

Hugo Andrade, Marshall Flores
Universidad de las Fuerzas Armadas “E.S.P.E” Ingeniería en electrónica y
Telecomunicaciones hdandrad@espe.edu.ec , jmflores15@espe.edu.ec

Resumen.- La electrónica digital ha marcado un gran avance en la tecnología durante los últimos 80 años. Es una rama de la electrónica que nos permite realizar circuitos lógicos a partir de las necesidades que se presenten en cualquier situación. Permittiéndonos aplicar dicho concepto para resolver o sustentar necesidades más avanzadas como el resolver operaciones aritméticas, multiplexor diferentes salidas o decodificar una serie de valores binarios, que incluso pueden ser de alta relevancia en la fundamentación acerca del funcionamiento de microprocesadores. En nuestro caso, al ser un entorno virtual en el cual vamos a trabajar, utilizaremos los componentes digitales que incorpora Tinkercad, para elaborar una calculadora básica que nos permita sumar y restar dos números de 8 bits (ingresados por el usuario), implementada en la plataforma Tinkercad utilizando principalmente el integrado 74283 (sumador de 4 bits), multiplexores 2 a 1 que nos permitan escoger la operación suma o resta, decodificadores y displays de 7 segmentos que nos permitan visualizar el resultado final en su salida. El propósito de este producto de unidad es analizar el funcionamiento de los circuitos lógicos aritméticos y combinacionales, haciendo énfasis en las aplicaciones que pueden tener en el ámbito de la electrónica y aplicándolo a un circuito, en nuestro caso, a la operatividad de un sistema selector de operaciones matemáticas, donde las entradas, corresponden a la variable de control y a los números A y B.

Introducción

A continuación se muestra como fue la realización de nuestro circuito comenzando por los diagramas que hemos utilizado

DIAGRAMAS

DIAGRAMA DE BLOQUES

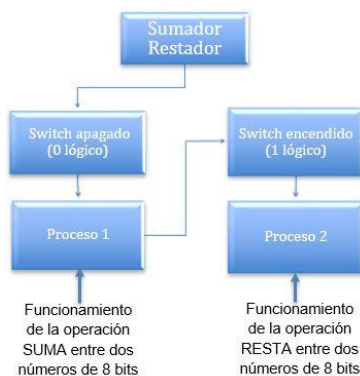


Diagrama 1.1 – Diagrama de bloques.

MAPARA DE VARIABLES

Variables utilizadas para el funcionamiento del circuito (Operaciones aritméticas seleccionadas mediante multiplexor)

Entradas

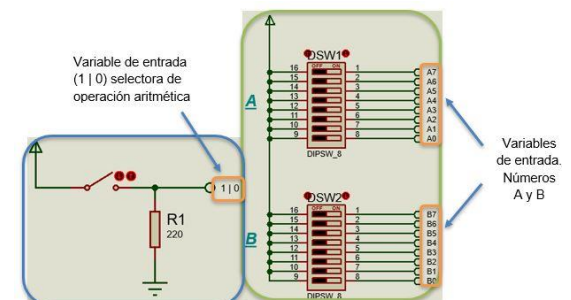


Figura 1.1 – Variables de entrada.

Salidas

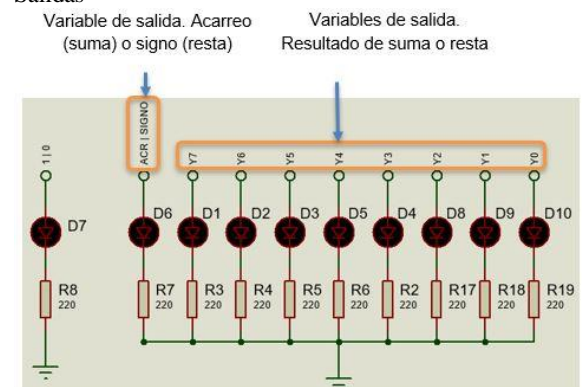


Figura 1.2 – Variables de salida.

La figura 1.1 y 1.2 nos ilustra de manera detallada todas las variables que han sido utilizadas para la implementación y el correcto funcionamiento del circuito (Operaciones aritméticas. Suma o resta) seleccionadas mediante un multiplexor.

En la figura 1.1 podemos observar las variables de entrada, las cuales corresponden al switch (1 | 0) y los dos dipswitch (Números A y B). Necesarios para realizar el control o selección de la operación aritmética a realizar y el ingreso de los valores en binario de A y B respectivamente.

La figura 1.2 ilustra las variables de salida (Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7) que se han utilizado para indicar el número binario resultante de la operación aritmética que se haya escogido realizar representadas por LEDs y también la variable (AC | SIGNO) que representa al bit de acarreo (suma) o el signo negativo (resta).

Interfaz de usuario

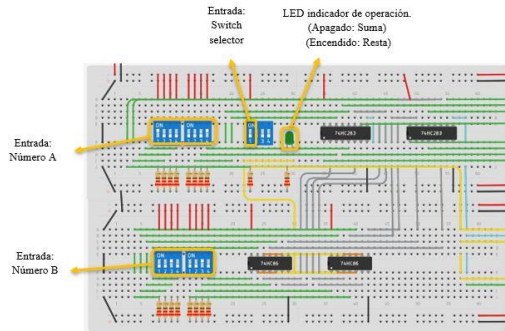


Figura 1.3 – Variables de entrada visibles en la interfaz de usuario.

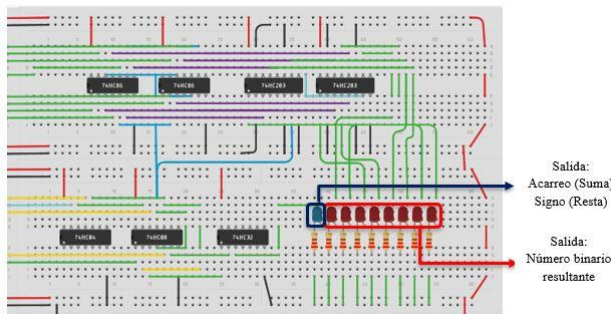


Figura 1.4 – Variables de salida visibles en la interfaz de usuario.

Donde:

A7 Y B7 son los bits más significativos y A0 Y B0 los bits menos significativos

Y7 es el bit más significativo y Y0 el menos significativo. A continuación, se presenta una tabla resumen de cada variable que se ha utilizado, el tipo y la función que realiza cada una en el circuito. Esto se ilustra en la tabla 1.3.

| Operaciones aritméticas seleccionadas mediante multiplexor | | |
|--|----------|--|
| Variables | Tipo | Función |
| 1 0 | Booleano | Variable de entrada encargada de controlar el funcionamiento de la suma (0) o la resta (1) |
| A0,A1,A2,A3 A4,A5,A6,A7 | Booleano | Variables de entrada que representan al número A de 8 bits. |
| B0,B1,B2,B3 B4,B5,B6,B7 | Booleano | Variables de entrada que representan al número B de 8 bits. |
| Y0,Y1,Y2,Y3 Y4,Y5,Y6,Y7 | Booleano | Variables de salida que representan al número resultante obtenido de la operación suma o resta. |
| ACR 1 SIGNO | Booleano | Variable de salida que representa el acarreo (en el caso que exista) al realizar la operación suma. Y al signo en el caso que se reste un número B de A, siendo B>A. |

Tabla 1.3 – Variables visibles y no visibles.

EXPLICACIÓN DEL CIRCUITO

OPERACIONES ARITMÉTICAS SELECCIONADAS MEDIANTE UN MULTIPLEXOR

El diseño del presente circuito, se basa en seleccionar, mediante un multiplexor con 1 variable de control, las operaciones correspondientes a la suma o resta de dos

números binarios de 8 bits, cuyos resultados se presentarán en formato decimal en displays de 7 segmentos. Es decir:

| S1 | Operación |
|----|-----------|
| 0 | Suma |
| 1 | Resta |

Tabla 1.4 – Operaciones mediante selectores

• CIRCUITO SUMADOR RESTADOR

El circuito sumador restador, consta principalmente de cuatro circuitos integrados (sumadores) 74HC283, los cuales son necesarios para realizar la operación suma o resta de dos números binarios de 8 bits.

Como primer punto tenemos la variable de control, la cuales determinarán la operación que se llevará a cabo. En este caso en particular, tenemos que para realizar la suma debe ejecutarse la acción (0) y para el caso de la resta (1). (Figura 1.5)

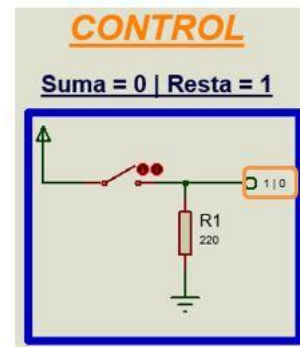


Figura 1.5 – Variables de control (Circuito Sumador | Restador)

Ahora bien, debido a que el circuito integrado 74HC283 (Sumador) necesita de un 1 ó 0 lógico en el pin correspondiente al carry de entrada (Co) para realizar la suma (0) o la resta (1) de los números binarios. Entonces procedemos a ingresar los valores ingresados de la variable de control (Figura 1.5) a dicho pin del sumador. De esta manera se realizará la suma únicamente cuando el interruptor esté en estado lógico BAJO y realizará la resta cuando se encuentre en estado lógico ALTO. El multiplexor escogerá el resultado de la operación que hayamos escogido (Figura 1.6). Esto último se explicará más adelante con más detalle.

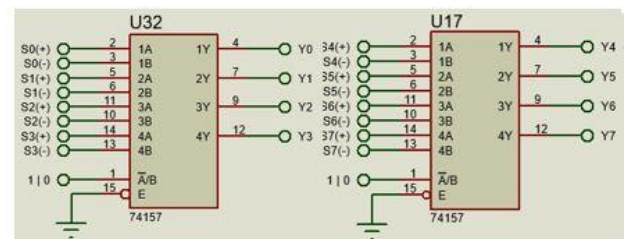
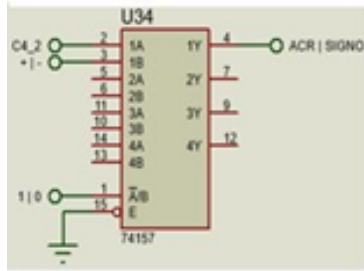


Figura 1.6 – Multiplexación de salidas



Por otro lado, tenemos las entradas de nuestro circuito, las cuales corresponden a los dos números de 8 bits A y B. Esto se ilustra en la figura 1.7.

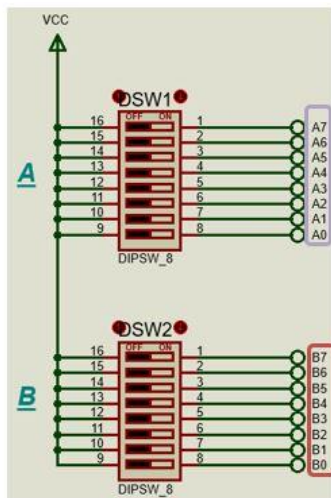


Figura 1.7 – Entradas número de 8 bits (A y B)

Donde A7 y B7 son las cifras más significativas (MSB) de ambos números, mientras que A0 y B0 son las cifras menos significativas (LSB).

Ahora bien, para realizar la suma o la resta de los números A y B, necesitamos realizar un cierto condicionamiento, para que funcione o la suma o la resta, dependiendo de lo que hayamos escogido, mediante el switch de la figura 1.5. Entonces para ello necesitamos de la utilización de las compuertas lógicas XOR, cuyo comportamiento lógico es necesario para, en el caso de la resta, realizar el complemento a 1 del número B.

Suponiendo que tenemos el switch de suma (0) o resta (1), y un número B, tal que:

Número B (en binario) = 10110001

Tendríamos las siguientes tablas de verdad, para cada caso:

| B | (-) | $B \oplus (-)$ | B | (-) | $B \oplus (-)$ |
|---|-----|----------------|---|-----|----------------|
| 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |

Tabla 1.6 – Ejemplo tabla de verdad XOR (Operación Resta)

La tabla de la izquierda ilustra lo que sucedería con el número B si realizamos la operación resta. Entonces Complemento a 1 de B = 01001110. Por otro lado, la tabla de la izquierda ilustra lo que sucedería con el número B si se realiza la operación suma. Es decir, el número sigue siendo el mismo.

Entonces, ya implementando esta fundamentación en el circuito, tenemos:

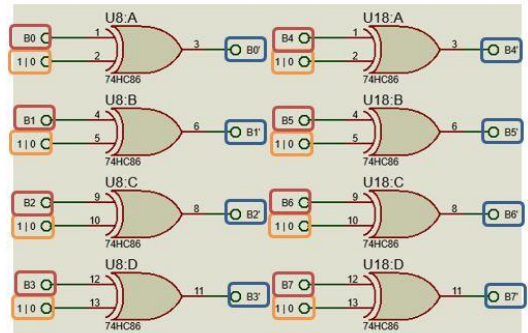


Figura 1.8 – Implementación de B en compuerta XOR

- Para la resta (bit que indica el signo)

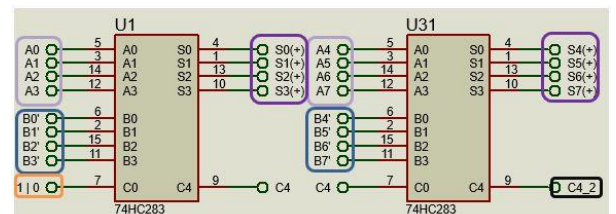


Figura 1.9 – Implementación y conexión de A y B al primer Circuito Integrado

Todo lo que se ha realizado hasta el momento nos indican las operaciones, suma o resta cuyos resultados son las cifras S0(+), S1(+), S2(+), S3(+), S4(+), S5(+), S6(+), S7(+) y el bit de acarreo resultante de la suma que es C4_2. Sin embargo, nos hace falta el bit que indica el signo (para la resta). Para ello necesitamos, conectar la salida C4_2 (figura 1.9). Tal como se ilustra a continuación.

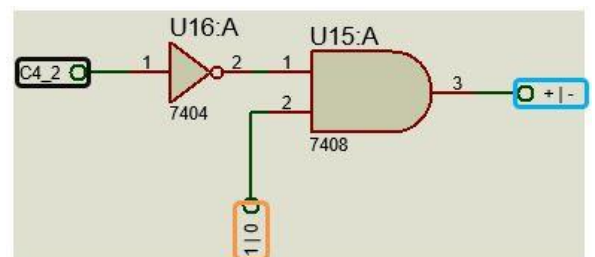


Figura 2.1 – Bit que indica el signo (+|-)

Básicamente, se realizó un condicionamiento, en el cual se ha conectado la salida C4 del circuito sumador a una compuerta lógica NOT y esta salida, conectada al primer pin de la compuerta AND. Mientras que en la segunda entrada se conectó la salida del switch de la figura 1.5. Para entender mejor su funcionamiento, ilustramos el siguiente ejemplo:

Sabemos que C4_2 puede tomar el valor de 1 o 0, dependiendo de la suma o resta que se realice entre los números A y B ingresados, entonces la tabla de verdad para la salida (+|-) nos queda de la siguiente manera:

| C4_2 | C4_2' (-) | Signo (+ -) |
|------|-----------|-------------|
| 1 | 0 | 1 |
| 0 | 1 | 1 |

Tabla 1.7 – Tabla de verdad para el signo (operación resta)

| C4_2 | C4_2' (+) | Signo (+ -) |
|------|-----------|-------------|
| 1 | 0 | 0 |
| 0 | 1 | 0 |

Tabla 1.8 – Tabla de verdad para el signo (operación suma)

Como podemos observar en la operación suma (Tabla 1.8), siempre vamos a obtener en la salida un 0, lo que representa al signo positivo. Mientras que en la operación resta (Tabla 1.7) tendremos un 1 (signo negativo) solamente cuando el bit de acarreo C4 sea 0, es decir, cuando el número A sea menor que el número B.

Hasta el momento sabemos que los dos primeros circuitos integrados se encargan de realizar la suma o resta entre los números A y B, siendo este último negado, es decir, con complemento a 1, por lo que tenemos un primer resultado. Sin embargo necesitamos de otros dos circuitos integrados (sumadores) conectados en cascada para realizar el complemento a 2 del resultado 1. Esto particularmente, para que el resultado final que obtengamos al realizar la operación resta, sea correcto.

Ahora bien, las conexiones, que necesitamos realizar en el tercer y cuarto circuito integrado, son similares a las del anterior, con algunas diferencias, que podemos notar a continuación.

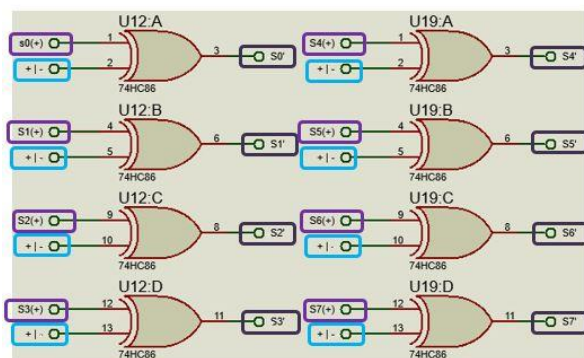


Figura 2.2 – Implementación del resultado 1 en compuerta XOR

Entonces implementando esta fundamentación en el tercer y cuarto circuito integrado (sumadores), tenemos lo ilustrado en la figura 2.3. Cabe recalcar que el bit del signo es conectado al pin de entrada C0, mientras que las entradas B0, B1, B2, B3, B4, B5, B6, B7 son conectadas a tierra (0 lógico), ya que estos pines no serán utilizados.

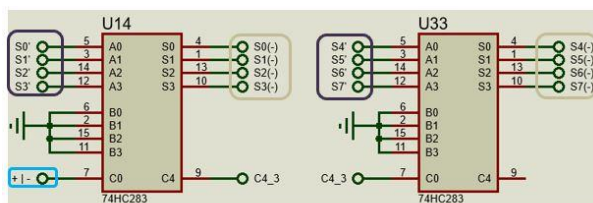


Figura 2.3 – Implementación y conexión de R1 al segundo Circuito Integrado

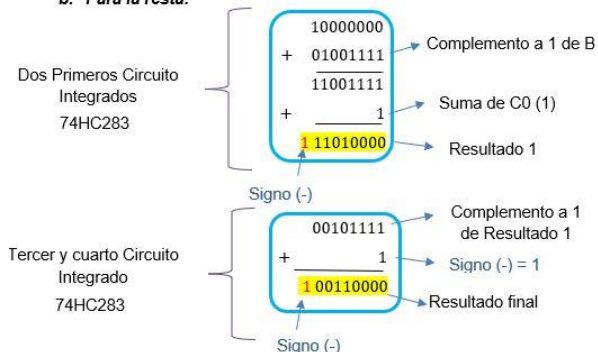
A continuación, ilustramos un ejemplo para entender de mejor manera lo que realiza cada circuito integrado:

- **A < B:** Ingresando el valor de A = 10000000 y B = 10110000

a. Para la suma:

$$\begin{array}{r} 10000000 \\ + 10110000 \\ \hline 100110000 \end{array}$$

b. Para la resta:

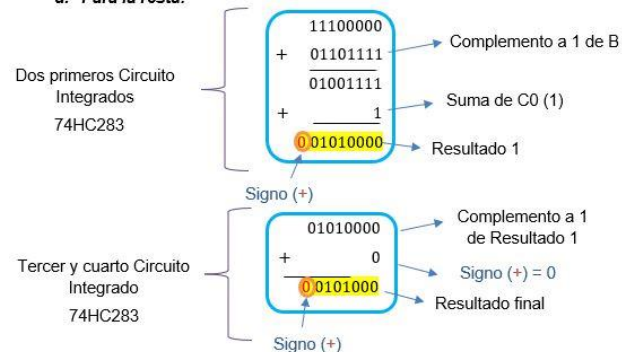


- **A > B:** Ingresando el valor de A = 11100000 y B = 10010000

c. Para la suma:

$$\begin{array}{r} 11100000 \\ + 10010000 \\ \hline 101110000 \end{array}$$

d. Para la resta:



Ahora bien, para realizar la selección de cada operación dependiendo del estado lógico de la variable de control (interruptor de la figura 1.5) se ha requerido de la utilización de tres circuitos integrados 74157 (Multiplexores). Tal como se muestra en la figura 2.4.

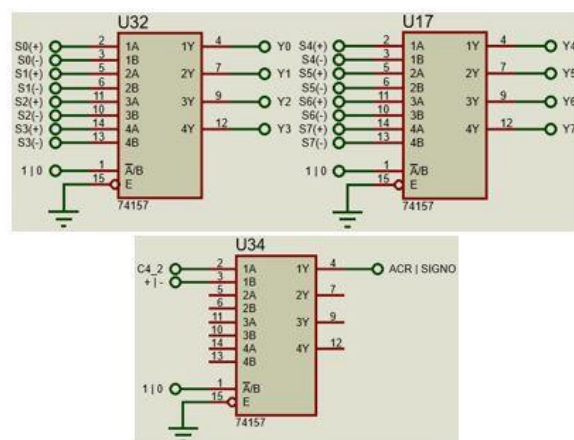


Figura 2.4 – Multiplexores

Cada circuito integrado contiene 4 multiplexores de 2 a 1, es decir 2 entradas a 1 salida, y 1 variable de selección, por lo

que, se escogerá una sola operación a realizar y el resultado se ilustra de la manera siguiente:

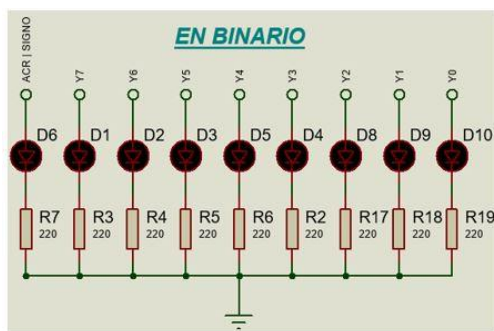
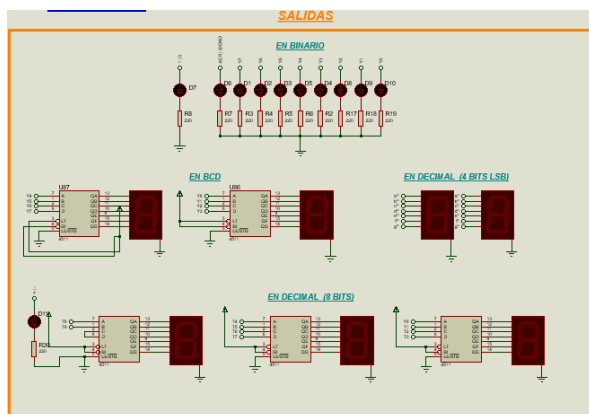


Figura 2.5 – Respuesta de operaciones en binario

Finalmente, para ilustrar el funcionamiento del circuito, en los diferentes displays de 7 segmentos (cátodo común) necesitamos utilizar decodificadores BCD (4511) y comparadores para realizar la transformación de binario a BCD, para los datos de salida, entonces el circuito visualizado de manera general se vería de la siguiente manera:



CONCLUSIONES

La implementación del circuito sobre la operatividad de un sistema selector de operaciones aritméticas de suma o resta, se lo ha realizado de tal manera que los datos de entrada, correspondientes a la variable de control (Switch selector) y a los números A y B en cuestión, se relacionen con los datos de salida entre sí por medio de un circuito combinacional que realiza la multiplexación de dos operaciones a realizar, proceso que se estableció en el cálculo de la suma o resta binaria, a través de circuitos lógicos aritméticos, cuyo circuito integrado es el 74283 y compuertas lógicas, tales como AND, OR, NOT y XOR, entonces, dependiendo del estado lógico en el que se encuentre la entrada selectora, obtuvimos el resultado en binario y en decimal, utilizando los decodificadores BCD 4511 y el display de 7 segmentos de cátodo común para este último caso.

El proceso de suma y resta binaria se basó principalmente en la fundamentación teórica correspondiente al complemento a 1 y/o complemento a 2. Debido a ello se utilizaron ciertos condicionamientos en los que realizamos la negación del número B a partir de compuertas lógicas XOR y este resultado agregarlo en el sumador 74283, realizando así la resta binaria, por otro lado en el caso de la suma se realiza el procedimiento normalmente sin negar al segundo valor

ingresado, obteniendo así el resultado de la operación que se haya escogido a realizar e ilustrándolo en LED's (respuesta binaria) y en los displays de 7 segmentos (respuesta en decimal).

El diseño del circuito se la realizó en base a todo lo investigado, lo cual tiene que ver con las funcionalidades que ofrecen los circuitos aritméticos y combinacionales. Cabe mencionar que esto es la base de una gran cantidad de aplicaciones en lo que respecta a sistemas digitales, ya que tienen una amplia relevancia e importancia, pues han sido implementados en procesadores para calcular índices de tablas, direcciones, etc. Sin duda los operadores aritméticos, junto con los circuitos combinacionales (en este caso multiplexores) han sido muy importantes en el avance de la electrónica y de la tecnología. Lo más esencial para nosotros es conocer cómo es su funcionamiento y mediante esta práctica se ha cumplido este objetivo. De hecho, al momento de realizar la investigación con los artículos de los autores investigados, podemos darnos cuenta de la gran cantidad de aplicaciones que tienen y que de cierta manera se relaciona con nuestro proyecto en cuestión.

Bibliografía:

[1] Privalov y K. Sohraby «Sci-Hub IEEEXplore» abril de 1998

Available:

<https://sci-hub.tw/https://ieeexplore.ieee.org/document/664263>

[2]JR Hoff, «Sci-Hub IEEEXplore» 24 de octubre de 2019

Available:

<https://sci-hub.tw/https://ieeexplore.ieee.org/document/8882342>

[3]R. J. Lycett, «Sci-Hub IEEEXplore» 13 de marzo de

Available:

<https://sci-hub.tw/https://ieeexplore.ieee.org/document/6479218>

Anónimo (26 de agosto del 2017) electronica.com Obtenido de:

<https://blog.ars-electronica.com.ar/2017/08/cd4511-decodificador-para-display-7.html>

Víctor González(15 de enero del 2020) piensa3d.com Obtenido de:

<https://piensa3d.com/que-es-un-multiplexor-como-funciona/>