

# DISEÑO DE CIRCUITOS DIGITALES

Glendalys Figueroa Freytes, Isaura Rivera

El objetivo de este experimento es el diseño de circuitos digitales a base de un problema real para la realización de un programa utilizando la tabla veritativa la cual representa cada una de las posibles combinaciones y el álgebra Booleana.

## I. Introducción

La electrónica moderna usa electrónica digital para realizar muchas funciones. Aunque los circuitos electrónicos pueden resultar muy complejos, en realidad se construyen de un número muy grande de circuitos. En un circuito digital se transmite información binaria entre estos circuitos y se consigue un circuito complejo con la combinación de bloques de circuitos simples (compuertas lógicas).

La mayoría de las compuertas tienen dos entradas y una salida. En cualquier momento, cada compuerta está en una de las dos condiciones binarias “low”(0) el cual corresponde a falso o “high”(1) el cual corresponde a verdadero, representados por diferentes niveles de voltaje.

Generalmente el estado lógico de una compuerta cambia mucho a medida que el circuito procesa los datos. En la mayoría de las compuertas lógicas, el estado bajo es aproximadamente cero voltios, mientras el estado alto es aproximadamente cinco voltios positivos. Los sistemas digitales se construyen usando tres compuertas logicas básicas. Estas compuertas son la compuerta AND, compuerta OR y la compuerta NOT (Figura 1). De estas tres compuertas lógicas se derivan otras tres: NOR, NAND y EOR (Figura 1).

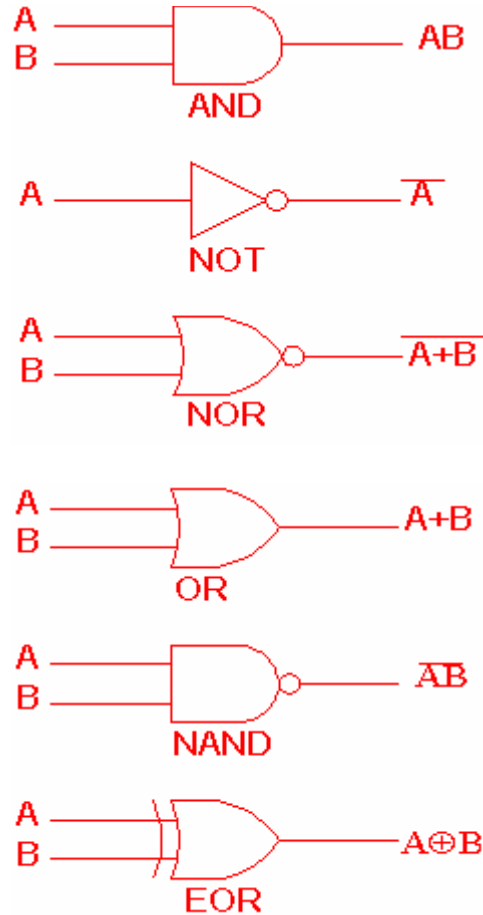


Figura 1: Símbolos de las compuertas lógicas.

## II. Experimento

Se resolvió el problema dado en clase mediante el álgebra Booleana, tablas veritativas y mapas de Carnot. Luego se hizo un programa con la función resultante del álgebra Booleana en Visual Basic.

### III. Problema

En la familia Pérez, German y Janet tienen dos niños, Joe y Susy. Cuando van a comer fuera ellos van a un restaurante que sirven solo hamburgers (0) o uno donde solo sirven pollo (1). Antes de salir la familia vota para decidir que restaurante escoger. La mayoría gana, excepto cuando mamá y papá están de acuerdo en este caso ellos deciden. Si hay otro empate todos van al restaurante de pollo. Diseñe un circuito lógico que automáticamente seleccione el restaurante cuando cada miembro vote y simplifíquelo utilizando álgebra booleana o “k-maps”. Luego ensamble el circuito utilizando la serie TTL (74) y utilice un LED para representar el “output”.

### IV. Análisis y datos

Se completó la siguiente tabla veritativa del problema anterior.

German (A)	Janet (B)	Joe (C)	Susy (D)	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	0	1
0	1	1	1	1
0	1	1	0	1
1	0	0	1	0
1	0	0	0	1
1	0	1	1	1
1	0	1	0	1
1	1	0	1	1
1	1	0	0	1
1	1	1	1	1
1	1	1	0	1

Simplifique la función utilizando álgebra Booleana o K-Maps.

Función Resultante:

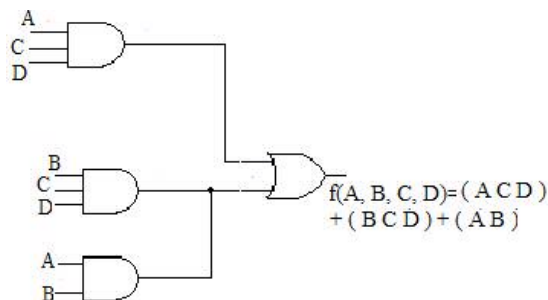
$$\overline{f(A, B, C, D)} = (\overline{A + C + D})(\overline{B + C + D})(\overline{A + B})$$

$$\overline{f(A, B, C, D)} = (\overline{A + C + D})(\overline{B + C + D})(\overline{A + B})$$

Función Simplificada:

$$f(A, B, C, D) = (A C D) + (B C D) + (A B)$$

### V. Esquemático del Circuito



### VII. Preguntas

1. ¿Cuáles son los componentes de la serie TTL que necesitó para ensamblar este circuito?

Los componentes de la serie TTL utilizados en Multisim para el diseño de el circuito fueron el 7408 para el AND y el 7432 para el OR.

2. ¿Qué ajustes le haría al circuito para mejorarlo?

Para mejorar el circuito se podría hacer el arreglo con exclusive OR (EOR) ó se pueden convertir todos los componentes del circuito en un mismo componente para ahorrar dinero.

3. Explique las diferencias entre la familia TTL y CMOS.

La familia del transistor-transistor-lógico (TTL) fue desarrollado en el uso de transistores tipo interruptores para operaciones lógicas y define los valores binarios como;

$$0 \text{ V a } 0.8 \text{ V} = 0$$
$$2 \text{ V ta } 5 \text{ V} = 1$$

Los TTL son la familia de circuitos integrados más numerosa aunque la familia CMOS está creciendo rápidamente. Los TTL son de bajo costo, pero requieren de mucha potencia y su fuente debe de ser de +5V. Las compuertas individuales deben requerir de 3 a 4 mA. Las versiones Schottky de baja potencia de los “chips” TTL requieren de solamente el 20% de la potencia, pero son más caros.

La familia del “complementary metal oxide semiconductor family” (CMOS) tienen equivalentes para la mayoría de los “chips” TTL. Los “chips” de CMOS poseen menor requerimientos de potencia (1 mA) y operan con un rango amplio de voltaje en la fuente (3 a 18V). Los números de los modelos CMOS tendrán una C en la mitad de éstos. Un ejemplo de esto es el CMOS 74C04 el cual es el equivalente al TTL 7404. Una gran desventaja es su extrema sensibilidad a la electricidad estática. Éstos deben ser protegidos de descargas estáticas

## VIII. Advance

### 1. Explique las compuertas lógicas elementales.

Como ya se había mencionado, los sistemas digitales se construyen usando tres compuertas lógicas básicas; AND, OR y NOT (Figura 1). De estas tres compuertas lógicas se derivan otras tres: NOR, NAND y EOR (Figura 1).

Una compuerta AND usualmente se compone de dos entradas y una salida la cual es igual al producto AND de las entradas lógicas; es decir,  $A \cdot B$ . La compuerta AND es un circuito que opera en forma tal que su salida es alta sólo cuando todas sus entradas son altas. En todos los otros casos la salida de la compuerta AND es baja.

En un circuito digital la compuerta OR es un circuito que tiene dos o más entradas y cuya salida es igual a la suma OR de las entradas. Las entradas A y B son niveles lógicos y la salida es un nivel de voltaje cuyo valor es el resultado de la operación OR de A y B ( $A+B$ ). Su salida es 0 únicamente cuando ambas entradas son 0. En los demás casos la salida es 1.

Un NAND está compuesto de un AND y un invertidor (NOT). La compuerta NAND opera igual a la de AND seguida de un invertidor. Por lo que su salida es baja cuando ambas entradas son altas. En los otros casos la salida del NAND es alta.

La compuerta NOR de dos entradas y una salida, es la combinación de la compuerta OR con un invertidor. De este modo, la operación de la compuerta NOR es que cuando todas sus entradas son bajas su salida es alta. En los demás casos la salida es baja.

La compuerta EOR (exclusive OR) regularmente posee dos entradas y una salida. La salida es 1 si y solo si A es diferente de B, si A y B son ambas 0 o ambas son 1 entonces la salida es 0.

La compuerta NOT, a la cual se le llama más comúnmente invertidor, siempre tiene una sola entrada y su nivel lógico de salida siempre es contrario al nivel lógico de la entrada.

### 2. ¿Cuáles son los números descriptivos de los componentes en la serie TTL y CMOS?

Algunos ejemplos de números utilizados para describir a los TTL son: 74, 74H, 74L, 74S, 74LS, 74AS y 74ALS. En los CMOS se utiliza una C en la numeración de los componentes. Para los CMOS se utilizan: 40, 140, 74C, 74HC, 74HCT.

## **V. Conclusión**

Mediante este trabajo se resolvió un problema de lógica digital a base de un problema real utilizando tablas veritativas, k-maps y álgebra booleana. Mediante el álgebra booleana se consiguió la función a utilizar en la realización de un programa en “visual basic”. También se estudió más a fondo y se diseñó los diferentes tipos de compuertas lógicas y sus funciones.

## **VI. Referencia**

1. <http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/icomp.html#c5>
2. [www.unicrom.com/Tut\\_circuitoslogicos.asp](http://www.unicrom.com/Tut_circuitoslogicos.asp)