

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

Sección C



Práctica #1 - Simulación de un visualizador de 7 segmentos (Display)

Grupo #2 - Integrantes

<i>Estudiante</i>	<i>Carnet</i>
<i>Henderson Migdo Baten Hernandez</i>	<i>201019694</i>
<i>Selim Idair Ergon Castillo</i>	<i>201801300</i>
<i>Jemima Solmaira Chavajay Quiejú</i>	<i>210801521</i>
<i>Giovanni Saul Concoha Cax</i>	<i>202100229</i>
<i>Johan Moises Cardona Rosales</i>	<i>202201405</i>
<i>Estiben Yair Lopez Leveron</i>	<i>202204578</i>

Introducción

La lógica booleana es fundamental para el diseño y funcionamiento de circuitos. Desde simples calculadoras hasta complejos sistemas de computación, la lógica booleana proporciona el marco conceptual para el procesamiento de información. En esta práctica, explicaremos cómo aplicar los principios de la lógica booleana para controlar un display compuesto por LEDs, utilizando tanto compuertas lógicas como la recreación de estas a través de transistores, nos centraremos en el uso de compuertas lógicas estándar, como AND, OR y NOT, para diseñar circuitos que controlan el encendido de los LEDs de acuerdo con las condiciones lógicas establecidas. Utilizaremos herramientas de simulación para verificar el comportamiento de estos circuitos antes de proceder a la implementación física.

Además de la recreación de estas compuertas utilizando transistores como componentes básicos. Analizaremos el funcionamiento de los transistores en diferentes configuraciones para emular el comportamiento de las compuertas lógicas estándar. Esto nos proporcionará una comprensión más profunda de cómo se construyen los circuitos lógicos a nivel de componentes electrónicos básicos.

Este proyecto es fundamental para comprender los fundamentos de la electrónica digital y los conceptos teóricos tales como la lógica booleana, la aplicación de Minterminos y Maxterminos y los métodos de simplificación como los Mapas de Karnaugh. Además de aplicar conceptos teóricos en la práctica, nos permitirá desarrollar habilidades en el diseño y la implementación de circuitos electrónicos, así como en el análisis de su funcionamiento.

Objetivos

- **Objetivo General:**

Aplicar los conocimientos aprendidos durante el laboratorio del curso, así como también los conocimientos básicos de electrónica para recrear la simulación de un display utilizando LED's. Además, implementar dichos conocimientos en diferentes circuitos según el segmento a recrear.

- **Objetivos Principales:**

- 1. Aplicar la lógica combinacional y la correcta interpretación de los mapas de Karnaugh, aplicando MINTERMS y MAXTERMS, según represente las tablas de verdad de cada segmento a recrear.*
- 2. Conocer el funcionamiento de las diferentes familias de circuitos: Familia lógica (AND, OR, NOT) y Familia TTL (Transistor-Transistor Logic) y el comportamiento que estas demuestran en los segmentos.*
- 3. Utilizar un sistema de control para la detección de errores por medio de las funciones de paridad.*

Funciones Booleanas

Una función booleana establece una dependencia entre una variable de salida S y un conjunto de variables de entrada $A, B, C...$ manejando para su control de entradas y salidas los números 1 y 0 (1 = entrada positiva, 0 = entrada negativa). Dependiendo del número de entradas, se establece el tamaño de la tabla de verdad que registra el comportamiento de la función, determinada por la fórmula:

$$2^n = \# \text{ de combinaciones posibles}$$

Donde n , representa el número de entradas.

Implementación en un display

Para la simulación de un display, se debe tener en cuenta la segmentación de un display:

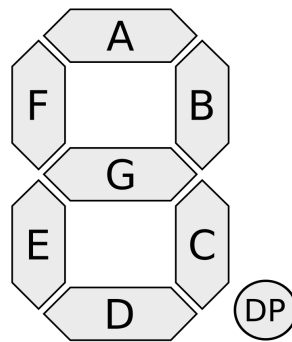


Imagen 1.1; Segmentos de un display

Basándonos en la segmentación, se desarrolló el procedimiento para la representación de la palabra “CAJETILLA”, resultando con la segmentación de cada letra de la siguiente forma:



Establecemos la tabla de verdad para cada letra de la palabra:

$n = 3$ entradas

$$2^3 = 8 \text{ combinaciones posibles}$$

Letra	A	B	C	a	b	c	d	e	f	g	db
C	0	0	0	1	0	0	1	1	1	0	0
A	0	0	1	1	1	1	1	1	0	1	0
J	0	1	0	0	1	1	1	0	0	0	0
E	0	1	1	1	0	0	1	1	1	1	0
T	1	0	0	0	0	0	0	1	1	1	0
I	1	0	1	0	0	0	0	1	1	0	0
L	1	1	0	0	0	0	1	1	1	0	0
L	1	1	1	0	0	0	1	1	1	0	1

Tabla 1.1: Tabla de verdad de la palabra "CAJETILLA"

Con las diferentes salidas obtenidas para cada segmento, se procede a utilizar los mapas de Karnaugh para su simplificación.

Ecuación Booleana

1. **Segmento a** $\rightarrow A'B'C' + A'B'C + A'BC$
2. **Segmento b** $\rightarrow A'B'C + A'BC'$
3. **Segmento c** $\rightarrow A'B'C + A'BC'$
4. **Segmento d** $\rightarrow A'B'C' + A'B'C + A'BC' + A'BC + ABC' + ABC$
5. **Segmento e** $\rightarrow A'B'C' + A'B'C + A'BC + AB'C' + AB'C + ABC' + ABC$
6. **Segmento f** $\rightarrow A'B'C' + A'BC + AB'C' + AB'C + ABC' + ABC$
7. **Segmento g** $\rightarrow A'B'C + A'BC + AB'C'$

Mapa 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. El mapa de Karnaugh consiste en una representación bidimensional de la tabla de verdad de la función a simplificar. Puesto que la tabla de verdad de una función de N variables posee 2^N filas, el mapa K correspondiente debe poseer también 2^N cuadrados. Las variables de la expresión son ordenadas en función de su peso siguiendo el **código Gray**.

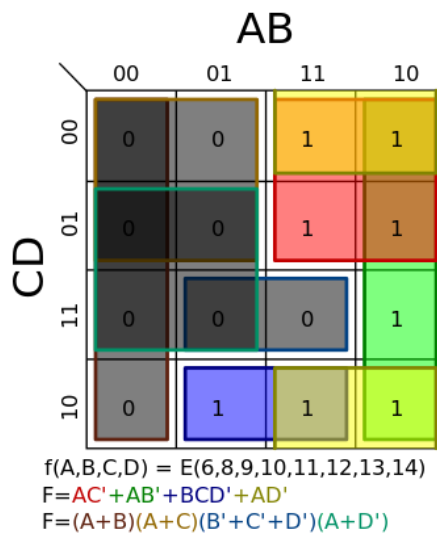


Imagen 1.2: Ejemplo Mapa de Karnaugh

Implementación de un mapa de Karnaugh

Basándonos en el tema anterior y en la tabla de verdad de la palabra “CAJETILLA”, se desarrolló el procedimiento para la simplificación de la ecuación booleana de cada segmento:

1. Segmento a:

A/BC	00	01	11	10
0	1	1	1	0
1	0	0	0	0

Tabla 1.2: Mapa de Karnaugh segmento a

$$\text{Minterms} \rightarrow S.a = (A'B') + (A'C)$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.a: (A+B)*(A+C')$$

2. Segmento B:

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>
<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>

Tabla 1.3: Mapa de Karnaugh segmento b

$$\text{Minterms} \rightarrow S.b = (A'B'C) + (A'BC')$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.b: (A')*(B+C)*(B'+C')$$

3. Segmento c:

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>
<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>

Tabla 1.4: Mapa de Karnaugh segmento c

$$\text{Minterms} \rightarrow S.c = (A'B'C) + (A'BC')$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.c: (B)*(C')*(A')$$

4. Segmento d:

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>
<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>

Tabla 1.5: Mapa de Karnaugh segmento d

$$\text{Minterms} \rightarrow S.d = (A'B') + (A'B) + (B) = A' + B$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.d: (A)*(B')$$

5. *Segmento e:*

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>0</i>
<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>

Tabla 1.6: Mapa de Karnaugh segmento e

$$\text{Minterms} \rightarrow S.e = (B') + (C) + (A)$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.e: (A+B+C')*(A+B'+C)$$

6. *Segmento f:*

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>
<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>

Tabla 1.7: Mapa de Karnaugh segmento f

$$\text{Minterms} \rightarrow S.f = (B'C') + (AB') + (BC) + (AB) = A + BC + B'C'$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.f: (A+B+C')*(A+B'+C')$$

7. *Segmento g:*

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>0</i>
<i>1</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>

Tabla 1.8: Mapa de Karnaugh segmento g

$$\text{Minterms} \rightarrow S.g = (AB'C') + (A'C)$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.g: (A+C')*(A'+B+C)$$

8. Segmento d.b:

<i>A/BC</i>	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>0</i>	0	0	0	0
<i>1</i>	0	0	1	0

Tabla 1.9: Mapa de Karnaugh segmento db

$$\text{Minterms} \rightarrow S.db = ABC$$

$$\text{Maxterms (Versión Espejo)} \rightarrow S.db: A' + B' + C'$$

MINTERMS y MAXTERMS

MINTERMS

Un minterm es un término en una expresión booleana en la que cada variable de entrada aparece exactamente una vez (complementada o no) y la función resultante es verdadera (1) para una combinación específica de valores de entrada. Cada minterm representa una sola combinación de entrada que hace que la función sea verdadera.

Por ejemplo, en una función booleana de tres variables (A , B , C), los minterms serían las combinaciones como $A'B'C'$, $AB'C'$, etc. donde cada variable (o su negación) aparece una vez.

MAXTERMS

Un maxterm es un término en una expresión booleana en la que cada variable de entrada aparece exactamente una vez (complementada o no) y la función resultante es falsa (0) para una combinación específica de valores de entrada. Cada maxterm representa una sola combinación de entrada que hace que la función sea falsa.

Por ejemplo, en una función booleana de tres variables (A , B , C), los maxterms serían las combinaciones como $A+B+C$, $A+B+C'$, etc. donde cada variable (o su negación) aparece una vez.

Los minterms y maxterms son útiles en el contexto de la simplificación de funciones lógicas mediante mapas de Karnaugh, donde se pueden agrupar términos adyacentes para encontrar formas más simples de representar la función lógica original.

Variables			Minterms	Maxterms
A	B	C		
0	0	0	$\overline{A}\overline{B}\overline{C} = m_0$	$A + B + C = M_0$
0	0	1	$\overline{A}\overline{B}C = m_1$	$A + B + \overline{C} = M_1$
0	1	0	$\overline{A}B\overline{C} = m_2$	$A + \overline{B} + C = M_2$
0	1	1	$\overline{A}BC = m_3$	$A + \overline{B} + \overline{C} = M_3$
1	0	0	$A\overline{B}\overline{C} = m_4$	$\overline{A} + B + C = M_4$
1	0	1	$A\overline{B}C = m_5$	$\overline{A} + B + \overline{C} = M_5$
1	1	0	$AB\overline{C} = m_6$	$A + \overline{B} + C = M_6$
1	1	1	$ABC = m_7$	$\overline{A} + \overline{B} + \overline{C} = M_7$

Imagen 1.3: Ejemplo de MINTERMS Y MAXTERMS

Funciones de Paridad

Una función de paridad es una función que toma una secuencia de bits como entrada y determina si el número total de unos en la secuencia es par o impar. Esto significa que la función evalúa si la suma de unos en la secuencia es divisible por dos.

Las funciones de paridad son útiles en muchos contextos, como la detección de errores en la transmisión de datos, donde pueden utilizarse para verificar si se ha encontrado un número impar de errores de bit en una secuencia transmitida.

Entrada	Paridad	Entrada	Paridad
1	1	1011	1
10	1	1100	0
11	0	1101	1
100	1	1110	1
101	0	1111	0
110	0	10000	1
111	1	10001	0
1000	1	10010	0
1001	0	10011	1
1010	0	10100	0

Imagen 1.4: Ejemplo función de paridad

Implementación de las funciones de paridad

En esta ocasión, se manejaron dos funciones (Función J y Función K). La función J será 1 cuando el número de bits '1' en las entradas es par. Por el contrario, la función K será 1 cuando el número de bits '0' en las entradas es par.

- Para Función J:

<i>A</i>	<i>B</i>	<i>C</i>	<i>Paridad</i>	<i>Ec. Booleana</i>
0	0	0	0	— — — —
0	0	1	0	— — — —
0	1	0	0	— — — —
0	1	1	1	$A'BC$
1	0	0	0	— — — —
1	0	1	1	$AB'C$
1	1	0	1	ABC'
1	1	1	0	— — — —

Tabla 1.10: Tabla de verdad de la función J

Salida J: $A'BC + AB'C + ABC'$

- Para Función K:

<i>A</i>	<i>B</i>	<i>C</i>	<i>Paridad</i>	<i>Ec. Booleana</i>
0	0	0	0	— — — —
0	0	1	1	$A'B'C$
0	1	0	1	$A'BC'$
0	1	1	0	— — — —
1	0	0	1	$AB'C'$
1	0	1	0	— — — —
1	1	0	0	— — — —
1	1	1	0	— — — —

Tabla 1.11: Tabla de verdad de la función K

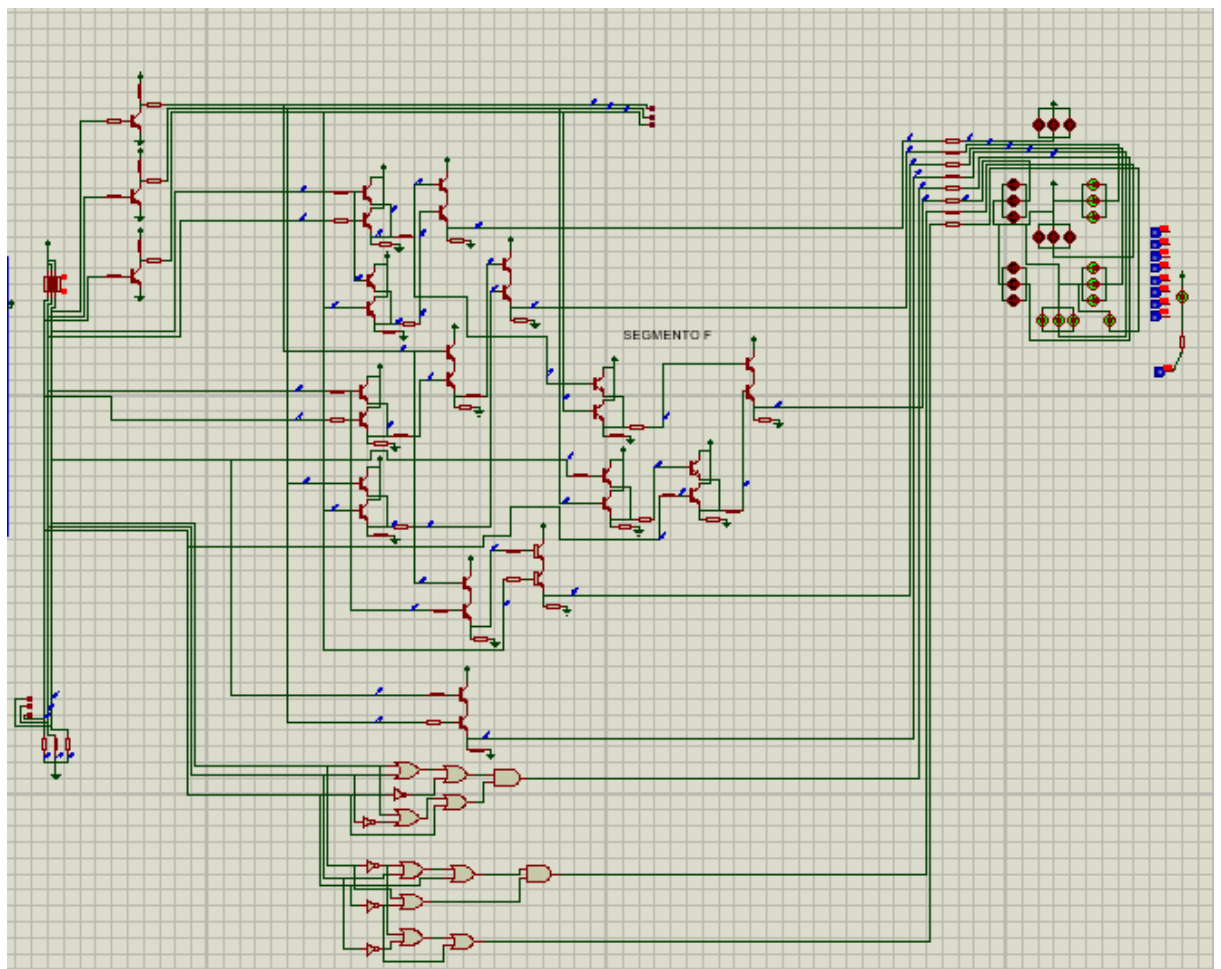
Salida K: $A'B'C + A'BC' + ABC'$

Diagramas del diseño del circuito

Tabla de verdad espejo:

	Letra	no.	A	B	C	a	b	c	d	e	f	g	.
1	c	0	0	0	0	0	0	0	0	1	1	1	1
2	a	1	0	0	1	0	1	0	0	0	0	0	1
3	j	2	0	1	0	1	1	1	0	0	0	1	1
4	e	3	0	1	1	0	0	0	0	1	1	0	1
5	t	4	1	0	0	1	0	0	1	1	1	0	1
6	i	5	1	0	1	1	0	0	1	1	1	1	1
7	l	6	1	1	0	1	0	0	0	1	1	1	1
8	l	7	1	1	1	1	0	0	0	1	1	1	0
9	a	8											

Circuito en Proteus espejo:

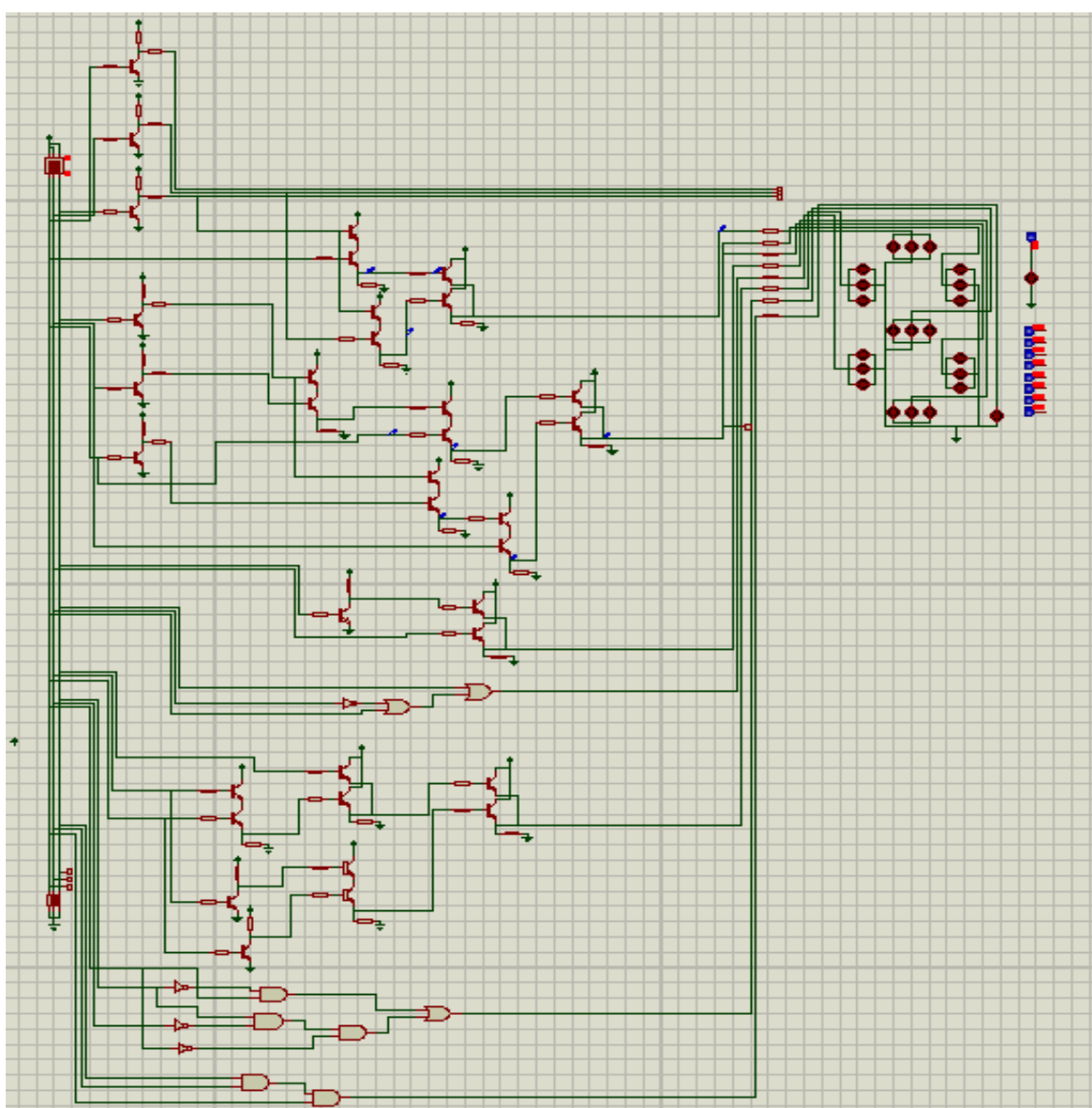


Fuente: Elaboración Propia

Tabla de verdad circuito normal:

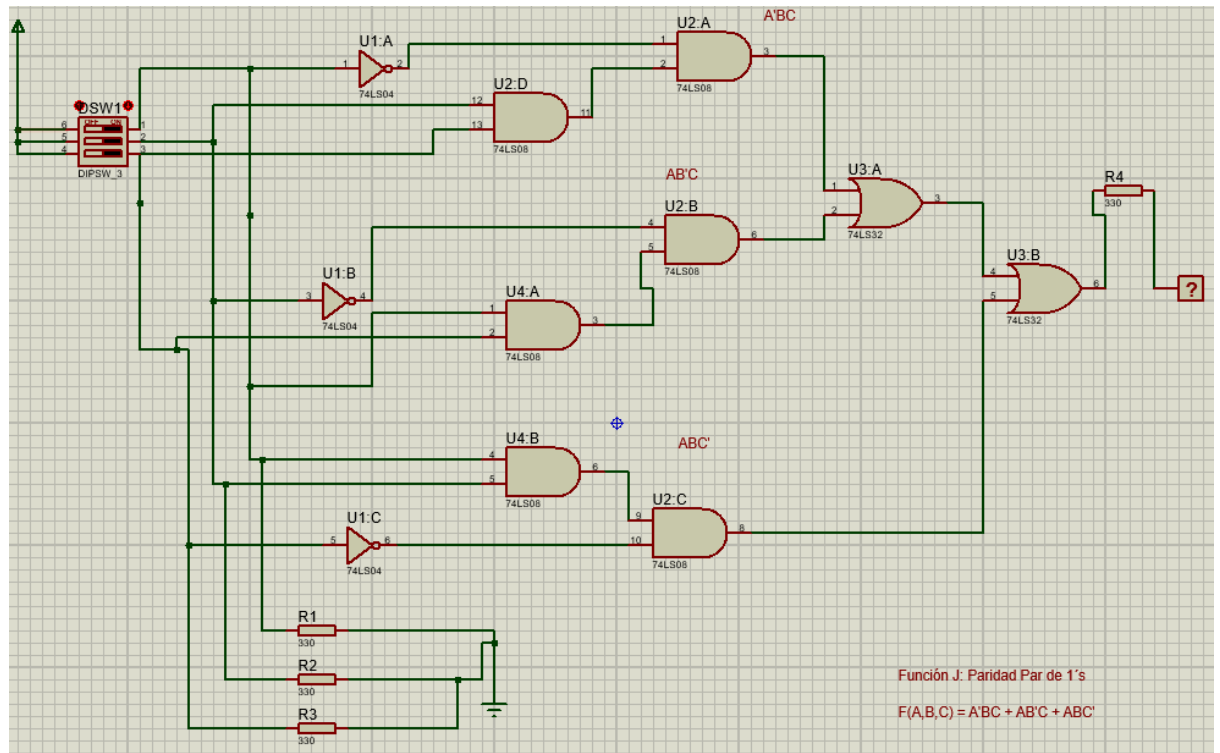
Letra	A	B	C	a	b	c	d	e	f	g	db
C	0	0	0	1	0	0	1	1	1	0	0
A	0	0	1	1	1	1	1	1	0	1	0
J	0	1	0	0	1	1	1	0	0	0	0
E	0	1	1	1	0	0	1	1	1	1	0
T	1	0	0	0	0	0	0	1	1	1	0
I	1	0	1	0	0	0	0	1	1	0	0
L	1	1	0	0	0	0	1	1	1	0	0
L	1	1	1	0	0	0	1	1	1	0	1

Circuito en Proteus Normal:



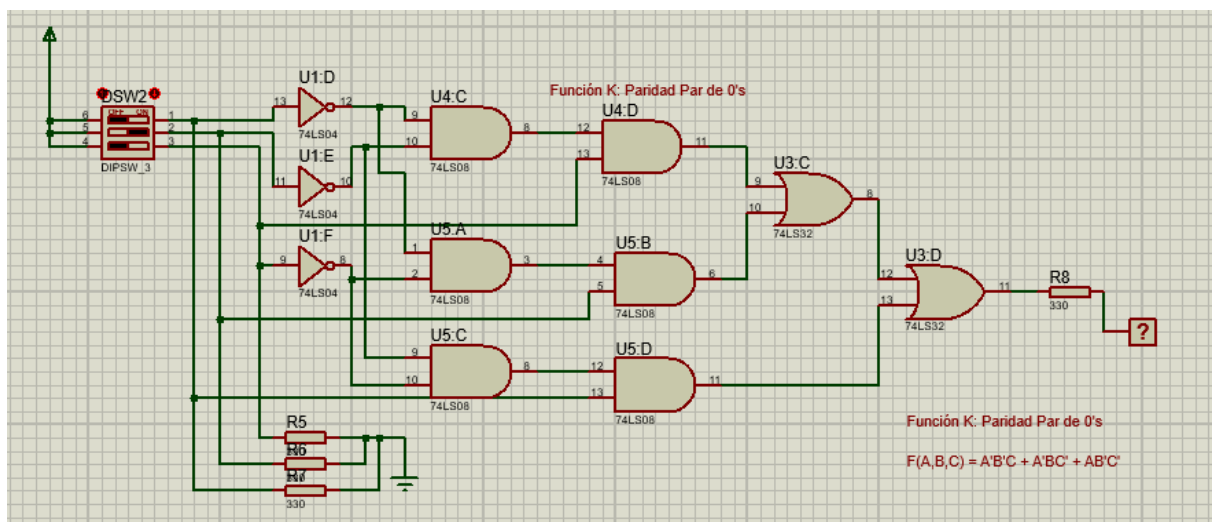
Fuente: Elaboración Propia

Circuito en Proteus Paridad J:



Fuente: Elaboración Propia

Circuito en Proteus Paridad K:



Fuente: Elaboración Propia

Listado de componentes y presupuesto

Componente: Transistor TTL			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>Transistor NPN 2n2222a</i>	<i>40</i>	<i>0.85</i>	<i>34</i>
Subtotal:			<i>34</i>
Componente: Compuerta Lógica			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>Compuerta AND 74ls08</i>	<i>10</i>	<i>6</i>	<i>60</i>
<i>Compuerta NOT 74ls04</i>	<i>10</i>	<i>7</i>	<i>70</i>
<i>Compuerta OR 74ls32</i>	<i>8</i>	<i>7</i>	<i>56</i>
Subtotal:			<i>186</i>
Componente: Protoboard			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>PB-MB102 transparente</i>	<i>5</i>	<i>52</i>	<i>260</i>
<i>Mitad de protoboard</i>	<i>4</i>	<i>32</i>	<i>128</i>
Subtotal:			<i>388</i>
Componente: Resistencia			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>Resistencia 3k Ω</i>	<i>40</i>	<i>0.60</i>	<i>24</i>
<i>Resistencia 5k Ω</i>	<i>40</i>	<i>0.60</i>	<i>24</i>
<i>Resistencia 10k Ω</i>	<i>90</i>	<i>0.60</i>	<i>54</i>

<i>Resistencia 330 Ω</i>	<i>10</i>	<i>0.75</i>	<i>7.50</i>
Subtotal:			<i>109.50</i>
Componente: Cables			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>DUPONT</i>	<i>10</i>	<i>1</i>	<i>10</i>
<i>AL-22N negro</i>	<i>11</i>	<i>2</i>	<i>22</i>
<i>AL-22RE rojo</i>	<i>11</i>	<i>2</i>	<i>22</i>
Subtotal:			<i>54</i>
Componente: DIP-SWITCH			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>SW-DIP3</i>	<i>4</i>	<i>3</i>	<i>12</i>
Subtotal:			<i>12</i>
Otros Componentes			
Descripción	Unidades	Precio Unidad (Q.)	Total
<i>Pinzas</i>	<i>1</i>	<i>— — —</i>	<i>0</i>
<i>Alicate</i>	<i>3</i>	<i>— — —</i>	<i>0</i>
<i>Pelador de cable M-398</i>	<i>1</i>	<i>39</i>	<i>39</i>
<i>Fuente de poder 5V</i>	<i>3</i>	<i>— — —</i>	<i>0</i>
<i>Multímetro</i>	<i>1</i>	<i>— — —</i>	<i>0</i>
<i>LED</i>	<i>30</i>	<i>1</i>	<i>30</i>
Subtotal:			<i>69</i>

Presupuesto Final: Q 852.50

Aporte por estudiante: Q 143.00

Conclusiones

En esta práctica, hemos explorado la aplicación de la lógica booleana en el control de un display de LEDs, utilizando compuertas lógicas estándar y la recreación de estas compuertas mediante transistores.

Demostramos que las compuertas lógicas estándar, como AND, OR y NOT, son herramientas importantes para diseñar circuitos que controlan el encendido de los LEDs de acuerdo con condiciones lógicas específicas.

Comprobamos la validez de nuestros diseños utilizando herramientas de simulación, lo que nos permitió detectar posibles errores y optimizar el rendimiento de nuestros circuitos antes de la implementación física.

Mediante la recreación de compuertas lógicas con transistores, pudimos profundizar nuestra comprensión de cómo funcionan estos componentes básicos y cómo se pueden combinar para realizar operaciones lógicas complejas.

La lógica booleana proporciona un marco sólido para el diseño de sistemas electrónicos, permitiendo la creación de circuitos que respondan de manera predecible a diferentes condiciones de entrada.

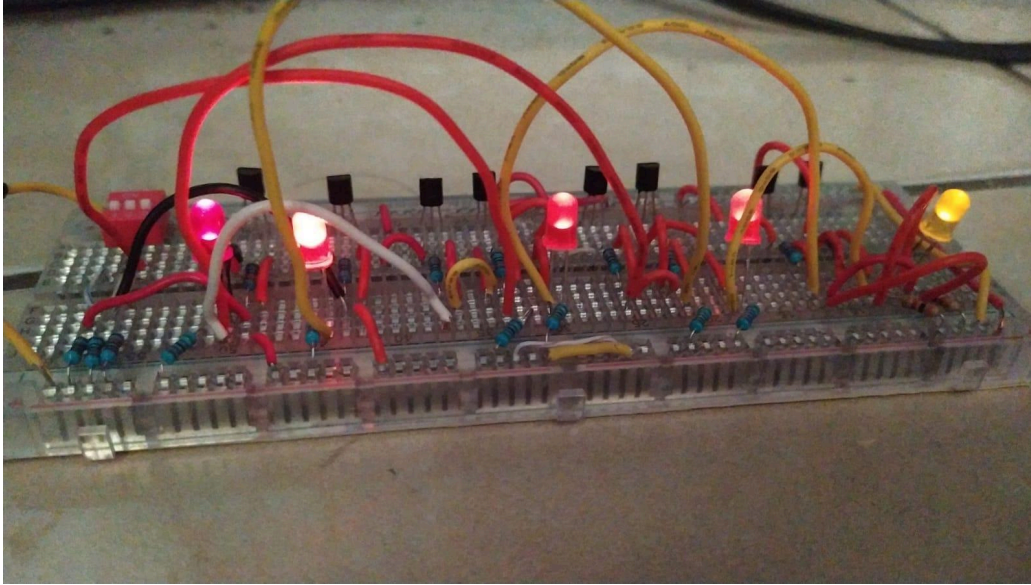
Los maxtérminos y mintérminos son representaciones alternativas de expresiones lógicas, que ofrecen una forma conveniente de describir conjuntos de condiciones de entrada que activan o desactivan una función lógica. La representación en maxtérminos es especialmente útil para la simplificación de funciones lógicas complejas mediante el uso de la ley de De Morgan y la identificación de condiciones en las que la función es cero, mientras que la representación en mintérminos se centra en las condiciones en las que la función es uno.

Los mapas de Karnaugh han demostrado ser una herramienta invaluable para simplificar expresiones lógicas de manera visual y sistemática. Estos diagramas nos permiten identificar patrones y agrupaciones de términos lógicos que pueden ser combinados para reducir la complejidad de una función booleana. Al utilizar técnicas de agrupamiento adecuadas, hemos podido simplificar nuestras expresiones lógicas de manera significativa, lo que resulta en circuitos más eficientes y fáciles de entender.

Anexos

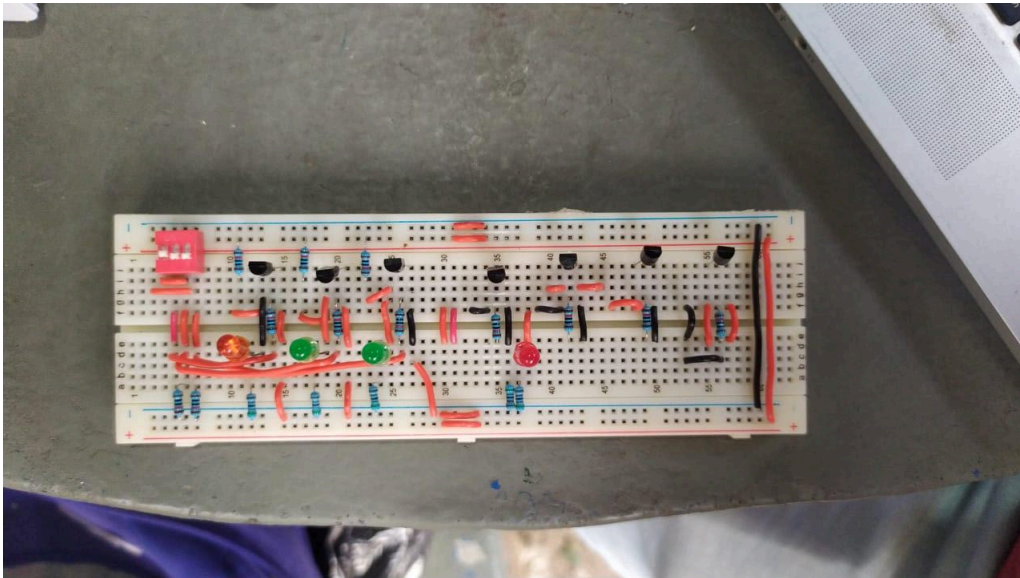
1. Fotografías de los circuitos físicos

Segmento A



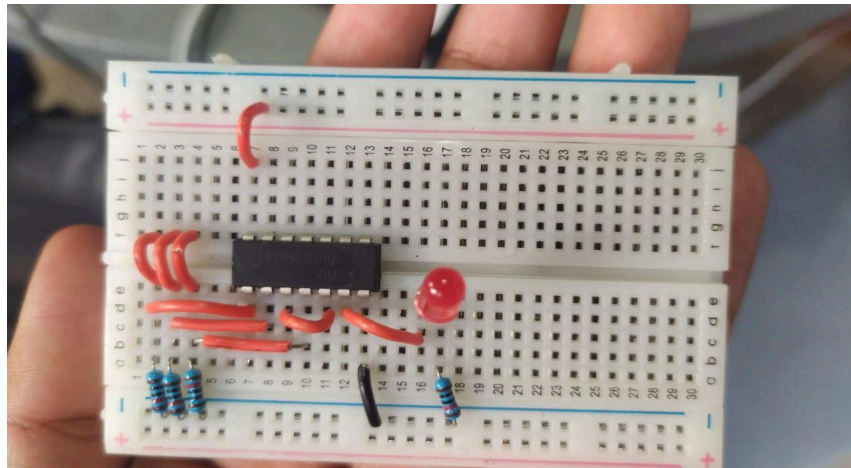
Fuente: Elaboración Propia

Segmento B y C



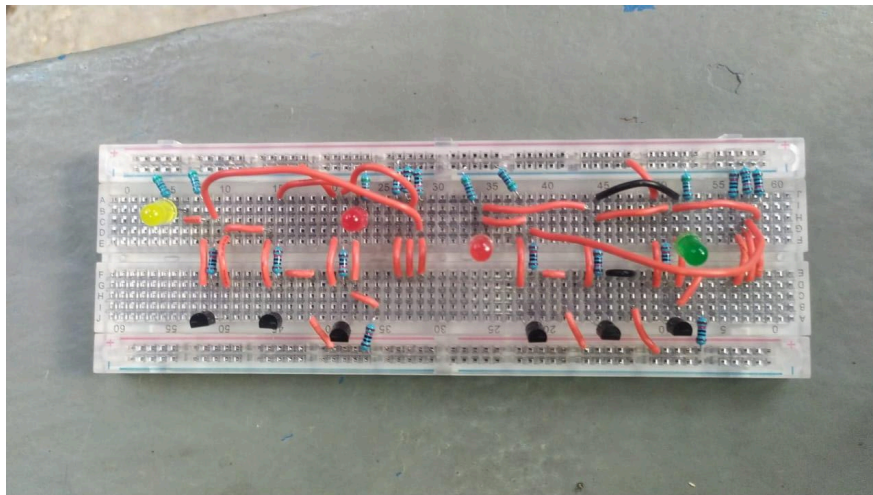
Fuente: Elaboración propia

Segmento pt Normal:



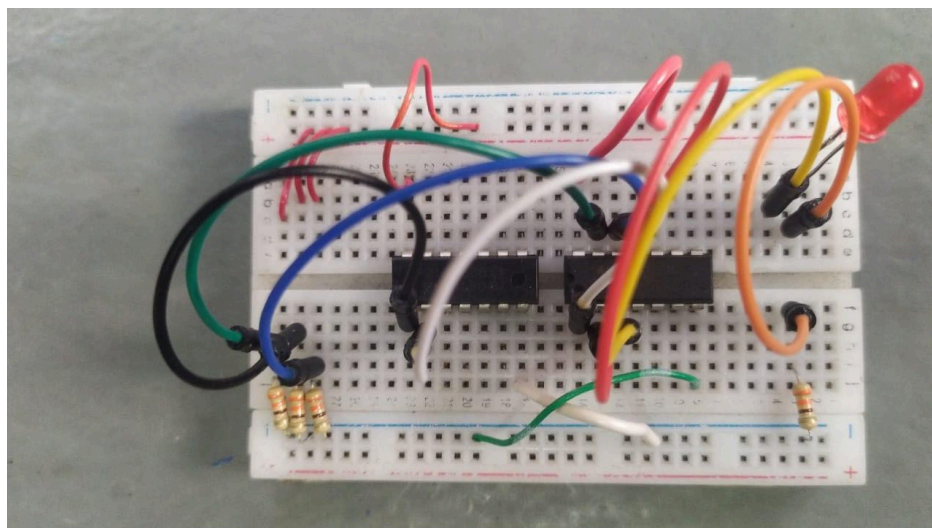
Fuente: Elaboración propia

Segmento D normal y espejo



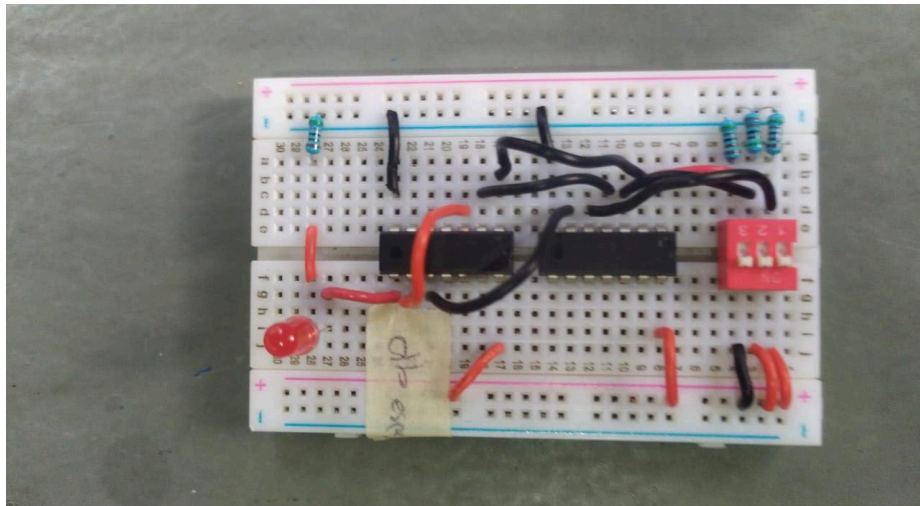
Fuente: Elaboración normal

Segmento E normal:



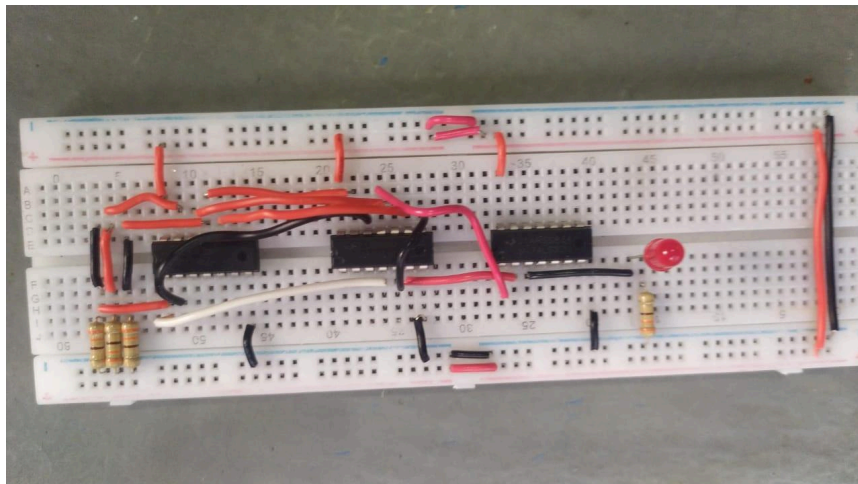
Fuente: Elaboración propia.

Segmento Pt



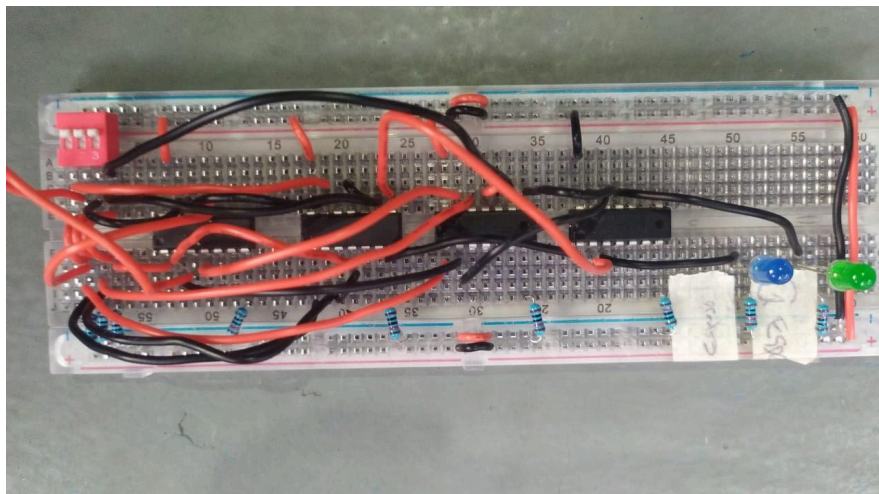
Fuente: Elaboración Propia

Segmento G normal



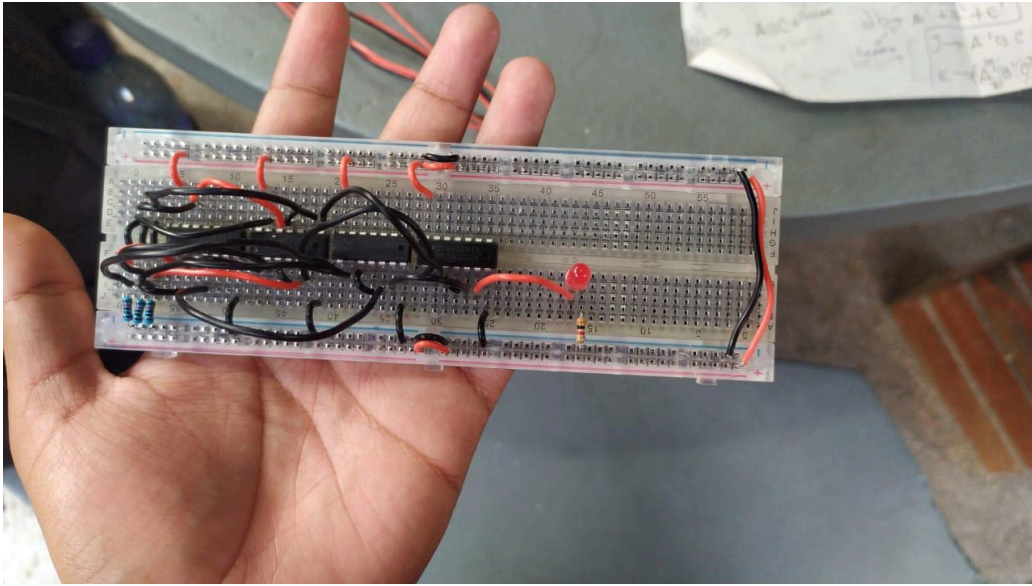
Fuente: Elaboración Propia

Segmento G y E espejo:



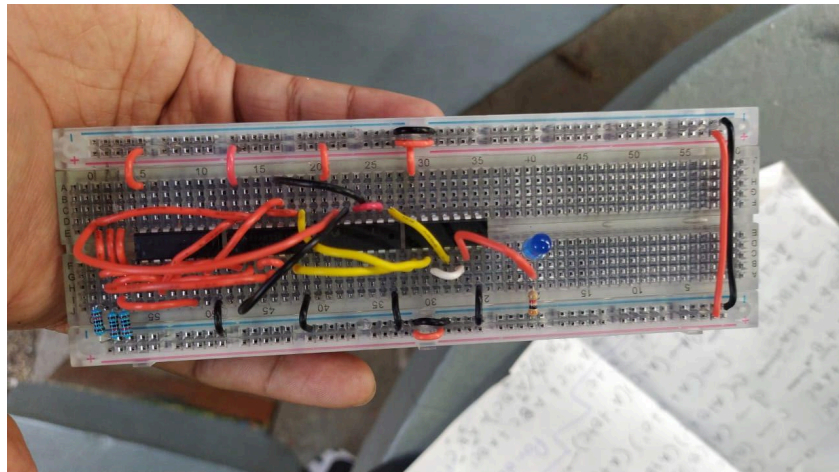
Fuente: Elaboración propia

Paridad J



Fuente: Elaboración Propia

Paridad K



Fuente: Elaboración propia

2. Enlace al video grupal de los circuitos físicos

<https://drive.google.com/drive/folders/1sczVGfF86nK4kvcyJuNMpRUZ5btmmS0t?usp=sharing>

Bibliografía

Minterminos y maxterminos - Cienciyt. (2020). Retrieved February 17, 2024, from

Cienciyt website:

<https://cienciyt.com/electronica/sistemas-digitales/minterminos-y-maxterminos/#:~:text=Los%20minterminos%20de%20una%20funci%C3%B3n,toma%20el%20valor%20de%200.>

Mini-maxi-términos y Mapas de Karnaugh. (2017, October 9). Retrieved February 17,

2024, from Laboratorios digitales website:

<https://labsdigitalpaulayfrancisco.wordpress.com/2017/10/09/mini-maxi-terminos-y-mapas-de-karnaugh/>

Dana, D., & Urribarri, K. (n.d.). *Arquitectura de Computadoras para Ingeniería*.

Retrieved from

https://cs.uns.edu.ar/~pmd/ac_ing/downloads/Slides/ACI-Clase-1.pdf

Administrador. (2018, January 24). Display 7 Segmentos ánodo y cátodo común.

Retrieved February 17, 2024, from HeTPro-Tutoriales website:

<https://hetpro-store.com/TUTORIALES/display-7-segmentos-anodo-catodo-comun/>

paguayo. (2022, August). El Display de 7 segmentos - MCI Capacitación. Retrieved

February 17, 2024, from MCI Capacitación website:

<https://cursos.mcielectronics.cl/2022/08/01/el-display-de-7-segmentos/>

raniellysvillarroel. (2019). Guia de Circuitos Basicos Para Proteus. Retrieved February

17, 2024, from Scribd website:

<https://www.scribd.com/doc/56647240/Guia-de-Circuitos-Basicos-Para-Proteus>