

ELECTRÓNICA DIGITAL: DISEÑO Y APLICACIONES

**COMPUERTAS LÓGICAS, CIRCUITOS COMBINACIONALES,
CIRCUITOS SECUENCIALES Y CONTROLADORES DIGITALES**

Ing. Jorge Antonio Polanía Puentes

Ingeniero Electrónico. Universidad Distrital - Colombia

Magister en Ingeniería Electrónica. UNAM- México

Copyright (© 2025, ing. Jorge Antonio Polanía, "Todos los derechos reservados")

CONTENIDO

| | |
|--|----|
| INTRODUCCIÓN..... | 5 |
| CAPÍTULO 1. COMPUERTAS LÓGICAS - TEORÍA | 9 |
| 1.1 NÚMEROS BINARIOS..... | 9 |
| 1.2 PUERTAS LÓGICAS BÁSICAS..... | 10 |
| 1.3 CIRCUITOS INTEGRADOS..... | 11 |
| 1.4 ALGEBRA DE BOOLE..... | 17 |
| 1.5 CIRCUITOS LÓGICOS..... | 19 |
| 1.6 CONVERSIÓN DE PUERTAS LÓGICAS | 20 |
| 1.7 DISEÑO CON PUERTAS NAND | 21 |
| 1.8 MAPAS DE KARNAUGH | 22 |
| CAPÍTULO 2. COMPUERTAS LÓGICAS - SIMULACIÓN | 25 |
| 2.1 CARACTERÍSTICAS ELÉCTRICAS..... | 25 |
| 2.2 ECUACIONES LÓGICAS DE LAS COMPUERTAS | 25 |
| 2.3 CIRCUITO LÓGICO CON COMPUERTAS..... | 26 |
| CAPÍTULO 3. COMPUERTAS LÓGICAS - LABORATORIO | 27 |
| 3.1 EQUIPO Y MATERIAL NECESARIO..... | 27 |
| 3.2 CARACTERÍSTICAS ELÉCTRICAS..... | 27 |
| 3.3 ECUACIONES LÓGICAS DE LAS COMPUERTAS | 28 |
| 3.4 CIRCUITO LÓGICO CON COMPUERTAS..... | 28 |
| 3.5 APLICACIÓN: PARQUEADERO | 29 |
| 3.6 APLICACIÓN: MICROROBOT..... | 30 |
| CAPÍTULO 4. SISTEMAS COMBINACIONALES..... | 32 |
| 4.1 CIRCUITOS ARITMÉTICOS..... | 32 |
| 4.2 COMPARADORES | 34 |
| 4.3 MULTIPLEXORES (MUX)..... | 36 |
| 4.4 DECODIFICADORES | 39 |
| CAPÍTULO 5. SIST. COMBINATORIOS - SIMULACIÓN | 44 |
| 5.1 SUMADOR DE 4 BITS..... | 44 |

| | |
|--|----|
| 5.2 DECODIFICADOR DECIMAL A BCD | 44 |
| 5.3 DECODIFICADOR BCD A DECIMAL | 45 |
| 5.4 DECODIFICADOR DE 7 SEGMENTOS | 46 |
| CAPÍTULO 6. SIST. COMBINATORIOS - LABORATORIO | 47 |
| 6.1 EQUIPO Y MATERIAL NECESARIO..... | 47 |
| 6.2 SUMADOR DE 4 BITS..... | 47 |
| 6.3 DECODIFICADOR DECIMAL A BCD | 48 |
| 6.4 DECODIFICADOR BCD A DECIMAL | 48 |
| 6.5 DECODIFICADOR DE 7 SEGMENTOS | 49 |
| CAPÍTULO 7. SISTEMAS SECUENCIALES - TEORÍA | 51 |
| 7.1 CELDA DE MEMORIA BINARIA..... | 52 |
| 7.2 EL FLIP FLOP O BÁSCULA | 52 |
| 7.3 CONTADORES..... | 56 |
| 7.4 REGISTROS DE CORRIMIENTO | 58 |
| CAPÍTULO 8. SISTEMAS SECUENCIALES - SIMULACIÓN | 60 |
| 8.1 CONTADOR DECIMAL DE 0 A 99 | 60 |
| 8.2 REGISTRO DE DESPLAZAMIENTO L/R | 60 |
| CAPÍTULO 9. SIST. SECUENCIALES - LABORATORIO | 62 |
| 9.1 EQUIPO Y MATERIAL NECESARIO..... | 62 |
| 9.2 CONTADOR DECIMAL DE 0 A 99 | 62 |
| 9.3 REGISTRO DE DESPLAZAMIENTO L/R | 63 |
| CAPÍTULO 10. CONTROLADORES DIGITALES - TEORÍA..... | 65 |
| 10.1 MÓDULOS DEL CONTROLADOR | 65 |
| 10.2 OBJETIVO DEL CONTROLADOR | 67 |
| 10.3 DIAGRAMA DE FLUJO..... | 67 |
| 10.4 CIRCUITO QUE EVALÚA LA SUMA ACUMULADA..... | 68 |
| 10.5 DIAGRAMA DE FLUJO DEL CONTROLADOR..... | 69 |
| 10.6 DISEÑO DEL CONTROLADOR CON LATCH..... | 70 |
| 10.7 SOLUCIÓN AL PROBLEMA - AUTÓMATA | 71 |
| 10.8 IMPLEMENTACIÓN CON LÓGICA COMBINATORIA..... | 73 |
| 10.9 CON MULTIPLEXORES Y DECODIFICADOR A LA SALIDA. | 73 |
| CAPÍTULO 11. CONTROLADORES CON SHIFT REGISTER..... | 75 |

| | |
|--|----|
| 11.1 SOLUCIÓN AL PROBLEMA - AUTÓMATA | 76 |
| CAPÍTULO 12. DISEÑO CON CONTADORES..... | 81 |
| 12.1 PASOS DE DISEÑO | 81 |
| 12.2 SOLUCIÓN AL PROBLEMA - AUTÓMATA | 82 |
| CAPÍTULO 13. CONTROLADORES - SIMULACIÓN..... | 85 |
| CAPÍTULO 14. CONTROLADORES - LABORATORIO | 87 |
| BIBLIOGRAFÍA RECOMENDADA | 90 |
| SOBRE EL AUTOR | 92 |

INTRODUCCIÓN

Del Bit al Controlador Digital: Fundamentos y Práctica

La electrónica digital es el lenguaje que sustenta la era de la información. Desde los microcontroladores que automatizan procesos industriales hasta los sistemas de comunicaciones, los dispositivos médicos y los vehículos inteligentes, todo se reduce al procesamiento de señales discretas, la lógica binaria y el diseño de circuitos secuenciales. Sin embargo, dominar este campo requiere más que memorizar tablas de verdad o simplificando expresiones booleanas; exige comprender cómo se traducen los principios teóricos en circuitos reales, simulados y funcionales.

Este libro, "Electrónica Digital: Diseño y Aplicaciones", nace con el propósito de cerrar la brecha entre la teoría académica y la implementación práctica. Con más de 30 años de experiencia en la enseñanza universitaria de la ingeniería electrónica, he estructurado este texto para guiar al lector desde los fundamentos de las compuertas lógicas hasta el diseño de controladores digitales complejos, integrando en cada etapa la teoría, la simulación en Proteus y el montaje físico en protoboard.

Propósito del Libro

A lo largo de este texto, se busca no solo transmitir conocimiento teórico sobre familias lógicas, álgebra de Boole y circuitos secuenciales, sino también desarrollar la capacidad de analizar, simplificar, simular e implementar sistemas digitales reales. Cada capítulo ha sido diseñado para mantener un equilibrio entre el rigor matemático y la claridad expositiva, priorizando la resolución de problemas prácticos y la aplicación en contextos industriales y de automatización.

¿Qué Encontrará en Este Libro?

El contenido está organizado en tres bloques temáticos que cubren el espectro completo de la electrónica digital:

◆ Bloque 1: Compuertas Lógicas y Álgebra Digital (Capítulos 1-3)

Números binarios, hexadecimales y representación de datos. Compuertas AND, OR, NOT, NAND, NOR y EXOR: símbolos, tablas de verdad y ecuaciones. Familias TTL (7400, 7404, 7408, 7432), niveles lógicos y resistencia pull-up. Álgebra de Boole, teoremas de De Morgan y mapas de Karnaugh. Diseño eficiente con puertas NAND y simulación/laboratorio práctico. Aplicaciones reales: control de parqueadero y navegación de microrobot.

◆ Bloque 2: Sistemas Combinacionales y Secuenciales (Capítulos 4-7)

Circuitos aritméticos: semisumadores y sumadores completos (74283). Comparadores digitales (7485), multiplexores (74151) y decodificadores (74138, 74147, 7442, 7447). Flip-flops: SR, D, JK, Maestro-Esclavo (7476). Contadores síncronos y asíncronos (74160, 74161, 74191). Registros de corrimiento (74174, 74175, 74198). Simulación en Proteus e implementación en protoboard con clock 555.

◆ Bloque 3: Controladores Digitales (Capítulos 8-10)

Diseño de máquinas de estados con flip-flops D, registros y contadores. Metodología paso a paso: mapas de estado presente/siguiente, mapas de salida, simplificación. Caso práctico completo: controlador digital para máquina expendedora de líquidos. Implementación con registro 74LS194 y contador 74LS161. Simulación y laboratorio de verificación.

Enfoque Pedagógico

Cada capítulo incluye:

- Explicaciones teóricas claras que establecen los fundamentos conceptuales.
- Ejemplos numéricos resueltos que ilustran la simplificación booleana, el diseño de circuitos y el cálculo de tiempos.
- Simulaciones en Proteus que permiten validar el diseño antes de la implementación física.
- Prácticas de laboratorio con listas de materiales, diagramas de conexión y espacios para registro de mediciones.
- Aplicaciones prácticas que conectan la teoría con el control industrial, la robótica básica y la automatización.

Se ha puesto especial énfasis en el diseño eficiente (uso de compuertas NAND, minimización con Karnaugh) y en la metodología de diseño de controladores, una competencia esencial para la ingeniería de control y la electrónica industrial.

Para Quién es Este Libro

Este texto está dirigido principalmente a:

- Estudiantes de ingeniería eléctrica, electrónica, mecatrónica, automática y de sistemas que cursan asignaturas de electrónica digital, lógica programable o sistemas digitales.

- Técnicos y tecnólogos en automatización, mantenimiento industrial y electrónica aplicada.
- Profesionales que requieren actualizar o reforzar sus fundamentos en diseño de circuitos secuenciales y controladores digitales.
- Docentes que buscan material de apoyo con teoría, simulación y guías de laboratorio integradas.
- Aficionados a la electrónica y la robótica con bases matemáticas y físicas sólidas.

Requisitos Previos

Para aprovechar al máximo este texto, se recomienda tener conocimientos básicos de:

- Circuitos eléctricos de corriente continua y alterna.
- Números binarios, hexadecimales y operaciones aritméticas básicas.
- Álgebra elemental y lógica proposicional.
- Uso básico de instrumentos de laboratorio (multímetro, fuente DC, osciloscopio).

Conexión con la Serie Técnica

Este libro complementa directamente mis publicaciones anteriores en Amazon KDP:

"Fundamentos de Circuitos Eléctricos" y "Semiconductores, Diodos y Transistores" establecen la base analógica.

"Amplificadores Transistorizados" y "Osciladores y Multivibradores" cubren el procesamiento de señales analógicas.

"Señales y Sistemas Continuos/Discretos" y "Aprende Control con MATLAB" introducen el modelado matemático y la simulación computacional.

"Electrónica Digital" cierra el ciclo con la implementación hardware de lógica y control, puente ideal hacia "Control de Motores con MATLAB" y "Dispositivos de Control Electrónico".

Nota del Autor

Con base en mi experiencia como Profesor Titular, Decano y Rector en la Universidad Surcolombiana, y la publicación de cursos virtuales y libros técnicos, he recopilado en este volumen los conceptos esenciales que todo profesional debe dominar en el área de electrónica digital y automatización. Mi deseo es que este libro no solo le ayude a aprobar un examen, sino a resolver problemas reales de lógica y control con seguridad y eficiencia.

Invito al lector a abordar este material con mente abierta y espíritu crítico. No se limite a copiar diagramas; busque comprender el comportamiento de las señales, verifique las simulaciones, mida los niveles lógicos en el protoboard y, sobre todo, no dude en modificar parámetros y experimentar. La electrónica digital cobra vida cuando se observa su respuesta en el osciloscopio y se valida su funcionamiento en una aplicación real.

Agradezco de antemano cualquier comentario, sugerencia o corrección que los lectores deseen compartir, con el fin de enriquecer futuras ediciones de este texto.

Ing. Jorge Antonio Polanía Puentes

Neiva, Colombia

2025

CAPÍTULO 1. COMPUERTAS LÓGICAS - TEORÍA

Las compuertas lógicas son bloques que realizan las operaciones básicas de la aritmética binaria del **álgebra de Boole**, esto es, realizan las operaciones AND, OR, NOT, NAND, NOR, etc. Estas compuertas internamente están construidas en forma de circuitos integrados básicamente con transistores y resistencias. Son pequeños sistemas que al aplicarles una señal de entrada binaria se produce en la salida una respuesta también binaria que va a depender del tipo de compuerta.

En esta unidad usted aprenderá a conocer los números binarios, los símbolos de cada una de las compuertas, su tabla de verdad y sus ecuaciones lógicas básicas del Algebra de Boole y a resolver problemas lógicos aplicando esta tecnología junto a los Mapas de Karnaugh.

1.1 NÚMEROS BINARIOS

Los números binarios son la base para el estudio de la Electrónica Digital. Así como los números decimales tienen como base el número 10, pues, el número 346, es igual a:

$$345 = 5 * 1 + 4 * 10 + 3 * 100 = 5 * 10^0 + 4 * 10^1 + 3 * 10^2$$

Los números binarios tienen como base el número 2 y sólo tienen los números 0 y 1 y se llaman binarios. El número binario 111001 tiene su representación decimal de la siguiente forma:

| HEX | DEC | 8 | 4 | 2 | 1 |
|-----|-----|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 2 | 0 | 0 | 1 | 0 |
| 3 | 3 | 0 | 0 | 1 | 1 |
| 4 | 4 | 0 | 1 | 0 | 0 |
| 5 | 5 | 0 | 1 | 0 | 1 |
| 6 | 6 | 0 | 1 | 1 | 0 |
| 7 | 7 | 0 | 1 | 1 | 1 |
| 8 | 8 | 1 | 0 | 0 | 0 |
| 9 | 9 | 1 | 0 | 0 | 1 |
| A | 10 | 1 | 0 | 1 | 0 |
| B | 11 | 1 | 0 | 1 | 1 |
| C | 12 | 1 | 1 | 0 | 0 |
| D | 13 | 1 | 1 | 0 | 1 |
| E | 14 | 1 | 1 | 1 | 0 |
| F | 15 | 1 | 1 | 1 | 1 |

EJEMPLO

En el cuadro anterior, está la representación de los números binarios en formato Hexadecimal que son del 0 al 9 y 10=A, 11=B, 12=C, 13=D, 14=E y 15=F.

El número decimal 57, en formato binario es igual a:

$$57 = 32 + 16 + 8 + 1 = 1\ 1\ 1\ 0\ 0\ 1$$

Note que la ponderación binaria de derecha a izquierda es: 1, 2, 4, 8, 16, 32, 64, etc (potencias de 2)

En formato hexadecimal:

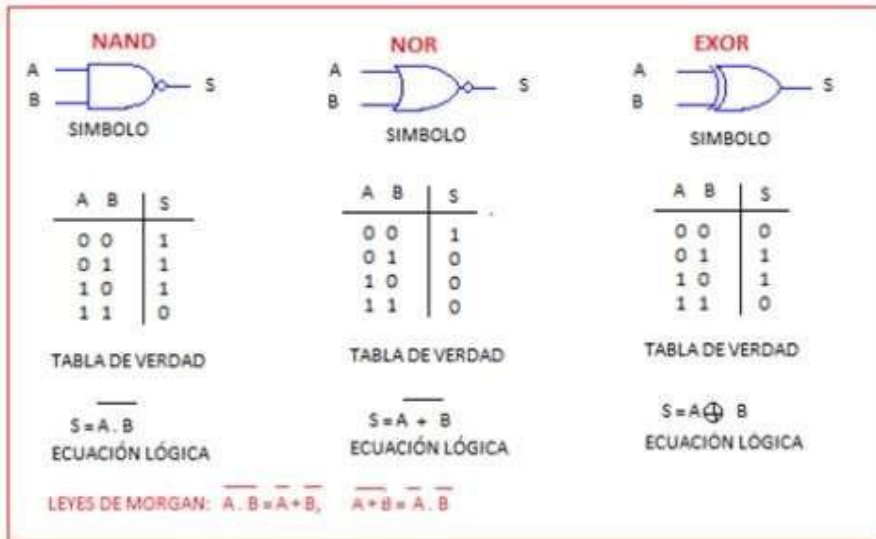
$$57_H = 0101(5)\ 0111(7) = 01010111$$

1.2 PUERTAS LÓGICAS BÁSICAS

Las puertas o compuertas lógicas básicas son: La puerta AND, la puerta OR y la puerta NOT. A continuación, se presenta su símbolo, la tabla de verdad que nos dice la salida dependiendo de la combinación de las entradas y su ecuación lógica. Observe que la puerta AND (Puerta Y) solo tiene una salida =1 o nivel alto si ambas entradas son 1. La puerta OR (puerta O) tiene una salida =1 si cualquiera o ambas de las entradas es 1. La puerta NOT o Inversor niega la entrada, esto es, si la entrada es 0 la salida es 1 y si es 1 la salida es 0.



Si a las compuertas anteriores se les niega la salida tenemos la configuración de otras compuertas como la NAND, la NOR. En el cuadro aparece su tabla de verdad con su ecuación lógica. La puerta EXOR es una puerta que se denomina OR EXCLUSIVO y su salida es 1 siempre que las entradas sean diferentes, o sea, si sus entradas son, 01 o 10 la salida es 1. Se puede decir que es un comparador. Si las entradas son diferentes la salida es 1 y si son iguales la salida es 0.



1.3 CIRCUITOS INTEGRADOS

| PUERTA | DESCRIPCIÓN | TTL |
|--------|--------------|------|
| AND | 4x2 ENTRADAS | 7408 |
| | 3x3 ENTRADAS | 7411 |
| | 2x4 ENTRADAS | 7421 |
| OR | 4x2 ENTRADAS | 7432 |
| NAND | 4x2 ENTRADAS | 7400 |
| | 3x3 ENTRADAS | 7410 |
| | 2x4 ENTRADAS | 7420 |
| NOT | 6x1 ENTRADA | 7404 |
| NOR | 4x2 ENTRADAS | 7402 |
| | 3x3 ENTRADAS | 7427 |
| | 2x4 ENTRADAS | 7425 |
| EX OR | 4x2 ENTRADAS | 7486 |

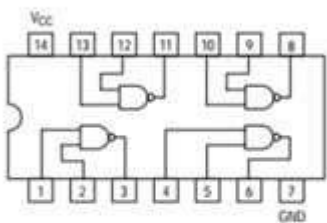
Las compuertas son fabricadas en circuitos integrados con base en transistores bipolares que se conoce con tecnología TTL (Transistor Transistor Logic), en donde en su interior pueden estar varias compuertas. Por ejemplo, el CI (circuito integrado) 7408 tiene en su interior 4 compuertas AND de 2 entradas, el 7411 tiene 3 puertas

AND de 3 entradas, el 7421 tiene 2 puertas AND de 4 entradas, el 7404 tiene 6 puertas NOT obviamente de 1 entrada, y así como se muestra en la siguiente tabla.

En el siguiente cuadro se presenta la distribución de las compuertas NAND en un circuito integrado. Observe que tiene 14 patas o pines, en donde la tierra (GND) es el pin 7 y la fuente (VCC) debe conectarse en el pin 14. La primera compuerta tiene como entradas los pines 1 y 2 y como salida el pin 3. Se puede apreciar la conexión de las demás compuertas en el integrado.

7400


Quad 2-Input NAND Gate



LOW POWER SCHOTTKY

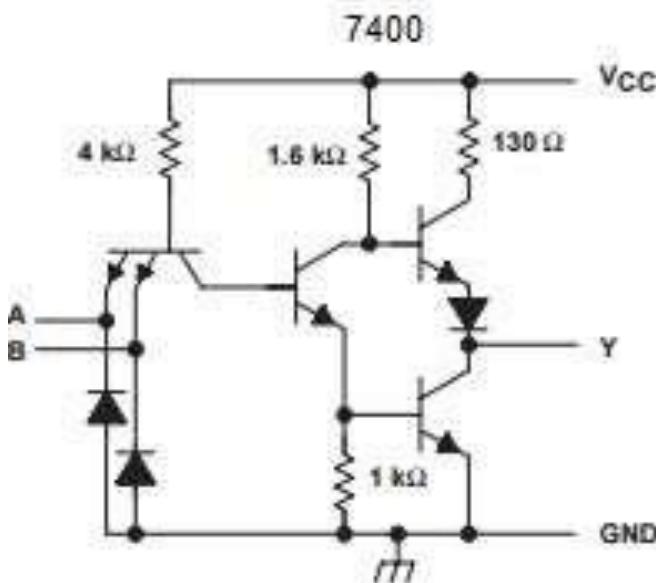
GUARANTEED OPERATING RANGES

| Symbol | Parameter | Min | Typ | Max | Unit |
|----------|-------------------------------------|------|-----|------|--------------------|
| V_{CC} | Supply Voltage | 4.75 | 5.0 | 5.25 | V |
| T_A | Operating Ambient Temperature Range | 0 | 25 | 70 | $^{\circ}\text{C}$ |
| I_{OH} | Output Current - High | | | -0.4 | mA |
| I_{OL} | Output Current - Low | | | 8.0 | mA |



<http://www.alldatasheet.com/datasheet-pdf/pdf/12608/ONSEMI/7400.html>

ESQUEMA ELÉCTRICO DE LA COMPUERTA 7400



CARACTERÍSTICAS ELÉCTRICAS

Bien importante es conocer las características eléctricas de los circuitos integrados. Me refiero a sus niveles de voltaje y corrientes. En el cuadro se observa,

- El nivel alto de voltaje de entrada (V_{IH}) es mínimo de 2V
- Nivel bajo de voltaje de entrada (V_{IL}) es máximo 0.8V
- El nivel alto de voltaje de salida (V_{OH}) es mínimo de 2.7V
- Nivel bajo de voltaje de salida (V_{OL}) es máximo 0.4V

| Symbol | Parameter | Limits | | | Unit | Test Conditions |
|----------|--|--------|-------|------|---------------|--|
| | | Min | Typ | Max | | |
| V_{IH} | Input HIGH Voltage | 2.0 | | | V | Guaranteed Input HIGH Voltage for All Inputs |
| V_{IL} | Input LOW Voltage | | | 0.8 | V | Guaranteed Input LOW voltage for All Inputs |
| V_{IK} | Input Clamp Diode Voltage | | -0.65 | -1.5 | V | $V_{CC} = \text{MIN}$, $I_{IK} = -18 \text{ mA}$ |
| V_{OH} | Output HIGH Voltage | 2.7 | 3.5 | | V | $V_{CC} = \text{MIN}$, $I_{OH} = \text{MAX}$, $V_{IK} = V_{IH}$ or V_{IL} per Truth Table |
| V_{OL} | Output LOW Voltage | | 0.25 | 0.4 | V | $I_{OL} = 4.0 \text{ mA}$ $V_{CC} = V_{CC} \text{ MIN}$, $V_{IK} = V_{IL}$ or V_{IH} per Truth Table |
| | | | 0.35 | 0.5 | V | |
| I_{IH} | Input HIGH Current | | | 20 | μA | $V_{CC} = \text{MAX}$, $V_{IK} = 2.7 \text{ V}$ |
| | | | | 0.1 | mA | $V_{CC} = \text{MAX}$, $V_{IK} = 7.0 \text{ V}$ |
| I_{IL} | Input LOW Current | | | -0.4 | mA | $V_{CC} = \text{MAX}$, $V_{IK} = 0.4 \text{ V}$ |
| I_{OS} | Short Circuit Current (Note 1) | -20 | | -100 | mA | $V_{CC} = \text{MAX}$ |
| I_{CC} | Power Supply Current Total, Output HIGH | | | 1.6 | mA | $V_{CC} = \text{MAX}$ |
| | Total, Output LOW | | | 4.4 | | |

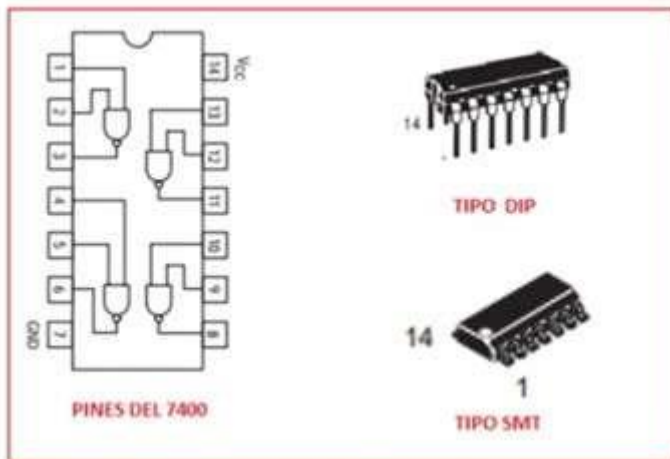
Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ($T_A = 25^\circ\text{C}$)

| Symbol | Parameter | Limits | | | Unit | Test Conditions |
|-----------|---------------------------------|--------|-----|-----|------|---|
| | | Min | Typ | Max | | |
| t_{PLH} | Turn-Off Delay, Input to Output | | 9.0 | 15 | ns | $V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$ |
| t_{PLL} | Turn-On Delay, Input to Output | | 10 | 15 | ns | |

TIPOS DE CIRCUITOS INTEGRADOS

Las puertas lógicas de tecnología bipolar TTL (Transistor Transistor Logic) o de tecnología MOSFET (Metal Oxido Semiconductor FET) vienen encapsuladas en circuitos integrados tipo DIP (paquete de doble línea) o tipo montaje superficial (SMT).



FAMILIAS DE CIRCUITOS INTEGRADOS

A continuación, se presentan las diferentes familias de circuitos integrados que se fabrican según el tipo de transistores utilizados ya sean BJT (bipolares) o del tipo MOS. Las puertas NAND pueden venir según sea su aplicación como: 7400, 74L00, 74S00, 74F00, etc

Se identifican con letras de referencia. Por ejemplo, 7400 tipo estándar, 74LS00 Schottky de baja potencia

L: TTL de baja potencia

LS: TTL Schottky de baja potencia

S: TTL Schottky

H: TTL de alta velocidad

AS: TTL Schottky avanzada

F: TTL Schottky avanzada de Fairchild

ALS: TTL Schottky avanzada de baja potencia

HC: CMOS de alta velocidad

HCT: CMOS de alta velocidad con entradas TTL

ACT: CMOS avanzada FACT con entrada TTL

AC: CMOS avanzada tecnología FACT

FCT: CMOS Tecnología FACT avanzada compatible TTL

NIVELES LÓGICOS DE CIRCUITOS INTEGRADOS

Los niveles lógicos de voltaje (nivel del cero y del 1), varían de acuerdo con la familia como se indica a continuación.

| TTL | | |
|------------------------------|--|--|
| FAMILIA | ENTRADA | SALIDA |
| TODAS | $V_{IH} = 2.0$ a $5.0V$ $V_{IL} = 0.0$ a $0.8V$ | $V_{OH} = 2.4$ a $5.0V$ $V_{OL} = 0.0$ a $0.4V$ |
| ESTÁNDAR | $I_{IH} = 40\mu A$ $I_{IL} = 1.6mA$ | $I_{OH} = 400\mu A$ $I_{OL} = 16mA$ |
| BAJA POTENCIA (LS) | $I_{IH} = 20\mu A$ $I_{IL} = 400\mu A$ | $I_{OH} = 400\mu A$ $I_{OL} = 8mA$ |
| AVANZADO BAJA POTENCIA (ALS) | $I_{IH} = 20\mu A$ $I_{IL} = 100\mu A$ | $I_{OH} = 400\mu A$ $I_{OL} = 8mA$ |
| AVANZADO FAST (AH) | $I_{IH} = 20\mu A$ $I_{IL} = 0.6mA$ | $I_{OH} = 1mA$ $I_{OL} = 20mA$ |
| CMOS | | |
| ALTA VELOCIDAD (HC) | $I_{IH} = 1\mu A$ $I_{IL} = 1\mu A$ | $I_{OH} = 4mA$ $I_{OL} = 4mA$ |
| AVANZADA FACT (AC/ACT) | $I_{IH} = 1\mu A$ $I_{IL} = 1\mu A$ | $I_{OH} = 24mA$ $I_{OL} = 24mA$ |

RESISTENCIA PULL UP DE ENTRADA

La resistencia que se debe conectar a la entrada de un circuito integrado depende del tipo de familia por sus niveles de voltaje como se indica en la siguiente tabla.

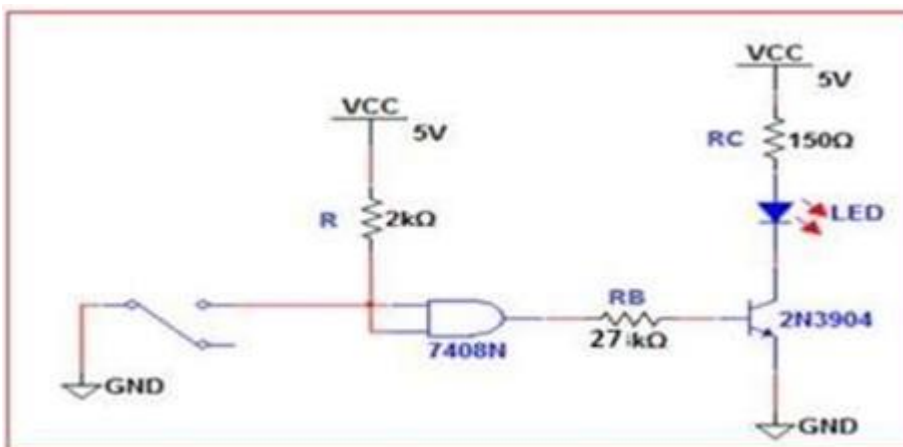
| SERIE TTL | MAX I _{IH} | MAX I _{IL} | RESISTENCIA |
|-------------|---------------------|---------------------|-------------|
| 54/74 | 40uA | -1.6mA | 4KΩ |
| 54ALS/74ALS | 20uA | -0.1mA | 40KΩ |
| 54AS/74AS | 20uA | -0.5mA | 8KΩ |
| 54H/74H | 50uA | -2.0mA | 2.8KΩ |
| 54LS/74LS | 20uA | -0.4mA | 18KΩ |
| 54S/74S | 50uA | -2.0mA | 2.8KΩ |

EJEMPLO

En el circuito de la figura, hallar la R_B que se debe colocar en la base del transistor.

Solución

Según la tabla anterior para un CI estándar la resistencia de entrada es de 4KΩ, pero como son dos entradas se coloca una de 2KΩ. Para el diodo: $V_D=2V$, $I_D=20mA$ para el transistor: $V_{CEsat} = 0.2V$, $\beta \approx 200$, $V_{BEsat} = 0.7V$



$$V_D = 2V, I_D = 20mA, V_{CESAT} = 0.2V, \beta \approx 200, V_{BESAT} = 0.7V$$

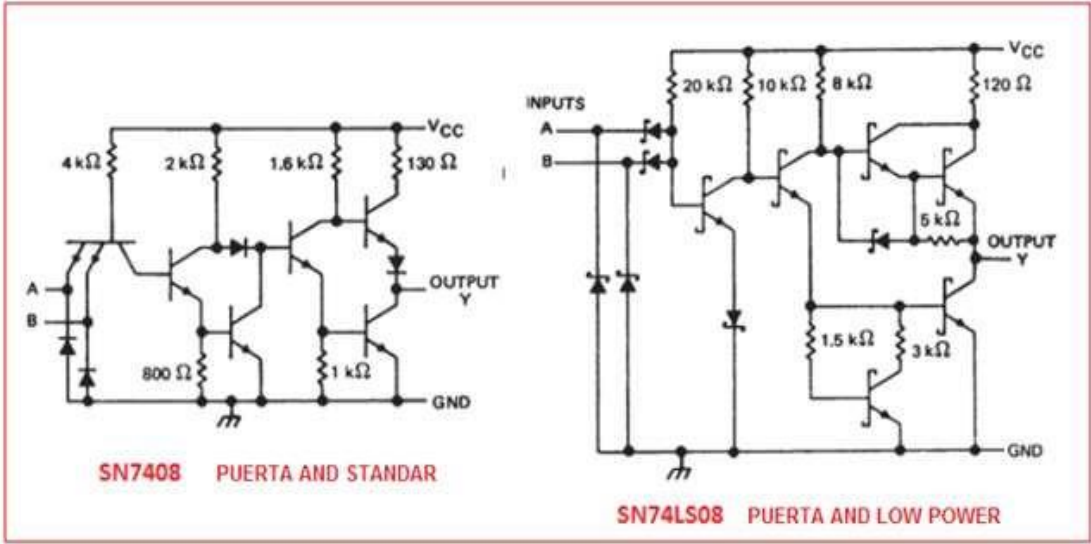
$$R_C \approx \frac{5V - 2V}{20mA} = 0.15K = 150\Omega \quad \frac{I_C}{\beta} = \frac{20mA}{200} = 0.1mA$$

Para que haya saturación del transistor es necesario que:

$$I_B > \frac{I_C}{\beta}, \text{ entonces, } R_B < \frac{5 - 0.7}{0.1} = 30K, \text{ se escoge } 27K$$

Respuesta: RB=27KΩ

DIAGRAMA DE LAS PUERTAS LÓGICAS



1.4 ALGEBRA DE BOOLE

George Boole británico nació en 1815 y murió en 1864. Como inventor del **álgebra de Boole**, que marca los fundamentos de la aritmética computacional moderna, Boole es considerado como uno de los fundadores del campo de las Ciencias de la computación. Se podría decir que es el padre de las operaciones lógicas y gracias a su álgebra hoy en día es posible manipular operaciones lógicas. Su teoría se base en postulados y teoremas lógicos.

En la lógica proposicional, las leyes de De Morgan (Augustus De Morgan, británico, 1806-1871) son un par de reglas de transformación que permiten la expresión de las conjunciones y disyunciones en términos de negación.

Las reglas se pueden expresar en español como:

1. La negación de la conjunción es la disyunción de las negaciones.
2. La negación de la disyunción es la conjunción de las negaciones.

En los siguientes cuadros se dan los Postulados y Teoremas básicos del Algebra de Boole. El apóstrofe significa entrada negada.

POSTULADOS

$A+0=A$, $A.1=A$,
 $0.A=0$, $1.A=A$
 $A.A=A$, $A.A'=0$,
 $(A')'=A$, $1+A=1$
 $0+A=A$, $A+A=A$,
 $A+A'=1$

TEOREMAS

$A+B=B+A$, conmutativas de la suma
 $A.B=B.A$, conmutativa de multiplicación
 $A+A.B=A.(1+B)=A.1=A$
 $A+A'.B=A+B$
 $A.B+A.B'=A.(B+B')=A.1=A$

LEYES DE MORGAN

$(A.B.C)'=A'+B'+C'$ $(A+B+C)'=A'.B'.C'$

EJEMPLO

Simplificar la siguiente expresión:

$$S=(A+AB).(B'+AB).[C'+(A.(A'+B'))+(B.(A'+B'))]+C.[C'+(A.(A'+B'))+(B.(A'+B'))]$$

Solución

$$S=(A.B'+A.AB+AB.B'+AB.AB).[C'+(AA'+A.B'+B.A'+BB')]+C.[C'+(AA'+A.B'+B.A'+BB')]$$

$$S=(A.B'+A.B+AB).[C'+(A.B'+B.A')]+C.[C'+(A.B'+B.A')]$$

$$S=(A.B'+A.B).[C'+(A.B'+B.A')]+C.[C'+(A.B'+B.A')]$$

$$S=(A.B'+A.B)C'+(AB'+AB).(AB'+BA')+C.C'+C'.(AB'+BA')$$

$$S=(A.B'+AB)C'+C(A.B'+A'.B)$$

$$S=AB'C'+ABC'+AB'CA'+A'BC$$


TABLAS DE VERDAD

Las Tablas de verdad es un recurso lógico para plasmar las ecuaciones lógicas en la tabla, siguiendo la metodología que una variable negada en la tabla es un 0 y una no negada es un 1. Por ejemplo,

Es la forma de dar una salida verdadera del circuito lógico según la función lógica o booleana que represente al circuito. Por ejemplo para el ejercicio anterior donde,

$$S = AB'C + ABC' + AB'C + A'BC$$

S=1 para las entradas lógicas dadas en la tabla de verdad con color rojo.



| ENTRADAS | | | SALIDA |
|----------|---|---|--------|
| A | B | C | S |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$A'BC$
 $AB'C'$
 $AB'C$
 ABC'

1.5 CIRCUITOS LÓGICOS

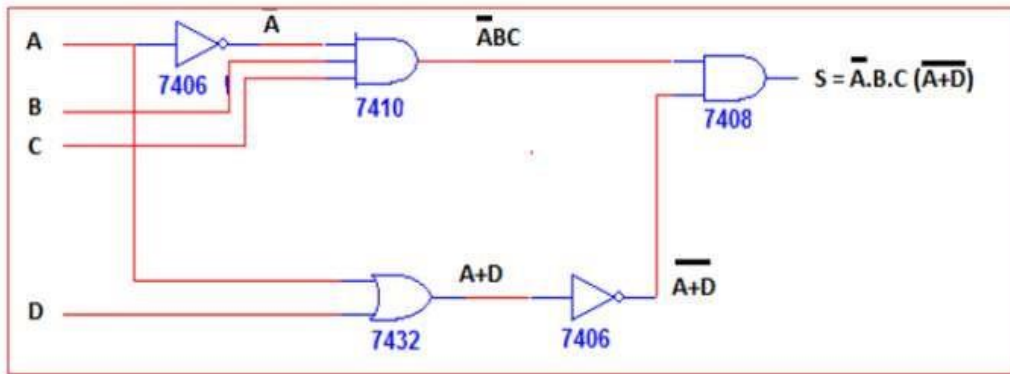
A partir de una expresión Booleana o ecuación lógica se puede implementar los circuitos con las puertas lógicas estudiadas. Por ejemplo, implementar la ecuación Booleana: $S = A'BC(A+D)'$

$A'BC$: se puede implementar con una puerta AND de 3 entradas A' , B y C

A' : requiere una NOT o inversor

$(A+D)'$: requiere una OR con entradas A, D y luego un inversor o puerta NOT

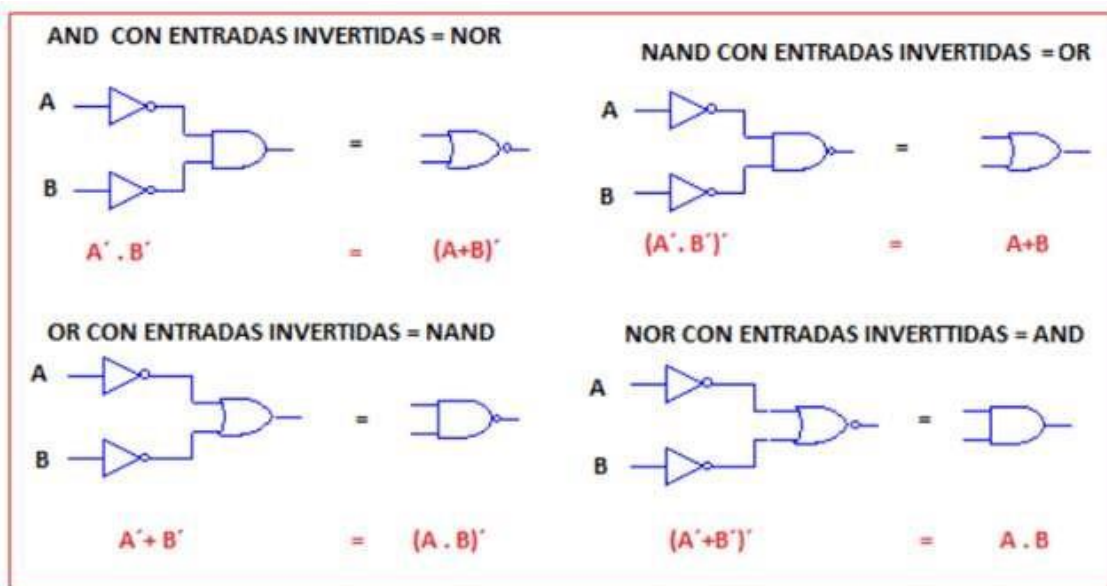
Estas dos salidas son las entradas de una AND con entradas $A'BC$ y $(A+D)'$



1.6 CONVERSIÓN DE PUERTAS LÓGICAS

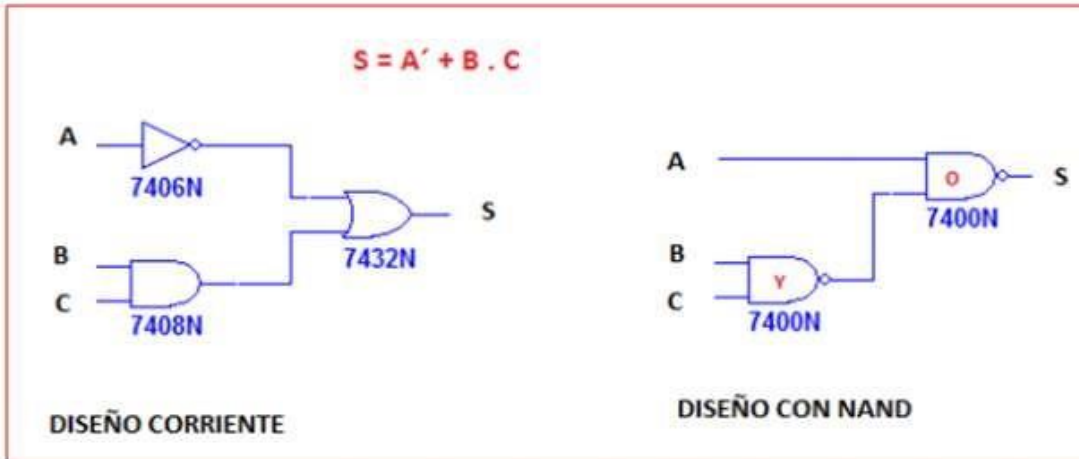
Para que el diseño de un circuito lógico sea más eficiente (utilizar menos circuitos integrados) se requiere con frecuencia convertir una puerta lógica en otra. Aquí se presentan algunas de ellas:

- Una NOR es una AND con entradas invertidas
- Una OR es una NAND con entradas invertidas
- Una NAND es una OR con entradas invertidas
- Una AND es una NOR con entradas invertidas



1.7 DISEÑO CON PUERTAS NAND

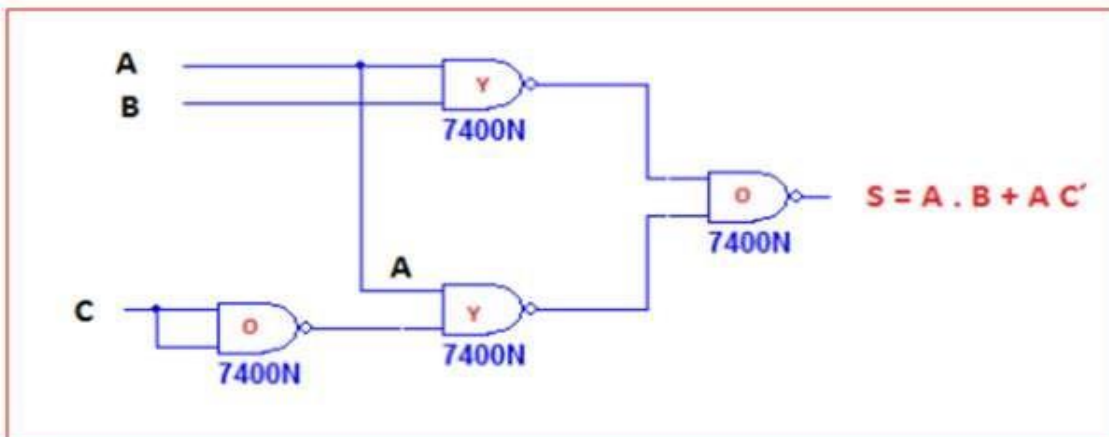
Es muy frecuente diseñar circuitos lógicos utilizando solamente puertas NAND. Por ejemplo, implementar la ecuación lógica $S=A'+B.C$ con puertas NAND. Observe que en el diseño corriente se necesitan tres CI, mientras que en el segundo diseño se necesita solamente un CI.



EJEMPLO

Implementar con puertas NAND la ecuación booleana: $S=A.B+A.C'$

De salida hacia la entrada el primer nivel de puertas es OR, el segundo nivel es NAND, el tercer nivel es nuevamente OR, etc. Observe para el último nivel de OR se debe invertir las entradas.



1.8 MAPAS DE KARNAUGH

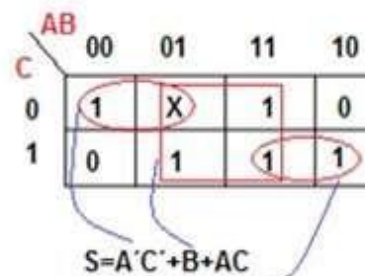
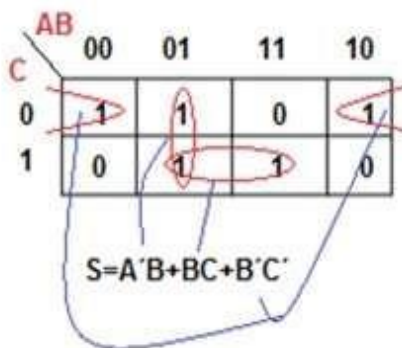
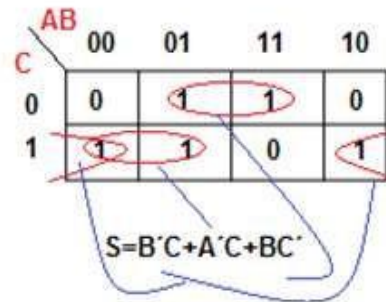
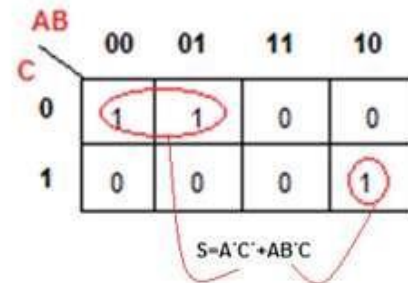
Los mapas de Karnaugh reducen la necesidad de hacer cálculos extensos para la simplificación de expresiones booleanas, aprovechando la capacidad del cerebro humano para el reconocimiento de patrones y otras formas de expresión analítica, permitiendo así identificar y eliminar condiciones muy inmensas.

Son mapas que se utilizan para simplificar circuitos, de una forma más sencilla que si se utiliza directamente el álgebra de Boole. Cada término de la función lógica es colocada en la casilla correspondiente: 0 corresponde a variable negada y 1 a variable no negada. $A'BC' = 010$

Las reglas de simplificación son las siguientes:

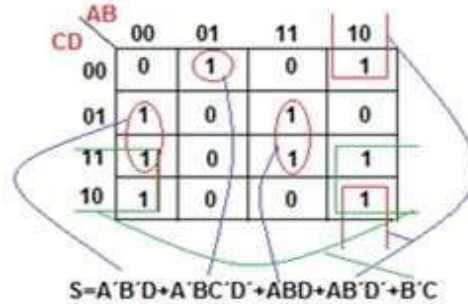
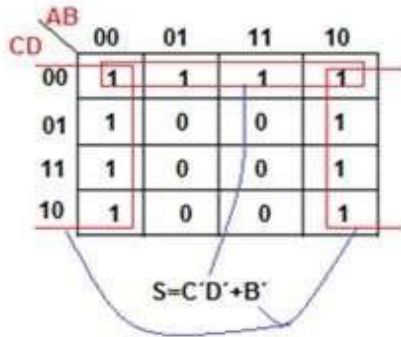
- Enlaces de dos elimina una variable
- Enlaces de cuatro elimina dos variables
- Enlaces de ocho elimina tres variables, etc...

MAPAS DE TRES VARIABLES



X: Don't care: indiferente

MAPAS DE CUATRO VARIABLES



EJEMPLO DE APLICACIÓN

Diseñar un circuito lógico que controle la puerta de un ascensor en un edificio de tres pisos. El circuito tiene cuatro entradas y una salida. Las entradas P1, P2, P3 indican el piso en que se está (P=1) y M el movimiento del ascensor (M=1 cuando se mueve). La salida S indica cuando la puerta se abre (S=1).

M=0 ascensor detenido

M=1 ascensor en movimiento

S=0 la puerta está cerrada

S=1 la puerta se abre

Son tres pasos:

1. Tabla de verdad
2. Mapa de Karnaugh
3. Circuito lógico

M P1 P2 P3 S

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | X |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | X |
| 0 | 1 | 1 | 1 | X |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

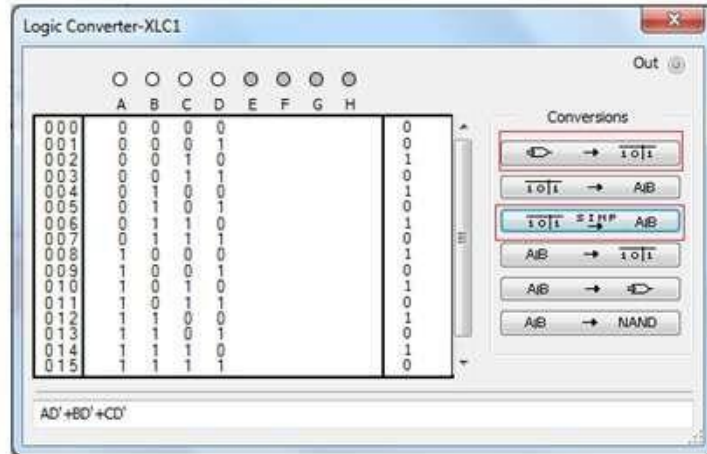
TABLA DE VERDAD

MAPA DE KARNAUGH

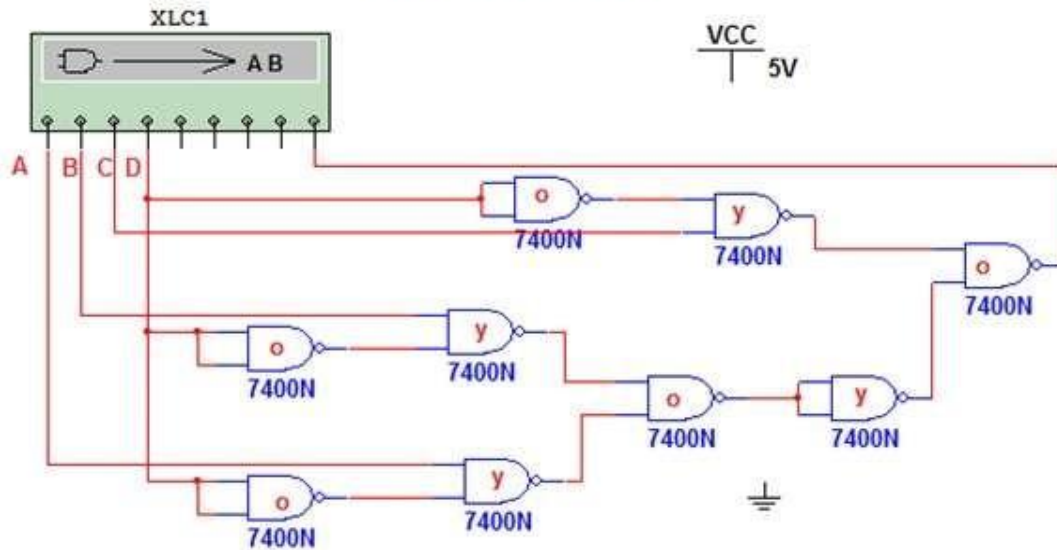
| | | | | |
|-----|------|----|----|----|
| | P2P3 | | | |
| MP1 | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | x | 1 |
| 01 | 1 | x | x | x |
| 11 | 0 | x | x | x |
| 10 | 0 | 0 | x | 0 |

$$S = M'P1 + M'P3 + M'P2 = M'(P1 + P2 + P3)$$

SIMULACIÓN CON MULTISIM



CIRCUITO LÓGICO



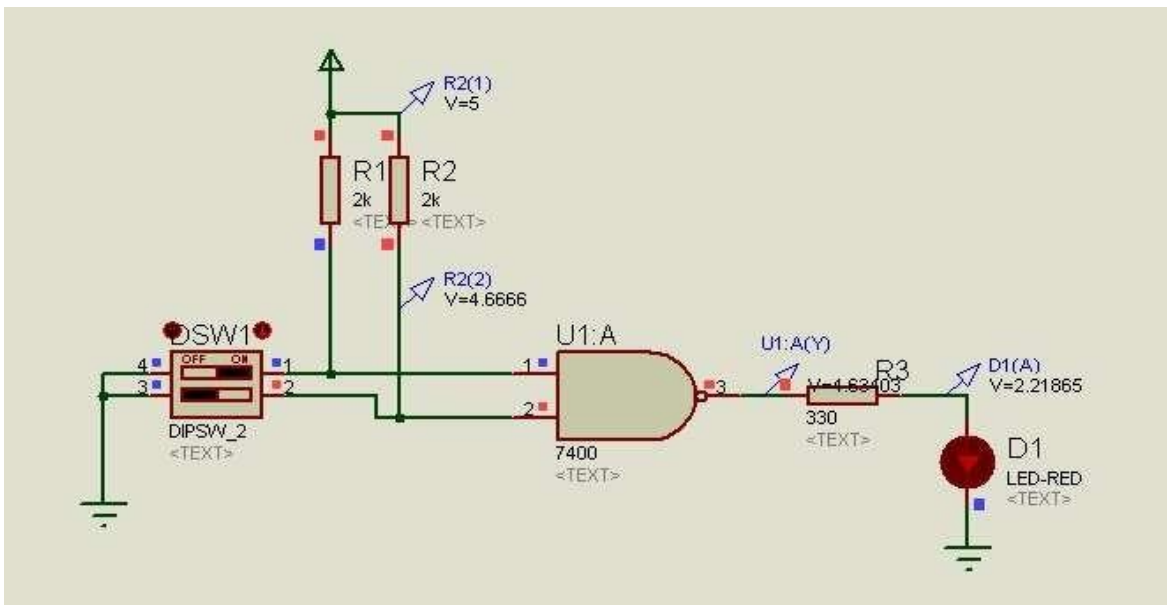
A=P1, B=P2, C=P3, D=M

CAPÍTULO 2. COMPUERTAS LÓGICAS - SIMULACIÓN

Vamos a simular con Proteus los siguientes circuitos digitales utilizando las compuertas lógicas básicas.

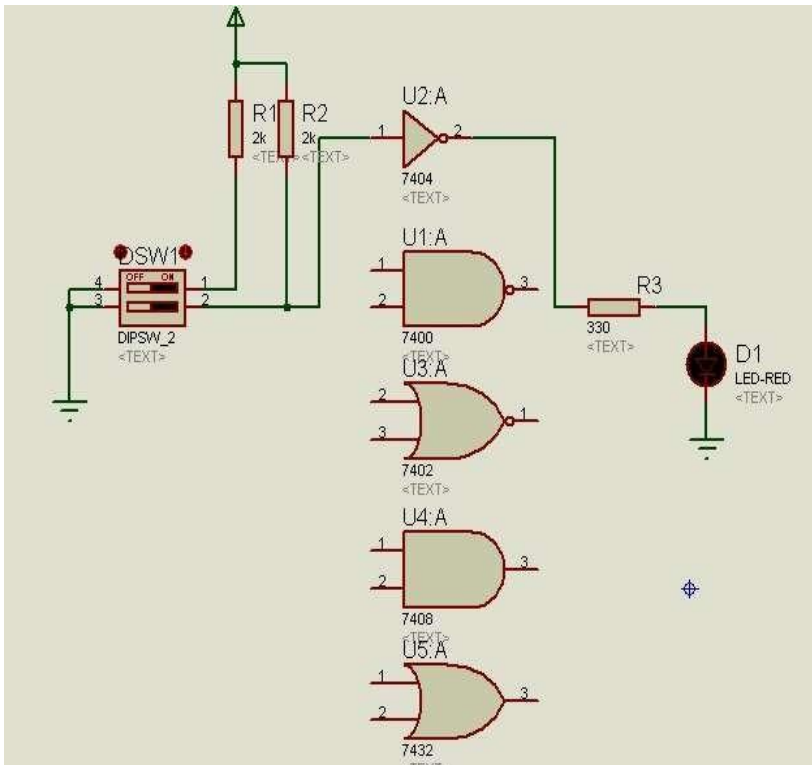
2.1 CARACTERÍSTICAS ELÉCTRICAS

Realice el circuito de la figura y mida los niveles eléctricos de entrada y salida de la compuerta 7400 (puerta NAND): I_{iL} , I_{iH} , V_{iL} , V_{iH} , I_{oL} , I_{oH} , V_{oL} , V_{oH} . Compruebe que los valores de R1, R2 y R3 son los correctos.



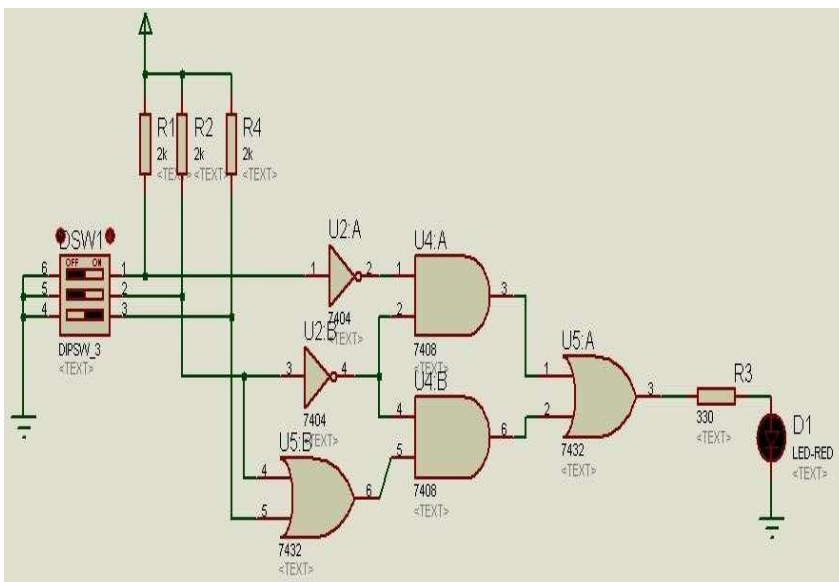
2.2 ECUACIONES LÓGICAS DE LAS COMPUERTAS

Con el circuito siguiente comprobar la Tabla de Verdad de las compuertas, NOT, NAND, NOR, AND y OR. Escriba su ecuación lógica.



2.3 CIRCUITO LÓGICO CON COMPUERTAS

Para el circuito de la figura, encuentre la Tabla de Verdad del circuito, con el Mapa de Karnaugh encuentre la ecuación o expresión lógica simplificada.



Repita el paso anterior diseñando el circuito con compuertas NAND solamente.

CAPÍTULO 3. COMPUERTAS LÓGICAS - LABORATORIO

El objetivo de la siguiente práctica es adquirir habilidades para que sean competentes en la realización de circuitos digitales usando las compuertas básicas AND, NAND, OR, NOR y NOT, comprobando sus características eléctricas, Tablas de Verdad, Mapa de Karnaugh y su ecuación lógica.

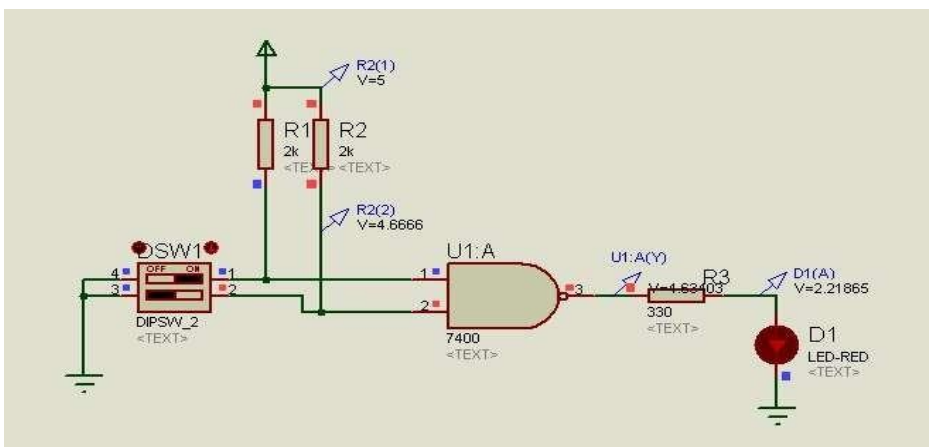
3.1 EQUIPO Y MATERIAL NECESARIO

Fuente de 5V

- Multímetro
- Protoboard
- Circuitos integrados: 7400, 7402, 7404, 7408, 7432
- DIPSWICH de 4 posiciones
- Un LED
- Resistencias a 1/4W de: 3x 2K Ω , 330 Ω
- Conectores

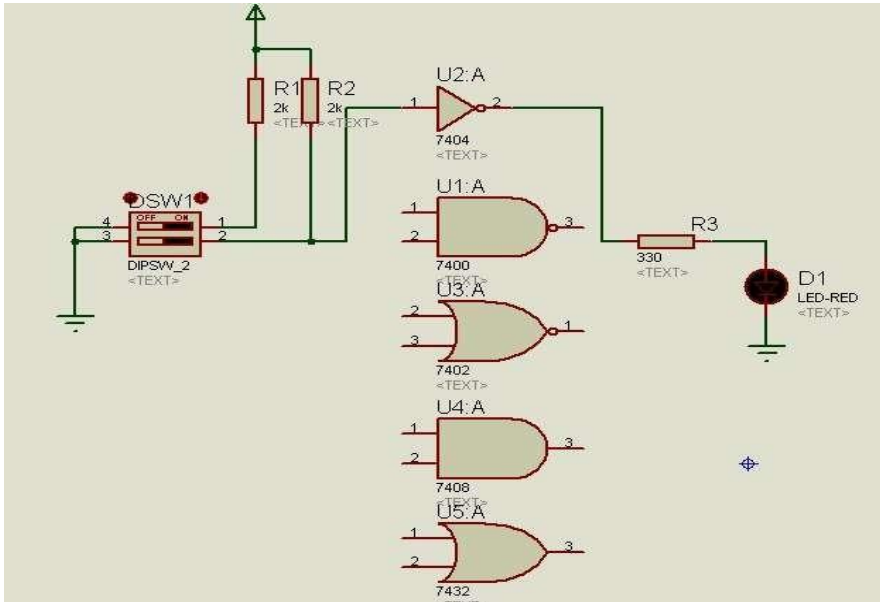
3.2 CARACTERÍSTICAS ELÉCTRICAS

Implemente el circuito de la figura y mida los niveles eléctricos de entrada y salida de la compuerta 7400 (puerta NAND): I_{iL} , I_{iH} , V_{iL} , V_{iH} , I_{oL} , I_{oH} , V_{oL} , V_{oH} . Compruebe que los valores de R1, R2 y R3 son los correctos.



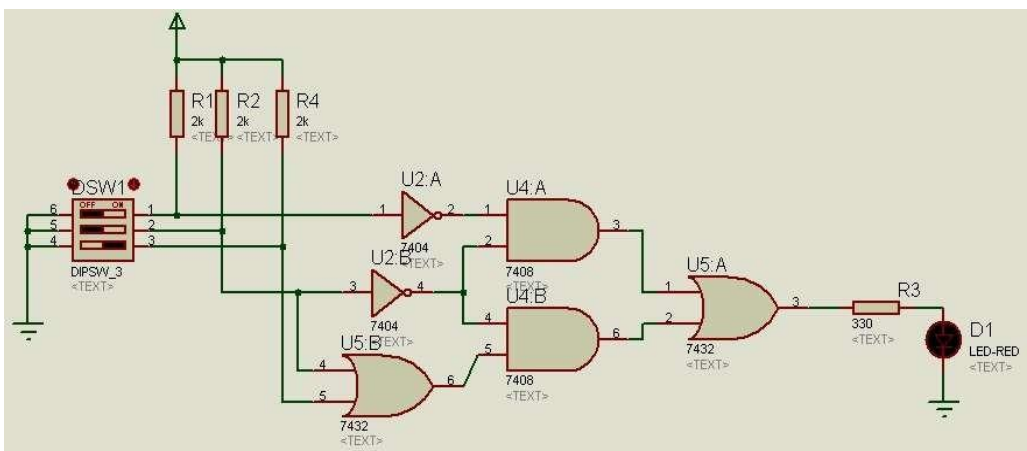
3.3 ECUACIONES LÓGICAS DE LAS COMPUERTAS

Con el circuito siguiente comprobar la Tabla de Verdad de las compuertas, NOT, NAND, NOR, AND y OR. Escriba su ecuación lógica.



3.4 CIRCUITO LÓGICO CON COMPUERTAS

Para el circuito de la figura, encuentre la Tabla de Verdad del circuito, con el Mapa de Karnaugh encuentre la ecuación o expresión lógica simplificada.



Repita el paso anterior diseñando el circuito con compuertas NAND solamente.

3.5 APLICACIÓN: PARQUEADERO

Diseñar un circuito lógico combinatorio para el control de entrada de autos en un parqueadero.



Entradas:

A: Detector de entrada del auto

B: Detector de retirada del ticket de entrada

C: Detector de presencia del auto que quiere entrar

D: Mando del operario para indicar "Parqueadero lleno" Salidas:

L1: Simula retirada del ticket

L2: Simula estado de la barra. Encendido está levantada, apagado está bajada

L3: Simula estado de la alarma

L4: Parqueadero lleno

a) Construcción de la Tabla de verdad

| D | C | B | A | L1 | L2 | L3 | L4 |
|---|---|---|---|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | X | X | X | 0 | 0 | 0 | 1 |

b) Obtención de las ecuaciones lógicas

Como las salidas son sencillas no es necesario realizar Mapas de Karnaugh

$L1 = D'CB'A' + D'CB'A = D'CB'(A' + A) = D'CB'$
 $L2 = D'CBA'$
 $L3 = D'CB'A$
 $L4 = D$

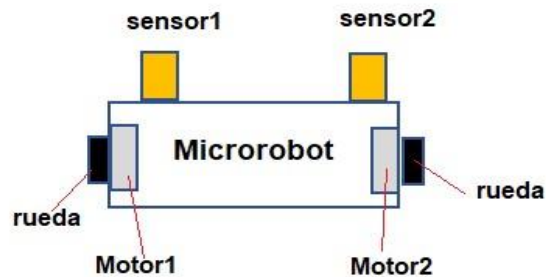
c) Implementación mediante compuertas lógicas

3.6 APLICACIÓN: MICROROBOT

Diseñar un circuito digital capaz de gobernar un Microrobot (móvil controlado) haciendo que este siga una pista negra pintada sobre un fondo blanco. El Microrobot tiene dos sensores (C1, C2) cuya salida es cero (=0) cuando leen blanco y uno (=1) cuando leen negro. Tiene dos motores de CC controlados por dos bits de esta forma:

| P | S | MOTOR |
|---|---|----------------|
| 0 | 0 | Parado |
| 0 | 1 | Parado |
| 1 | 0 | Giro izquierdo |
| 1 | 1 | Giro derecho |

P bit de poner (encender)
 S bit de giro (signo)



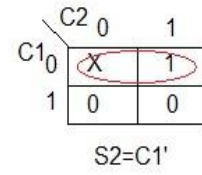
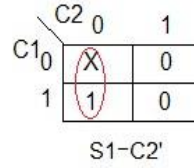
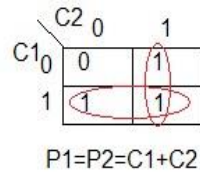
Algoritmo. Si los dos sensores detecten negro ($C1=C2=1$) el robot debe avanzar, si el sensor de la izquierda detecta negro ($C1=1$) y el de la derecha blanco ($C2=0$) girará a la izquierda y si sucede lo contrario girará a la derecha.

Sistema. El siguiente es el sistema por diseñar,

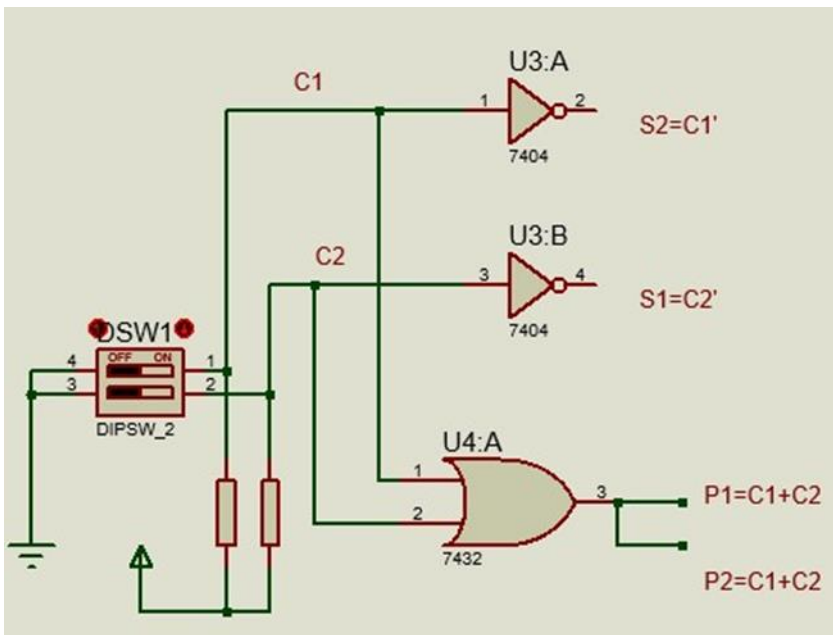


Tabla de verdad y mapas de Karnaugh.

| C1 | C2 | P1 | S1 | P2 | S2 |
|----|----|----|----|----|----|
| 0 | 0 | 0 | X | 0 | X |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |



Implementación.



CAPÍTULO 4. SISTEMAS COMBINACIONALES

En Electrónica digital se tienen sistemas combinatorios y sistemas secuenciales. Un sistema combinatorio es un arreglo de compuertas lógicas con un conjunto de entradas y salidas que transforma la información binaria de la entrada en otra salida binaria cuya correspondencia se describe mediante una tabla de verdad que muestre la relación binaria entre las variables de entradas y las variables de salida. Puede especificarse también con una ecuación lógica o Booleana, una por cada variable de salida. El análisis de un circuito combinatorio comienza con un diagrama de circuito lógico determinado y culmina con un conjunto de funciones booleanas o una tabla de verdad. En un sistema combinatorio la salida depende únicamente de las entradas presentes. En esta unidad se aprenderá a analizar los diferentes circuitos integrados que se encuentran como son los circuitos aritméticos, los comparadores digitales, los decodificadores, multiplexers, etc.

En esta unidad usted aprenderá a utilizar los diferentes circuitos integrados que se han fabricado para resolver problemas de la electrónica en forma digital con base en el análisis de sistemas combinatorios tales como: Circuitos aritméticos, comparadores digitales, multiplexores, decodificadores binarios y decodificadores de siete segmentos. Mucho entusiasmo porque de aquí comenzamos con la electrónica moderna la de la computación.

4.1 CIRCUITOS ARITMÉTICOS

1. SEMISUMADOR (HALF ADDER)

El semisumador (Half Adder) es un circuito combinatorio formado por compuertas que tiene como función sumador dos bits (dos entradas) y obtener como resultado la suma y el carry o lleva (dos salidas). A continuación, se presenta su tabla de verdad, su ecuación lógica, su símbolo y su circuito lógico.

TABLA DE VERDAD

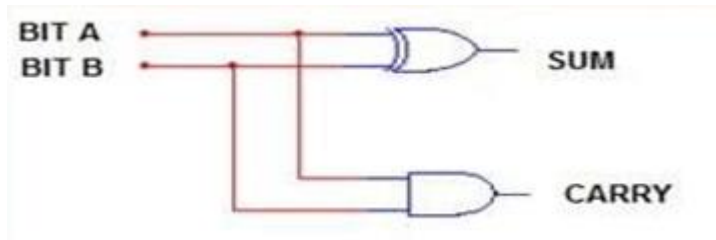
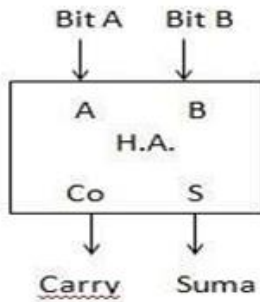
| A | B | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

S=SUMA C=CARRY

ECUACIÓN LÓGICA

$$S=A'B+AB'=A \oplus B$$

$$C=A.B$$



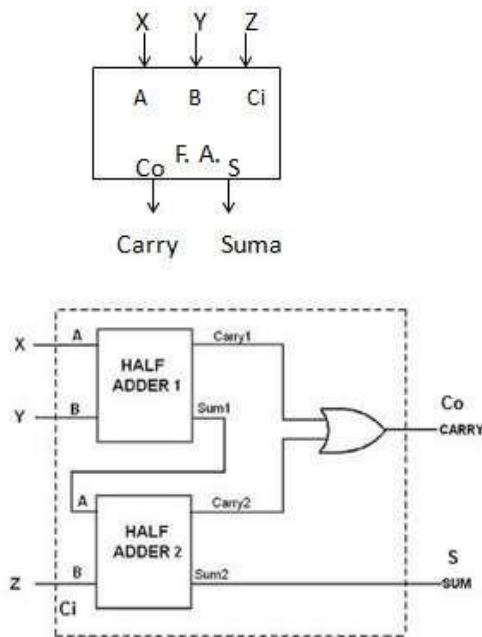
La suma es el Or-Exclusivo de las entradas y el carry es un AND.

2. SUMADOR COMPLETO (FULL ADDER)

Tiene tres entradas, los dos bits a sumar con el carry anterior (Ci) y dos salidas, la suma y el nuevo carry (Co). Abajo se tiene la tabla de verdad, el símbolo y el circuito lógico formado por dos semisumadores.

FULL ADDER
TABLA DE VERDAD

| A | B | Ci | S | Co |
|---|---|----|---|----|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



3. SUMADOR BINARIO DE 4 BITS

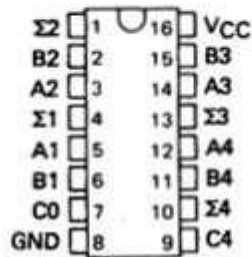
Este sumador es el circuito integrado 74283, suma dos números binarios de 4 bits con el carry anterior (Co), o sea, en total 9 entradas y se obtiene a la salida los 4

bits del resultado de la suma y el nuevo carry (C4), en total 5 salidas. Es un circuito integrado de 16 pines con la tierra (pin 8) y la fuente Vcc (pin 16). A continuación, se presenta la distribución de los pines del circuito, su diagrama y su símbolo lógico.

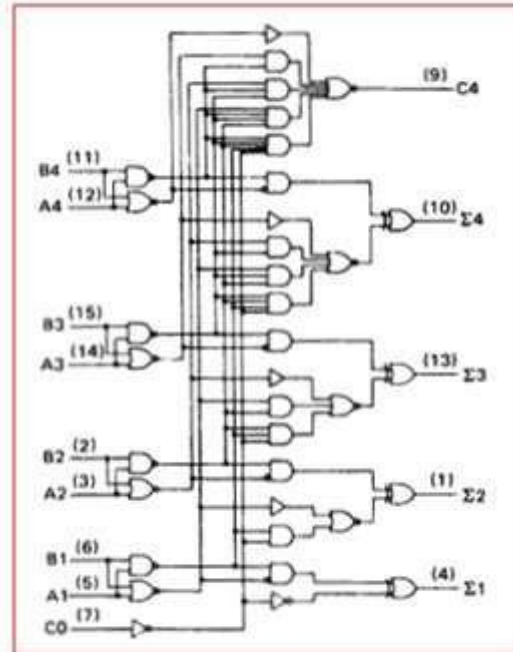
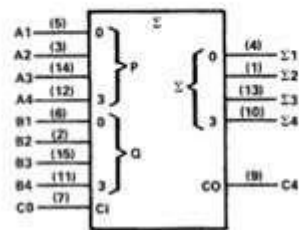
74283: SUMADOR BINARIO DE 4 BITS

DIAGRAMA LÓGICO

CIRCUITO INTEGRADO



SÍMBOLO LÓGICO



Por ejemplo, si $A = 1001$ ($A_4 A_3 A_2 A_1$) y $C_0 = 1$, $A = 9$ en decimal $B = 1101$ ($B_4 B_3 B_2 B_1$) $B = 13$ en decimal, el resultado es:

$S = 0111$ ($\Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1$), $C_4 = 1$, suma = $10111 = 16+4+2+1=23$

4.2 COMPARADORES

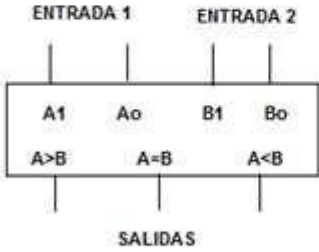
Un comparador digital compara dos números binarios activando su salida = 1 si es mayor, igual o menor tal como se indica en la tabla de verdad de abajo. En este caso tenemos dos números binarios cada uno de 2 bits.

Compara la magnitud relativa de dos números binarios. Su salida se activa si: $A < B$, $A = B$, $A > B$. A continuación se tiene un comparador de dos bits.

TABLA DE VERDAD

| A1 | A0 | B1 | B0 | A>B | A=B | A<B |
|----|----|----|----|-----|-----|-----|
| 0 | 0 | 0 | 0 | | 1 | |
| 0 | 0 | 0 | 1 | | | 1 |
| 0 | 0 | 1 | 0 | | | 1 |
| 0 | 0 | 1 | 1 | | | 1 |
| 0 | 1 | 0 | 0 | 1 | | |
| 0 | 1 | 0 | 1 | | 1 | |
| 0 | 1 | 1 | 0 | | | 1 |
| 0 | 1 | 1 | 1 | | | 1 |
| 1 | 0 | 0 | 0 | 1 | | |
| 1 | 0 | 0 | 1 | 1 | | |
| 1 | 0 | 1 | 0 | | 1 | |
| 1 | 0 | 1 | 1 | | | 1 |
| 1 | 1 | 0 | 0 | 1 | | |
| 1 | 1 | 0 | 1 | 1 | | |
| 1 | 1 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 1 | | 1 | |

SÍMBOLO LÓGICO



1. COMPARADOR DE 4 BITS

El circuito integrado 7485 es un comparador digital de dos números binarios de 4 bits como entradas (A0 A1 A2 A3, B0 B1 B2 B3).

EL 7485:COMPARADOR DE 4 BITS

El 7485 es un comparador de dos números (A,B) de 4 bits. Su salida se activa si: $A < B$, $A = B$, $A > B$.

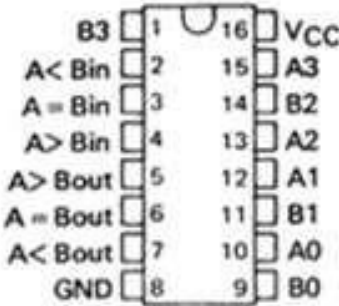
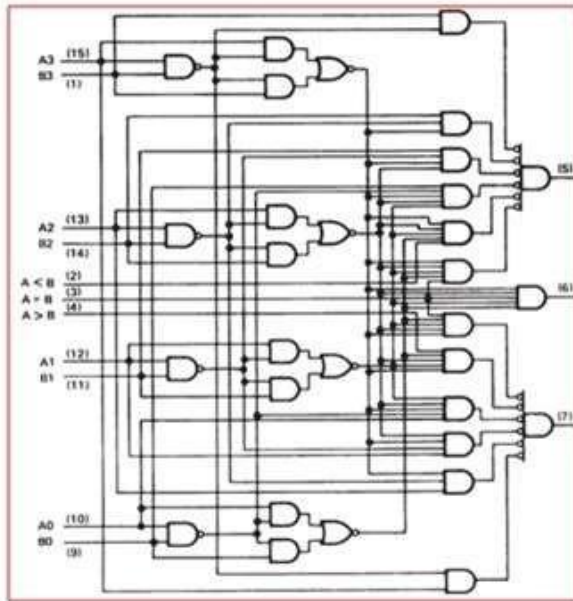


DIAGRAMA LÓGICO



| A | B | $I_{A<B}$ | $I_{A=B}$ | $I_{A>B}$ | $O_{A<B}$ | $O_{A=B}$ | $O_{A>B}$ |
|-------|---|-----------|-----------|-----------|-----------|-----------|-----------|
| A < B | * | * | * | | 1 | 0 | 0 |
| A > B | * | * | * | | 0 | 0 | 1 |
| A = B | 1 | 0 | 0 | | 1 | 0 | 0 |
| A = B | * | 1 | * | | 0 | 1 | 0 |
| A = B | 0 | 0 | 1 | | 0 | 0 | 1 |
| A = B | 0 | 0 | 0 | | 1 | 0 | 1 |
| A = B | 1 | 0 | 1 | | 0 | 0 | 0 |

4.3 MULTIPLEXORES (MUX)

Son circuitos digitales combinatorios que selecciona a la salida una de las posibles entradas a través de un pin o pines de selección.

1. MUX DE 2 A 1

En el circuito de abajo, se tiene un Mux de dos entradas D1 y D2, requiere solo un pin de selección S que es otra entrada, por tanto, en su tabla de verdad aparece 8 posibles combinaciones de la 000 a la 111 (entradas S D1 D0) y la salida W. Observe que cuando S= 0, W=D1, y cuando S= 1, W = D0.

Es un circuito combinacional diseñado para seleccionar una sola de las varias entradas a la salida.



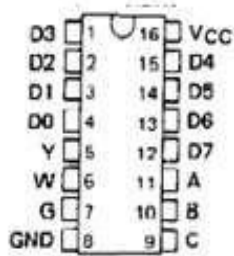
TABLA DE VERDAD

| S | D1 | D0 | W |
|---|----|----|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

S=0, selecciona D1, W=D1
S=1, selecciona D0, W=D0

2. MUX DE 8 A 1: 74151

El circuito integrado 74151 es un Mux de 8 a 1, esto quiere decir que de 8 posibles entradas D0...D7 podemos seleccionar una de ellas a través de tres pines de selección C, B, A. O sea que si CBA=101 estamos seleccionando D5 (lo que hay en esta entrada aparece en la salida Y). La otra salida W es la negación de Y. Este circuito necesita que el pin G (habilitador esté en 0 para su funcionamiento).



SÍMBOLO LÓGICO

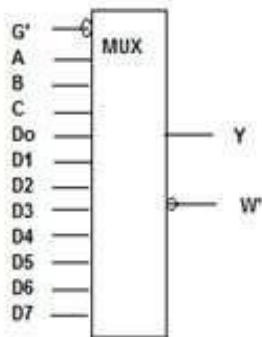


TABLA FUNCIONAL

| C | B | A | G' | Y | W |
|---|---|---|----|----------------|------------------|
| X | X | X | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | D ₀ | D ₀ ' |
| 0 | 0 | 1 | 0 | D ₁ | D ₁ ' |
| 0 | 1 | 0 | 0 | D ₂ | D ₂ ' |
| 0 | 1 | 1 | 0 | D ₃ | D ₃ ' |
| 1 | 0 | 0 | 0 | D ₄ | D ₄ ' |
| 1 | 0 | 1 | 0 | D ₅ | D ₅ ' |
| 1 | 1 | 0 | 0 | D ₆ | D ₆ ' |
| 1 | 1 | 1 | 0 | D ₇ | D ₇ ' |

D₀...D₇: datos de entrada
 C,B,A: Líneas de selección
 Y: Salida W: Salida negada
 G': Habilitador

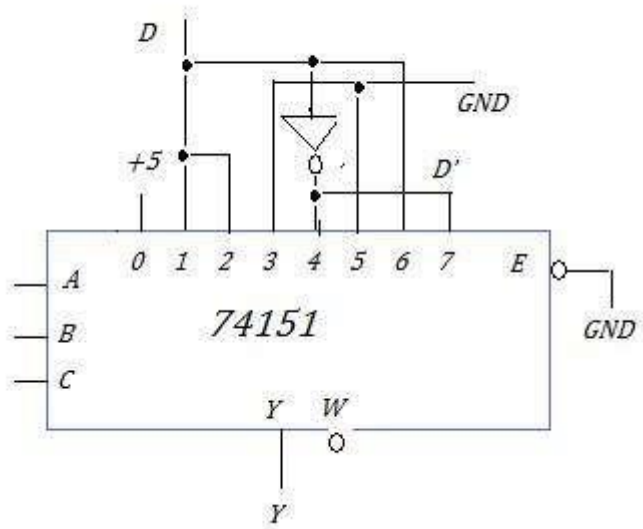
EJEMPLO

De la siguiente tabla de verdad, implemente la expresión usando un Mux de 8 a 1

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Hacer el mapa de Karnaugh haciendo uso del mapeo con variables de entrada MEV

| AB | 00 | 01 | 11 | 10 |
|-----|-----------------|----------------|-----------------|-----------------|
| C 0 | 1 ⁰ | D ² | D ⁶ | D' ⁴ |
| C 1 | D' ¹ | 0 ³ | D' ⁷ | 0 ⁵ |



EJEMPLO

Implementar con un Mux la siguiente función lógica:

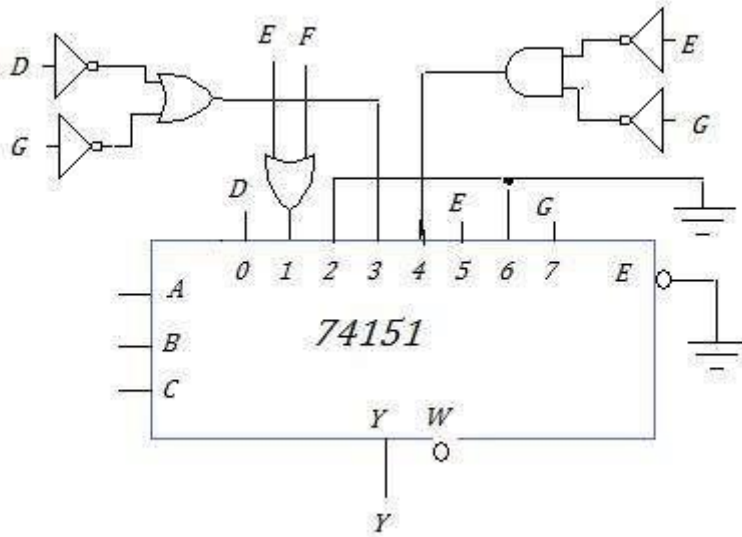
$$Y = A'BC'D + AB'CE + A'B'C(E+F) + ABCG + AB'C'(E'G') + A'BC(D'+G')$$

Se dejan las entradas A, B, C para seleccionar el Mux y las variables D,E,F,G como datos de entrada.

Solución

Separamos las variables de selección y las de entrada:

$$Y = A'BC'(D) + AB'C(E) + A'B'C(E+F) + ABC(G) + AB'C'(E'G') + A'BC(D+G')$$



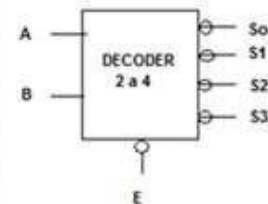
4.4 DECODIFICADORES

Los decodificadores son circuitos combinatorios que activa la salida que corresponde al número binario de la entrada. Por ejemplo, el circuito de abajo tiene dos entradas binarias (A y B), o sea, que puede ser 00 (=0), 01(=1), 10 (=2) o 11 (=3) la salida se activa colocando un 0 en la salida correspondiente So, S1, S2, S3. Para que funcione se requiere que el pin E=0.

TABLA DE VERDAD

| A | B | E | So | S1 | S2 | S3 |
|---|---|---|----|----|----|----|
| X | X | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |

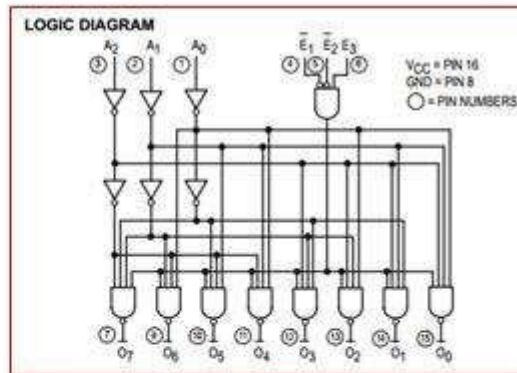
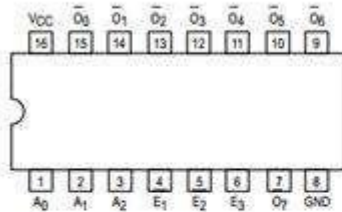
Un decodificador es un dispositivo que al aplicarle a la entrada un código binario se activa la salida que corresponde a ese código. Un decodificador de 2 a 4 (dos entradas y cuatro salidas). Entradas: 00, 01, 10, 11



1. DECODIFICADOR DE 3 A 8

El 74138 es un circuito integrado que decodifica un número binario de 3 bits a la entrada A0A1A2 en 8 salidas O0.....O7. La salida correspondiente se activa según el código presente en la entrada. Por ejemplo, si A2A1A0=011 (LHH) se activa O3=L.

74138: DECODER 1 DE 8



TRUTH TABLE

| INPUTS | | | OUTPUTS | | | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| E ₁ | E ₂ | E ₃ | A ₀ | A ₁ | A ₂ | O ₀ | O ₁ | O ₂ | O ₃ | O ₄ | O ₅ | O ₆ | O ₇ |
| H | X | X | X | X | X | H | H | H | H | H | H | H | H |
| X | H | X | X | X | X | H | H | H | H | H | H | H | H |
| X | X | L | X | X | X | H | H | H | H | H | H | H | H |
| L | L | H | L | L | L | L | H | H | H | H | H | H | H |
| L | L | H | H | L | L | H | L | H | H | H | H | H | H |
| L | L | H | L | H | L | H | H | L | H | H | H | H | H |
| L | L | H | H | H | L | H | H | H | L | H | H | H | H |
| L | L | H | L | L | H | H | H | H | L | H | H | H | H |
| L | L | H | H | L | H | H | H | H | H | L | H | H | H |
| L | L | H | L | H | H | H | H | H | H | H | L | H | H |
| L | L | H | H | H | H | H | H | H | H | H | H | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | H | L |

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

EJEMPLO

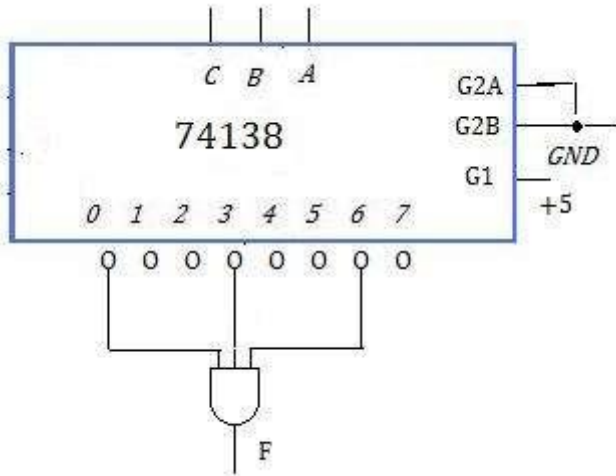
Implementar la función derivada de la siguiente tabla de verdad.

| | A | B | C | F |
|----|---|---|---|---|
| m0 | 0 | 0 | 0 | 0 |
| m1 | 0 | 0 | 1 | 1 |
| m2 | 0 | 1 | 0 | 1 |
| m3 | 0 | 1 | 1 | 0 |
| m4 | 1 | 0 | 0 | 1 |
| m5 | 1 | 0 | 1 | 1 |
| m6 | 1 | 1 | 0 | 0 |
| m7 | 1 | 1 | 1 | 1 |

Como hay menos ceros que unos se utiliza F':

$$F' = m_0 + m_3 + m_6$$

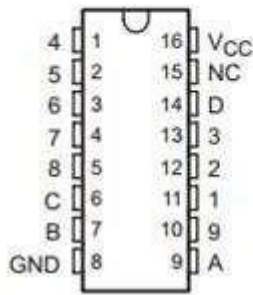
$$F = m_0' \cdot m_3' \cdot m_6' \text{ (Morgan)}$$



2. DECODIFICADOR DECIMAL

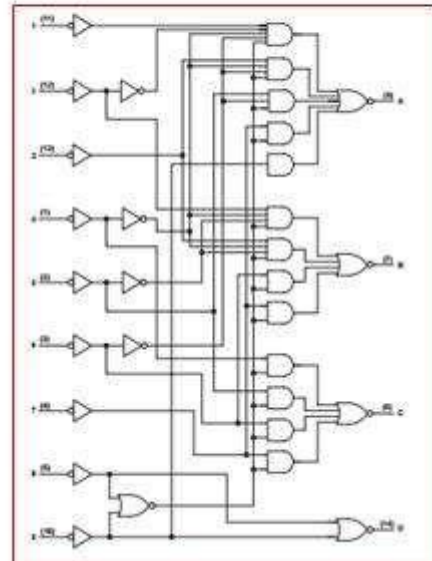
El 74147 es un decodificador decimal que tiene como entradas las líneas de 1 al 9 y como salidas el código binario que corresponde a la entrada activada. Funciona con lógica negativa (activa con nivel bajo).

74147: DECODER DE 10 (DECIMAL) A 4 (BINARIO)



| INPUTS | | | | | | | | | OUTPUTS | | | |
|--------|---|---|---|---|---|---|---|---|---------|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | D | C | B | A |
| H | H | H | H | H | H | H | H | H | H | H | H | H |
| X | X | X | X | X | X | X | X | L | L | H | H | L |
| X | X | X | X | X | X | X | L | H | L | H | H | H |
| X | X | X | X | X | X | L | H | H | H | L | L | L |
| X | X | X | X | L | H | H | H | H | H | L | L | H |
| X | X | X | L | H | H | H | H | H | H | L | H | L |
| X | X | L | H | H | H | H | H | H | H | H | L | L |
| X | L | H | H | H | H | H | H | H | H | H | L | H |
| L | H | H | H | H | H | H | H | H | H | H | H | L |

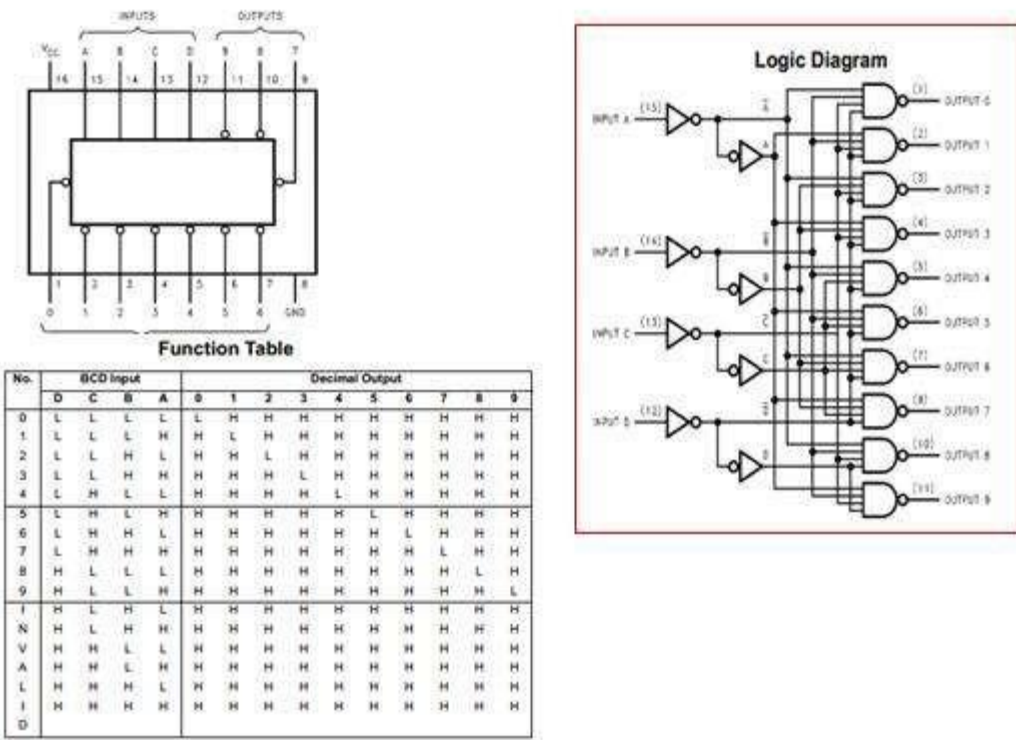
H = high logic level, L = low logic level, X = irrelevant



3. DECODIFICADOR BCD

BCD quiere decir Binario Codificado en Decimal. Este circuito 7442 es un decodificador cuya entrada es un número binario de 4 bits y su salida el correspondiente número decimal.

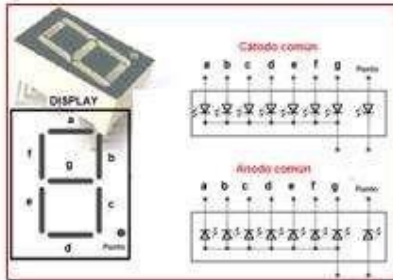
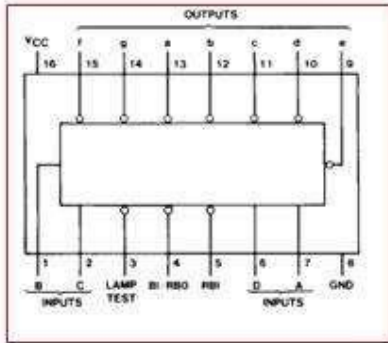
7442: DECODER DE BCD A DECIMAL



3. DECODIFICADOR BCD A 7 SEGMENTOS

Es un decodificador cuya entrada es un número binario BCD y su salida son las líneas (a, b, c, d, e, f, g) que se activan para formar el número decimal en un display de 7 segmentos.

7447: DECODER/DRIVER DE BCD A 7 SEGMENTOS



Function Table

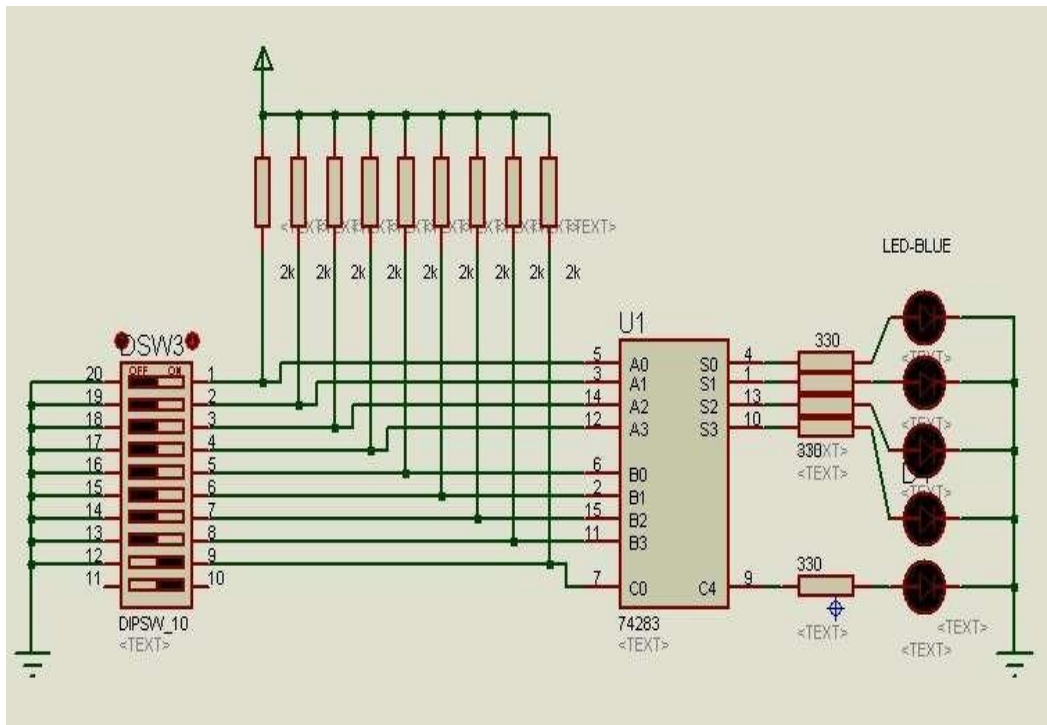
| Decimal or Function | Inputs | | | | | BI/RBO (Note 1) | Outputs | | | | | | | Note | |
|---------------------|--------|-----|---|---|---|-----------------|---------|---|---|---|---|---|---|------|----------|
| | LT | RBI | D | C | B | | A | a | b | c | d | e | f | | g |
| 0 | H | H | L | L | L | L | H | L | L | L | L | L | L | H | (Note 2) |
| 1 | H | X | L | L | L | H | H | L | L | L | H | H | H | H | |
| 2 | H | X | L | L | H | L | H | L | L | H | L | L | H | L | |
| 3 | H | X | L | L | H | H | H | L | L | L | L | H | H | L | |
| 4 | H | X | L | H | L | L | H | H | L | L | H | H | L | L | |
| 5 | H | X | L | H | L | H | H | L | H | L | L | H | L | L | |
| 6 | H | X | L | H | H | L | H | H | H | L | L | L | L | L | |
| 7 | H | X | L | H | H | H | H | L | L | L | H | H | H | H | |
| 8 | H | X | H | L | L | L | H | L | L | L | L | L | L | L | |
| 9 | H | X | H | L | L | H | H | L | L | L | H | H | L | L | |
| 10 | H | X | H | L | H | L | H | H | H | L | L | H | L | L | |
| 11 | H | X | H | L | H | H | H | H | H | L | L | H | H | L | |
| 12 | H | X | H | H | L | L | H | H | L | H | H | H | L | L | |
| 13 | H | X | H | H | L | H | H | L | H | H | L | H | L | L | |
| 14 | H | X | H | H | H | L | H | H | H | H | L | L | L | L | |
| 15 | H | X | H | H | H | H | H | H | H | H | H | H | H | H | |
| BI | X | X | X | X | X | X | L | H | H | H | H | H | H | H | (Note 3) |
| RBI | H | L | L | L | L | L | L | H | H | H | H | H | H | H | (Note 4) |
| LT | L | X | X | X | X | X | H | L | L | L | L | L | L | L | (Note 5) |

H = HIGH level, L = LOW level, X = Don't Care

CAPÍTULO 5. SIST. COMBINATORIOS - SIMULACIÓN

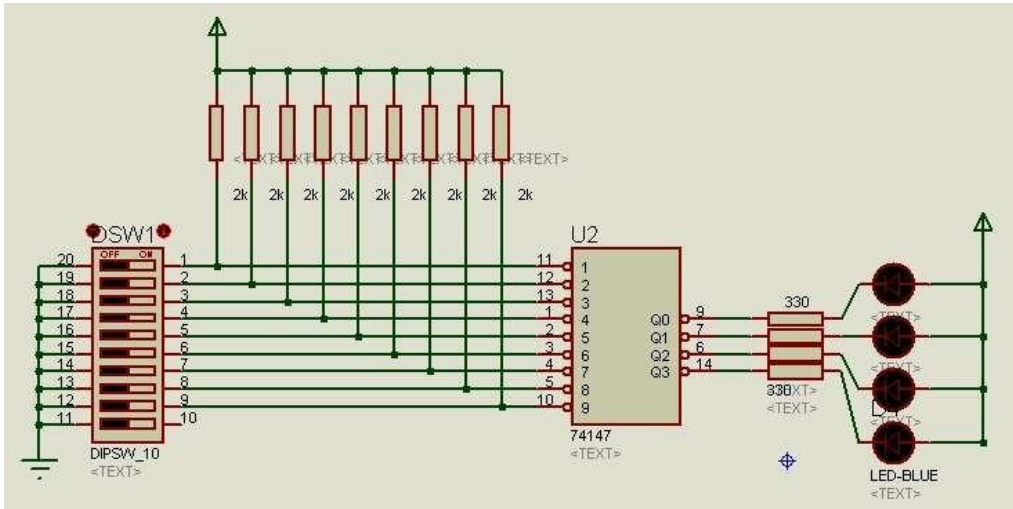
5.1 SUMADOR DE 4 BITS

Realice la simulación del circuito sumador de 4 bits usando el circuito integrado 74283, tal como se indica en la figura. Operando el Dipswitch cargue datos binarios para el sumador $A=A_3A_2A_1A_0$ y para el sumando $B=B_3B_2B_1B_0$ y opere la suma con carry de entrada C_0 . Observe el resultado de la operación en $S=S_3S_2S_1S_0$ y en el carry de salida C_4 . Compruebe estos resultados.



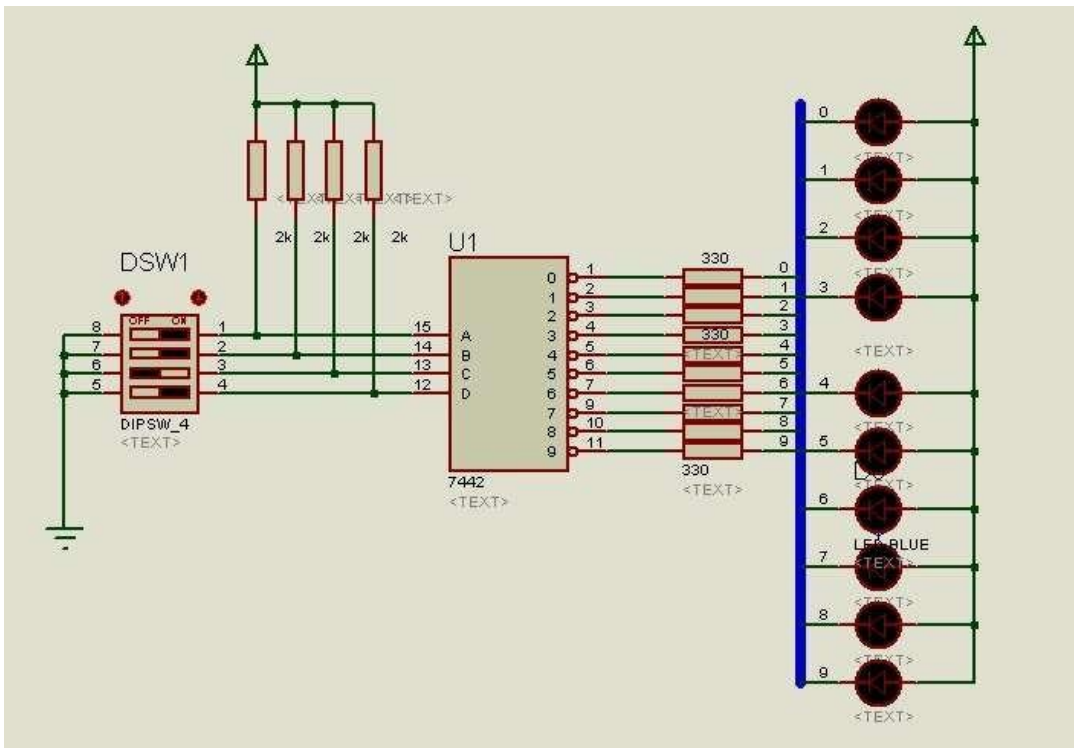
5.2 DECODIFICADOR DECIMAL A BCD

Realice la simulación del circuito decodificador decimal a BCD usando el circuito integrado 74147. Con el Dipswitch active cada una de las líneas de entrada del decodificador 1, 2, 3,.....,9 colocando un 0 y observe el resultado de la decodificación en los Leds de salida $Q=Q_3Q_2Q_1Q_0$. Haga la Tabla de Verdad de esta codificación.



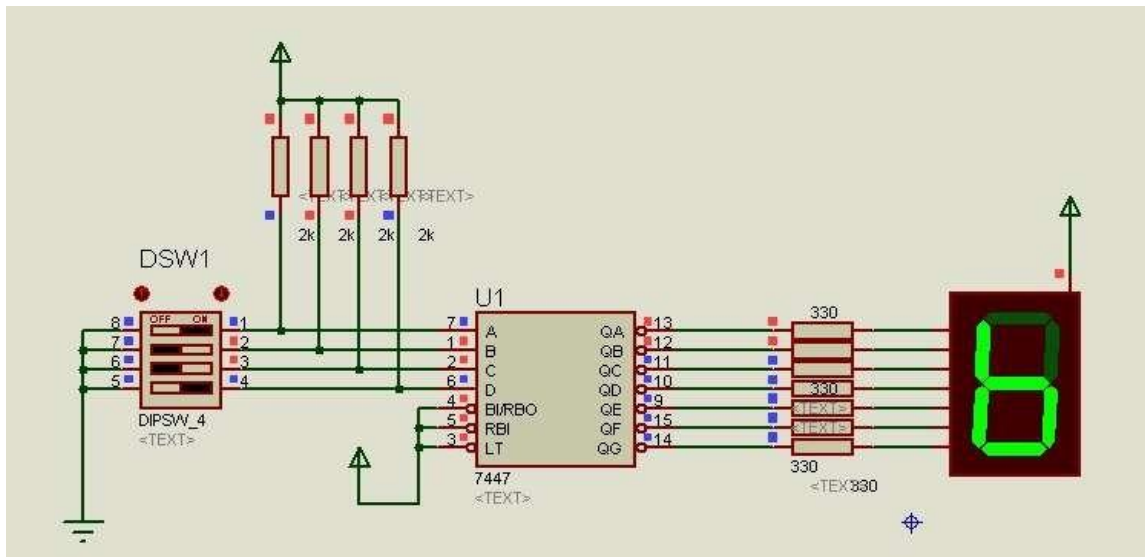
5.3 DECODIFICADOR BCD A DECIMAL

Realice la simulación del circuito decodificador BCD a decimal usando el circuito integrado 7442. Introduzca un dato binario de 4 bits a la entrada del decodificador ABCD, y observe la activación de las líneas de salida 0,1,2,3,.....9. Construya la Tabla de Verdad de este decodificador.



5.4 DECODIFICADOR DE 7 SEGMENTOS

Realice la simulación del circuito mostrado en la figura, es un decodificador BCD a 7 segmentos de ánodo común. Introduzca a la entrada un código binario de 4 bits ABCD y observe el display de 7 segmentos. Explique la operación de este display y cuál sería la diferencia con un display de cátodo común.



CAPÍTULO 6. SIST. COMBINATORIOS - LABORATORIO

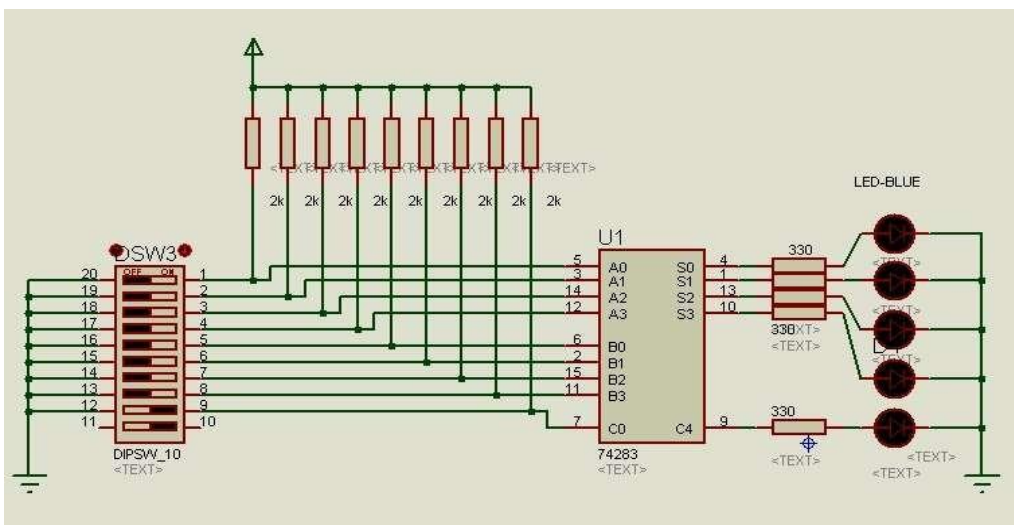
El objetivo de la siguiente práctica es adquirir habilidades para investigar y comprobar el funcionamiento de los circuitos aritméticos como el sumador, los decodificadores decimales, BCD y de 7 segmentos y obtener su operación con la definición de la Tabla de Verdad correspondiente.

6.1 EQUIPO Y MATERIAL NECESARIO

- Fuente de 5V
- Multímetro
- Protoboard
- Circuitos integrados: 74283, 74147, 7442, 7447
- DIPSWICH de 10 posiciones
- 10 LEDs
- Resistencias a 1/4W de: 10x 2KΩ, 10x330Ω
- Un display de 7 segmentos ánodo común
- Conectores

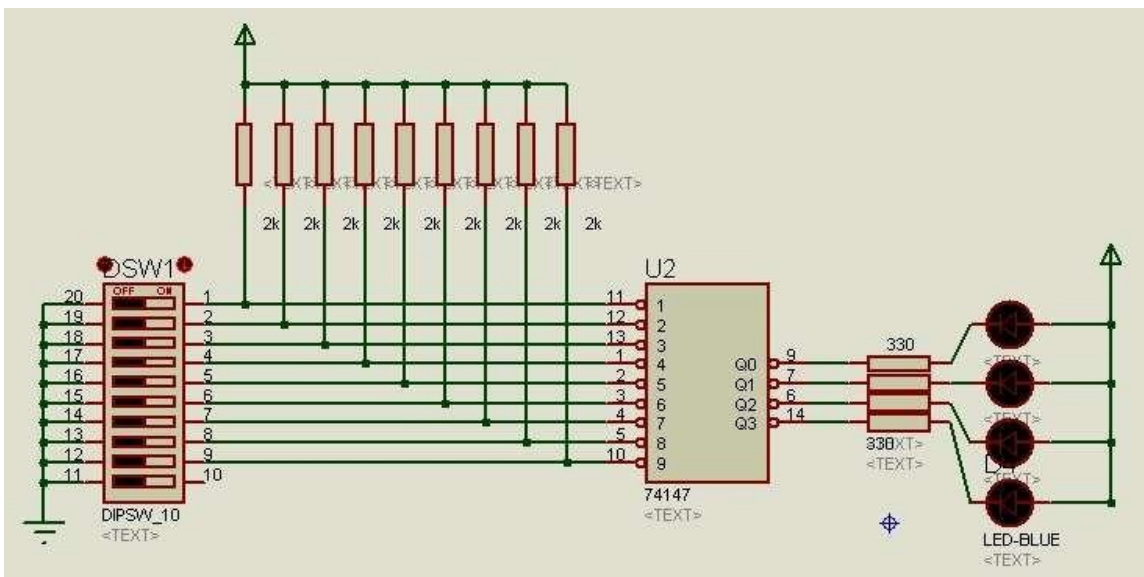
6.2 SUMADOR DE 4 BITS

Realice la implementación del circuito sumador de 4 bits usando el circuito integrado 74283, tal como se indica en la figura. Operando el Dipswitch cargue datos binarios para el sumador $A=A_3A_2A_1A_0$ y para el sumando $B=B_3B_2B_1B_0$ y opere la suma con carry de entrada C_0 . Observe el resultado de la operación en $S=S_3S_2S_1S_0$ y en el carry de salida C_4 . Compruebe estos resultados.



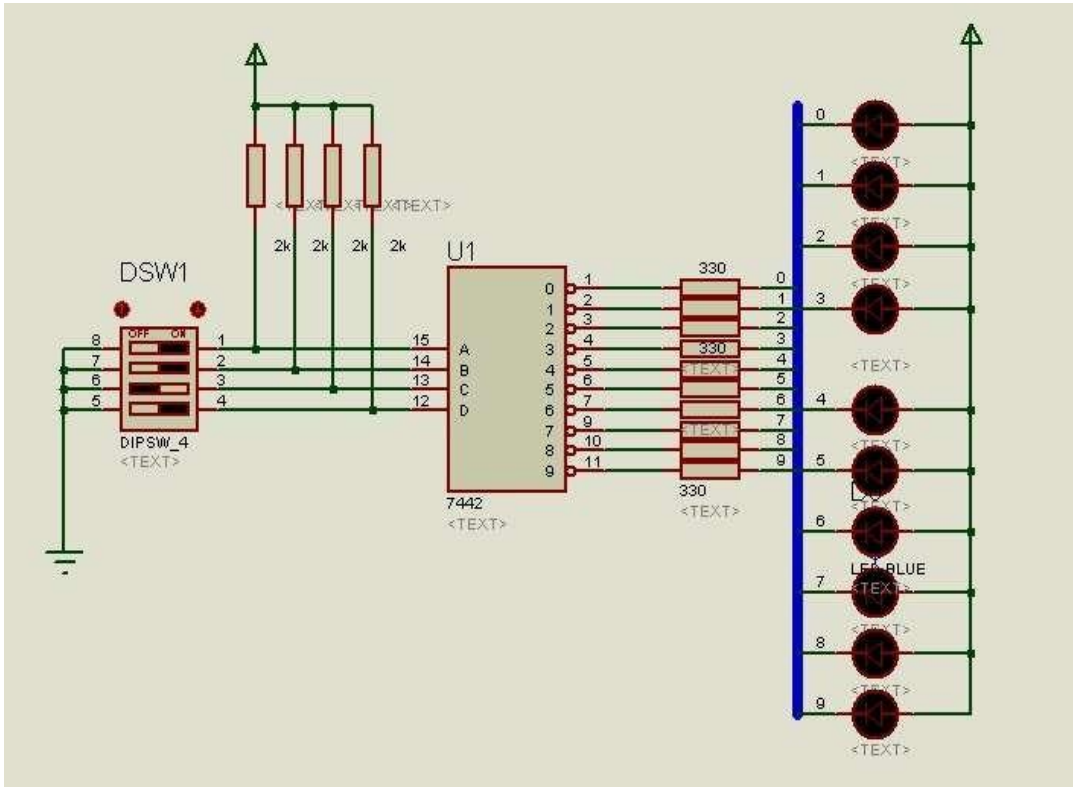
6.3 DECODIFICADOR DECIMAL A BCD

Realice la implementación del circuito decodificador decimal a BCD usando el circuito integrado 74147. Con el Dipswitch active cada una de las líneas de entrada del decodificador 1, 2, 3,.....,9 colocando un 0 y observe el resultado de la decodificación en los Leds de salida Q=Q3Q2Q1Q0. Haga la Tabla de Verdad de esta codificación.



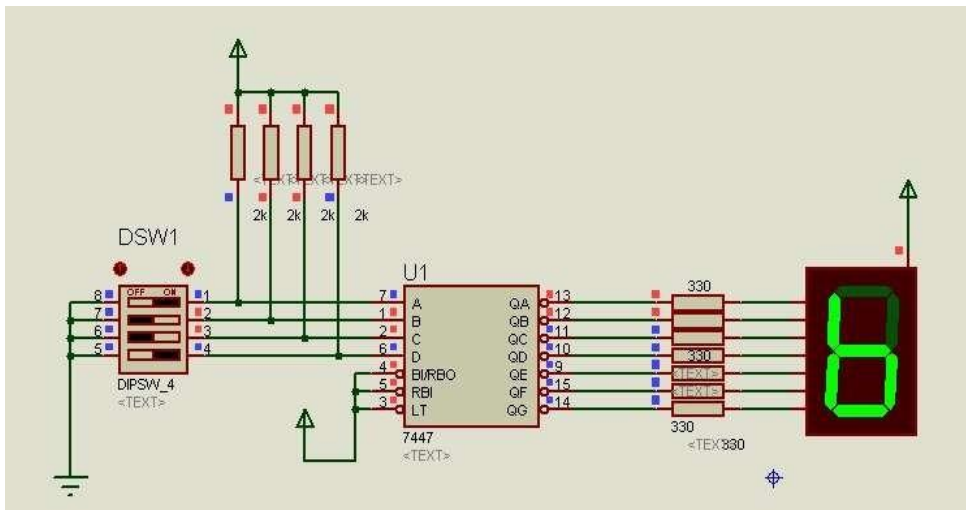
6.4 DECODIFICADOR BCD A DECIMAL

Realice la implementación del circuito decodificador BCD a decimal usando el circuito integrado 7442. Introduzca un dato binario de 4 bits a la entrada del decodificador ABCD, y observe la activación de las líneas de salida 0,1,2,3,.....9. Construya la Tabla de Verdad de este decodificador.

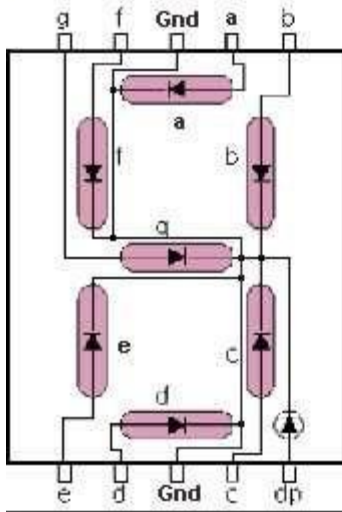


6.5 DECODIFICADOR DE 7 SEGMENTOS

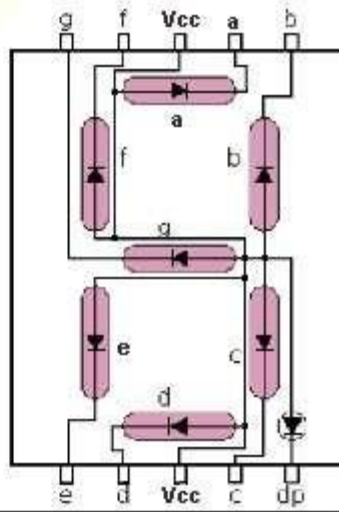
Realice la implementación del circuito mostrado en la figura, es un decodificador BCD a 7 segmentos de ánodo común. Introduzca a la entrada un código binario de 4 bits ABCD y observe el display de 7 segmentos. Explique la operación de este display y cuál sería la diferencia con un display de cátodo común.



Common Cathode



Common Anode



CAPÍTULO 7. SISTEMAS SECUENCIALES - TEORÍA

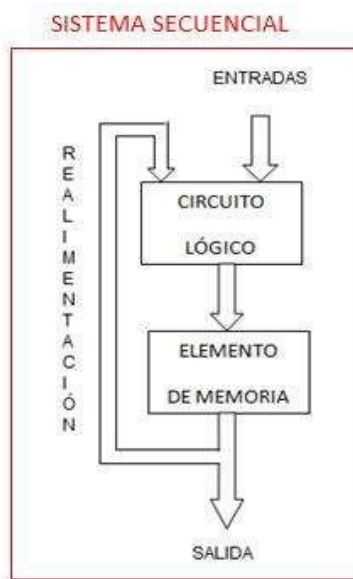
Como vimos en la unidad anterior, un sistema combinatorio se identifica porque la salida del sistema depende estrictamente de las entradas. Un sistema secuencial es un sistema que además de depender de las entradas depende de sus salidas pasadas, esto es, hay un lazo de realimentación. En sistemas secuenciales, el sistema debe tener algún elemento de memoria, debe haber al menos una realimentación del elemento de memoria al sistema de entrada y es de naturaleza cíclica.

En esta unidad se estudiarán los elementos básicos de un circuito secuencial como los flip-flop que son los elementos de memoria, los contadores y los registros digitales. El aprendizaje como en todas las unidades de los cursos se realiza mediante la metodología de estudiar su teoría, simulaciones en laboratorio virtual, prácticas en laboratorio y la correspondiente evaluación.

Un sistema secuencial es un sistema que además de depender de las entradas depende de sus salidas pasadas, esto es, hay un lazo de realimentación. En sistemas secuenciales, el sistema debe tener algún elemento de memoria, debe haber al menos una realimentación del elemento de memoria al sistema de entrada y es de naturaleza cíclica.

En esta unidad se estudiarán los elementos básicos de un circuito secuencial como los flip-flop que son los elementos de memoria, los contadores y los registros digitales.

| |
|--|
| <p>Un sistema combinatorio se identifica por:</p> <ol style="list-style-type: none">1. La salida del sistema debe ser estrictamente una función de las entradas.2. No puede existir lazo de realimentación de la salida a la entrada. <p>Un sistema secuencial se identifica por:</p> <ol style="list-style-type: none">1. El sistema debe tener algún elemento de memoria2. El sistema debe tener al menos una realimentación del elemento de memoria al sistema de entrada3. Tiene una naturaleza cíclica |
|--|



7.1 CELDA DE MEMORIA BINARIA

El elemento básico de una memoria es la celda que almacena un bit. La tabla funcional describe el circuito lógico de la celda, mediante la cual se plantea su ecuación lógica que es igual a $Q = S + R'Q$. La nueva salida (estado posterior) depende de las entradas y de la salida pasada (estado anterior).

La celda binaria es el circuito lógico que almacena un bit.

ECUACIÓN LÓGICA
 $Q = S + R'Q$, $Q' = R'Q$

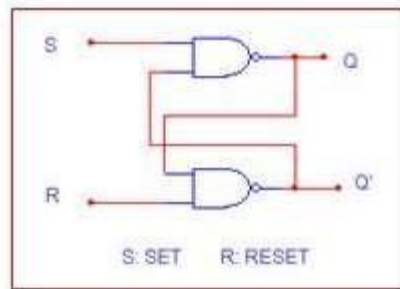


TABLA FUNCIONAL

| S | R | Q | Q' | Q |
|---|---|---|----|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

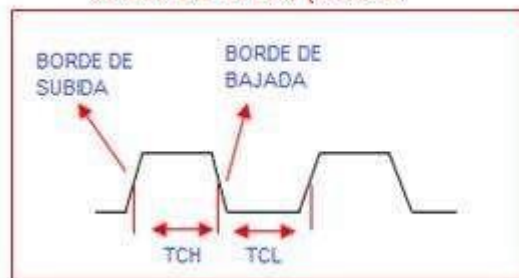
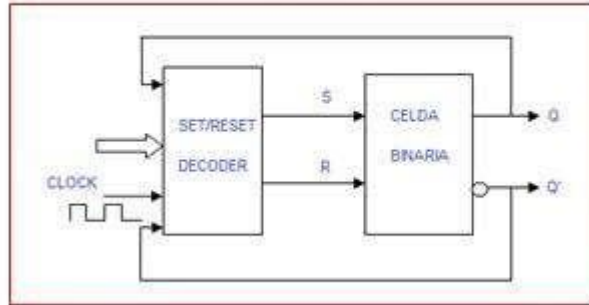
ESTADO ANTERIOR   ESTADO POSTERIOR

7.2 EL FLIP FLOP O BÁSCULA

Está compuesto por una celda binaria de almacenamiento de un bit que tiene dos entradas S (set) y R (reset), el decodificador y la señal de reloj (clock).

Un Flip Flop es un circuito lógico que tiene dos estados estables que se utiliza para almacenar información, es un multivibrador biestable. Se requiere de un reloj de sincronización (CK), en el cual es importante el tiempo de subida y el tiempo de bajada. Hay flip flop que almacenan la entrada con el borde o flanco de subida y otros que la almacenan con el flanco de bajada.

BLOQUE FUNCIONAL



1. FLIP FLOP: SET- RESET

Tiene dos entradas S y R, la señal de reloj para sincronizar el almacenamiento y dos salidas Q y Q'. La tabla funcional nos presenta el estado posterior Q_{n+1} de la salida en función de las entradas S, R y del estado anterior de esta salida Q_n . Lo importante en este flip flop es que $Q_{n+1}=1$ si $S=1$ y $R=0$, $Q_{n+1}=0$ si $S=0$ y $R=1$.

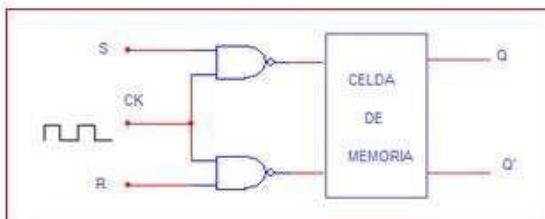


TABLA FUNCIONAL

| S | R | Q_n | Q_{n+1} | |
|---|---|-------|-----------|------------|
| 0 | 0 | 0 | 0 | NO OPERA |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | OPERA |
| 0 | 1 | 1 | 0 | RESET |
| 1 | 0 | 0 | 1 | OPERA |
| 1 | 0 | 1 | 1 | SET |
| 1 | 1 | 0 | X | DON'T CARE |
| 1 | 1 | 1 | X | DON'T CARE |

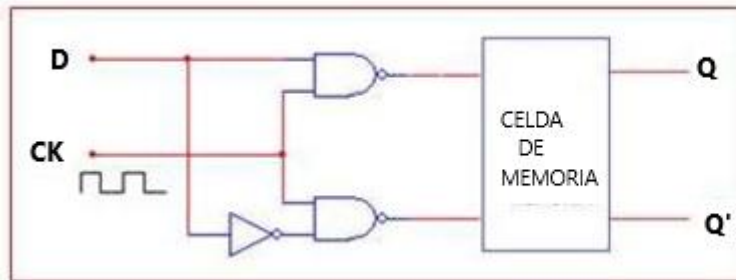
TABLA DE EXCITACIÓN

| Q_n | Q_{n+1} | S | R |
|-------|-----------|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

Q_n : Antes del borde de subida
 Q_{n+1} : Después del borde de subida

2. FLIP FLOP: TIPO D (LATCH)

Es un flip flop que almacena el bit de entrada con el flanco de subida del reloj. Lo que hay en D aparece en Q.



Generalmente se pasa la información de la entrada D a Q con el borde de subida del reloj.

TABLA FUNCIONAL

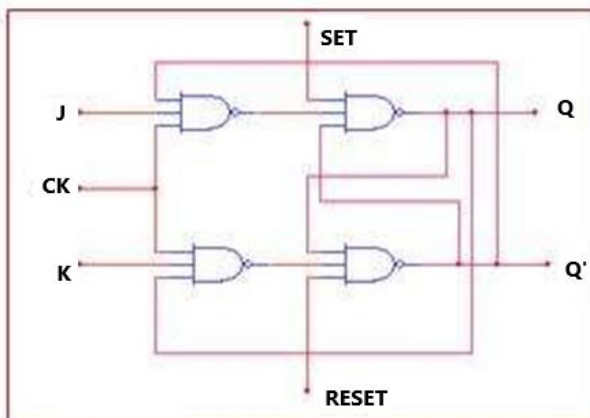
| D | Q_n | Q_{n+1} |
|---|-------|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

TABLA DE EXCITACIÓN

| Q_n | Q_{n+1} | D |
|-------|-----------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

3. FLIP FLOP JK

Es un mejoramiento al flip flop SR. Cuando $J=1, K=0, Q_{n+1}=1$, $J=0, K=1, Q_{n+1}=0$, $J=0, K=0, Q_{n+1}=Q_n$, $J=1, K=1, Q_{n+1}=Q_n'$



En el flip flop RS cuando las entradas estaban ambas en nivel alto, su salida era indeterminada (don't care), el JK evita esta indeterminación.

TABLA FUNCIONAL

| J | K | Q_n | Q_{n+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 |

TABLA DE EXCITACIÓN

| Q_n | $\rightarrow Q_{n+1}$ | J | K |
|-------|-----------------------|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

4. FLIP FLOP JK MAESTRO- ESCLAVO

En la figura tenemos su diagrama lógico y su tabla de excitación. Con el flanco positivo del reloj se almacena el bit en el maestro y con el flanco de bajada se almacena en el esclavo.

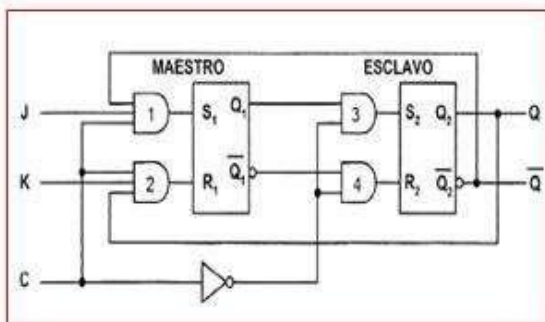


TABLA DE EXCITACIÓN

| Q_n | $\rightarrow Q_{n+1}$ | J | K |
|-------|-----------------------|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

Un flip-flop maestro-esclavo se realiza a partir de dos biestables SR en cascada, de forma que la señal de reloj entra al biestable maestro y la señal de reloj negada entra al esclavo. Sólo el biestable maestro está habilitado cuando el reloj es 1. Durante todo ese intervalo de tiempo, sus salidas irán acorde con sus entradas. Cuando llega el flanco negativo de reloj, se habilita el biestable esclavo (y se deshabilita el maestro), que toma la salida del maestro. Por tanto, justamente después del flanco negativo de reloj, la salida del biestable esclavo será la equivalente a la salida almacenada en el biestable maestro..

5. CIRCUITO INTEGRADO 7476

Este circuito tiene en su interior dos flip flop JK maestro esclavo, tal como se presenta en el diagrama de conexión. El pin PRE colocado en nivel bajo pone la salida en nivel alto y el pin CLR en nivel bajo pone la salida en nivel bajo.

DIAGRAMA DE CONEXIÓN

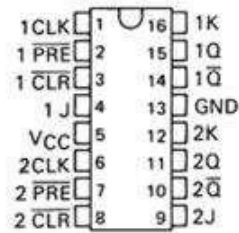
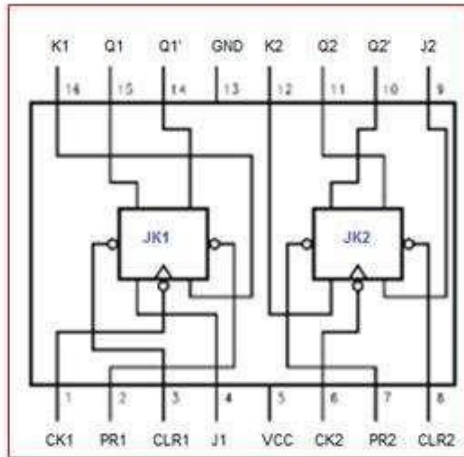


TABLA FUNCIONAL

| INPUTS | | | | | OUTPUTS | |
|--------|-----|-----|---|---|----------------|----------------|
| PRE | CLR | CLK | J | K | Q | \bar{Q} |
| L | H | X | X | X | H | L |
| H | L | X | X | X | L | H |
| L | L | X | X | X | H [†] | H [†] |
| H | H | | L | L | Q ₀ | \bar{Q}_0 |
| H | H | | H | L | H | L |
| H | H | | L | H | L | H |
| H | H | | H | H | TOGGLE | |

7.3 CONTADORES

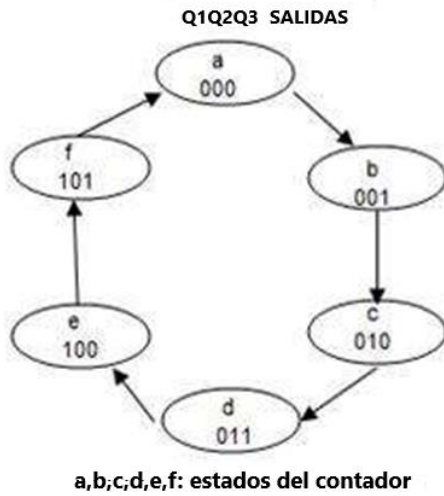
Son de los circuitos secuenciales más importantes en la electrónica digital. Su función es contar un número de pulsos que le llegan por el reloj y también son usados como temporizadores. Pueden ser síncronos o asíncronos, ascendentes o descendentes y binarios o decimales. Están compuestos por flip flops y un decodificador a la entrada. Cada vez que llega un pulso de reloj el contador cambia su estado tal como se aprecia en el diagrama de estados de la figura.

Son usados ampliamente para realizar una de las siguientes funciones:

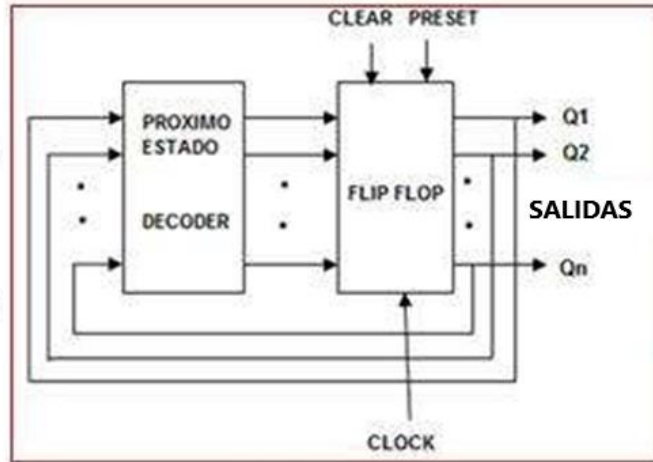
1. Convertir un número de pulsos de entrada en un determinado código y almacenarlo.
2. Como elementos temporizadores

Los contadores pueden ser síncronos o asíncronos, ascendentes o descendentes, binarios o decimales.

DIAGRAMA DE ESTADOS

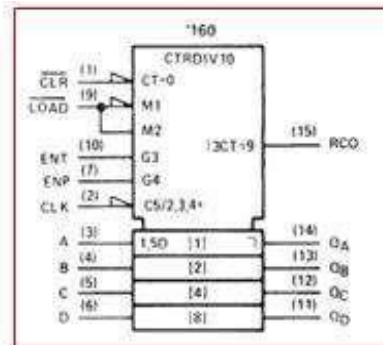
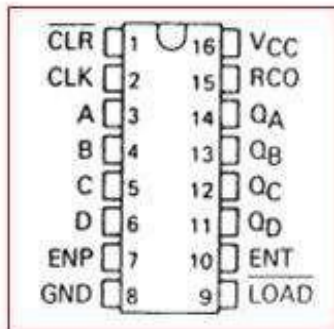


BLOQUES FUNCIONALES



74160/162: CONTADOR DECIMAL SÍNCRONO

74161/163: CONTADOR BINARIO SÍNCRONO

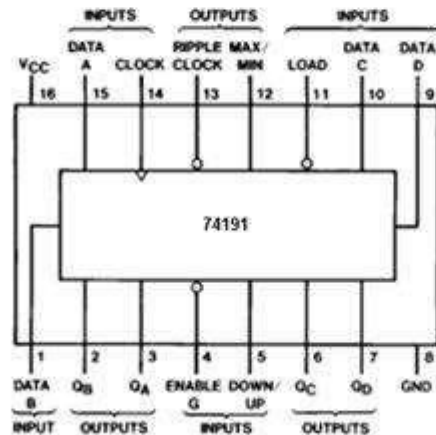


El contador 74160/162 son contadores decimales, cuentan de 000 a 1001 y vuelve a 0. El 74161/163 son contadores binarios, o sea, cuentan de 0000 a 1111 y vuelve a 0000.

Las entradas A, B, C, D se utilizan para cargar un dato desde donde debe comenzar la cuenta al habilitar la línea LOAD=0. Con CLEAR se borra la salida del contador. Para que funcione el contador ENT y ENP deben estar en nivel H. CLK es el reloj mediante el cual a su velocidad funciona el contador. RCO es una línea de salida que indica salida de carry.

74191: CONTADOR BINARIO DE 4 BITS UP/DOWN

Es un contador binario de 4 bits ascendente/descendente, que se puede programar en las entradas A, B, C, D. La línea DOWN/UP permite la programación ascendente o descendente.



7.4 REGISTROS DE CORRIMIENTO

Son circuitos síncronos que se utilizan para almacenamiento temporal de datos. Un registro de corrimiento básico es un conjunto de *flip-flops* conectados de tal forma que los números binarios almacenados en él son desplazados de un *flip-flop* al siguiente con cada pulso de reloj aplicado.

Son especificador por:

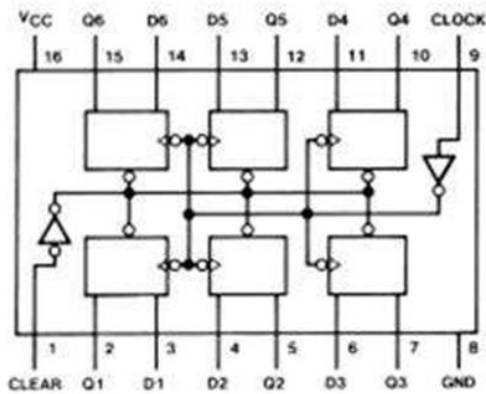
1. Número de bits
2. Modo de operación: PIPO: Entrada paralela – salida paralela, PISO: Entrada paralela – salida serial
SIPO: Entrada serial – salida paralela, SISO: Entrada serial – salida serial, UNIVERSAL: Cualquier configuración

Generalmente son disparados por el flanco positivo del reloj.

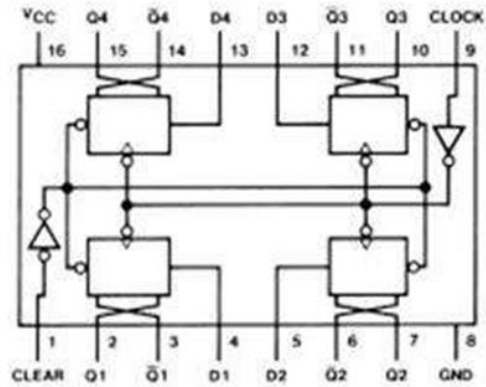
El 74174 es un registro conformado por seis FF tipo D con entradas D1..D6 y seis salidas Q1..Q6 con CLEAR.

El 74175 es un registro conformado por cuatro FF tipo D con entradas D1..D4 y salidas Q1..Q4 y Q1'...Q4' (salidas negadas) con CLEAR.

74174: SEIS FF TIPO D



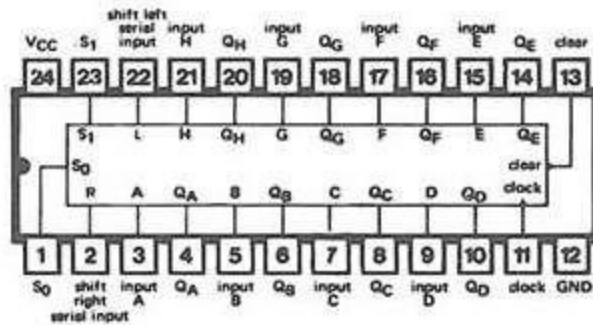
74175: CUATRO FF TIPO D



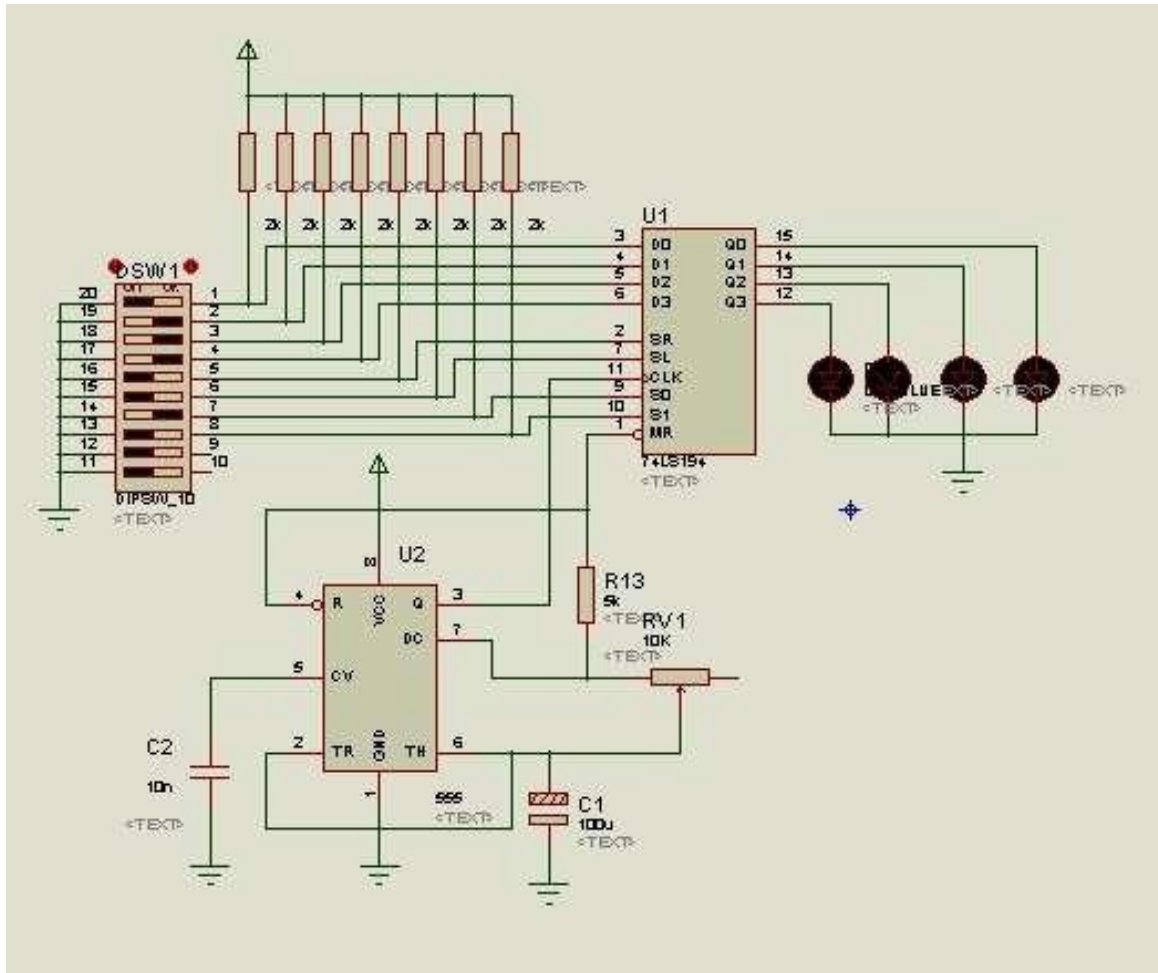
74198: SHIFT REGISTER DE 8 BITS R/L

Es un registro de corrimiento síncrono de carga paralela, con modos de corrimiento a la derecha o a la izquierda determinado por la selección S0 y S1. Este circuito es útil para la transferencia de datos serial-serial, serial-paralela, paralela-serial y paralela-paralela. La línea MR es el clear.

| S1 | S0 | MODO OPERACIÓN |
|----|----|----------------------|
| 0 | 0 | Reloj inhibido |
| 0 | 1 | Corre a la derecha |
| 1 | 0 | Corre a la izquierda |
| 1 | 1 | Carga paralela |



Simule el circuito de la figura y hágalo funcionar de acuerdo a la Tabla funcional dada. Saque sus propias conclusiones.



CAPÍTULO 9. SIST. SECUENCIALES - LABORATORIO

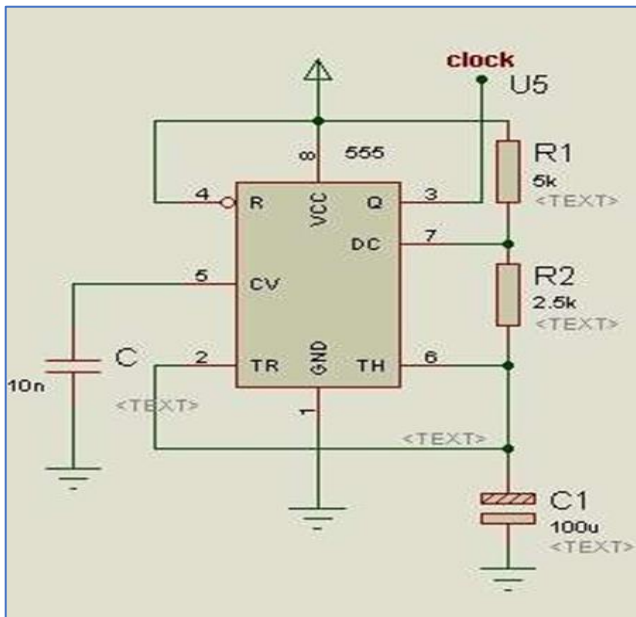
En esta unidad vamos a hacer dos implementaciones en Protoboard para consolidar los conceptos teóricos sobre circuitos secuenciales.

9.1 EQUIPO Y MATERIAL NECESARIO

- Fuente de 5V
- Protoboard
- Circuitos integrados: 555, 2x74LS90, 2x74LS47, 74LS194
- DIPSWICH de 10 posiciones
- 4 LEDs
- Resistencias a 1/4W de: 10x 2K Ω , 10x330 Ω , 5K, 2.5K
- Capacitores de 10 nF y de 100 uF
- POT de 10K
- 2 display de 7 segmentos ánodo común
- Conectores

9.2 CONTADOR DECIMAL DE 0 A 99

Implemente el siguiente circuito y póngalo a funcionar con el reloj realizado con el 555 que se presenta en la figura. Compruebe la frecuencia de oscilación del reloj mediante las fórmulas obtenidas en unidades anteriores. Primero implemente el circuito de las unidades y luego el circuito completo con las decenas. Varíe la frecuencia del reloj. Implemente el siguiente reloj:



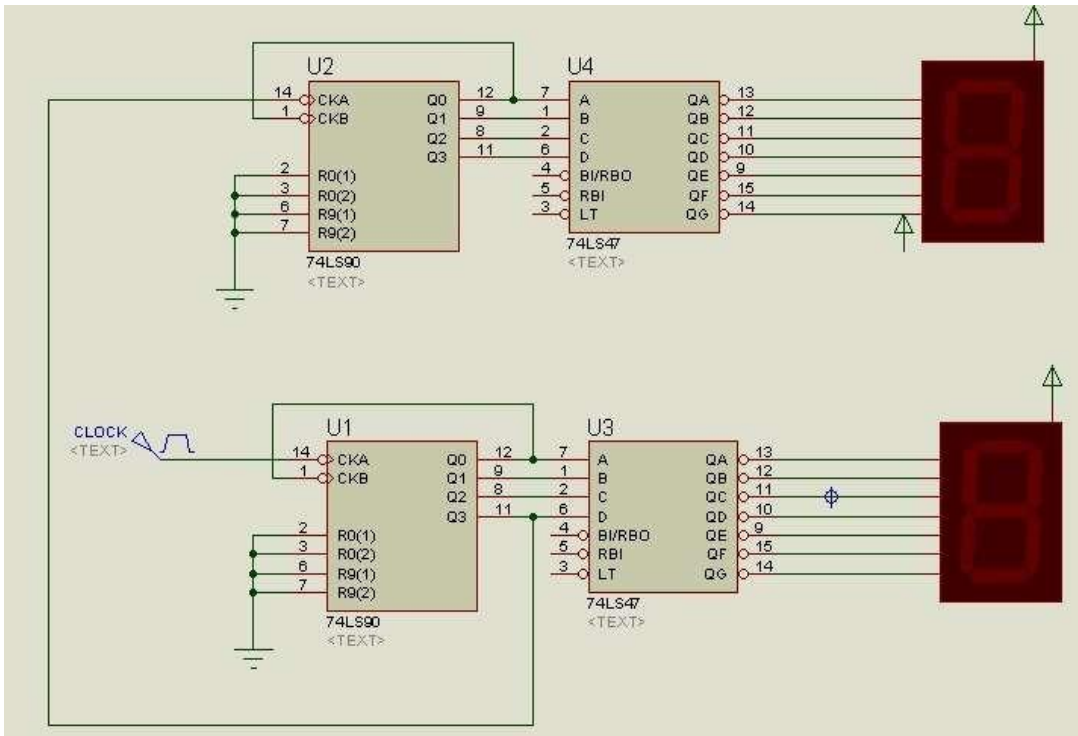
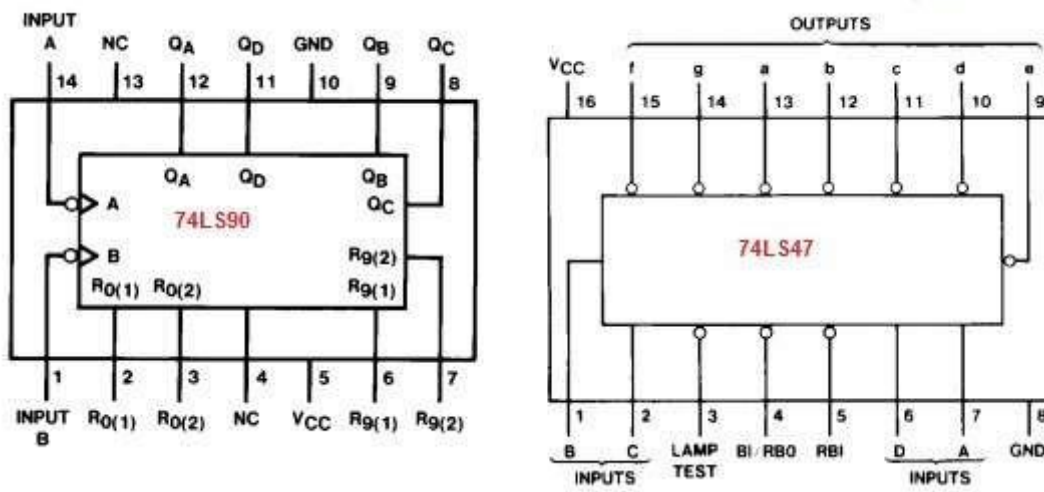


DIAGRAMA DE LOS CIRCUITOS INTEGRADOS



9.3 REGISTRO DE DESPLAZAMIENTO L/R

Implemente el circuito de la figura y hágalo funcionar de acuerdo con la Tabla funcional dada. saque sus propias conclusiones.

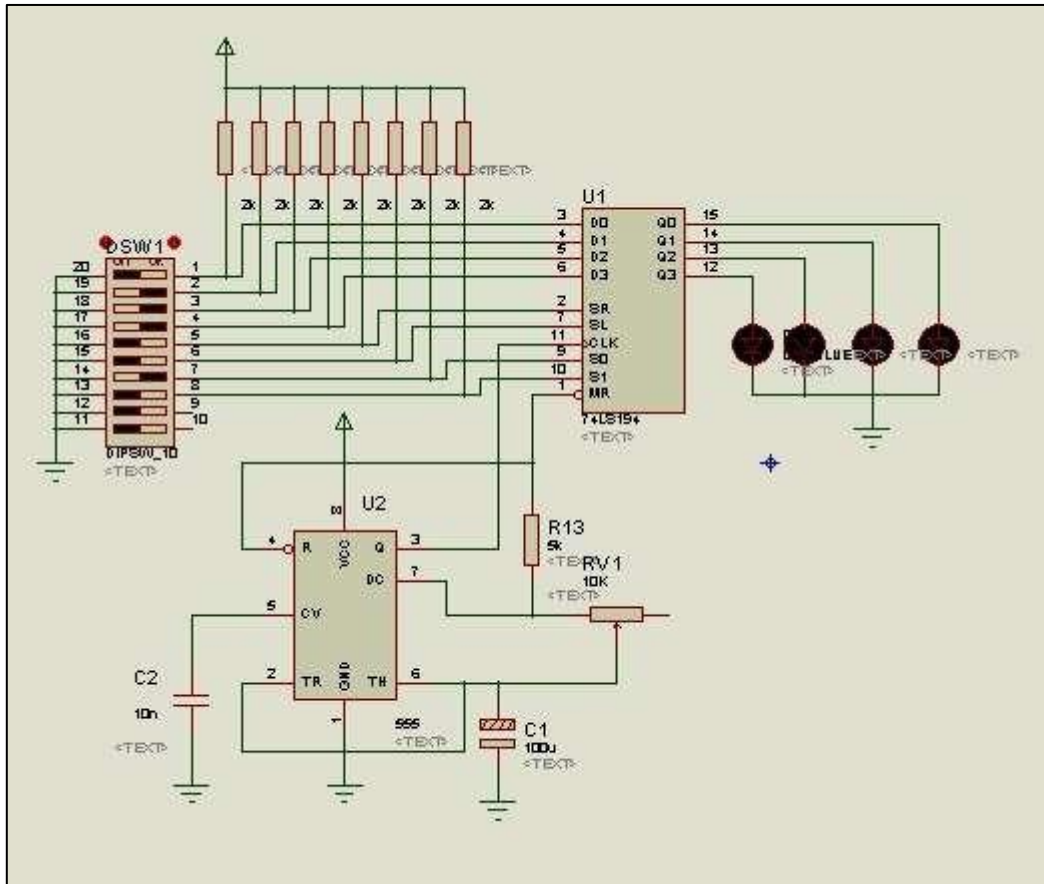
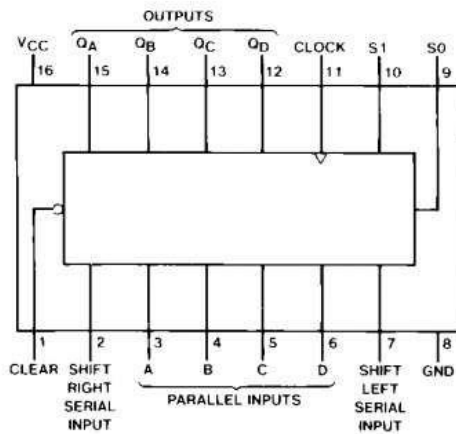


DIAGRAMA DE PINES Y TABLA FUNCIONAL



| Clear | Mode | | Clock | Inputs | | | | Outputs | | | | |
|-------|------|----|-------|--------|-------|----------|---|----------------|-----------------|-----------------|-----------------|-----------------|
| | S1 | S0 | | Serial | | Parallel | | Q _A | Q _B | Q _C | Q _D | |
| | | | | Left | Right | A | B | | | | | C |
| L | X | X | X | X | X | X | X | X | L | L | L | L |
| H | X | X | L | X | X | X | X | X | Q _{A0} | Q _{B0} | Q _{C0} | Q _{D0} |
| H | H | H | ↑ | X | X | a | b | c | a | b | c | d |
| H | L | H | ↑ | X | H | X | X | X | H | Q _{An} | Q _{Bn} | Q _{Cn} |
| H | L | H | ↑ | X | L | X | X | X | L | Q _{An} | Q _{Bn} | Q _{Cn} |
| H | H | L | ↑ | H | X | X | X | X | Q _{Bn} | Q _{Cn} | Q _{Dn} | H |
| H | H | L | ↑ | L | X | X | X | X | Q _{Rn} | Q _{Cn} | Q _{Dn} | L |
| H | L | L | X | X | X | X | X | X | Q _{A0} | Q _{B0} | Q _{C0} | Q _{D0} |

CAPÍTULO 10. CONTROLADORES DIGITALES - TEORÍA

En esta unidad se aprenderá a diseñar controladores digitales con base en la lógica combinatoria y secuencial estudiada en las unidades anteriores. Los diseños se harán con base en flip flop tipo D, registros de corrimiento y contadores. Este tipo de controladores nos permitirá realizar circuito de control para la automatización de procesos y se expone su aplicación en una máquina de servicios de líquidos.

La metodología como en todas las unidades de los cursos se realiza mediante explicaciones teóricas, simulaciones en laboratorio virtual, prácticas en laboratorio y la correspondiente evaluación

Los controladores digitales son circuitos que utilizando flip flops, registros o contadores pueden realizar un control a una máquina con base en la lógica combinatoria o secuencial vista en las anteriores unidades. Se aplicará esta metodología para resolver el problema de automatización de una máquina de servicios de líquido. Si se quiere profundizar en este tema sugiero el libro de Fletcher William "An engineering approach to digital design" editado por Prentice Hall.

10.1 MÓDULOS DEL CONTROLADOR

Para su estudio se ha considerado una **máquina de servicios de líquidos** que tendrá los siguientes módulos:

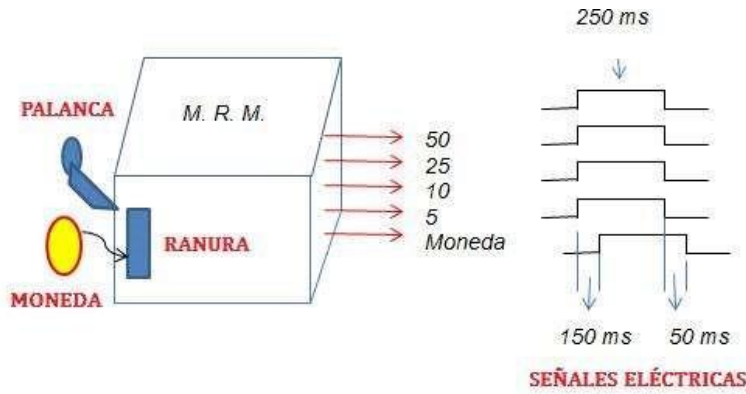
1. Módulo de recepción de monedas
2. Módulo de cambio y
3. Módulo de servicio

Los módulos tienen las siguientes características:

1. MÓDULO RECEPCIÓN DE MONEDAS

Tiene una sola ranura para la inserción de monedas

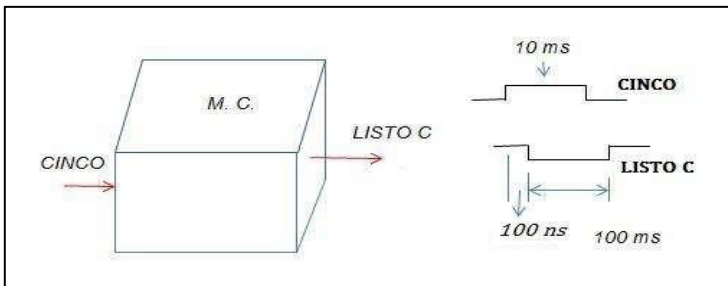
- a) Hace detección electrónica de monedas
- b) Reconoce monedas de 50¢, 25¢, 10¢ y 5¢ centavos de dólar
- c) Tiene un sistema para prevenir el sobreflujo de monedas
- d) tiene una palanca para devolución de monedas
- e) Existen señales eléctricas asociadas a las monedas reconocidas



1. MÓDULO DE CAMBIO

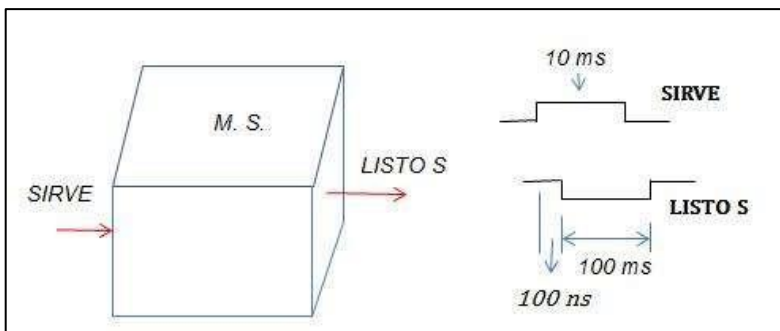
Acepta una señal tipo TTL (CINCO) para entregar una moneda de 5¢ como cambio.

- Tiene una señal eléctrica de salida que indica cuándo el módulo de cambio está listo para iniciar un nuevo proceso de cambio (LISTO C)



2. MÓDULO DE SERVICIO

- Acepta una señal (SIRVE) y
- Entrega una señal de listo servicio (LISTO S)

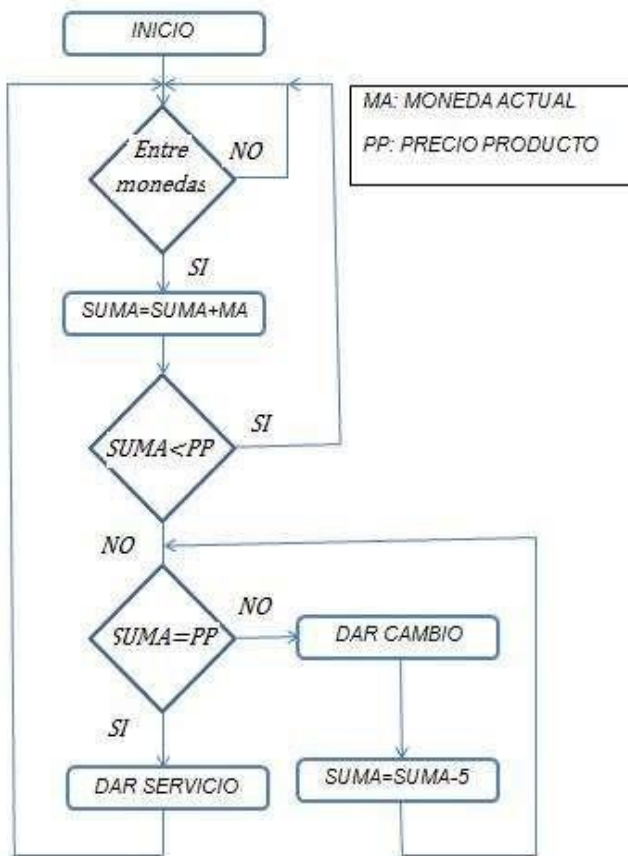


10.2 OBJETIVO DEL CONTROLADOR

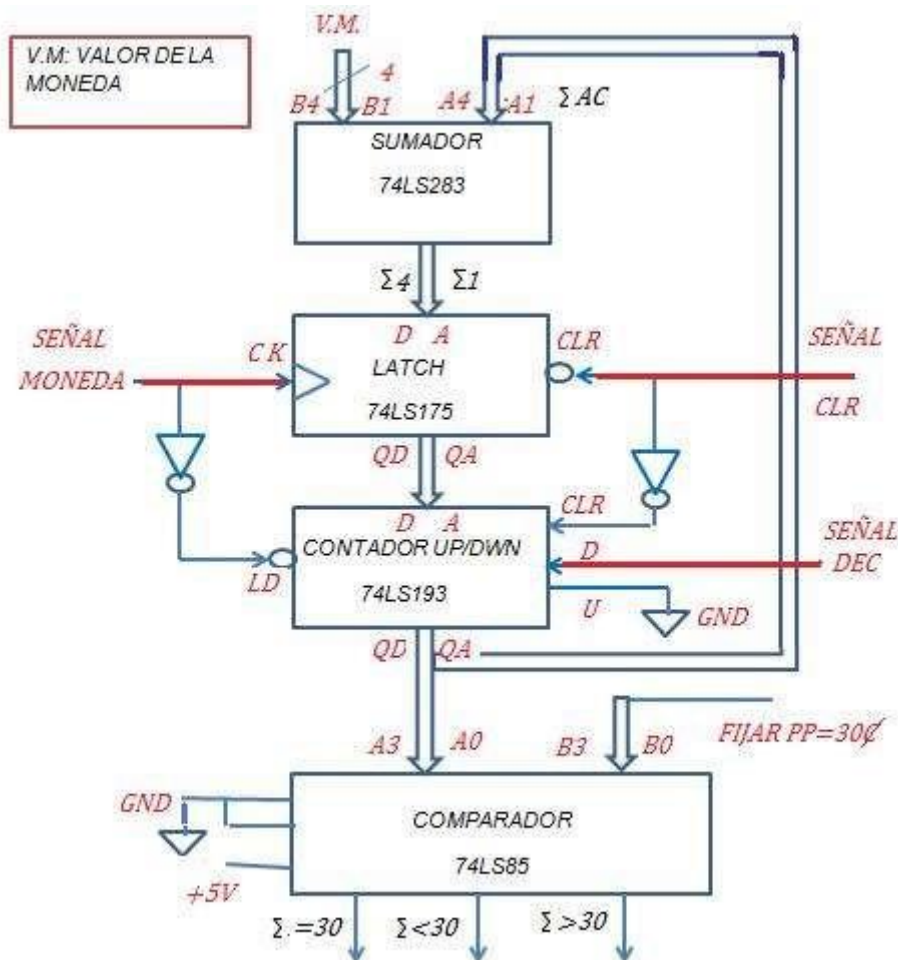
Diseñar un controlador digital que interactúe con los módulos tal como se indica en la figura.



10.3 DIAGRAMA DE FLUJO



10.4 CIRCUITO QUE EVALÚA LA SUMA ACUMULADA

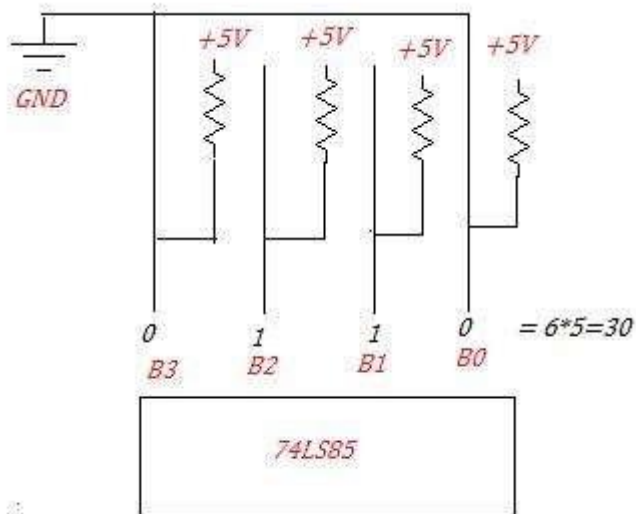


HARDWARD ADICIONAL

Este es el hardware que se debe adicionar al circuito anterior.

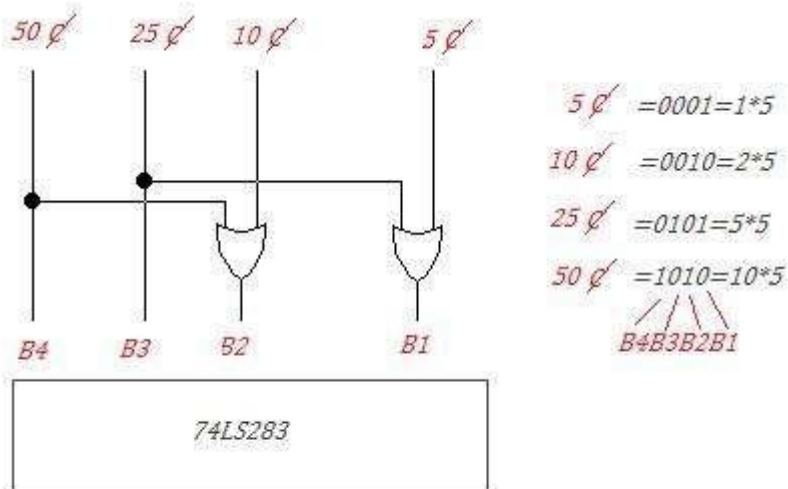
FIJAR PP = 30¢

Dato binario B3B2B1B0 de entrada al comparador 74LS85 que es el precio del producto.

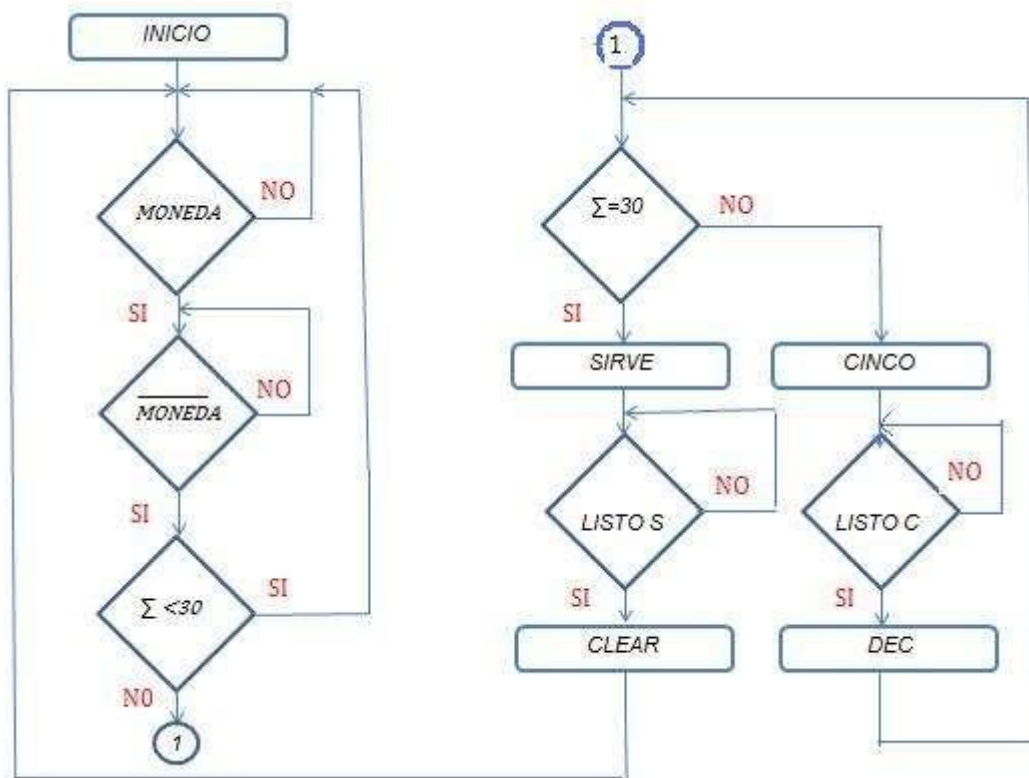


V.M. VALOR DE LA MONEDA

Dato binario B4B3B2B1 de entrada al sumador de 4 bits 74LS283 que da un código según el valor de la moneda.



10.5 DIAGRAMA DE FLUJO DEL CONTROLADOR



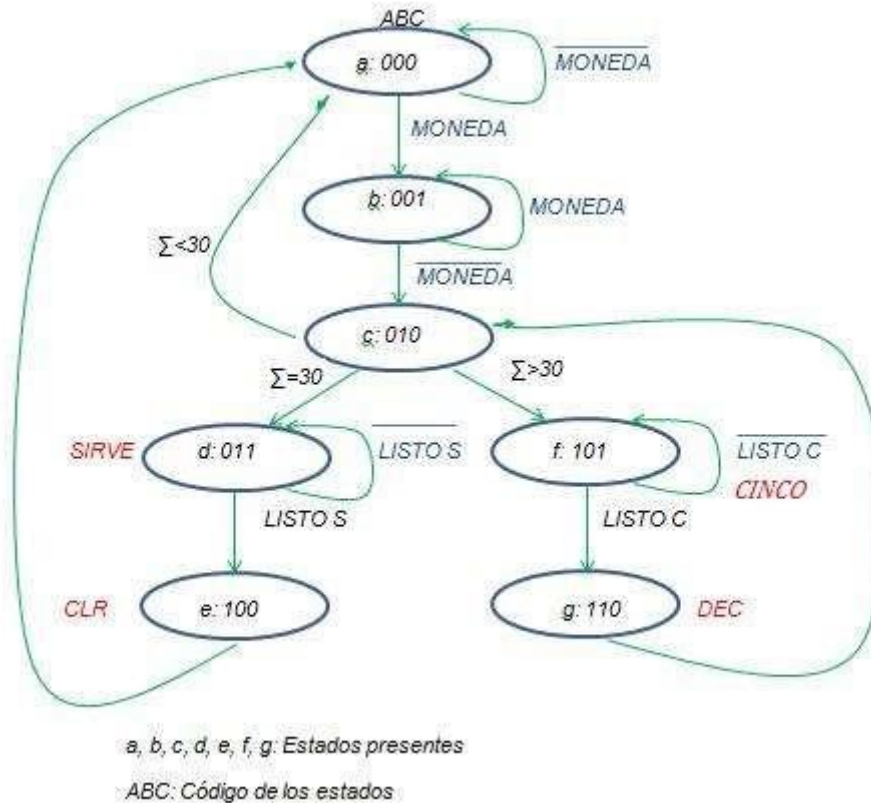
10.6 DISEÑO DEL CONTROLADOR CON LATCH

PASOS DE DISEÑO

1. Asignar códigos a los estados del autómata respetando la regla de distancia mínima utilizada en los Mapas de Karnaugh
2. Hacer el mapa del estado presente (E.P.)
3. Hacer el mapa del estado siguiente (E.S.) indicando los estados siguientes así como las variables que indican el cambio. Indicar el código del estado siguiente.
4. Obtener los mapas del estado siguiente para cada una de las variables de estado del problema en cuestión
5. Hacer mapas de salidas
6. Implementar la solución más adecuada con:
 - a) Lógica combinatoria para codificadores de entrada y salida
 - b) Multiplexores como decodificadores de entrada y decodificadores para las salidas
 - c) Combinaciones de a) y b)

10.7 SOLUCIÓN AL PROBLEMA - AUTÓMATA

1. Realizar el autómata donde se presenten sus estados. Se recomienda asignar códigos con distancia mínima, esto es, 000, 001, 011, etc. De esta forma se simplifican las funciones en los mapas.



2. Mapa del estado presente (M.E.P.)

| | | | | | |
|----------|--|------------|----|----|----|
| | | A B | | | |
| C | | 00 | 01 | 11 | 10 |
| 0 | | a | c | g | e |
| 1 | | b | d | * | f |

M.E.P.

3. Mapa del estado siguiente (M.E.S.)

| | | | | | |
|----------|---|--------------------------------|--|-------|------------------|
| | | <i>AB</i> | | | |
| | | <i>C</i> | 00 | 01 | 11 |
| <i>C</i> | 0 | MON' -> a:000 MON -> b:001 | $\Sigma < \rightarrow$ a:000 $\Sigma = \rightarrow$ d:011 $\Sigma > \rightarrow$ f:101 | c:010 | a:000 |
| | 1 | MON' -> b:001 MON' -> c:010 | LISTO S' -> d:011 LISTO S -> e:100 | * | LISTO C -> g:110 |

M.E.S.

4. Mapa del estado siguiente de las variables.

Las variables de los estados con don't care (*) se dejan = 0.

| | | | | | |
|----------|---|-----------|------------|----|----|
| | | <i>AB</i> | | | |
| | | <i>C</i> | 00 | 01 | 11 |
| <i>C</i> | 0 | 0 | $\Sigma >$ | 0 | 0 |
| | 1 | 0 | LISTO S | 0 | 1 |

$$D_A = A'BC'(\Sigma >) + A'BC(\text{LISTO S}) + AB'C$$

| | | | | | |
|----------|---|-----------|------------|----|---------|
| | | <i>AB</i> | | | |
| | | <i>C</i> | 00 | 01 | 11 |
| <i>C</i> | 0 | 0 | $\Sigma =$ | 1 | 0 |
| | 1 | MON' | LISTO S' | 0 | LISTO C |

$$D_B = A'BC'(\Sigma =) + ABC' + A'B'C(\text{MON}') + A'BC(\text{LISTO S}') + AB'C(\text{LISTO C})$$

| | | | | | |
|----------|---|-----------|-------------|----|----------|
| | | <i>AB</i> | | | |
| | | <i>C</i> | 00 | 01 | 11 |
| <i>C</i> | 0 | MON | $\Sigma <'$ | 0 | 0 |
| | 1 | MON | LISTO S' | 0 | LISTO C' |

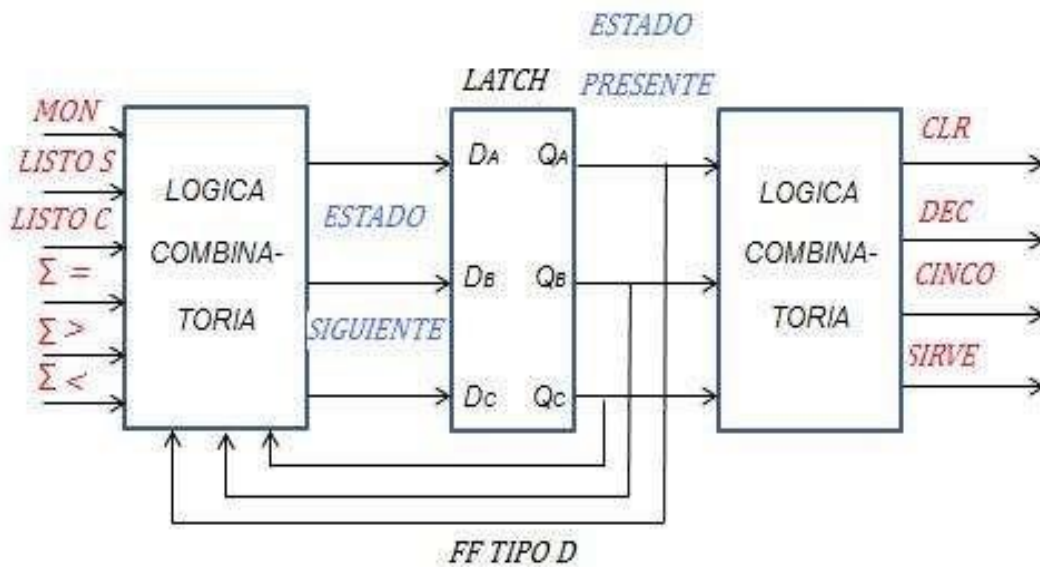
$$D_C = A'B'(MON) + A'BC'(\Sigma <') + A'BC(\text{LISTO S}') + AB'C(\text{LISTO C}')$$

5. Mapa de salidas

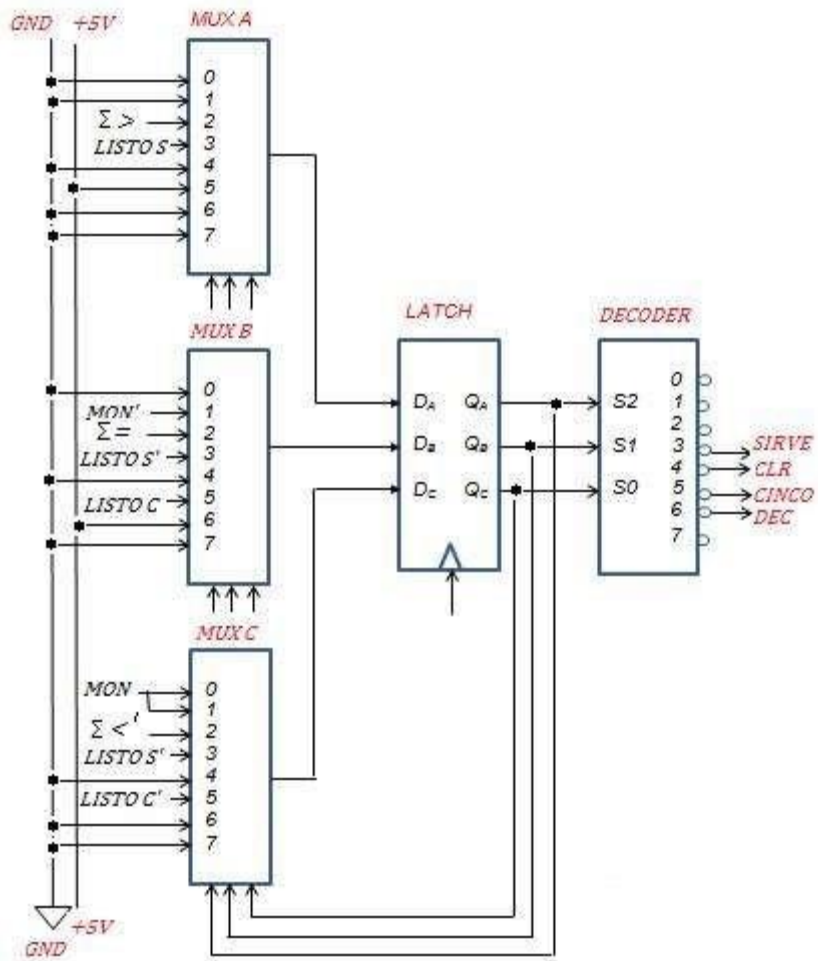
| | | | | |
|----------|-----------|-------|-----|-------|
| | <i>AB</i> | | | |
| <i>C</i> | 00 | 01 | 11 | 10 |
| 0 | * | * | DEC | CLR |
| 1 | * | SIRVE | * | CINCO |

$$DEC = ABC' \quad SIRVE = A'B'C \quad CLR = AB'C' \quad CINCO = AB'C$$

10.8 IMPLEMENTACIÓN CON LÓGICA COMBINATORIA

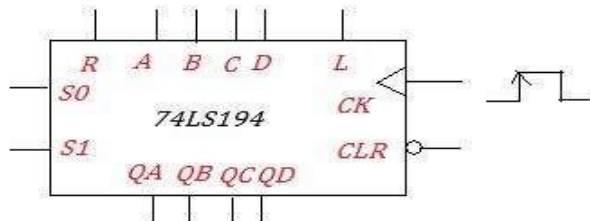


10.9 CON MULTIPLEXORES Y DECODIFICADOR A LA SALIDA.



CAPÍTULO 11. CONTROLADORES CON SHIFT REGISTER

Se utiliza el registro de corrimiento 74LS194.



Tiene las siguientes características:

- Síncrono y actúa con el flanco de subida
- Permite cargar valores (LOAD) de bits por las entradas A, B, C, D
- Permite corrimientos a la izquierda (SL) y a la derecha (SR)
- Define un estado HOLD por tiempo indefinido
- Tiene 4 modos de control fijados por S0 y S1

| S0 | S1 | ACCIÓN |
|----|----|--------|
| 0 | 0 | HOLD |
| 0 | 1 | SL |
| 1 | 0 | SR |
| 1 | 1 | LOAD |

Posibles instrucciones de control

- SL0 (L=0), SL1 (L=1), SL0C, SL1C (Condicionados)
- SR0, SR1 (Incondicionados), SR0C (R=0), SR1C (R=1)
- BI (Brinco incondicional)
- BC (Brinco condicionado)

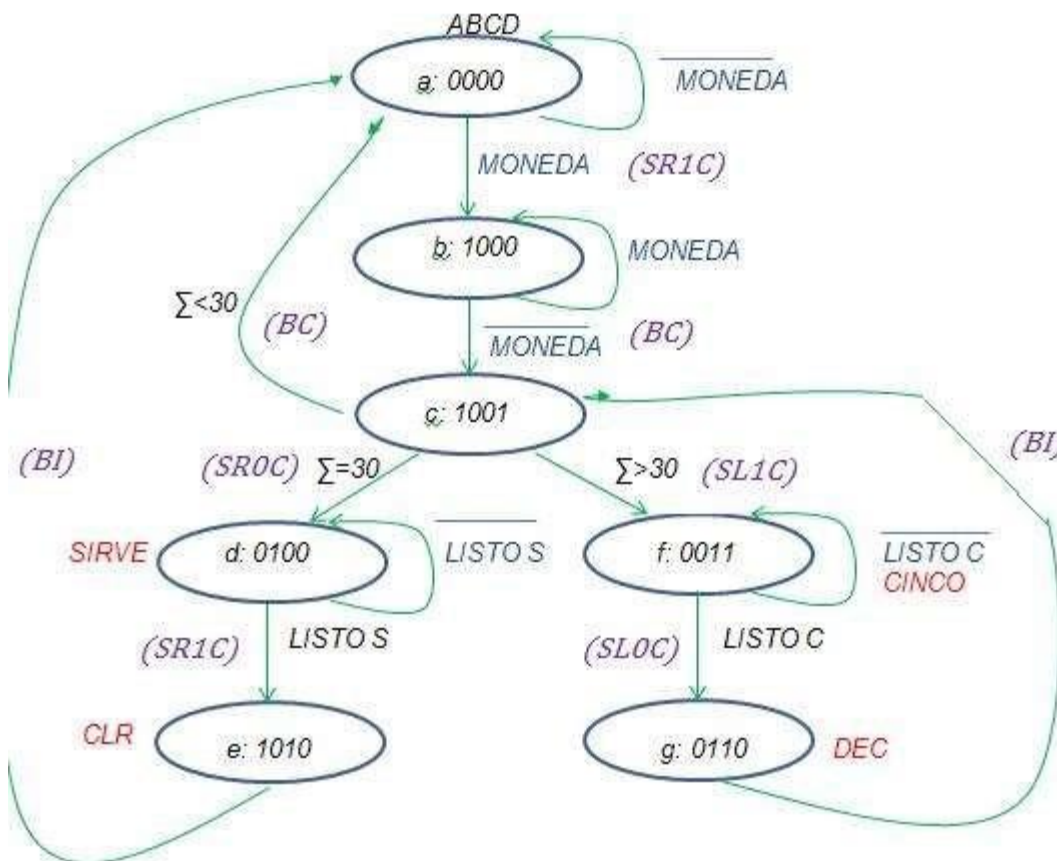
PASOS DE DISEÑO

- Asignar código a los estados del autómata, tratando de usar en lo posible algún tipo de instrucción de las definidas
- Hacer el mapa de M.E.P.

- c) Hacer el mapa de M.E.S. indicando los tipos de acción del registro, así como la variable de condición
- d) Hacer mapas para los estados de control S0 y S1
- e) Hacer los mapas de los bits de carga en paralelo, así como, para R y L
- f) Hacer los mapas de las salidas
- g) Implementar con lógica combinatoria a la entrada, registro y decodificador a la salida.

11.1 SOLUCIÓN AL PROBLEMA - AUTÓMATA

- a) Asignar código a los estados del autómata, tratando de usar en lo posible algún tipo de instrucción de las definidas



a, b, c, d, e, f, g: Estados presentes

ABCD: Código de los estados

b) Hacer el mapa de M.E.P.

| | | | | | |
|-----------|----|-----------|----|----|----|
| | | <i>AB</i> | | | |
| | | 00 | 01 | 11 | 10 |
| <i>CD</i> | 00 | a | d | * | b |
| | 01 | * | * | * | c |
| | 11 | f | * | * | * |
| | 10 | * | g | * | e |

c) Hacer el mapa de M.E.S. indicando los tipos de acción del registro, así como la variable de condición

| | | | | | |
|-----------|----|---------------|---------------|----|--|
| | | <i>AB</i> | | | |
| | | 00 | 01 | 11 | 10 |
| <i>CD</i> | 00 | SR1C, MON | SR1C, LISTO S | * | BC, MON' c |
| | 01 | * | * | * | BC, $\Sigma < a$ SR0C, $\Sigma =$ SL1C, $\Sigma >$ |
| | 11 | SL0C, LISTO C | * | * | * |
| | 10 | * | BI, c | * | BI, a |

d) Hacer mapas para los estados de control S0 y S1. Líneas de control para el registro de corrimiento 74LS194.

| | | | | | |
|-----------|----|-----------|---------|----|-------------|
| | | <i>AB</i> | | | |
| | | 00 | 01 | 11 | 10 |
| <i>CD</i> | 00 | MON | LISTO S | 1 | MON' |
| | 01 | 1 | 1 | 1 | $\Sigma >'$ |
| | 11 | 0 | 1 | 1 | 1 |
| | 10 | 1 | 1 | 1 | 1 |

CONTROL S0

| | | | | | |
|----|----|---------|----|----|-------------|
| AB | | 00 | 01 | 11 | 10 |
| | | 0 | 0 | 1 | MON' |
| CD | 00 | 0 | 0 | 1 | MON' |
| | 01 | 1 | 1 | 1 | $\Sigma ='$ |
| | 11 | LISTO C | 1 | 1 | 1 |
| | 10 | 1 | 1 | 1 | 1 |

CONTROL S1

Explicación: Observando el autómata,

estado a:

si MON=0 -->HOLD, entonces S0, S1=0 0 si MON=1 -->SR, entonces S0, S1=1 0, deducción: S0= MON, S1=0

estado b:

si MON=1 -->HOLD, entonces S0, S1=0 0 si MON=0 -->BC, entonces S0, S1=1 1, deducción: S0= MON', S1=MON'

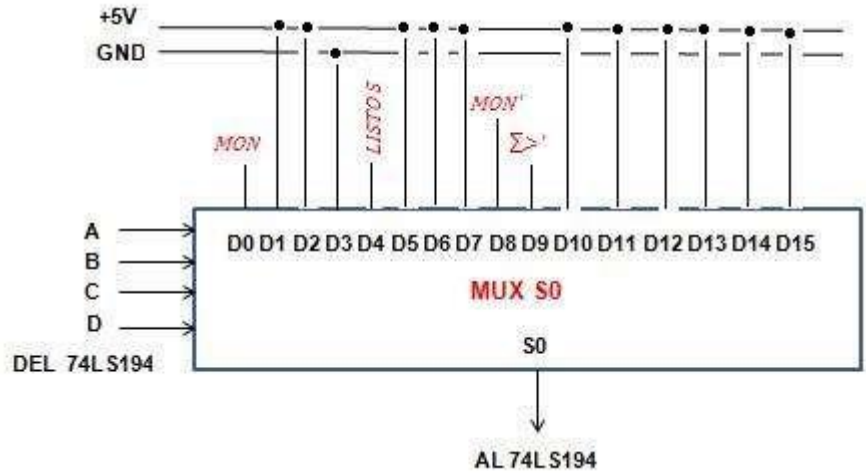
estado c:

si $\Sigma <$ -->BC, entonces S0, S1=1 1 si $\Sigma =$ -->SR, entonces S0, S1=1 0, si $\Sigma >$ -->SL, entonces S0, S1=0 1, deducción: S0= $\Sigma >$ ', S1= $\Sigma =$ '

estado e, g: son BI, S0, S1=1 1

Los demás estados se dejan para brincos, S0, S1 = 11

Estas salidas de control para el registro se implementan con multiplexores. Por ejemplo, para S0:



e) Hacer los mapas de los bits de carga en paralelo A, B, C, D (brincos), así como, para R y L (corrimientos).

Observar los brincos del M.E.S. En estos mapas los bits verdes corresponden a los brincos. Los don't care se ponen en ceros para evitar brincos no programados.

| | | | | | |
|----|----|----|----|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| CD | 00 | 0 | 0 | 0 | 1 |
| | 01 | 0 | 0 | 0 | 0 |
| | 11 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 1 | 0 | 0 |

A

| | | | | | |
|----|----|----|----|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| CD | 00 | 0 | 0 | 0 | 0 |
| | 01 | 0 | 0 | 0 | 0 |
| | 11 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 0 | 0 | 0 |

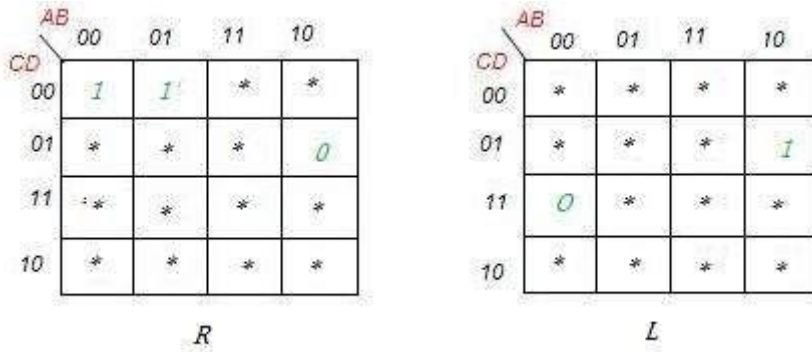
B

| | | | | | |
|----|----|----|----|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| CD | 00 | 0 | 0 | 0 | 0 |
| | 01 | 0 | 0 | 0 | 0 |
| | 11 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 0 | 0 | 0 |

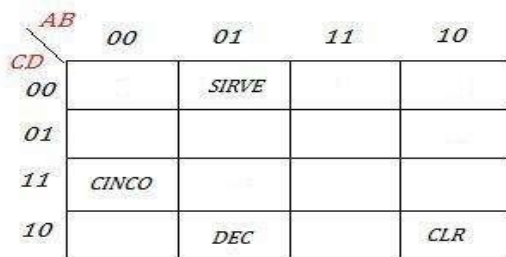
C

| | | | | | |
|----|----|----|----|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| CD | 00 | 0 | 0 | 0 | 1 |
| | 01 | 0 | 0 | 0 | 0 |
| | 11 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 1 | 0 | 0 |

D

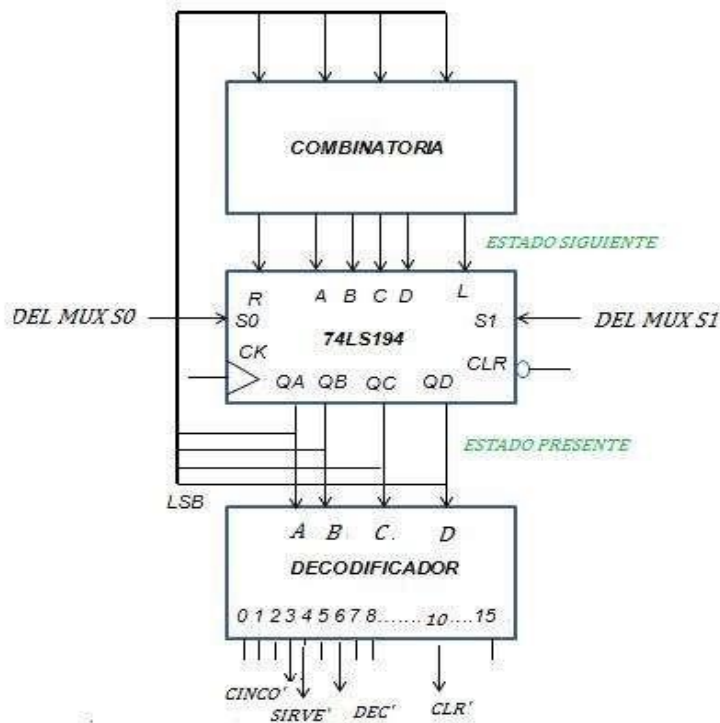


f) Hacer los mapas de las salidas



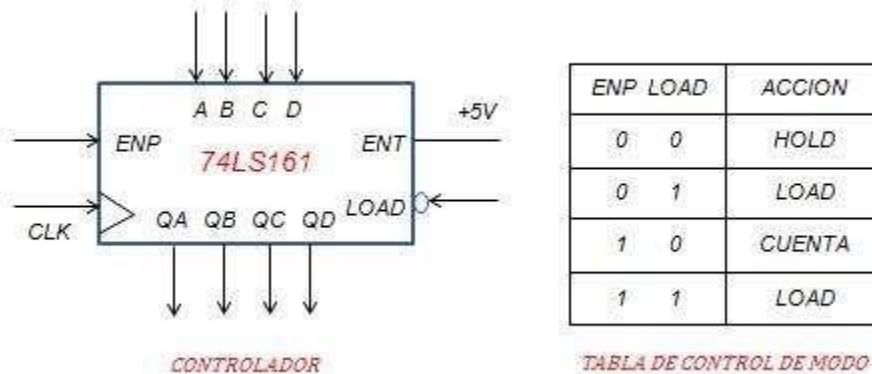
g) Implementar con lógica combinatoria a la entrada, registro y decodificador a la salida.

↓



CAPÍTULO 12. DISEÑO CON CONTADORES

Se utiliza el contador 74LS161 que tiene la siguiente tabla de control de modo.



Características del 74LS161:

- Es de 4 bits y posibilidad de conexión en cascada
- Tiene habilitador de cuentas ENP
- Permite hacer cargas en paralelo con LOAD
- Clear asíncrono
- Carga con el flanco de subida del reloj

Instrucciones o comandos:

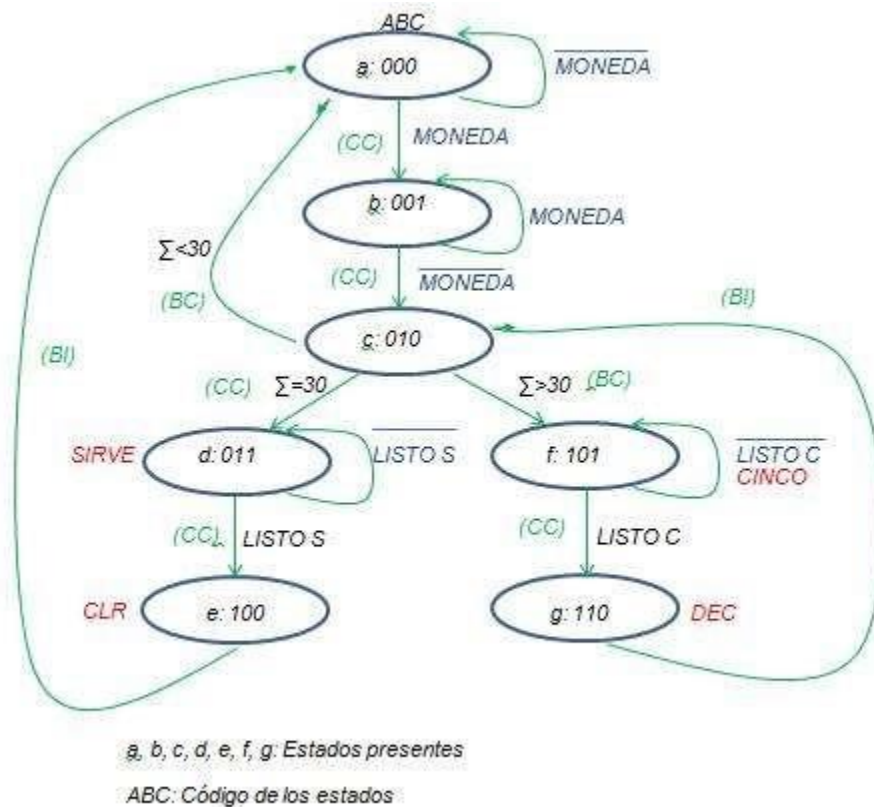
- CC, CI: cuenta condicional e incondicional
- BC, BI: brinco condicional e incondicional

12.1 PASOS DE DISEÑO

1. Asignar códigos y comandos al autómata
2. Hacer el mapa M.E.P.
3. Hacer el mapa M.E.S. En caso de que se tengan brinco indicar la dirección respectiva. Indicar también las variables que indican los cambios
4. Hacer mapa de control de modos para ENP y LOAD
5. Hacer mapas de cargas en paralelo y de salida
6. Implementar el controlador

12.2 SOLUCIÓN AL PROBLEMA - AUTÓMATA

1) Asignar códigos y comandos al autómata



- 1) Hacer el mapa M.E.P.
- 2) Hacer el mapa M.E.S. En caso de que se tengan brincos indicar la dirección respectiva. Indicar también las variables que indican los cambios

| | | | | | |
|---|---|----|----|----|----|
| | | AR | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | a | c | g | e |
| | 1 | b | d | * | f |

M.E.P

| | | | | | |
|---|---|---------|---|--------------------|--------------------|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | CC, MON | $\Sigma < 30, BC \rightarrow a$ $\Sigma = 30, CC$ $\Sigma > 30, BC \rightarrow f$ | Bl $\rightarrow c$ | Bl $\rightarrow a$ |
| | 1 | CC, MON | CC, LISTO S | * | CC, LISTO |

M.E.S

3) Hacer mapa de control de modos para ENP y LOAD

| | | | | | |
|---|---|------|---------|----|---------|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | MON | 1 | * | * |
| | 1 | MON' | LISTO S | * | LISTO C |

ENP

| | | | | | |
|---|---|----|-------------|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | 0 | $\Sigma ='$ | 1 | 1 |
| | 1 | 0 | 0 | * | 0 |

LOAD

Estado a:

MON -> CC ENP, LOAD= 1 0 MON' -> HLT, ENP, LOAD= 0 0 deducción para los mapas: ENP: MON, LOAD: 0

Estado c:

$\Sigma <$ -> BRINCO ENP, LOAD= * 1

$\Sigma =$ -> CUENTA ENP, LOAD= 1 0 $\Sigma >$ -> BRINCO ENP, LOAD= * 1 deducción para los mapas: ENP:1, LOAD: $\Sigma ='$

Estado g:

BRINCO ENP, LOAD= * 1

4) Hacer mapas de cargas en paralelo A, B, C (brincos) y de salida

| | | | | | |
|---|---|----|------------|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | * | $\Sigma >$ | 0 | 0 |
| | 1 | * | * | * | * |

$A = A'(\Sigma >)$

| | | | | | |
|---|---|----|----|----|----|
| | | AB | | | |
| | | 00 | 01 | 11 | 10 |
| C | 0 | * | 0 | 1 | 0 |
| | 1 | * | * | * | * |

$B = AB$

| | | | | | | |
|----------|-----------|---|------------|----|----|----|
| | <i>AB</i> | | 00 | 01 | 11 | 10 |
| <i>C</i> | 0 | * | $\Sigma >$ | 0 | 0 | |
| | 1 | * | * | * | * | |

$C = A'(\Sigma >)$

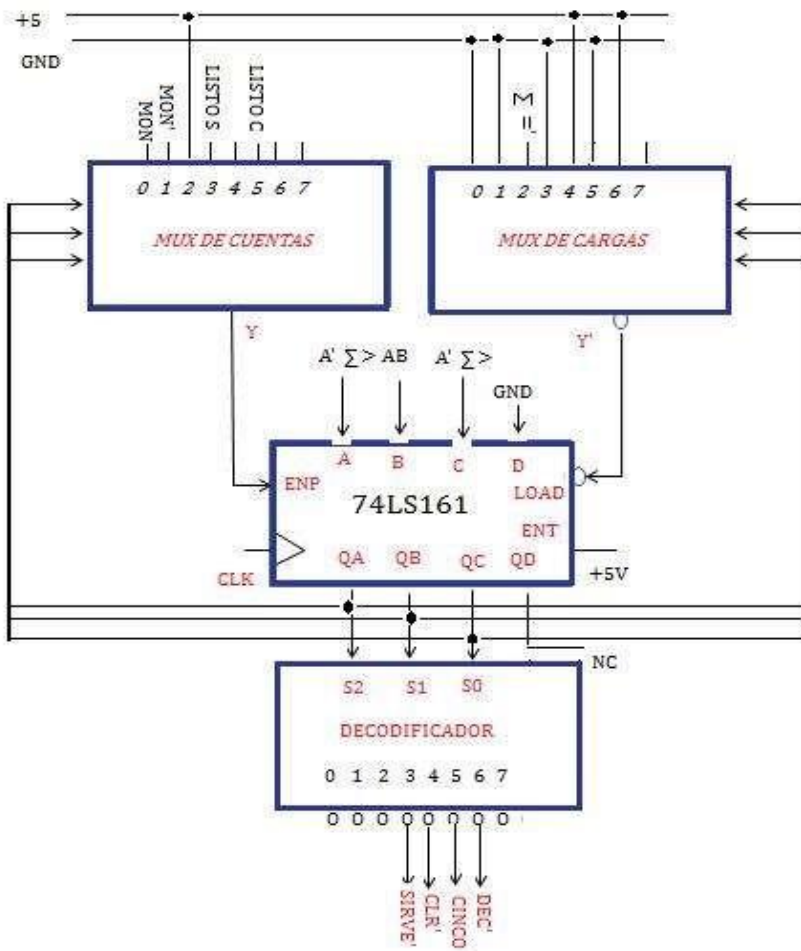
Estado c:

$\Sigma < \rightarrow a, ABC = 000 \quad \Sigma > \rightarrow f, \rightarrow ABC = 101$ deducción: $A = \Sigma >, B = 0, C = \Sigma >$

| | | | | | | |
|----------|-----------|--|-------|-----|-------|----|
| | <i>AB</i> | | 00 | 01 | 11 | 10 |
| <i>C</i> | 0 | | | DEC | CLR | |
| | 1 | | SIRVE | | CINCO | |

SALIDAS

5) Implementar el controlador



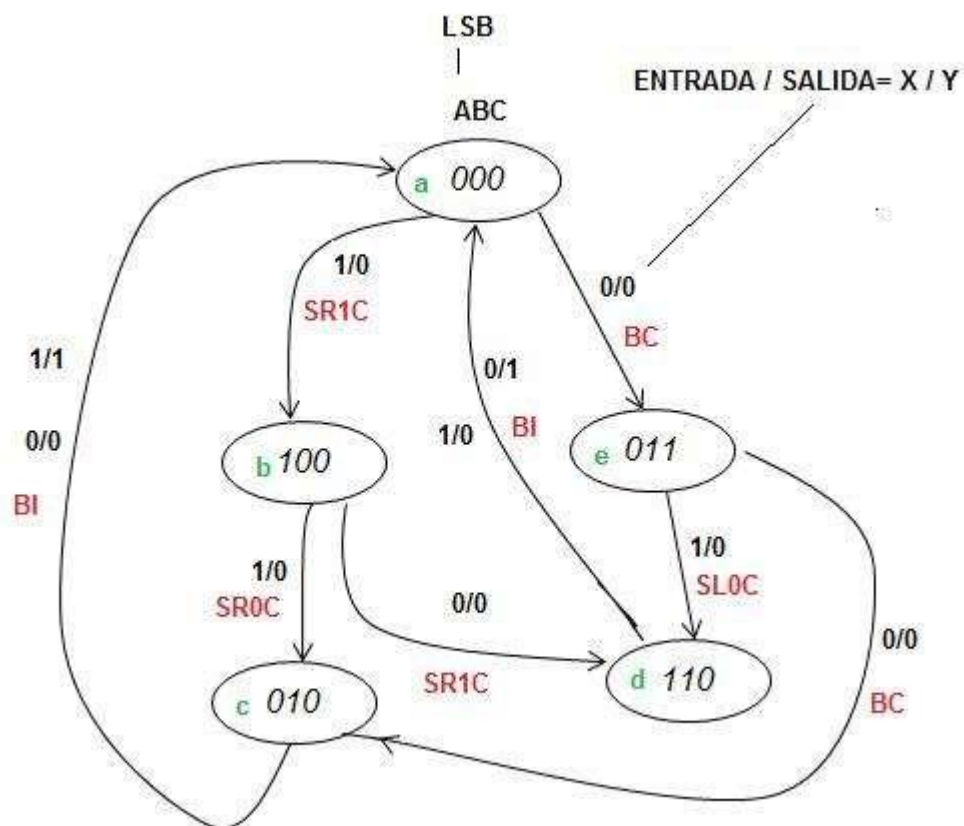
CAPÍTULO 13. CONTROLADORES - SIMULACIÓN

En esta unidad vamos a hacer dos simulaciones en Proteus para consolidar los conceptos teóricos sobre controladores digitales utilizando como controlador el registro de corrimiento 74194.

CONTROLADOR DIGITAL CON REGISTRO SR 74194

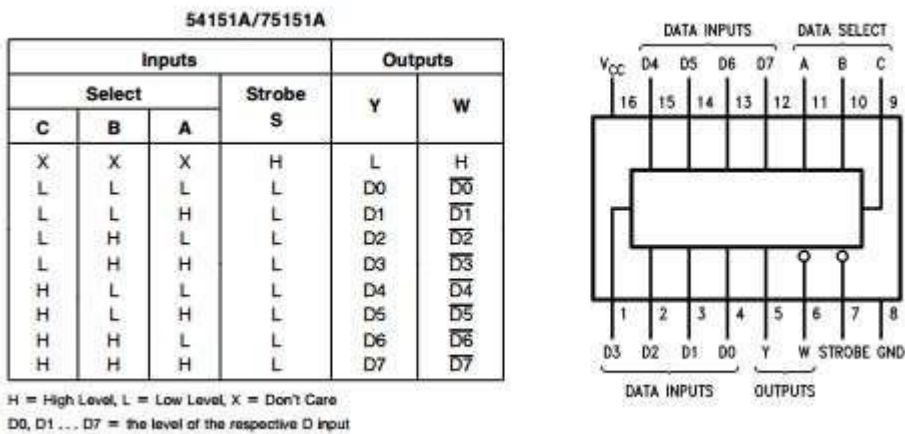
Simule el controlador digital que implemente el siguiente autómata utilizando como controlador el circuito integrado SR 74194.

AUTÓMATA

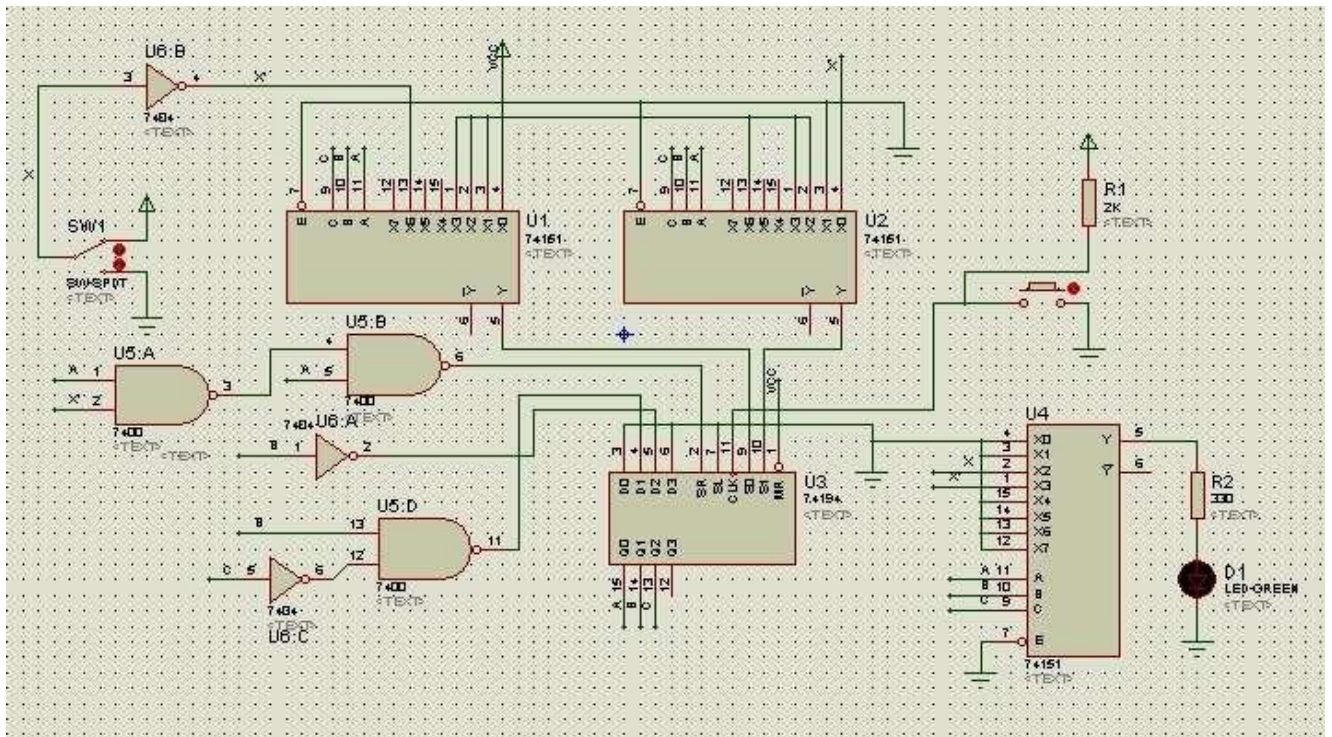


METODOLOGÍA

1. Haga el Mapa de estados presente MEP
2. Haga el Mapa de estados siguiente MES
3. Haga el Mapa para: S1, S0, R, L
4. Haga los Mapas para los brincos: A, B, C
5. Haga el Mapa de salidas
6. Simule el siguiente circuito que es la solución del autómata 7. Compruebe el funcionamiento del autómata.



CIRCUITO



CAPÍTULO 14. CONTROLADORES - LABORATORIO

En esta unidad vamos a hacer la implementación del controlador digital usando el registro de corrimiento 74194.

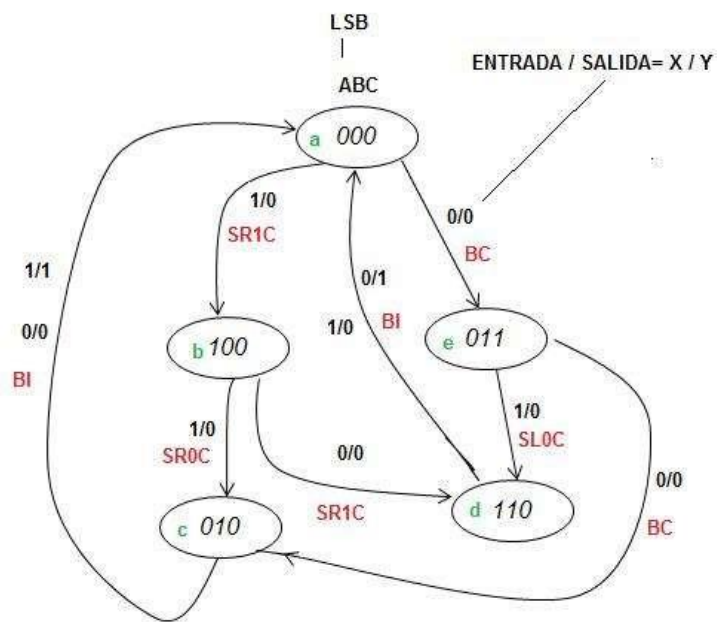
EQUIPO Y MATERIAL NECESARIO

- Fuente de 5V
- Protoboard
- Circuitos integrados: 555, 3x74LS151, 74LS194, 74LS00, 74LS04
- Switch SP2T
- Un pulsador
- 4 LEDs
- Resistencias a 1/4W de: 4x330Ω
- Conectores

CONTROLADOR DIGITAL CON REGISTRO SR 74194

Implemente el controlador digital del autómata de la figura, utilizando como controlador el circuito integrado SR 74194.

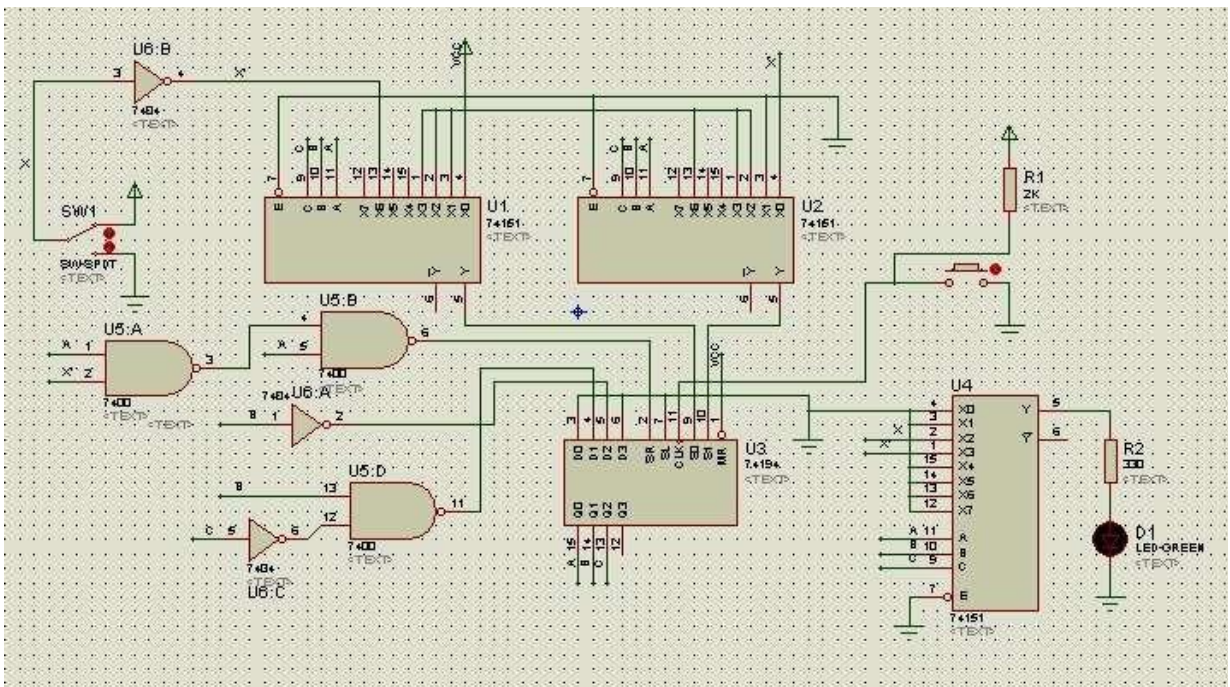
1. AUTÓMATA



2. METODOLOGÍA

1. Haga el Mapa de estados presente MEP
2. Haga el Mapa de estados siguiente MES
3. Haga el Mapa para: S1, S0, R, L
4. Haga los Mapas para los brincos: A, B, C
5. Haga el Mapa de salidas
6. Simule el siguiente circuito que es la solución del autómata
7. Compruebe el funcionamiento del autómata.

3. CIRCUITO



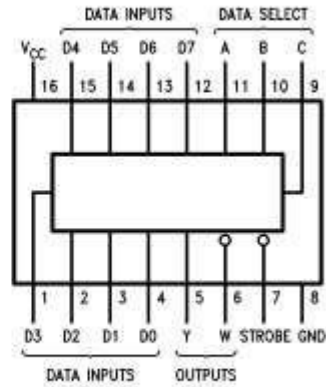
CIRCUITOS INTEGRADOS

74151 MUX

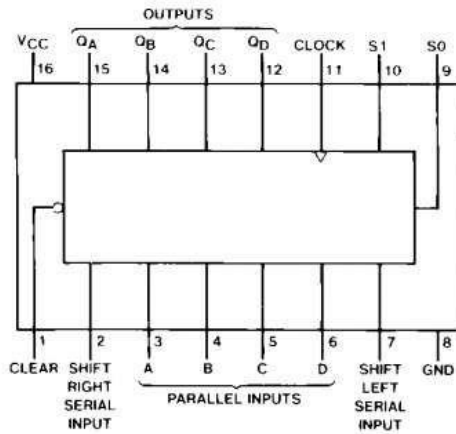
54151A/75151A

| Inputs | | | | Outputs | |
|--------|---|---|-------------|---------|-----------------|
| Select | | | Strobe S | Y | W |
| C | B | A | | | |
| X | X | X | H | L | H |
| L | L | L | L | D0 | $\overline{D0}$ |
| L | L | H | L | D1 | $\overline{D1}$ |
| L | H | L | L | D2 | $\overline{D2}$ |
| L | H | H | L | D3 | $\overline{D3}$ |
| H | L | L | L | D4 | $\overline{D4}$ |
| H | L | H | L | D5 | $\overline{D5}$ |
| H | H | L | L | D6 | $\overline{D6}$ |
| H | H | H | L | D7 | $\overline{D7}$ |

H = High Level, L = Low Level, X = Don't Care
 D0, D1 ... D7 = the level of the respective D input



74194 S-R



| Clear | Inputs | | | | Outputs | | | | | | | |
|-------|--------|----|-------|--------|---------|----------|---|----|-----|-----|-----|-----|
| | Mode | | Clock | Serial | | Parallel | | QA | QB | QC | QD | |
| | S1 | S0 | | Left | Right | A | B | | | | | C |
| L | X | X | X | X | X | X | X | X | L | L | L | L |
| H | X | X | L | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| H | H | H | ↑ | X | X | a | b | c | a | b | c | d |
| H | L | H | ↑ | X | H | X | X | X | H | QA0 | QB0 | QC0 |
| H | L | H | ↑ | X | L | X | X | X | L | QA0 | QB0 | QC0 |
| H | H | L | ↑ | H | X | X | X | X | QA0 | QB0 | QC0 | H |
| H | H | L | ↑ | L | X | X | X | X | QA0 | QB0 | QC0 | L |
| H | L | L | X | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |

BIBLIOGRAFÍA RECOMENDADA

LIBROS DE TEXTO Y REFERENCIAS ACADÉMICAS

- Floyd, Thomas L. (2015). Fundamentos de Sistemas Digitales (11ª ed.). México: Pearson Educación.
- Referencia estándar en universidades técnicas; excelente para compuertas, familias lógicas y diseño secuencial.
- Tocci, Ronald J.; Widmer, Neal S. & Moss, Gregory L. (2013). Sistemas Digitales: Principios y Aplicaciones (11ª ed.). México: Pearson.
- Cobertura completa de circuitos combinacionales, secuenciales, contadores y registros con enfoque práctico.
- Mano, M. Morris & Kime, Charles R. (2016). Fundamentos de Diseño Lógico y de Computadoras (5ª ed.). Madrid: Cengage Learning.
- Texto académico riguroso para álgebra de Boole, simplificación, mapas de Karnaugh y síntesis de máquinas de estado.
- Roth, Charles H. & Kinney, Larry L. (2014). Fundamentos de Diseño Lógico (7ª ed.). México: Cengage Learning.
- Ideal para diseño sistemático de circuitos secuenciales, diagramas de estados y minimización de lógica.
- Wakerly, John F. (2006). Diseño Digital: Principios y Prácticas (4ª ed.). México: Pearson.
- Enfoque moderno en diseño con circuitos integrados, temporización, metastabilidad y buenas prácticas de PCB.

DISEÑO PRÁCTICO, SIMULACIÓN Y LABORATORIO

- Kleitz, William. (2017). Electrónica Digital: Un Enfoque Práctico (8ª ed.). Madrid: Pearson. Guía paso a paso para montaje en protoboard, uso de osciloscopio, generación de clocks y depuración de circuitos.
- Labrosse, Jean J. (2010). Embedded Systems: Introduction to ARM® Cortex™-M Microcontrollers. (Referencia complementaria para transición a microcontroladores). Útil como puente hacia sistemas embebidos modernos tras dominar lógica discreta.
- Proteus Design Suite. (2023). User Manual & Simulation Guide. LabCenter Electronics. Documentación oficial para simulación de circuitos digitales, generación de formas de onda y validación antes del montaje físico.

HOJAS DE DATOS Y REFERENCIAS DE FABRICANTES

- Texas Instruments. (2023). Logic Guide (SCLD001). Dallas: TI Semiconductor.
- Catálogo actualizado de familias 74HC, 74HCT, 74LV, 74LVC con parámetros eléctricos, diagramas de pines y aplicaciones.

- NXP Semiconductors. (2022). CMOS Logic Data Handbook. Eindhoven: NXP.
- Especificaciones técnicas, niveles de voltaje, consumo y recomendaciones de manejo para circuitos integrados CMOS.
- ON Semiconductor. (2021). TTL/CMOS Logic IC Selection Guide. Phoenix: ON Semi.
- Guía de selección, compatibilidad de niveles y notas de aplicación para interfaces entre familias lógicas.

PUBLICACIONES DEL AUTOR EN AMAZON KDP

Esta serie complementa el presente texto, formando una ruta formativa integral desde los fundamentos eléctricos hasta el control digital avanzado.

- Polanía Puentes, Jorge Antonio. (2025). Fundamentos de Circuitos Eléctricos: Teoría y Práctica. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Semiconductores, Diodos y Transistores. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Amplificadores Transistorizados: Multietapas, Video y Sintonizados. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Osciladores y Multivibradores Transistorizados. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Señales y Sistemas Continuos. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Señales y Sistemas Discretos. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Aprende Control con MATLAB. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Control de Motores con MATLAB. Amazon KDP.
- Polanía Puentes, Jorge Antonio. (2025). Instalaciones Eléctricas Residenciales. Amazon KDP.

RECURSOS DIGITALES Y CURSOS DE APOYO

- All About Circuits. (2023). Digital Logic Textbook. Recuperado de <https://www.allaboutcircuits.com>
- Khan Academy. (2023). Ingeniería Eléctrica: Lógica Digital. Recuperado de <https://es.khanacademy.org>
- Coursera / edX. (2023). Digital Systems Design & FPGA Prototyping. Cursos universitarios abiertos.

SOBRE EL AUTOR

1. FORMACIÓN ACADÉMICA

Secundaria: Bachiller, 1969. Colegio Nacional Santa Librada de Neiva Pregrado: Ingeniero Electrónico, 1976. Universidad Distrital de Santafé de Bogotá. Tesis Meritoria. Matrícula profesional: CN206-45500 del Consejo Profesional Nacional de Ingenierías Eléctrica, Mecánicas y Profesiones afines. Postgrado: Magister en Ingeniería Electrónica, 1991. Universidad Nacional Autónoma de México. Mención Honorífica. Convalidación Resolución No 823 de 1994 del ICFES. Cursos de postgrado: Microprocesadores y sistemas de desarrollo. Sistemas de comunicación de datos. Interconexión de redes usando TCP/IP. Sistemas de telecomunicaciones por satélite. HTML Scripting. Autoevaluación institucional. Programación Turbo Pascal. Planeación institucional. Planeación y desarrollo institucional.

2. EXPERIENCIA ACADÉMICA

Experiencia académica de 30 años como Profesor de Tiempo Completo de la Universidad Surcolombiana adscrito a la Facultad de Ingeniería, del 7 de febrero de 1977 al 30 de marzo de 2007. Profesor Titular desde el 1 de diciembre de 1993. Profesor de Pregrado en el Programa de Ingeniería Electrónica de las asignaturas: Electrónica Analógica, Arquitectura de computadores. Electrónica Digital, Microcontroladores. Programación en Matlab. Control digital. Señales y sistemas. Procesamiento digital de señales. Energías Alternativas. Profesor de Postgrado en la Maestría de Gestión e Ingeniería Ambiental de las asignaturas: Energética Ambiental y Simulación de Sistemas Ambientales.

3. EXPERIENCIA ADMINISTRATIVA

Experiencia administrativa de más de 10 años en los siguientes cargos: Decano de la Facultad de Ingeniería, Universidad Surcolombiana. 1984-1986 Decano de la Facultad de Ingeniería, Universidad Surcolombiana. 1987-1988 Jefe de la Oficina de Planeación, Universidad Surcolombiana. 1988 - 1989 Rector de la Universidad Surcolombiana. 1997 - 2000 Director del Postgrado de Automatización industrial. 2000 Jefe de Programa de Ingeniería Electrónica. 2000 - 2001 Director del Departamento de Ingeniería Electrónica. 2006

4. EXPERIENCIA INVESTIGATIVA

Coordinador del grupo de investigación Nuevas tecnologías. Categoría C Colciencias. 2005. Proyectos de investigación: "Simulación de algoritmos de control digital con Matlab". 2005. "Diseño y construcción de un electrocardiógrafo digital inalámbrico". 2006. Diseño e implementación de un sistema de riego inteligente para un cultivo de mango". 2007. Hoja de vida CvLAC COLCIENCIAS

5. PRODUCCIÓN INTELECTUAL

Cursos publicados en forma virtual en la web www.ceduvirt.com: Circuitos eléctricos. Semiconductores. Electrónica básica. Electrónica industrial. Electrónica digital. Microcontroladores. Señales y sistemas. Control digital con Matlab, Procesamiento de señales. Procesos con Matlab y Simulink. Teoría de sistemas. Energía solar. Aplicaciones en Ingeniería ambiental.