de la Figura A.2, además construimos la tabla de verdad A.1.1 y el símbolo de la compuerta OR como el que se observa en la Figura A.1, que contiene todos los símbolos de las compuertas más comunes.

Figura A.2 *Circuito representando la función OR*



En términos circuitales esto significa que para que encienda la lámpara del circuito es necesario que al menos una llave esté cerrada.

Podemos ver que para que la lámpara prenda debe estar cerrada la llave A o (OR) la llave B, también se puede dar el caso que ambas estén cerradas y obtener como resultado que la lámpara encienda.

Tabla A1.1 *Tabla de verdad de la Función OR*

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Por último, escribiremos la ecuación y símbolo con el que representamos a la operación suma.

La función se lee como:

$$Z = A \text{ “OR” } B \quad \textbf{(A.1)}$$

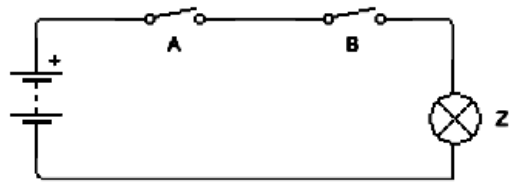
En la práctica utilizamos la expresión: Z = A + B (recordar que el + representa función lógica).

A.2.2. Función AND

La denominada multiplicación booleana es equivalente a la función lógica AND. En el álgebra de Boole, un término producto es un producto de los valores de sus variables de entrada. Para obtener un 1 como resultado de esta operación, el valor de las variables de entrada deberá ser iguales a 1. Para obtener un 0 como resultado, basta con que alguna de las variables de entrada presente un valor igual a 0.

Al igual que en el caso anterior representamos lo expresado con un circuito con llaves como el de la Figura A.3, tabla de verdad y símbolo como se muestra en la Figura A.1.

Figura A.3 *Circuito representando la función AND*



Como se ve en el circuito, tener a la lámpara en estado de encendido debemos tener a la llave A y a la llave B cerradas. En el caso de que alguna de las llaves. La tabla de verdad será entonces como la A.1.2.

Tabla A1.2 *Tabla de verdad de la Función AND*

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Por último, escribimos la ecuación y símbolo con el que representamos a la operación producto:

$$Z = A \text{ “AND” } B \quad \textbf{(A.2)}$$

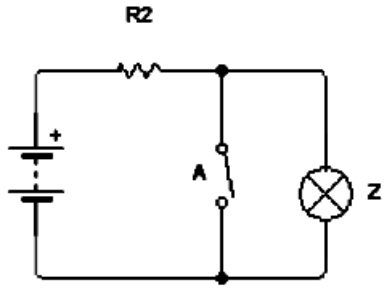
Utilizamos la expresión: Z = A . B (recordar que el punto en esta ecuación no representa una multiplicación aritmética, sino a la función lógica AND).

A.2.3. Función NOT

La negación booleana es equivalente a la compuerta NOT (inversor lógico). Esta función nos dará un 1 lógico en su salida, cuando la variable de entrada valga 0 y viceversa, podremos decir entonces, que la salida tendrá el valor que no tiene la entrada.

Al igual que en los casos anteriores, representamos la explicación anterior con un circuito con llaves como el de la Figura A.4, la tabla de verdad será como A.1.3 y el símbolo como en la Figura A.1.

Figura A.4 *Circuito representando la función NOT*



Vemos en el circuito, que la lámpara estará encendida cuando la llave A se encuentre abierta y que estará apagada cuando la llave A se encuentre cerrada.

Tabla A1.3 *Tabla de verdad de la Función NOT*

A	$\bar{A}$
0	1
0	0

Por último, escribimos la ecuación y símbolo con el que representamos a la operación producto:

$$Z = \text{“N O”}\ A \quad \textbf{(A.3)}$$

En la práctica utilizamos la expresión: $Z = \bar{A}$

A.2.4. Otras funciones

El resto de las funciones representadas como compuertas de la Figura A.1 se pueden obtener a partir de la combinación de las funciones AND, OR y NOT.

Por ejemplo, NOR es $Z = \overline{A + B}$, NAND es $Z = \bar{A} \cdot \bar{B}$, OR exclusiva es $Z = \bar{A}B + A\bar{B}$.

A.3. Leyes y reglas del álgebra de Boole

El álgebra de Boole utiliza las leyes conmutativas de la suma y del producto, también la ley asociativa y la distributiva. Tener presente que el número de variables no se limita a la cantidad mostrada en lo ejemplos, usted puede implementarlo para la cantidad de variables que necesite.

- **Idempotencia:** Las operaciones producto o suma de la variable consigo misma, resultan equivalentes a trabajar con la variable.
  Para el producto: $A \cdot A = A$
  Para la suma: $A + A = A$

- **Elemento de identidad:**
  $0 + A = A$
  $1 \cdot A = A$

- **Propiedad conmutativa** (el resultado no depende del orden):
  $A + B = B + A$
  $A \cdot B = B \cdot A$

- **Propiedad asociativa** (el resultado no depende del modo de asociación):
  $A + (B + C) = (A + B) + C = A + B + C$
  $A \cdot (B\,C) = (A \cdot B) \cdot C = A\,B\,C$

- **Propiedad distributiva** (una operación se distribuye en una asociación):
  $A \cdot (B + C) = A \cdot B + A \cdot C$
  $A + (B \cdot C) = (A + B) \cdot (A + C)$

- **De Morgan**:
  $\overline{A \cdot B} \equiv \bar{A} + \bar{B}$
  $\overline{A + B} \equiv \bar{A} \cdot \bar{B}$

- **Otras propiedades:**
  $A + \bar{A} = 1$
  $A \cdot \bar{A} = 0$
  $A + (A \cdot B) = A$
  $A \cdot (A + B) = A$
  $A + (\bar{A} \cdot B) = A + B$
  $A \cdot (\bar{A} + B) = A \cdot B$

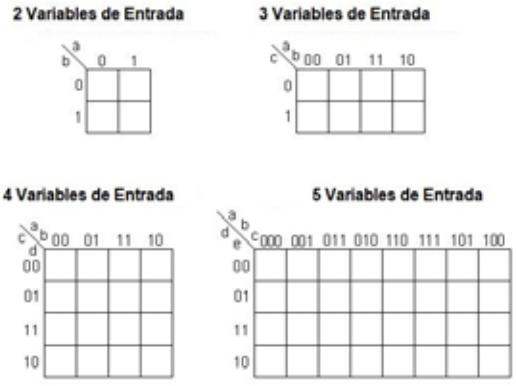
Apéndice B
MAPAS DE KARNAUGH

Un mapa de Karnaugh es un diagrama utilizado para la simplificación de funciones algebraicas booleanas. Fue inventado en 1953 por Maurice Karnaugh, un físico y matemático de los laboratorios Bell.

Estos diagramas reducen la necesidad de hacer cálculos extensos para la simplificación de expresiones booleanas, y aprovechan la capacidad del cerebro en el reconocimiento de patrones, permitiendo identificar y eliminar condiciones muy inmensas.

Es una representación bidimensional de la tabla de verdad de la función a simplificar. Puesto que la tabla de verdad de una función de N variables posee 2^N filas, el mapa K correspondiente debe poseer también 2^N espacios. Las variables de la expresión son ordenadas en función de su peso y siguiendo el código Gray, de manera que sólo una de las variables varía entre celdas adyacentes. En la Figura B.1 se muestran los ejemplos para armar mapas de Karnaugh para 2, 3, 4 y 5 variables.

Figura B.1. *Mapas de Karnaugh para tablas de verdad de 2,3,4 y 5 variables de entrada*



La transferencia de los términos de la tabla de verdad al mapa de Karnaugh se realiza de forma directa, colocando un 0 o un 1, dependiendo del valor que toma la función en cada fila. Se pueden fácilmente realizar a mano con funciones de hasta 6 variables, para funciones de mayor cantidad de variables es más eficiente el uso de software especializado.

Una vez diagramado el mapa de Karnaugh, la siguiente tarea es la de seleccionar conjuntos de términos denominados subgrupos de manera que se obtenga el menor número de subgrupos posible. Estos subgrupos se seleccionan formando grupos de rectángulos que encierren a los unos del mapa, las áreas deben ser potencia de 2 (ej. 1, 2, 4, 8, ...) y se debe tratar de agrupar el mayor número de unos posible.

Hay que tomar en cuenta al hacer estos grupos de unos (subgrupos) lo siguiente:

- Debemos usar todos los unos del mapa.
- Es más eficiente crear el menor número de grupos.
- Los unos pueden estar en varios grupos a la vez.
- El número de unos dentro de un grupo debe ser cualquier potencia de 2.
- Cuanto más grande sea un grupo, la simplificación de la función será mejor.
- No es necesario que todos los grupos tengan el mismo tamaño.
- Qué términos seleccionar va dependiendo de cómo se quiera realizar la simplificación, puesto que esta puede realizarse por mini términos o por maxitérminos.

Ejemplos de aplicación de los mapas se pueden ver en la Sección 2.1.4 aplicados a hallar la ecuación del Flip Flop JK y en la Sección 4.4, donde se usan para poder obtener el circuito correspondiente a un contador sincrónico y a un generador pseudoaleatorio.

BIBLIOGRAFÍA

Arduino software Guide (2020). **https://www.arduino.cc/en/Guide/Introduction**

Bertoloto, D. (1990-2010). Notas de Clases: *"Técnicas Digitales II"*. Material no publicado.

Baker, R. J. (2010) CMOS: *Circuit Design, Layout, and Simulation* (3ª ed.). New York: John Wiley & Sons.

Eccles, W. H., Jordan, F. W. (1919, setiembre,19). *A trigger relay utilizing three-electrode thermionic vacuum tubes*, The Electrician, 83, p.298.

Eccles, W. H., Jordan, F. W. (solicitada 1918; publicada 1920). *Improvements in ionic relays*. Patente Inglesa number: GB 148582.

Floyd, T. L. (2007). *Digital Fundamentals Prentice*, New Jersey: Pearson Prentice Hall.

Flowers, T. H. (julio 1983). *The Design of Colossus*. IEEE Annals of the History of Computing, 5, (3), p.239.

Fogiel, M., Gu, Y. L. (1998). *The Electronics problem solver*, 1. Research & Education Assoc. p. 1223.

Fuller, B. (2012, 8, 15). *Hans Camenzind, 555 timer inventor, dies*. EE Times.

Gates, E. D. (12 de enero de 2000). *Introduction to electronics* (4ª ed.). Delmar Thomson (Cengage) Learning. p.299.

Godse, A. P., Godse. D. A. (2009). *Digital Electronics*. Peth: Technical Publications.

Gray, P. R., Hurst, P. J, Lewis, S. H, Meyer, R (2009). *Analysis and Design of Analog Integrated Circuits*. New York: John Wiley & Sons, Inc.

Jain R. P. (2006). *Modern Digital Electronics*. New Delhi: McGraw-Hill Science.

Linear1 Databook (1972). Sunnyvale: Signetics Corporation.

Maini, A. K. (2007). *Digital Electronics: Principles and Integrated Circuits*. New Delhi: Wiley India Pvt. Limited,

Morris M., Mano, R., Ciletti, M. D. (2017). *Digital Design: With an Introduction to Verilog, HDL*, New Jersey: Pearson Education, Inc.

Motorola. (1991). *CMOS logic data*. USA: Series C MOTOROLA INC.

Patiño, A. A. (2014). *Guía para Diseño y Elaboración de sistemas digitales con Arduino R3*. Pereira: Universidad Tecnológica de Pereira.

Proteus Design Suite. **https://www.labcenter.com/**

Pugh, E. W., Johnson, L. R.; Palmer, J. H. (1991). *IBM's 360 and early 370 systems*. Cambridge, London: MIT Press.

Rabaey, J. M., Chandrakasan, A., Nikolic, B. (2004). *Digital Integrated Circuits: A design perspective*. (2ª ed.). New Jersey: Prentice Hall Press,

Sah, C. T., Wanlass, F. (2019, julio, 6). *1963: Complementary MOS Circuit Configuration is Invented. Computer*. Archivo extraído History Museum.

Sonde, B. S. (1992). *Introduction to System Design Using Integrated Circuits*. New Delhi: New Age International. Publisher.

Texas Instruments (enero de 1994, revisado abril de 2015). *Principles of Data Acquisition and Conversion*. **https://www.ti.com/lit/an/sbaa051a/sbaa051a.pdf**

Torrente Artero, O. (2013). *Arduino. Curso Práctico de Formación*. Madrid: Alfaomega.

Wright, M. (2006, junio, 22). *Milestones That Mattered: CMOS pioneer developed a precursor to the processor*. EDN.

RESUMEN

Este libro contiene las nociones básicas necesarias para comprender los circuitos secuenciales. Comienza, en el primer capítulo con las definiciones y términos que se utilizan en la electrónica digital. Luego, en el capítulo 2 se explica el concepto de Flip Flop de manera constructiva. Dicho concepto es fundamental para poder entender el funcionamiento de los circuitos secuenciales y poder armar distintas aplicaciones.

Los capítulos 3, 4 y 5 exploran las aplicaciones de estos dispositivos: registros, contadores y osciladores, mientras que los capítulos 5 y 6 explican los conceptos relacionados con la conversión y digitalización de señales analógicas.

Finalmente, el capítulo 7 brinda una breve introducción al próximo paso en el diseño de circuitos digitales: los sistemas programables. Todos los capítulos anteriores son claves para poder explotar de lleno todas las características de estos dispositivos que, desde hace muchos años son fundamentales tanto para la industria, como en el desarrollo tecnológico a distintas escalas.

En todos los capítulos se presentan propuestas de ejercicios, prácticas de laboratorio y/o proyectos que permitan aplicar o consolidar los conocimientos presentados, teniendo en cuenta siempre la disponibilidad local y el costo de los dispositivos propuestos y empleados. Al final de este libro se presenta una serie de apéndices que contienen un resumen de los conocimientos previos que son necesarios para la comprensión de los circuitos secuenciales.

Palabras claves: **Flip Flop, circuitos secuenciales, contadores, osciladores, sistemas programables**