

---

## Trabajo Práctico de Laboratorio N° 2

---

*Grupo 5:*

Matías LARROQUE  
Leg. 56597

Lucero Guadalupe FERNANDEZ  
Leg. 57485

Manuel MOLLÓN  
Leg. 58023

Ezequiel VIJANDE  
Leg. 58057

*Profesor:*

Kevin DEWALD  
Pablo WUNDES

Entregado: 18 de Octubre de 2018

# 1. IMPLEMENTACIÓN DE TABLA DE VERDAD CON COMPUERTAS LÓGICAS

El objetivo fue implementar la tabla de verdad con compuertas lógicas discretas. Se sabe que simplificando ya sea con máxterminos o mintérminos se llega a la expresión mínima, es decir a la máxima simplificación; por lo que se procedió a simplificar con mintérminos, resultando en lo siguiente, acorde al mapa de Karnaugh 1.1:

$$Y = \overline{A}B + \overline{B}C \quad (1.1)$$

c \ ab	00	01	11	10
	0	1	0	0
0	0	1	0	0
1	1	1	0	1

La expresión anterior es la que representa el menor costo pues se implementa con la menor cantidad posible de compuertas (ver figura 1.1). Sin embargo, se observa que al transicionar entre los grupos  $(A, B, C) = (0, 1, 1)$ , -en verde-, y  $(A, B, C) = (0, 0, 1)$ , -en azul-, es decir variando el estado de la variable B, el comportamiento será incierto, pues no está cubierta por ningún mintérmino. Para medir ésto, se conectó la entrada A a masa, y C, a +Vcc (5 V). La señal B se simuló con una onda cuadrada generando así la transición entre los susodichos estados. Esto se puede ver en la figura 1.2; la entrada es la señal amarilla, y la verde la salida, que se encuentra en un nivel de tensión alto, es decir un 1 lógico, como era de esperar. Por otro lado, cuando B varía de 5 V a 0 V, se observa un comportamiento indeseado, un glitch, que se debe a los tiempos de delay propios de las compuertas lógicas.

Cuadro 1.1: Mapa de Karnaugh de menor costo.

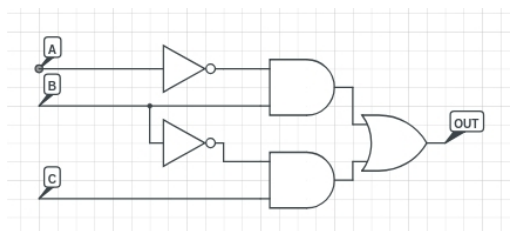


Figura 1.1: Circuito utilizado de menor costo.

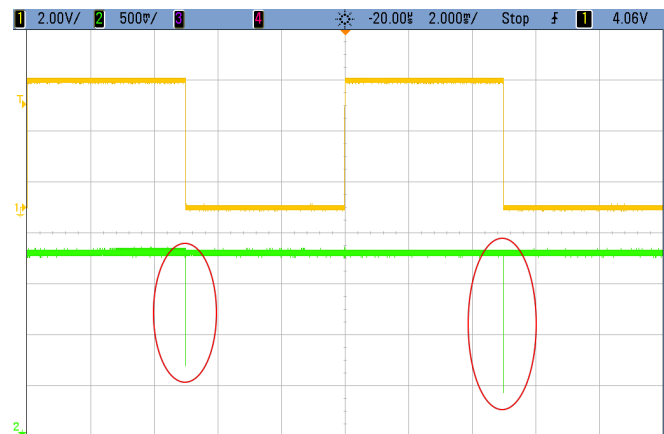


Figura 1.2: Glitch observado señalado en rojo al variar el estado de B.

Para evitar este inconveniente lo que se puede hacer es agregar un grupo al mapa de Karnaugh entre los estados problemáticos, para asegurarse que la salida, al transicionar entre los mismos, se mantenga siempre en el mismo valor; ver figura 1.3. Entonces, agregando un mintérmino, la función queda simplificada a:

$$Y = \overline{A}B + \overline{B}C + \overline{A}C \quad (1.2)$$

Lo que da como resultado el circuito 1.4, que si bien requiere una compuerta más (un OR); y además un AND de tres entradas, evita que haya glitches.

		ab			
c		00	01	11	10
	0	0	1	0	0
	1	1	1	0	1

Figura 1.3: Mapa de Karnaugh sin glitches.

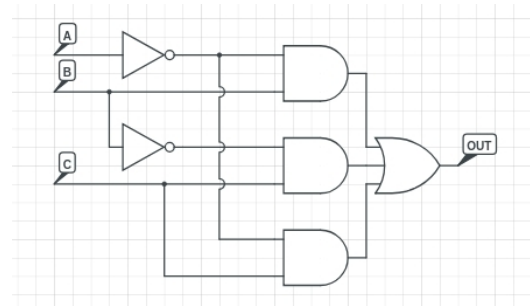


Figura 1.4: Circuito utilizado de mayor costo.

## 2. ENTRADAS FLOTANTES EN COMPUERTAS TTL Y CMOS

Se implementaron los siguientes circuitos:

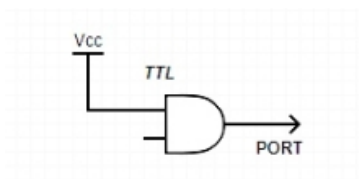


Figura 2.1: Compuerta AND TTL con entrada flotante.

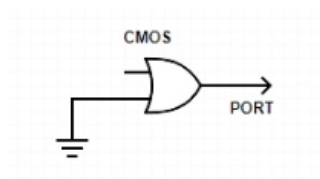


Figura 2.2: Compuerta OR CMOS con entrada flotante.

En primer lugar, en el caso de la compuerta TTL se conectó una entrada a  $V_{cc}$  (5V) y se dejó la otra flotante (figura 2.1), es decir no se le realizó ninguna conexión y se midió la salida, que se observó constante como un 1 lógico (aproximadamente 5V).

En el caso de la CMOS, se dejó también un input flotante y otro se lo conectó a masa, como se ve en la figura 2.2. Se observó en este caso ruido de línea de 50Hz, con amplitud que variaba entre 0-0.8V, que no se encontraba entre las tensiones high y low de salida del CMOS ( $V_{OH}$  y  $V_{OL}$ )<sup>1</sup>, es decir, la salida era indefinida (no era ni un 1 ni un 0 lógico).

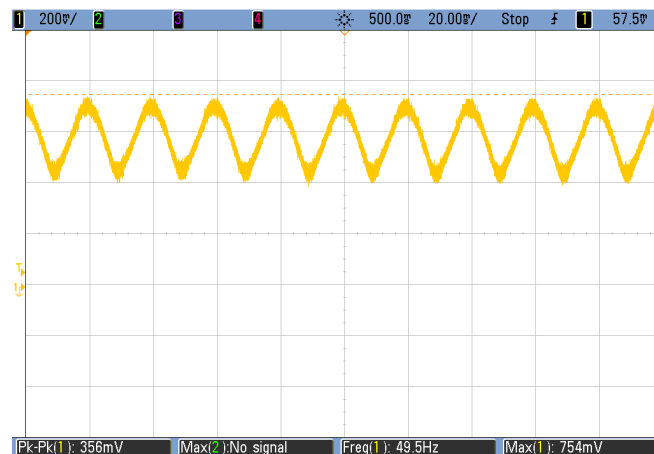
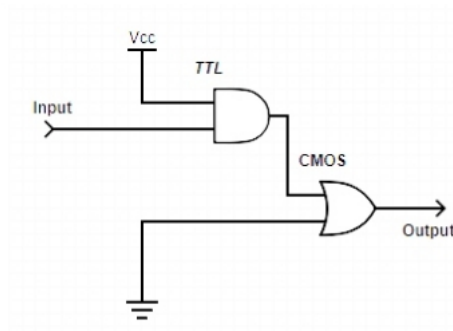


Figura 2.3: Ruido observado del CMOS con entrada flotante.

<sup>1</sup> Datos sacados de la datasheet: ver figura 2.5.



Cuadro 2.1: Circuito implementado.

Para el CMOS este comportamiento puede deberse a que tiene alta impedancia de entrada por lo que al dejar una entrada flotante funciona como antena, induciéndose corrientes, por ejemplo, por ruido, lo que genera tensiones a la entrada de la compuerta ocasionando un nivel de tensión high o low incierto a la salida. Para que no ocurra ésto, el fabricante recomienda conectar las entradas que no se utilizan ya sea a GND o a Vcc, para evitar estados operacionales indefinidos.<sup>2</sup>

A continuación se implementó el circuito del cuadro 2.1. No se observó ningún comportamiento indeseado, sin embargo, si se observan las figuras 2.4 y 2.5 que muestran los niveles de tensión high y low de entrada y salida para una compuerta TTL y CMOS ( $V_{IH}/V_{IL}$  y  $V_{OH}/V_{OL}$ ), se pueden encontrar problemas si, por ejemplo, la tensión de entrada al circuito es de 3V; ésto ocasionaría que la salida de la TTL estuviera entre

los 2.4V-5V para un 1 lógico, que podría presentar cierta incompatibilidad con los valores de tensión de entrada de la CMOS, ya que podría caer dentro de la zona indefinida, resultando en una salida incierta del circuito.

Posibles soluciones a este inconveniente serían utilizar mismas familias de compuertas, es decir, ambas TTL (LS) o CMOS (HC), lo que corrige estas ambigüedades. Se implementó el nuevo circuito y no se encontraron problemas, como era esperable.

Figura 2.4: Tensiones  $V_{OH}$ ,  $V_{OL}$ ,  $V_{IH}$ ,  $V_{IL}$  para una compuerta TTL. Figura 2.5: Tensiones  $V_{OH}$ ,  $V_{OL}$ ,  $V_{IH}$ ,  $V_{IL}$  para una compuerta OR CMOS.

A modo de síntesis, una entrada flotante en un TTL es considerada como un 1 lógico a la entrada, y un input flotante en CMOS tiene un comportamiento incierto a la salida. Si bien, el circuito combinado con compuertas TTL y CMOS funciona de manera correcta, los fabricantes *Texas Instruments*<sup>3</sup> y *Fairchild*<sup>4</sup> no pueden asegurar el valor de salida del integrado y recomiendan conectar todos los inputs sin utilizar a un valor de tensión conocido, sea GND, Vcc o cortocircuitarla con otra entrada que esté en uso, dependiendo de la función del circuito.

<sup>2</sup>Referirse a la hoja de datos del fabricante: <http://www.ti.com/lit/ds/symlink/sn54hc32-sp.pdf>.

<sup>3</sup>Nota de aplicación del fabricante: Haseloff, E. (1997). Designing with Logic. *Texas Instruments*, 6-7. Extraído de <http://www.ti.com/lit/an/sdya009c/sdya009c.pdf>.

<sup>4</sup>Nota de aplicación del fabricante: (1984). Application Note 363. Designing with TTL. *Fairchild*, 2. Extraído de <https://www.fairchildsemi.com/application-notes/AN/AN-363.pdf>.