UNIVERSIDAD SIMÓN BOLÍVAR

DEPARTAMENTO DE ELECTRÓNICA Y CIRCUITOS

LABORATORIO DE CIRCUITOS DIGITALES EC-2072

INFORME - PRÁCTICA #1

INTRODUCCIÓN A LA LÓGICA COMBINATORIA

Profesor

Mauricio Pérez

Estudiante

Giancarlo Torlone 20-10626

ÍNDICE

INTRODUCCIÓN

En la siguiente práctica de laboratorio se diseña un circuito de lógica combinatoria que indica si una letra pertenece o no a un apellido previamente escogido al momento de ingresar alguno de los códigos binarios de 4 bits establecidos en una tabla dada. El apellido escogido deberá tener por lo menos 6 letras y si alguna de ellas se repite, deberá ser un máximo dos veces.

MARCO TEÓRICO

Álgebra Booleana

Es una rama especial del álgebra que se usa principalmente en electrónica digital. La álgebra booleana fue inventada en 1854 por el matemático inglés George Boole.

El álgebra de Boole es un método para simplificar los circuitos lógicos (o a veces llamados circuitos de conmutación lógica) en electrónica digital.

Por lo tanto, también se llama como "Cambio de álgebra". Podemos representar el funcionamiento de los circuitos lógicos utilizando números, siguiendo algunas reglas, que son bien conocidas como "Leyes del álgebra de Boole".

También podemos hacer los cálculos y las operaciones lógicas de los circuitos aún más rápido siguiendo algunos teoremas, que se conocen como "Teoremas del álgebra de Boole". Una función booleana es una función que representa la relación entre la entrada y la salida de un circuito lógico.

La lógica booleana sólo permite dos estados del circuito, como True y False. Estos dos estados están representados por 1 y 0, donde 1 representa el estado "Verdadero" y 0 representa el estado "Falso".

Leyes fundamentales del álgebra booleana

OR

$$A + 0 = A$$
$$A + 1 = 1$$
$$A + A = A$$
$$A + \overline{A} = 1$$

AND

$$A \cdot 0 = 0$$
$$A \cdot 1 = A$$
$$A \cdot A = A$$
$$A \cdot \overline{A} = 0$$

NOT

$$\overline{\overline{A}} = A$$

- **Leyes conmutativas**

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

- **Leyes asociativas**

$$(A + B) + C = A + (B + C)$$
$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

- **Leyes distributivas**

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$
$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

- **Otras identidades útiles**

$$A + (A \cdot B) = A$$
$$A \cdot (A + B) = A$$
$$A + (A \cdot B) = A + B$$
$$(A + B) \cdot (A + B) = A$$
$$(A + B) \cdot (A + C) = A + (B \cdot C)$$
$$A + B + (A \cdot B) = A + B$$
$$(A \cdot B) + (B \cdot C) + (B \cdot C) = (A \cdot B) + C$$
$$(A \cdot B) + (A \cdot C) + (B \cdot C) = (A \cdot B) + (B \cdot C)$$

Teorema de De Morgan

Las leyes o teoremas de Morgan son herramientas esenciales tanto en la lógica proposicional como en el álgebra de Boole. De manera general, se define como la equivalencia que existe entre dos proposiciones lógicamente equivalentes. Su aplicación permite simplificar expresiones booleanas.

- **Primer Teorema:** El complemento de un producto de variables es igual a la suma de los complementos de las variables.

$$\overline{(A \cdot B)} \equiv (\overline{A} + \overline{B})$$

- **Segundo Teorema:** El complemento de una suma de variables es igual al producto de los de los complementos de las variables.

$$\overline{A + B} \equiv \overline{A} \cdot \overline{B},$$

Simplificación de funciones booleanas

Al usar los teoremas de De Morgan y leyes booleanas, podemos simplificar las expresiones booleanas, mediante las cuales podemos reducir el número requerido de compuertas lógicas a implementar. Podemos simplificar la función Boolean utilizando dos métodos:

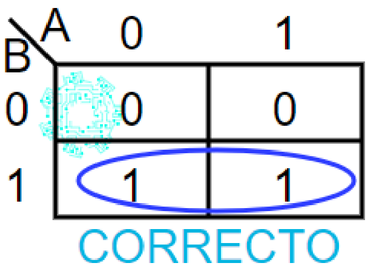
1. El método algebraico: mediante el uso de identidades (leyes booleanas).
2. El método gráfico: utilizando el método del Mapa de Karnaugh.

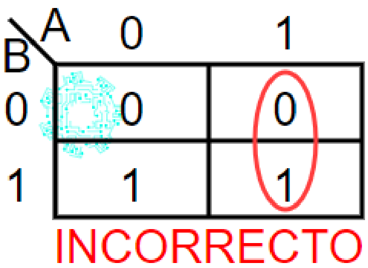
Mapas de Karnaugh

El mapa de Karnaugh o mapa-k es un diagrama utilizado para la simplificación de funciones algebraicas Booleanas, permitiendo de manera gráfica reconocer patrones y así reduce la necesidad de hacer cálculos extensos para la simplificación de expresiones booleanas.

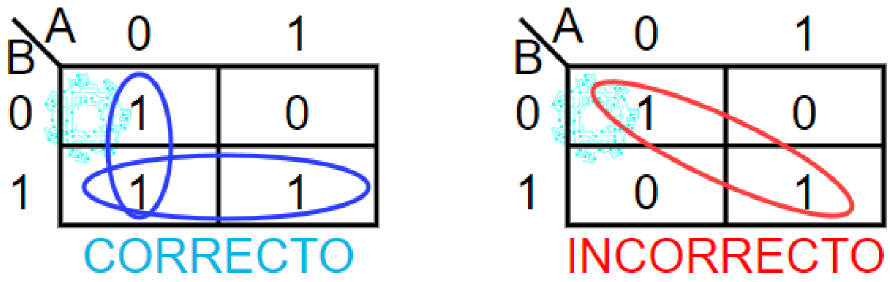
Reglas del Mapa de Karnaugh

1. Las agrupaciones o el término a considerar únicamente será del número “1”.

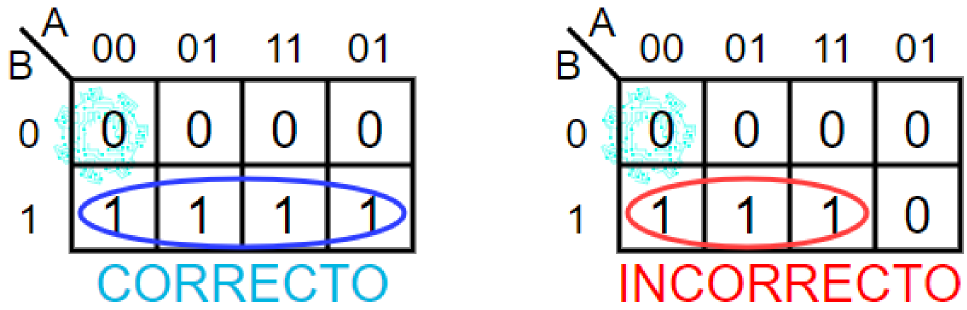




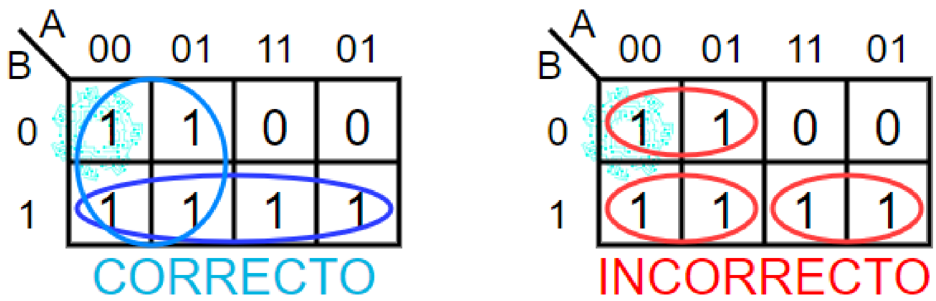
2. Las agrupaciones únicamente se deben hacer en horizontal y vertical.



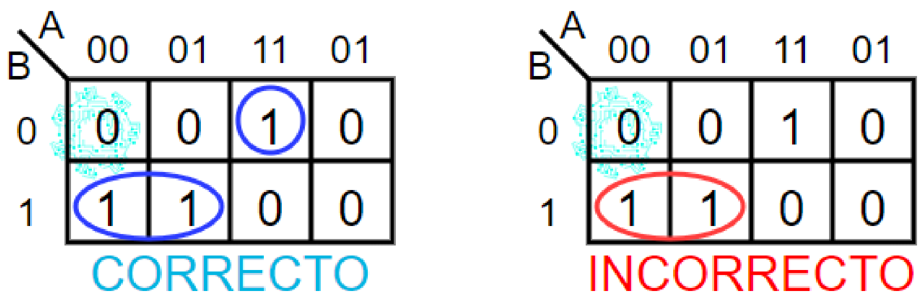
3. Las agrupaciones a considerar deben contener 2n elementos. Es decir cada agrupación que contiene cada grupo tendrá 1, 2, 4,8,…, 2n cantidad de número de uno o unos.



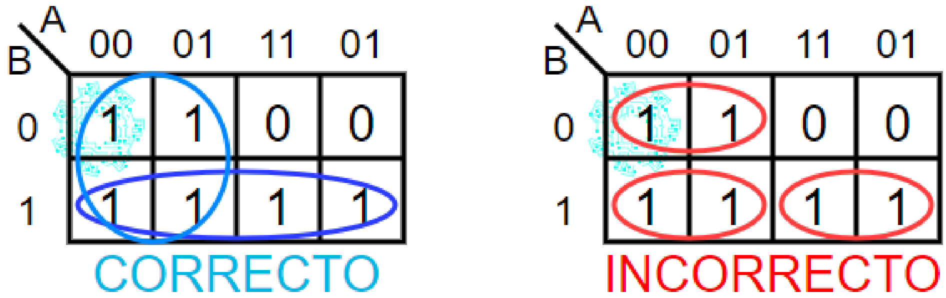
4. Para una mejor simplificación se debe considerar el grupo más grande posible.



5. Se debe considerar todo número “1”



6. Es posible solapar grupos de “1”.



7. La formación de grupos también se pueden producir con las celdas extremas de la tabla.
8. Debemos considerar el menor número de agrupaciones o grupos posibles obedeciendo las reglas anteriores.

ANÁLISIS DE RESULTADOS

Apellido escogido: TORLONE

Tabla asignada a las letras a utilizar:

Letra	Código	Letra	Código	Letra	Código	Letra	Código
A	0000	E	0100	J	1000	O	1100
B	0001	F	0101	L	1001	P	1101
C	0010	G	0110	M	1010	R	1110
D	0011	I	0111	N	1011	T	1111

Figura 1. Tabla con los códigos para cada letra

Nota: La letra U fue sustituida por la letra T debido al apellido escogido.

Tabla de Verdad:

Dec	Hex	A	B	C	D	Fun
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	0
4	4	0	1	0	0	1
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	0
8	8	1	0	0	0	0
9	9	1	0	0	1	1
10	A	1	0	1	0	0
11	B	1	0	1	1	1
12	C	1	1	0	0	1
13	D	1	1	0	1	0
14	E	1	1	1	0	1
15	F	1	1	1	1	1

Figura 2. Tabla de la verdad para el apellido TORLONE

Mapa de Karnaugh:

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	0	1	1
10	0	1	1	0

Figura 3. Mapa de Karnaugh para obtener la función lógica

Función Lógica:

F = A*B*C + A*|B*D + B*|C*|D

Circuito lógico:

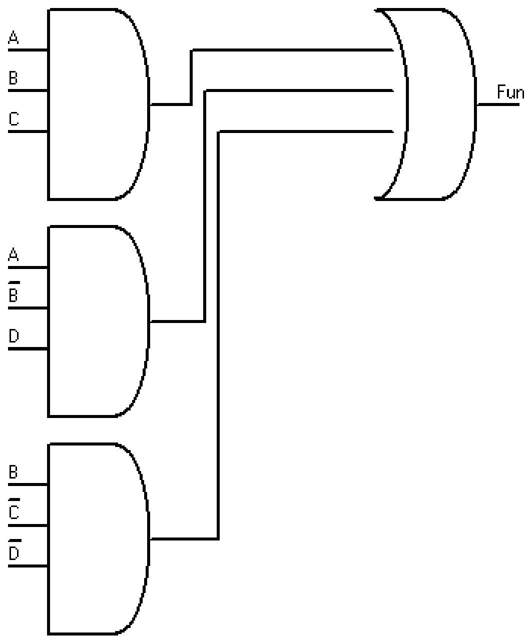


Figura 4. Esquemático del circuito lógico

Tabla de Datos de LEDs

Color	Caída de tensión (VLED) V	Intensidad máxima (ILED) mA	Intensidad media (ILED)mA
Rojo	1.6	20	5 – 10
Verde	2.4	20	5 – 10
Amarillo	2.4	20	5 – 10
Naranja	1.7	20	5 – 10

Figura 5. Tabla de datos de los LEDs para calcular la resistencia de protección

Resistencias de protección para los LEDS:

- **LED Verde:** 220 Ω
- **LED Rojo:** 290 Ω

Circuito final en PROTEUS:

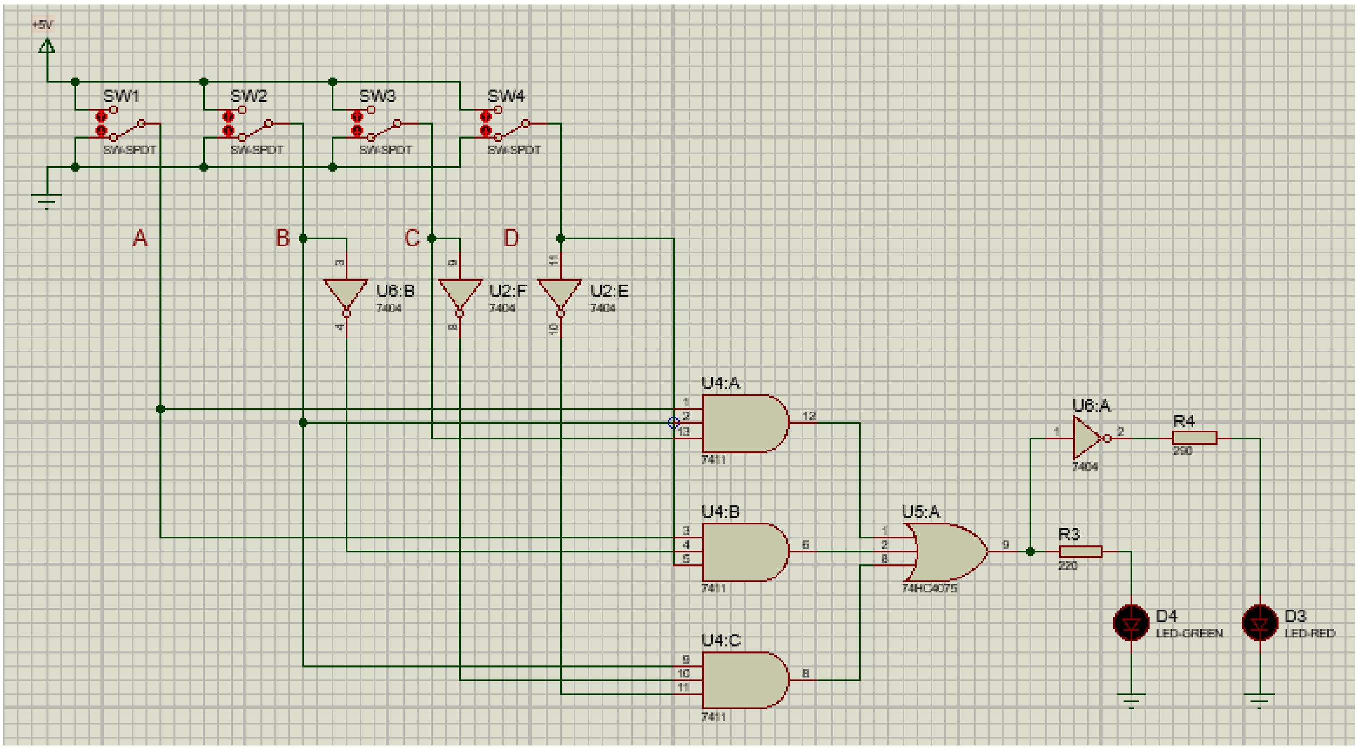


Figura 6. Circuito lógico en simulador PROTEUS

- Puertas AND de 3 entradas 7411
- Puerta OR de 3 entradas 74HC4075
- Puertas NOT 7404

- Bit más significativo: SW1
- Bit menos significativo: SW4

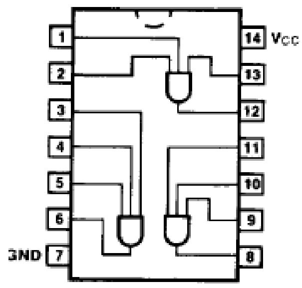
El circuito se comporta de la forma esperada. Para introducir los códigos de las letras del apellido se utilizan los switches donde los +5V representan un 1 y ground o tierra representan un 0.
El circuito detecta si las letras pertenecen o no al apellido TORLONE, de manera que, si la letra pertenece se enciende el LED Verde, y si no pertenece se enciende el LED Rojo.

ANEXOS

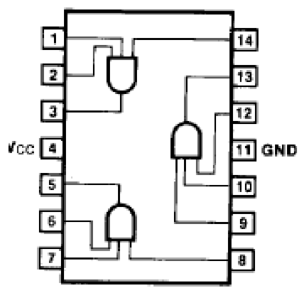
- **Anexo A**

  Esquemático de la puerta AND de 3 entradas 7411

CONNECTION DIAGRAMS
PINOUT A

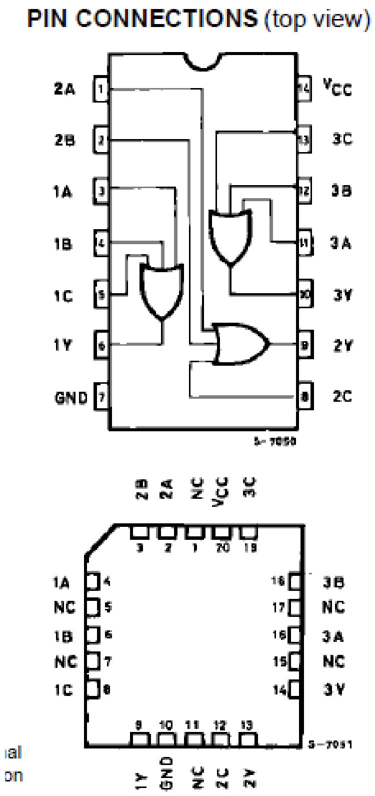


PINOUT B



- **Anexo B**

  Esquemático de la puerta OR de 3 entradas 74HC4075



- **Anexo C**

  Esquemático de la puerta NOT 7404