



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Departamento de
Ingeniería en Sistemas Computacionales



Unidad de aprendizaje:
Fundamentos de diseño digital

PRÁCTICA 2

Minimización algebraica

Equipo 2

Grupo: 3CV2

Integrantes:

González Cárdenas Ángel Aquilez
Hernández Reyes Diego Alberto

Profesor: Barrón Vera Josué Emanuel

Fecha de realización: 19 de septiembre de 2023
Fecha de entrega: 25 de septiembre de 2023

Índice

1. Desarrollo	2
1.1. Comparador de dos bits	2
1.2. Generador de código Grey de cuatro bits	5
2. Observaciones y conclusiones	12

En el presente reporte de laboratorio correspondiente a la segunda práctica, se aborda el diseño y la verificación de un comparador con magnitud de dos bits, así como el diseño y simplificación de un generador de código Gray para cuatro bits.

Objetivo

Objetivo: Al terminar la sesión, los integrantes del equipo contarán con la habilidad de diseñar circuitos combinatorios a partir de un enunciado.

Los alumnos utilizarán los siguientes materiales y equipo:

Equipo

- 10 LEDS de colores
- Dip switch
- Fuente de Alimentación
- Manual de especificaciones *FAST and LS TTL* de MOTOROLA®

Material

- *Protoboard*
- 1 C. I. 74LS00
- 1 C. I. 74LS02
- 1 C. I. 74LS04
- 1 C. I. 74LS08
- 1 C. I. 74LS32
- 1 C. I. 74LS86

1. Desarrollo

1.1. Comparador de dos bits

El comparador de dos bits se caracteriza por poseer dos entradas (A, B) y tres salidas correspondientes a las operaciones $A < B$, $A = B$ y $A > B$.

La Figura 1 ilustra la tabla de verdad que indica los valores para los cuales son verdaderas las operaciones.

#	A	B	$F_1 = A < B$	$F_2 = A = B$	$F_3 = A > B$
0	0	0	0	1	0
1	0	1	1	0	0
2	1	0	0	0	1
3	1	1	0	1	0

$\bar{A} \cdot B$ $AB + \bar{A}\bar{B}$ $A \cdot \bar{B}$

Figura 1: Tabla de verdad del comparador de dos bits

Después, con los valores para los cuales las salidas producen un valor alto, o verdadero, se determina qué operación produce el resultado deseado realizando la suma de productos correspondiente.

Para F_1 (que se produce por la operación $A < B$), es verdadera solo cuando A es un valor bajo y B un valor alto. De modo que la expresión que nos entregará el resultado deseado será:

$$\overline{A}B$$

De manera análoga para F_2 tenemos:

$$AB + \overline{A}(\overline{B})$$

Y para F_3

$$A\overline{B}$$

Una vez determinadas las expresiones que otorgan los resultados correctos, se obtuvieron los siguientes circuitos lógicos:

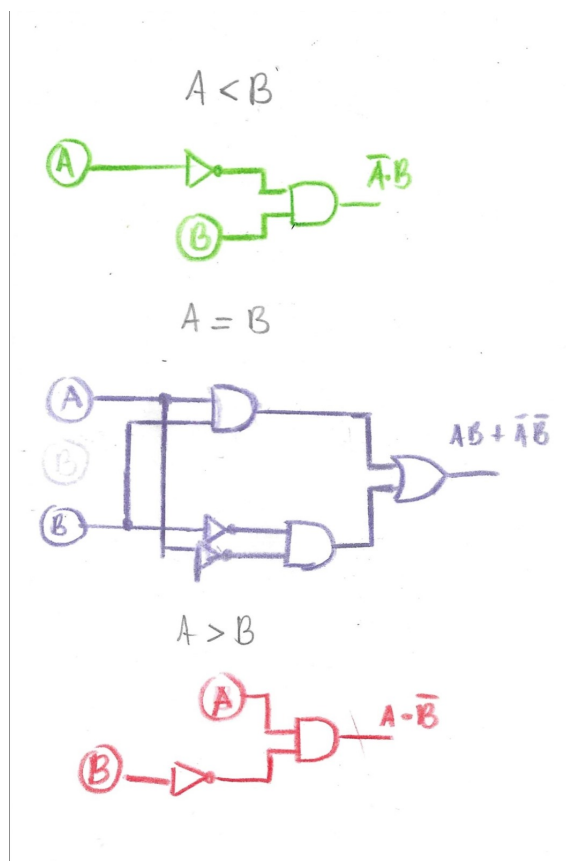


Figura 2: Circuitos lógicos del comparador de dos bits

Por consiguiente, el circuito en funcionamiento se muestra en las Figuras 3 y 4:

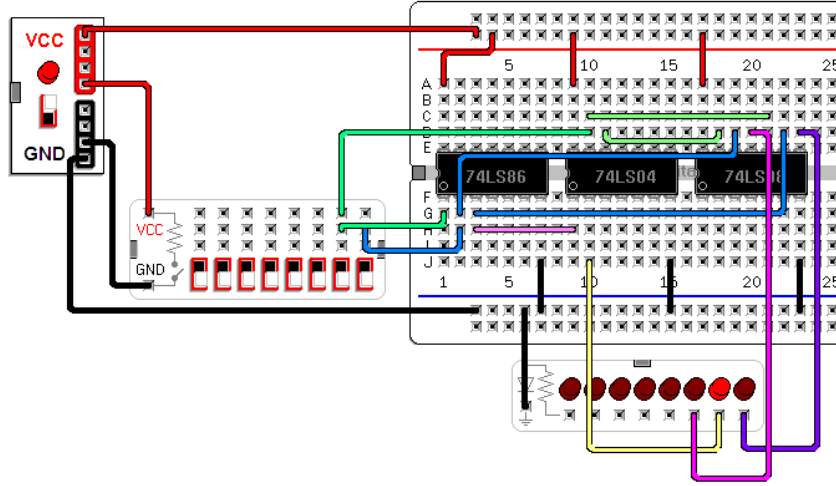


Figura 3: Circuito comparador $A = 0, B = 0, F_2 = 1$

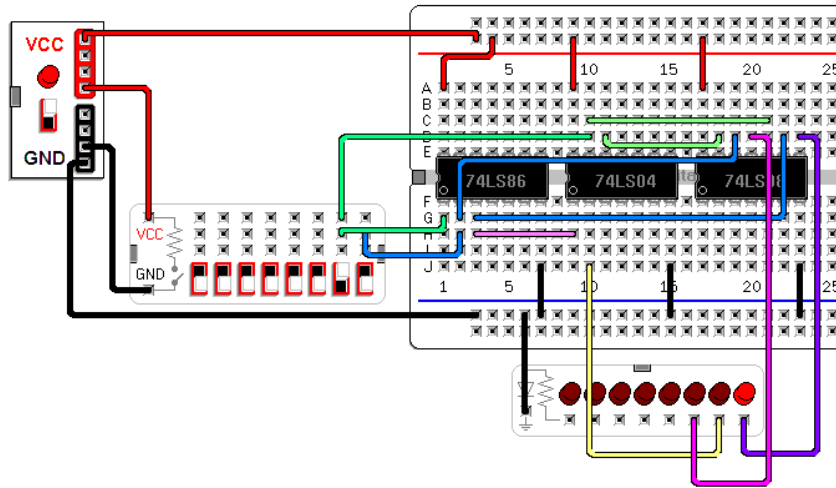


Figura 4: Circuito comparador $A = 1, B = 0, F_3 = 1$

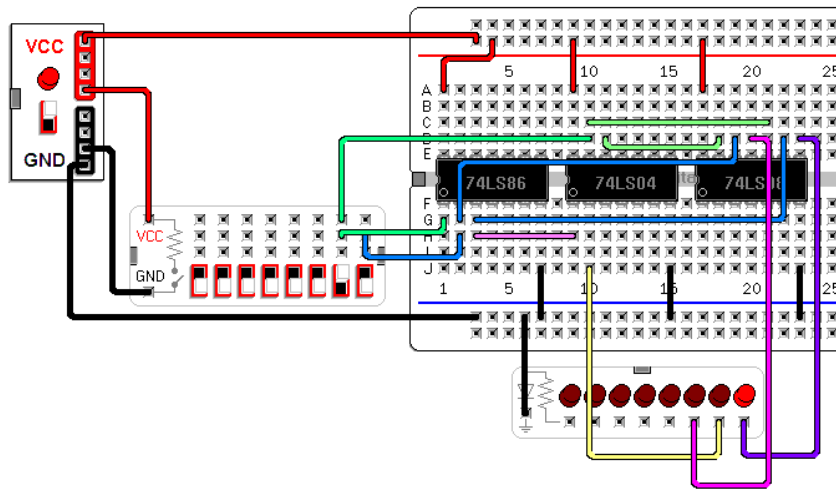


Figura 5: Circuito comparador $A = 1, B = 1, F_2 = 1$

1.2. Generador de código Grey de cuatro bits

La principal característica del código Grey es el cambio en solo un bit entre la salida anterior y la siguiente, que se puede apreciar en la tabla de verdad correspondiente al código Grey para cuatro entradas de la Figura 6:

#	A	B	C	D	F1	F2	F3	F4
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Figura 6: Tabla de verdad del código Grey

De manera similar al ejercicio anterior, se obtiene la suma de productos para los cuales la salida es positiva, esto es: la suma de minterminos. Como las salidas son cuatro, se obtuvieron cuatro enunciados distintos, lo que significa cuatro circuitos independientes entre sí.

Por simple inspección, se aprecia que la salida para F_1 será equivalente a los valores de entrada para A , lo cual se comprueba en la Figura 7 que muestra la simplificación realizada:

$$\begin{aligned}
 F_1 &= A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD \\
 &= A\bar{B}(\bar{C}\bar{D} + \bar{C}D + C\bar{D} + CD) + AB(\bar{C}\bar{D} + \bar{C}D + C\bar{D} + CD) \\
 &= (A\bar{B} + AB)(\bar{C}\bar{D} + \bar{C}D + C\bar{D} + CD) \\
 &= (A(\bar{B} + B))(\bar{C}(\bar{D} + D) + C(\bar{D} + D)) \\
 &= (A)(\bar{C} + C) \\
 F_1 &= A
 \end{aligned}$$

Figura 7: Simplificación para F_1

La principal motivación para realizar la simplificación se encuentra en la reducción de componentes a utilizar cuando se construyen los circuitos, como se puede ver en la Figura 8 al reducirse drásticamente el número de componentes:

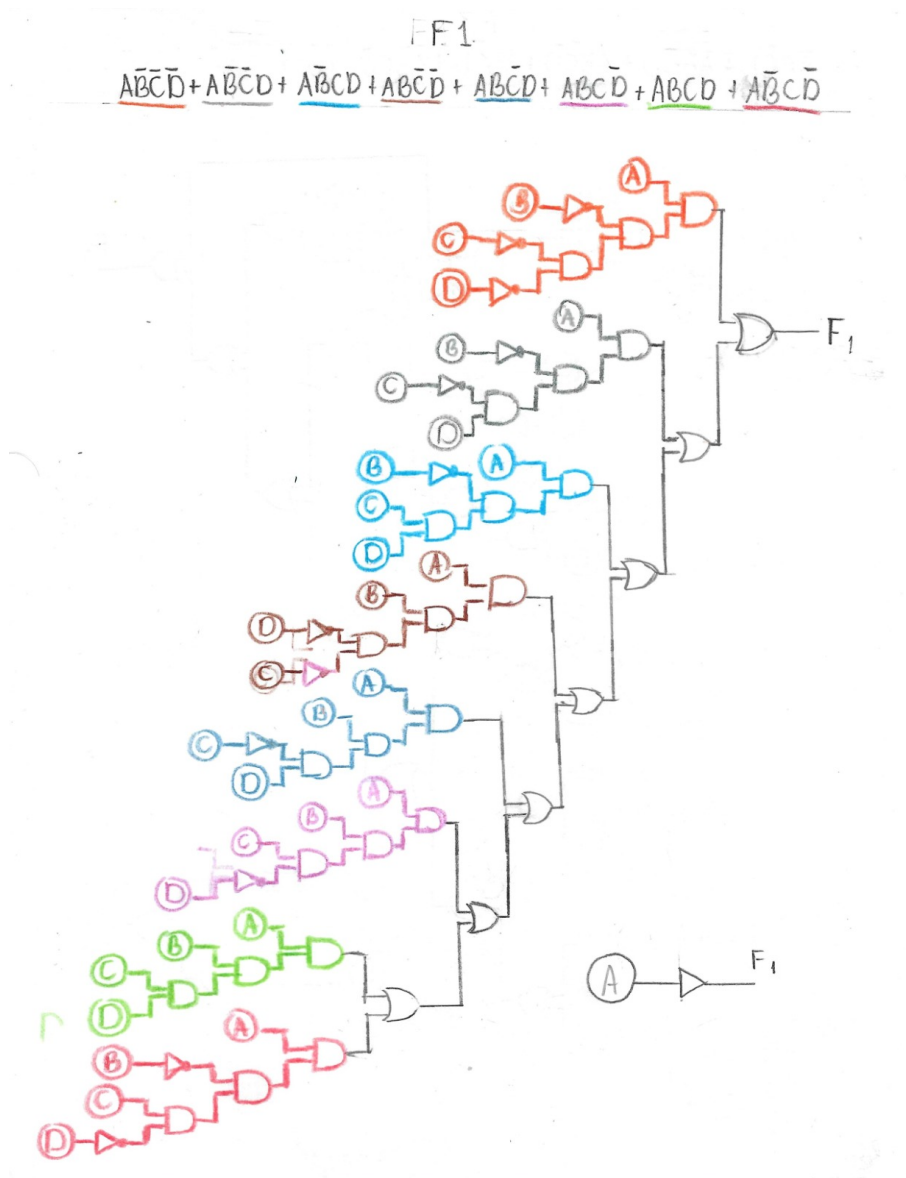


Figura 8: Simplificación para el circuito de F_1

De manera análoga, se determinan la suma de minterminos para cada salida y se simplifican a su mínima expresión equivalente.

$$\begin{aligned}
 F_2 &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD \\
 &= \bar{A}\bar{B}[\bar{C}\bar{D} + \bar{C}D + C\bar{D} + CD] + A\bar{B}[\bar{C}\bar{D} + \bar{C}D + C\bar{D} + CD] \\
 &= \bar{A}\bar{B}(\bar{C}(\bar{D}+D) + C(\bar{D}+D)) + A\bar{B}(\bar{C}(\bar{D}+D) + C(\bar{D}+D)) \\
 &= \bar{A}\bar{B}(\bar{C}+C) + A\bar{B}(\bar{C}+C) \\
 &= \bar{A}\bar{B} + A\bar{B} \\
 &= A \oplus B
 \end{aligned}$$

Figura 9: Simplificación para F_2

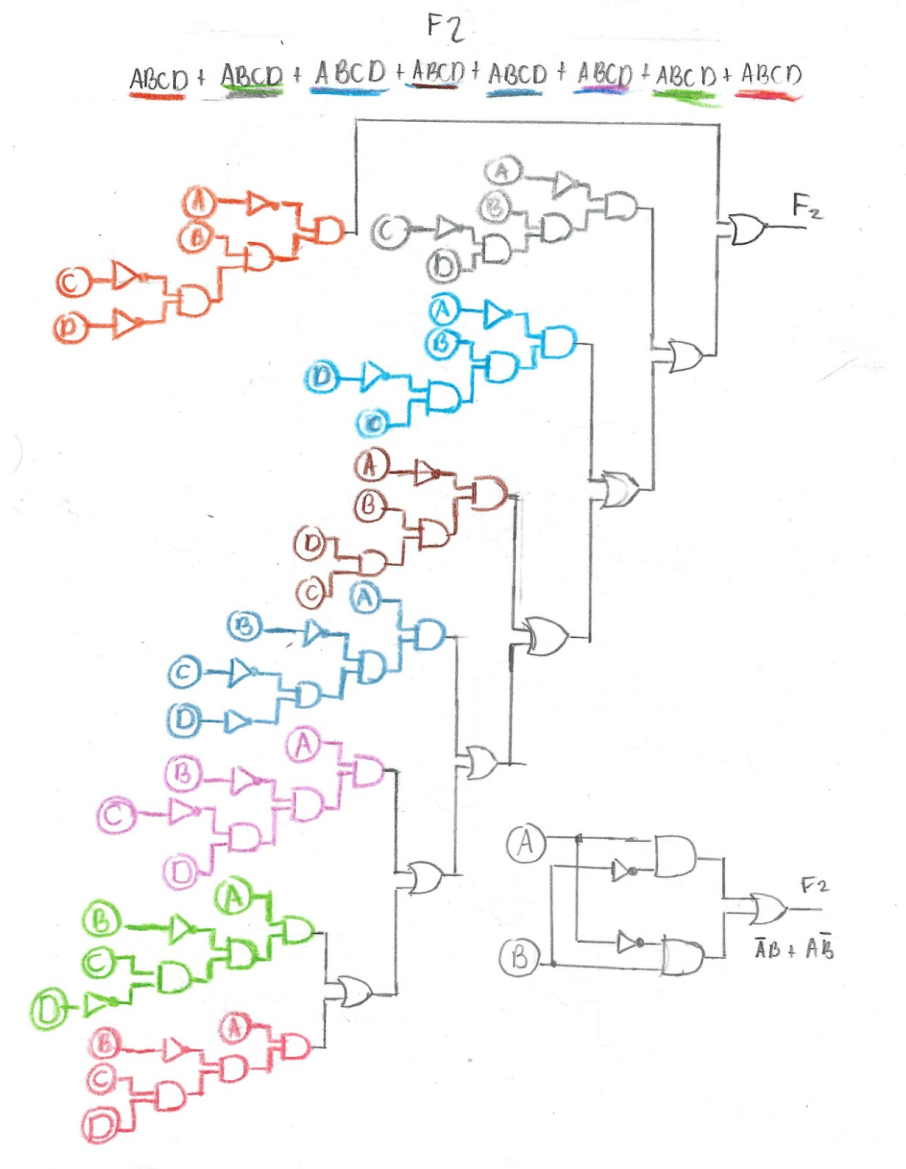


Figura 10: Simplificación para el circuito de F_2

$$\begin{aligned}
 & \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD \\
 & + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD \\
 & = BC(\bar{A}\bar{D} + \bar{A}D + A\bar{D} + AD) + \bar{B}C(\bar{A}\bar{D} + \bar{A}D + A\bar{D} + AD) \\
 & = BC(\bar{A}(\bar{D} + D) + A(\bar{D} + D)) + \bar{B}C(\bar{A}(\bar{D} + D) + A(\bar{D} + D)) \\
 & = BC(\bar{A} + A) + \bar{B}C(\bar{A} + A) = BC + \bar{B}C = F_3
 \end{aligned}$$

Figura 11: Simplificación para F_3

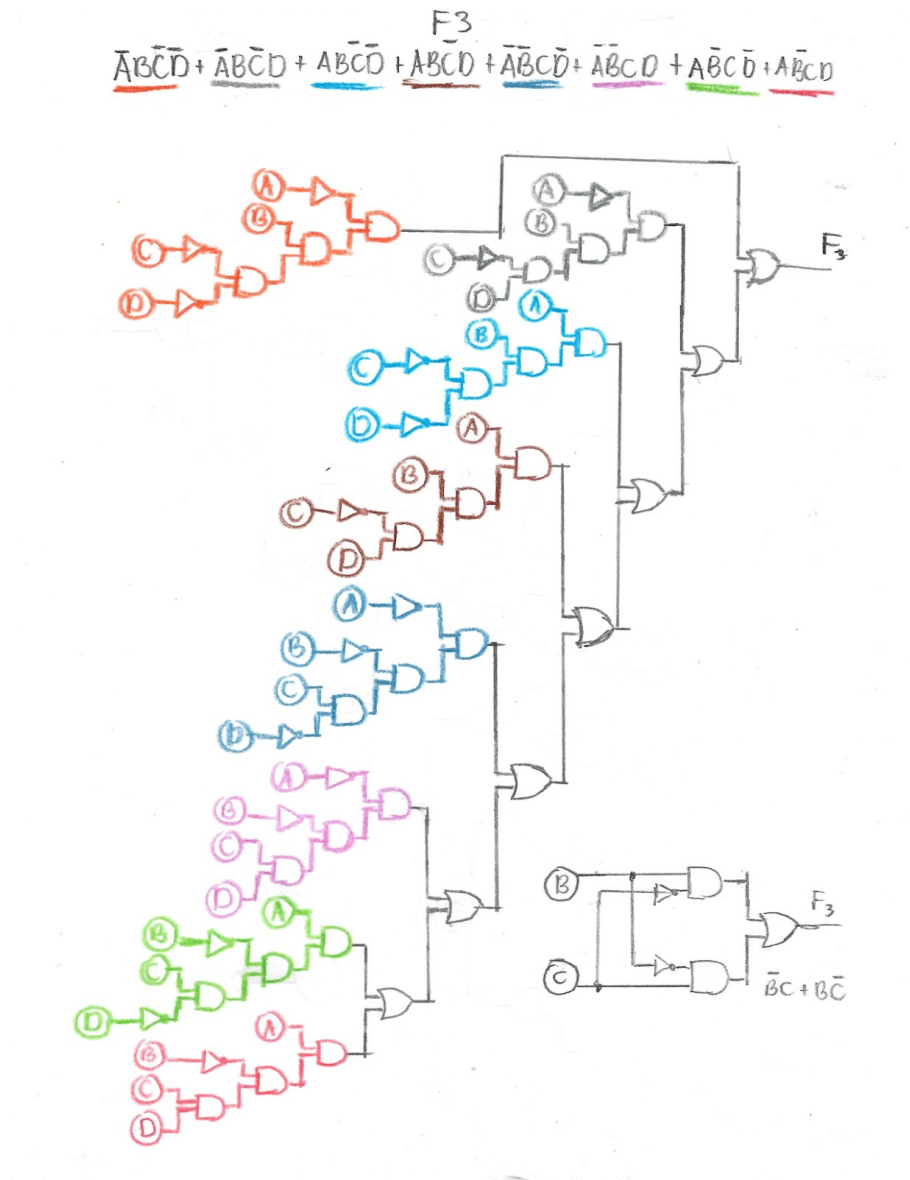


Figura 12: Simplificación para el circuito de F_3

$$\begin{aligned}
 & \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + A\bar{B}\bar{C}D + AB\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B C\bar{D} + A\bar{B}C\bar{D} + ABC\bar{D} \\
 & \bar{C}D + (\bar{A}\bar{B} + A\bar{B} + A\bar{B} + AB) + C\bar{D}(\bar{A}\bar{B} + \bar{A}B + AB + AB) \\
 & \bar{C}D(\bar{A}(\bar{B}+B) + A(\bar{B}+B)) + C\bar{D}(\bar{A}(\bar{B}+B) + A(\bar{B}+B)) \\
 & \bar{C}D(\bar{A}+A) + C\bar{D}(\bar{A}+A) \\
 & \bar{C}D + C\bar{D} = F_4
 \end{aligned}$$

Figura 13: Simplificación para F_4

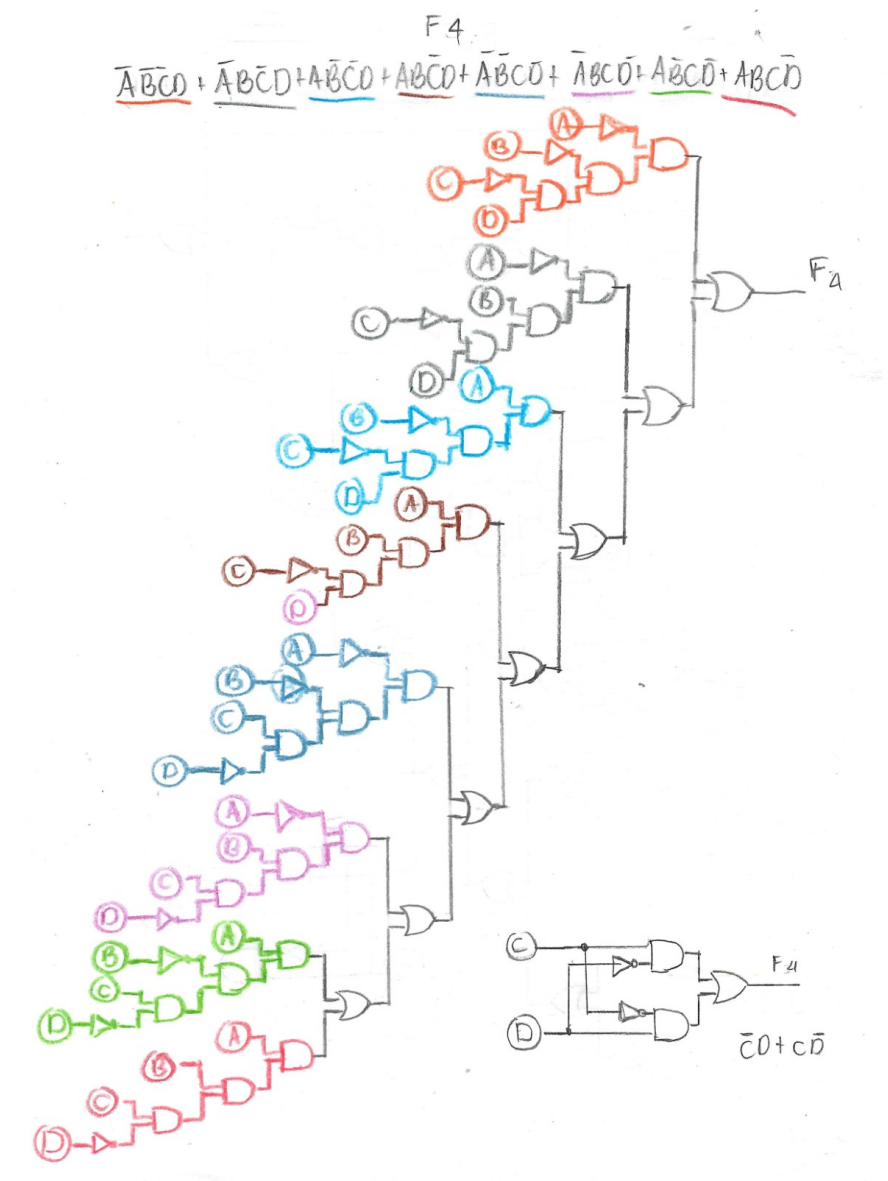


Figura 14: Simplificación para el circuito de F_4

Una vez obtenidos los equivalentes simplificados se procedió a construir el circuito, que se aprecia en la Figura 15:

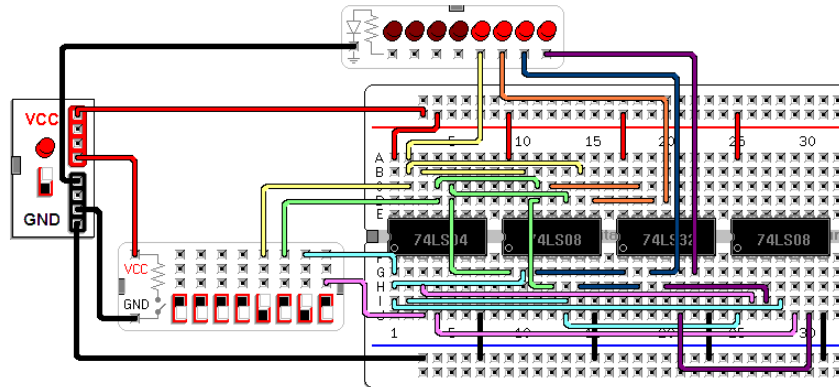


Figura 15: $A = 1, B = 0, C = 1, D = 0$, Equivalente Grey: 1111

Este primer circuito fue construido a partir de las expresiones simplificadas sin hacer uso de la compuerta *XOR*.

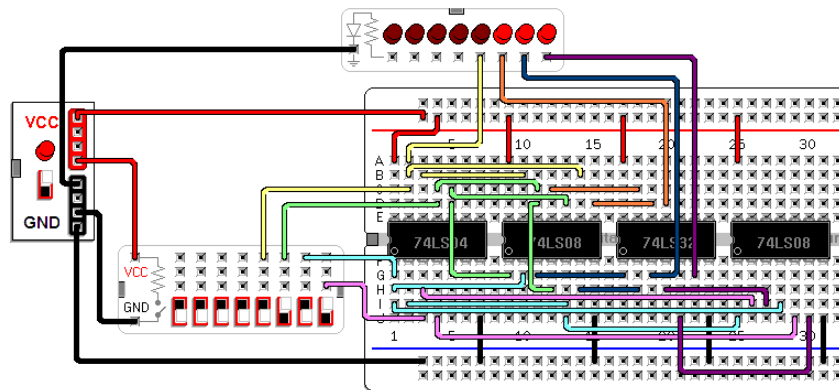


Figura 16: Entrada: 0101, Grey: 1111

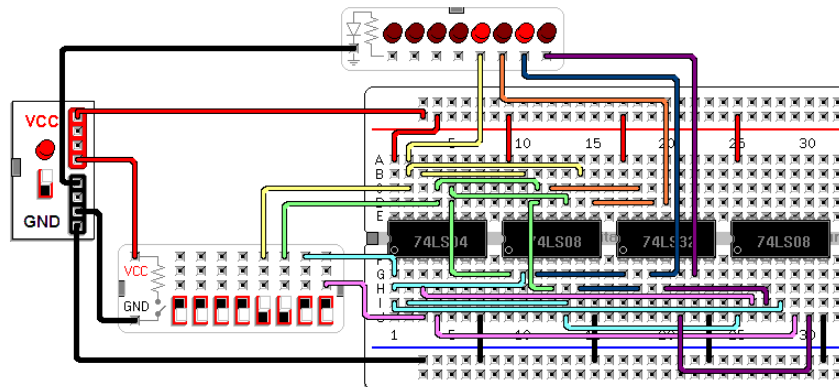


Figura 17: Entrada: 1100, Grey: 1010

Ahora, utilizando una compuesta *XOR* y sus equivalentes para cada salida, se obtuvo el siguiente circuito:

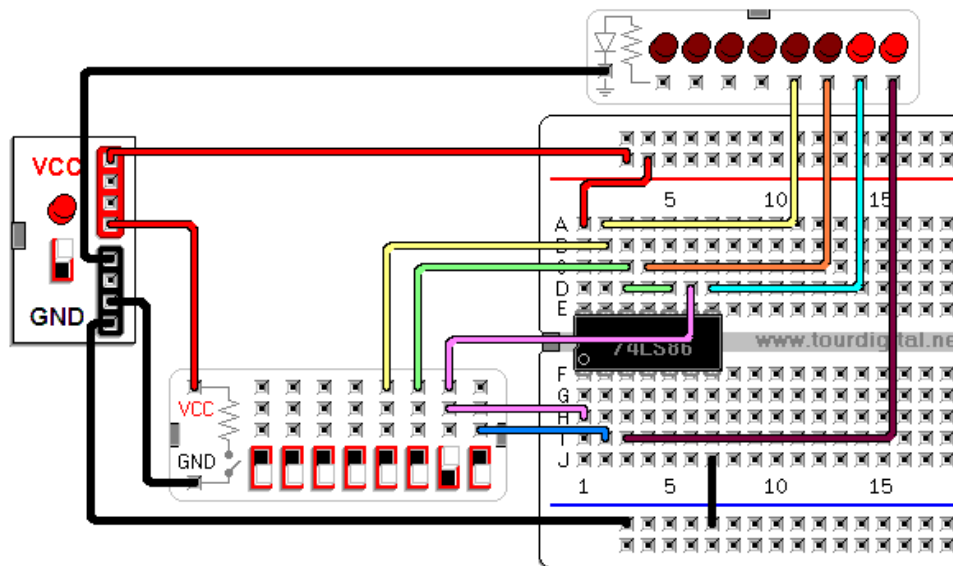


Figura 18: Entrada: 0010, Grey: 0011

Se vuelve evidente la importancia de la simplificación y el ahorro de componentes en comparación con las expresiones donde solo se utilizaron *AND*, *OR* y *NOT*.

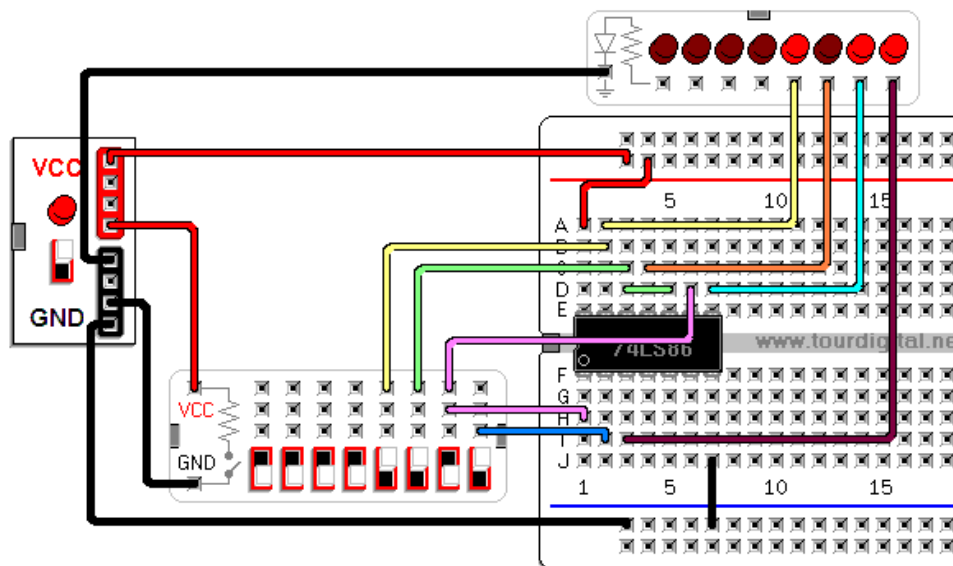


Figura 19: Entrada: 1101, Grey: 1011

2. Observaciones y conclusiones

González Cárdenas Ángel Aquilez

La práctica de laboratorio involucró la construcción de un circuito comparador de dos bits y un generador de código Gray de cuatro bits, los cuales destacaron la importancia de la simplificación en el diseño de circuitos lógicos y el ahorro que esto puede representar en términos del número de compuertas utilizadas.

Primero, se obtuvo la tabla de verdad de cada circuito y se extrajeron los minterminos de las expresiones lógicas correspondientes. Posteriormente, se llevó a cabo la simplificación de estas expresiones a su forma más reducida. Este paso de simplificación es crítico en el diseño de circuitos digitales, ya que permite reducir la complejidad y la cantidad de componentes necesarios para implementar una función lógica.

El ahorro que se logra a través de la simplificación es de gran importancia tanto en términos de costos como de eficiencia. Menos compuertas lógicas significa menos componentes electrónicos, menos espacio físico requerido y, en última instancia, menor consumo de energía. Además, la simplificación conlleva una mejora en la velocidad de operación del circuito, ya que menos compuertas implican menos retardos en la señal.

Al concluir la práctica, se demostró la importancia de la simplificación en el diseño de circuitos digitales, resaltando cómo la optimización de las expresiones lógicas puede llevar a un ahorro significativo en términos de recursos.

Hernández Reyes Diego Alberto

Al concluir la práctica, se construyeron los circuitos de un circuito comparador de dos bits y un generador de código Gray de cuatro bits. Se resaltó la importancia de construir circuitos a partir de expresiones simplificadas para lograr un diseño eficiente y confiable.

En este proceso, al simplificar las expresiones lógicas de cada salida, se obtuvo una representación más clara y compacta de las funciones que debían ser implementadas en los circuitos. Al construir los circuitos basándose en estas expresiones simplificadas, se logra una serie de ventajas:

- Mayor confiabilidad: Las expresiones simplificadas reducen la probabilidad de errores en la implementación física, ya que implican menos puertas lógicas y, por lo tanto, menos posibilidades de fallos.
- Eficiencia de recursos: Al construir circuitos a partir de expresiones simplificadas, se utiliza un número menor de componentes electrónicos, lo que resulta en un uso más eficiente de los recursos y reduce los costos de producción.
- Facilita el mantenimiento: Los circuitos basados en expresiones simplificadas son más fáciles de mantener y depurar, ya que la lógica subyacente es más clara y comprensible.

- Mejora el rendimiento: La simplificación de las expresiones lógicas puede conducir a un menor retardo en el circuito, lo que se traduce en un mejor rendimiento y tiempos de respuesta más rápidos.
- Ahorro de energía: Menos compuertas lógicas en el circuito implican un menor consumo de energía, lo que es especialmente importante en aplicaciones de bajo consumo energético.

En conclusión, la construcción de circuitos a partir de expresiones simplificadas es esencial para lograr diseños de circuitos digitales eficientes, confiables y rentables. Esta práctica subrayó la importancia de la simplificación como una herramienta fundamental, permitiendo la implementación de circuitos más efectivos y robustos.