

Ensayo: ¿Por qué odio a Kezn1tDeus?

Nicolás Aguilera (1.5 horas), Lucas Mesias (1 hora)

Departamento de Ingeniería Informática

Universidad de Santiago de Chile, Santiago, Chile

nicolas.aguilera.g@usach.cl, lucas.mesias@usach.cl

Resumen—El presente informe muestra la resolución de un problema que contempla una unidad aritmética lógica (ALU). En primer lugar se introducen el problema y su contexto, luego se presentan los antecedentes y el marco teórico dentro del cual se trabajará. A continuación se realiza una explicación de la solución y cómo se llegó a ella, para luego dar paso a los resultados del laboratorio y finalizar con la conclusión del trabajo realizado.

Palabras claves—c circuito, ALU, SimulIDE.

I. INTRODUCCIÓN

El siguiente informe tiene como objetivo la resolución de problemas de circuitos lógicos, mediante el uso del simulador de circuitos eléctricos y lógicos SimulIDE. Esto a través de la recreación de los circuitos mostrados en el enunciado y la interacción con ellos a través de la interfaz del programa para su posterior descripción y análisis.

Los objetivos de este informe se presentan a continuación:

- **Comprender** el funcionamiento de la memoria de programa y de una unidad aritmética lógica, desde ahora ALU por sus siglas en inglés (arithmetic-logic unit).

II. ANTECEDENTES

SimulIDE es un simulador en tiempo real de circuitos electrónicos, que permite experimentar y manipular circuitos simples y microcontroladores, soportando PIC, AVR y Arduino. Sin embargo no está diseñado para analizar circuitos, si no más bien para experimentar de manera rápida y sencilla con distintos circuitos y modelos electrónicos.

Las piezas utilizadas en este laboratorio son las siguientes[1][2]:

- **Voltaje fijo:** Proporciona voltaje de manera fija al circuito.
- **Interruptor:** Permite habilitar o deshabilitar el flujo de corriente en el circuito.
- **Conexión a tierra:** Su principal función es mantener el circuito a salvo, evitando diferencias de potencial.
- **Resistencia:** Sirve para limitar la corriente que circula por el circuito.
- **Led:** Circuito que se enciende al paso de corriente.
- **Visualizador de 7 segmentos:** Por lo general se utilizan para mostrar letras y/o números como salida de un circuito.

- **Circuito 74HC283:** Es un sumador de dos números de 4 bits cada uno.
- **Circuito 74HC157:** Es un multiplexor de dos números de 4 bits cada uno.

III. MATERIALES Y MÉTODOS

III-A. Materiales

Para el desarrollo de esta actividad solamente se hizo uso del SimulIDE en su versión 1.0.0 sin herramientas o softwares adicionales.

III-B. Métodos

A continuación se describen los métodos utilizados para llegar a la solución del laboratorio.

En primer lugar, el problema se comenzó por la recreación del circuito a través de la interfaz de SimulIDE, esto buscando en la biblioteca de piezas y componentes para cada una de las partes necesarias para recrear el circuito. Luego, se procedía a efectuar la simulación del circuito encendiendo las fuentes de voltaje y observando cómo se comportaba en caso de prender o apagar alguna de las fuentes del circuito. Opcionalmente, para estudiar el comportamiento de algunas partes del circuito (como el conjunto de operadores lógicos en la parte baja o en la parte derecha del circuito, de la salida), se implementó en Python probando todas las combinaciones posibles observando algún patrón. Finalmente se analiza el funcionamiento y los resultados de este.

El conjunto de las 4 puertas lógicas que se ubican al lado derecho del multiplexor fue analizado con una herramienta generadora de tablas, usando `"((a nor b) nand (c nor d)) nand ((a nor b) nand (c nor d))"` como entrada para la herramienta.

IV. RESULTADOS

Los resultados obtenidos consisten en el comportamiento del circuito tras diferentes pruebas con las entradas de voltajes en cada uno de estos. Los componentes utilizados en este circuito son los mismos descritos en el apartado de Antecedentes.

En las figuras 1-4 se muestran ejemplos de ejecución del circuito y cómo estos varían dependiendo del tramo de voltajes que se encuentren encendidos. Si lo separamos por secciones hay 5:

- La fijación de 2 números binarios que van del 0 al 15
- El circuito 74HC283 que suma dichos números
- El conjunto inferior de compuertas lógicas
- El circuito 74HC157 multiplexor de dos números de 4 bits
- El conjunto de compuertas lógicas que modifican la salida superior del circuito.

En la primera sección se fijan 2 números binarios de 4 bits, donde las fuentes de voltaje representan cada uno de estos, cada una representando la potencia de 2 del número que representa, desde la fuente superior siendo la potencia de 0, hasta el inferior siendo la potencia de 3, conectados a estos números hay visualizadores de 7 segmentos que permiten ver estos números con mayor facilidad.

La segunda sección es un sumador (4-bit binary full adder with fast carry) de números binarios de 4 bits, cada conector del lado izquierdo es una potencia de 2 que compone el número, la cual se suma con su par correspondiente, a este par de 4 bits se le agrega una entrada C_{in} , la cual permanece en estado LOW o 0, lo que señala que no hay intención de hacer “carry in”. En el lado derecho de este componente se encuentran los conectores de salida, el grupo superior de 4 conectores es el número de 4 bits resultante de la suma, añadiendo un quinto conector C_{out} , que almacena la potencia de 2 a la cuarta, para los casos en que la suma de los números de entrada excede a los 4 bits de salida, si dicho límite es excedido, este conector encenderá la luz led inferior. Un ejemplo del circuito sumando dos números se puede apreciar en la Figura 1.

La tercera sección es un conjunto de operaciones lógicas que obtienen como entrada 7 conectores de distintos bits de los números generados, por lo que se definió que usa 2 números de 4 bits de entrada, este conjunto no realiza una operación matemática conocida, tampoco fue posible encontrar un patrón en los resultados entregados por este al generar todas las combinaciones de entradas posibles. Este circuito entrega 4 conectores de salida, los cuales son configurados como un número binario de 4 bits en el multiplexor.

El multiplexor es un componente que recibe como entrada 2 pares de señales de 4 bits, y retorna solo 1 de estas señales, esto se decide con el conector marcado como S, si este es LOW o 0, la señal A o I_0 , será la retornada, en el caso contrario, la señal B o I_1 es retornada.

La última sección del circuito encenderá la luz led si el número de 4 bits que retorna el multiplexor es 0. Sabemos esto gracias a la tabla de verdad generada con la herramienta de dCode, mostrada en la Figura 3. En la Tabla 1 se muestra un resumen de algunos de los resultados obtenidos al activar el voltaje superior que se encuentra entre los visualizadores de 7 segmentos.

V. CONCLUSIONES

Luego de realizar la experiencia, se puede decir que se han cumplido parcialmente los objetivos planteados en la

N1	N2	Resultado
0	0	8
0	1	10
0	2	8
0	5	11
0	10	10
1	0	8
1	2	8
1	3	2
1	4	9
2	0	9
2	1	11
2	2	9
2	4	9

Tabla I: Resultados de la operación realizada entre dos números de 4 bits al activar el voltaje superior del circuito.

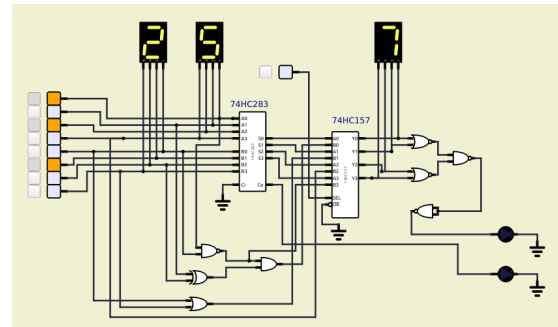


Figura 1: Suma de los dos números de 4 bits, representados por las fuentes de voltaje, mientras la fuente de voltaje superior está apagada.

introducción de este informe, es decir, utilizar el simulador para circuitos electrónicos y lógicos SimulIDE para la búsqueda de una solución y entender el funcionamiento de una ALU. En particular, este último fue difícil de comprender ya que si bien se comprendió el funcionamiento de todas las partes del circuito, el subcircuito inferior dió problemas ya que al entregar resultados que no siguen un patrón o que a primera vista no tienen un sentido fácil de descifrar, fue la parte a la que se dedicó la mayor parte del análisis. Es por eso que se cumplen parcialmente los objetivos, ya que puede ser que efectivamente ese circuito tenga un patrón que no se ha podido descifrar.

Por otro lado, el reconstruir el circuito en SimulIDE requirió de una pequeña investigación sobre circuitos ya que el original estaba construido en Proteus y por lo tanto, difieren un poco en su construcción.

REFERENCIAS

- [1] Philips, “74hc/hct157 quad 2-input multiplexer,” *Philips*, vol. 1, no. 1, pp. 1–4, 1990.
- [2] —, “74hc/hct283 4-bit binary full adder with fast carry,” *Philips*, vol. 1, no. 1, pp. 1–4, 1990.
- [3] “Boolean truth table generator,” <https://www.dcode.fr/boolean-truth-table>, accessed: 2022-12-15.

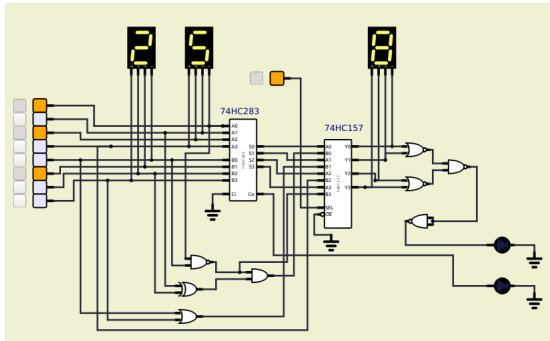


Figura 2: Mismo caso de la Figura 1, a excepción de que la fuente superior se encuentra encendida, entregando otro resultado que no es la suma.

t↓	t↓	t↓	t↓	t↓
a	b	c	d	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Figura 3: Tabla de verdad generada por dCode[3]