

# INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.13 ELECTRÓNICA III

---

## Trabajo práctico 2

---

*Grupo 3*

STEWART HARRIS, María Luz	57676
PARRA, Rocío	57669
PIERDOMINICI, Matías Nicolás	57498
GONZÁLEZ ORLANDO, Tomás Agustín	57090

*Profesores*

DEWALD, Kevin  
WUNDES, Pablo Enrique

Presentado: 18/10/2018

# Índice

<b>1</b>	<b>Ejercicio 1</b>	<b>2</b>
1.1	Análisis de resultados . . . . .	2
<b>2</b>	<b>Ejercicio 2</b>	<b>3</b>
<b>3</b>	<b>Ejercicio 3</b>	<b>4</b>
<b>4</b>	<b>Ejercicio 4</b>	<b>5</b>
<b>5</b>	<b>Ejercicio 5</b>	<b>6</b>
5.1	Compuerta <i>and</i> TTL . . . . .	6
5.2	Compuerta <i>or</i> CMOS . . . . .	6
5.3	Compuerta CMOS cargando a la compuerta TTL . . . . .	7
5.4	Conclusiones . . . . .	7
<b>6</b>	<b>Ejercicio 6</b>	<b>7</b>
6.1	Flip flop D . . . . .	8
6.1.1	Funcionamiento . . . . .	8
6.1.2	Circuito lógico . . . . .	8
6.1.3	Implementación y medición . . . . .	8
6.2	Latch SR . . . . .	8
6.2.1	Funcionamiento . . . . .	8
6.2.2	Circuito lógico . . . . .	9
6.2.3	Implementación y medición . . . . .	9
<b>7</b>	<b>Ejercicio 7</b>	<b>9</b>
7.1	Contador síncronico . . . . .	9
7.1.1	Funcionamiento . . . . .	9
7.1.2	Implementación y Medición . . . . .	9
7.1.3	Máxima velocidad de conteo . . . . .	10
7.2	Contador asíncronico . . . . .	10
7.2.1	Funcionamiento . . . . .	10
7.2.2	Implementación . . . . .	10
7.2.3	Máxima velocidad de conteo . . . . .	11
<b>8</b>	<b>Ejercicio 8</b>	<b>11</b>
8.1	Sensor HC-SR04 . . . . .	11
8.2	Circuito Implementado . . . . .	11
8.2.1	Generador de pulso . . . . .	12
8.2.2	Contador de tiempo . . . . .	12
8.2.3	Meas ready . . . . .	12
8.2.4	Prototipo . . . . .	12
8.2.5	Circuito . . . . .	13
8.2.6	Simulación . . . . .	13

## Ejercicio 1

En esta sección se implementaron compuertas *not* a partir de transistores BJT (de sus siglas en inglés, *bipolar junction transistor*). Particularmente, se decidió utilizar BJTs de tipo NPN, formando circuitos de dos familias distintas:

- RTL (*resistor-transistor logic*): utiliza resistencias en la malla de entrada y transistores como *switch*.
- TTL (*transistor-transistor logic*): utiliza transistores tanto para el *switching* como para la amplificación.

De estas definiciones surge que en TTL se requiere mayor cantidad de transistores por compuerta, pero se disipará menos potencia en resistencias pues las corrientes serán menores. Para comparar otros factores de su funcionamiento, se armaron los siguientes circuitos:

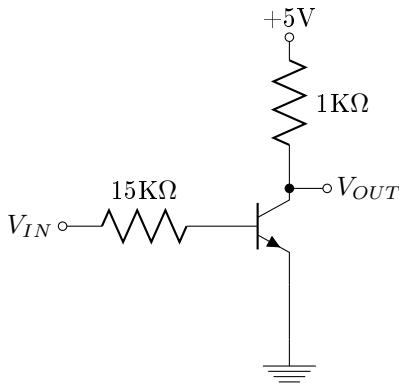


Figura 1: Compuerta *not* RTL

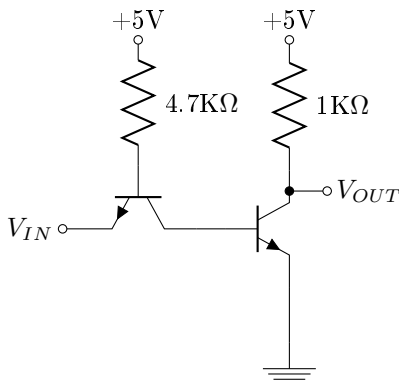


Figura 2: Compuerta *not* TTL

Se utilizaron transistores BC547 y resistencias de metal-film, y los circuitos se armaron en una *printed circuit board*.

## Análisis de resultados

Las mediciones se realizaron el circuito con ondas cuadradas de 5V de amplitud, con el nivel bajo en 0V. Las tensiones necesarias para calcular el margen de ruido se midieron utilizando el modo XY del osciloscopio con permanencia infinita, mientras que para los tiempos el modo *single*. Todas las mediciones se realizaron en dos condiciones para cada compuerta: en vacío, y con un capacitor de 1nF como carga.

Los resultados de las mediciones se encuentran en las tablas 1 y 2. Las magnitudes medidas fueron:

- HLIV (*high-level input voltage*): máxima tensión de entrada con la cual la pendiente de  $V_{IN}(V_{OUT}) = -1$ , es decir, la mínima con la que se interpreta un 1 en la entrada.
- LLIV (*low-level input voltage*): mínima tensión de entrada con la cual la pendiente de  $V_{IN}(V_{OUT}) = -1$ , es decir, la máxima con la que se interpreta un 0 en la entrada.
- HLOV (*high-level output voltage*): tensión de salida cuando  $V_{IN} = HLIV$ , es decir, la mínima con la cual puede considerarse que hay un 1 en la salida.
- LLIV (*low-level output voltage*): tensión de salida cuando  $V_{IN} = LLIV$ , es decir, la mínima con la cual puede considerarse que hay un 0 en la salida.
- Noise margin: diferencia entre el *high/low level input* y *output voltage*, es decir rango de tensiones que pueden hallarse a la entrada pero no a la salida (pero que tienen comportamiento definido para la salida) para cada nivel lógico.
- PD (*propagation delay*): tiempo que transcurre entre que la tensión de entrada está al 50% entre su valor bajo y alto, y que lo opuesto ocurre en la salida.
- TT (*transition time*): tiempo que tarda la salida en transicionar de un 10% a un 90% de la tensión alta o viceversa.
- Max. out. curr (*maximum output current*): corriente de salida máxima, que por ser la carga puramente capacitiva, se calculó como  $i_c = C \cdot \frac{dV_C}{dt}$ , obteniendo esta derivada a través de las funciones matemáticas del osciloscopio.

RTL	En vacío	Con carga
HLIV (V)	0.80	0.80
LLIV (V)	0.57	0.57
HLOV (V)	4.80	4.80
LLOV (V)	0.27	0.27
Noise margin H (V)	4.0	4.0
Noise margin L (V)	0.3	0.3
PD (H-L) (ns)	3855	4380
PD (L-H) (ns)	65	140
TT (H-L) (ns)	134	224
TT (L-H) (ns)	520	3860
Max. out. curr. (mA)	-	51

Tabla 1: Mediciones para la compuerta RTL

TTL	En vacío	Con carga
HLIV (V)	0.63	0.63
LLIV (V)	0.55	0.55
HLOV (V)	4.85	4.85
LLOV (V)	0.21	0.21
Noise margin H (V)	4.22	4.22
Noise margin L (V)	0.34	0.34
PD (H-L) (ns)	804	1445
PD (L-H) (ns)	10	15
TT (H-L) (ns)	<13	70
TT (L-H) (ns)	115	3120
Max. out. curr. (mA)	-	16

Tabla 2: Mediciones para la compuerta TTL

En primer lugar, debe tenerse en cuenta que las mediciones de los tiempos de transición están limitados por el *rise time* mínimo del generador para ondas cuadradas, que es de  $13\text{ns}^1$ , con lo cual para algunas mediciones sólo podemos afirmar que el valor obtenido es menor a este último.

Se puede apreciar que, para ambos circuitos, la carga no afecta los valores de tensiones de entrada ni de salida que se consideran como 1 o 0, pero causa que se extiendan todos los tiempos, tanto de propagación como de transición. Esto es consistente con el hecho de que la carga capacitiva limita la variación de tensión para una misma corriente.

Se observa además que prácticamente todos los parámetros medidos indican una mejor *performance* de la compuerta TTL respecto de la RTL. La principal ventaja de este último tipo de tecnología era el menor requerimiento en cantidad de transistores, lo cual era significativo cuando estos componentes eran considerablemente más caros que el resto del circuito. Actualmente, puesto que ya no es necesario el uso de transistores discretos sino que puede re-

currirse a circuitos integrados y su precio ha disminuido, las compuertas RTL están prácticamente obsoletas.

El única medición realizada que favorece a la compuerta RTL es la de máxima corriente de salida: una mayor corriente máxima indica un mayor *fanout*. Esta es una de las principales limitaciones de las compuertas TTL.

## Ejercicio 2

Cuando se utilizan compuertas lógicas provenientes de distintas familias, es importante tener en cuenta si son compatibles entre ellas. En esta sección, estudiaremos cómo interactúan compuertas *nor* HC (CMOS), HCT (CMOS compatible con TTL) y LS (TTL).

Cargando cada compuerta con cada una de las otras dos sucesivamente (con la otra entrada conectada a *ground*, de forma tal que la salida fuese la entrada negada), se realizaron *DC sweeps* con tensiones de 0V a 5V (puesto que esta fue la tensión que se utilizó como  $V_{CC}$ ) para observar qué problemáticas podían presentarse. Todas las combinaciones funcionaban de acuerdo a lo esperado para compuertas *nor*, salvo cuando se cargó la compuerta HC con la LS.

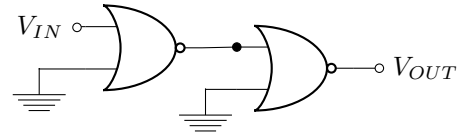


Figura 3: Conexión entre las compuertas

En este caso, se observó un valor inesperado de -0.22V en la salida cuando:

- *low to high*: la tensión de entrada se encontraba entre 2.33V y 2.47V
- *high to low*: la tensión de entrada se encontraba entre 2.25V y 2.38V

Este es el rango de tensiones donde, de acuerdo al fabricante<sup>2</sup> se encuentra el *low level input voltage*, y donde la salida cambiaba de estado cuando la carga de la HC era la HCT (2.4V). Esto sugeriría que al hacer interactuar una compuerta CMOS con una TTL sin una interfaz adecuada, pueden obtenerse a la salida tensiones que no reflejan adecuadamente el circuito lógico que se supone que se está representando.

<sup>1</sup>Información obtenida de la hoja de datos del generador Agilent 33220A (consultado: 16/10/18).

<sup>2</sup>Información obtenida de la hoja de datos del integrado (consultado: 12/10/18).

<sup>3</sup>De acuerdo a la nota de aplicación de Fairchild Semiconductors: *An Introduction to and Comparison of 74HCT TTL Compatible CMOS Logic* (consultado: 13/10/18).

Efectivamente, esta es una de las aplicaciones de la familia de compuertas HCT<sup>3</sup>. Si bien según la nota consultada, suele haber problemas al cargar una compuerta TTL con una CMOS y no tanto al revés (al contrario de lo que se obtuvo, si bien el glitch hallado se presentaba en un rango de tensiones acotado), se reconocen las incompatibilidades entre ambas tecnologías.

En particular, se pueden suscitar inconvenientes debido a que la tensión de salida *high* de las compuertas TTL es mayor que la de entrada *high* de las CMOS para algunos de los valores que pueden tomar estos parámetros, pero no para todos los admitidos por el fabricante. Por ejemplo, para  $V_{CC} = 4.5V$ , para el integrado 74LS02 no se garantiza que la tensión de salida alta de las compuertas sea superior a 2.5V, si bien típicamente es de 3.5V<sup>4</sup>. En cuanto a la HC, las tensiones de entrada necesarias para salida alta se encuentran entre 2.4V y 3.15V. Por lo tanto, existe un rango de valores en que no se cumple que el primer parámetro sea mayor que el segundo.

En conclusión, si bien en este caso la tensión de entrada necesaria para obtener una salida alta en HC era de 2.4V con  $V_{CC} = 5V$ , con lo cual la salida alta de la compuerta LS, medida en 3.9V, podía inducir a su vez un cambio en la salida de la HC (en este caso, puesto que las compuertas eran de tipo *nor*, de 1 a 0). Sin embargo, sólo porque en este caso no se presentó este problema no implica que sea una buena práctica realizar conexiones de este tipo. De hecho, se registró otro tipo de problema que no se observó para la *gate* HCT. Por lo tanto, de suscitarse la necesidad de hacer interactuar una compuerta CMOS con una TTL, se debe recordar que hay compuertas CMOS especialmente diseñadas para ser compatibles con las tensiones de salida más bajas de las TTL.

### Ejercicio 3

Se denomina *hazard* a los *glitches* causados por la estructura del circuito y los delays de propagación de las compuertas. Estos ocurren cuando la salida toma momentáneamente un valor que no se corresponde con lo establecido por la tabla de verdad del circuito. Existen dos tipos de *hazards*: dinámicos y estáticos.

- Dinámico: cuando la salida cambia, oscila entre 0 y 1 momentáneamente antes de establecerse. Sucede cuando existen múltiples caminos para que una señal se propague.

- Estático: la salida cambia cuando debería mantenerse. Al usar el método de mapas de Karnaugh con suma de productos para obtener la función lógica de mínimo costo de una tabla de verdad, usualmente quedan 1 adyacentes que no comparten grupo. Las transiciones entre estos estados tienen un riesgo de *hazards* estáticos.

Se pueden observar ejemplos de ambos tipos en la figura 4.

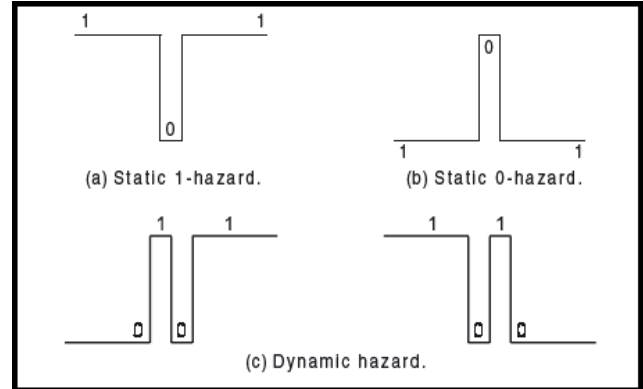


Figura 4: *Glitches* estáticos y dinámicos<sup>5</sup>

Para evitar los *hazards* estáticos, se usan funciones lógicas resultantes de agregar grupos extras al mapa de Karnaugh de manera tal que todos los 1 adyacentes compartan grupo. Estos grupos extra son redundantes, por lo que van a aumentar el costo de la función.

A	B\C			
	00	01	11	10
0	0	1	1	1
1	0	1	0	0

Figura 5: Mapa de Karnaugh utilizado. Resolución de menor costo por suma de productos. Si se quisiera reducir el riesgo de *hazards*, se debería agregar otro grupo que contenga a 001 y 011.

Del mapa de Karnaugh de la figura 5 se obtiene la función lógica de menor costo de la salida:

$$Y = A' \cdot B + B' \cdot C$$

Se implementó esta función y se pudieron observar *hazards*, como muestras las figuras 6 y 7<sup>6</sup>.

<sup>4</sup>Según la hoja de datos proporcionada (consultado: 12/10/18).

<sup>5</sup>Imagen extraída de: <http://www.electronicengineering.nbcafe.in/hazards-in-digital-circuit/> (13/10/18)

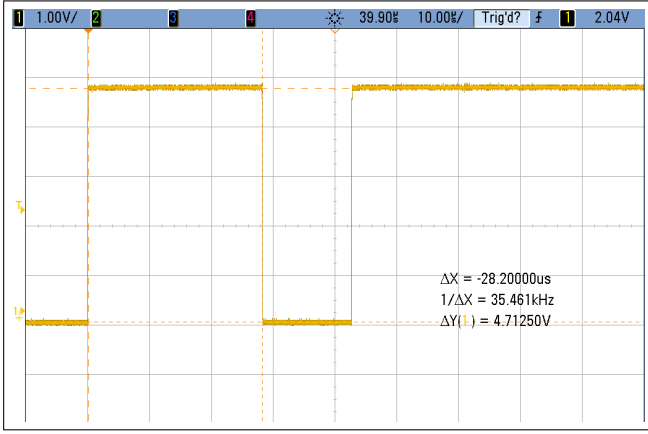


Figura 6: *Hazard* dinámico medido en la transición  $000 \rightarrow 001$ , donde la salida cambia de 0 a 1.

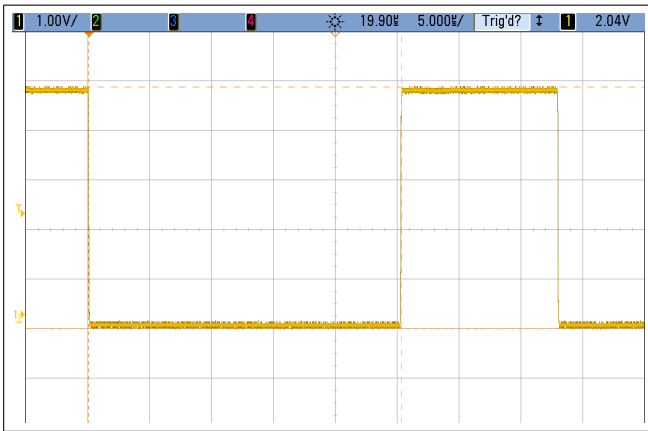


Figura 7: *Hazard* dinámico medido en la transición  $011 \rightarrow 111$ , donde la salida cambia de 1 a 0.

## Ejercicio 4

En esta sección, estudiaremos la variación de los parámetros característicos de una compuerta lógica al cargar su salida, y al introducir capacitores de desacople. A su vez, se observará el comportamiento del circuito a frecuencias altas ( $100\text{kHz}$ ), particularmente qué sucede con la tensión de alimentación. La compuerta que utilizaremos es una 74HC02, es decir una *nor* de tecnología CMOS.

Para observar el comportamiento del circuito al conectarle una carga, se armó el circuito que se encuentra ilustrado en la figura 8.

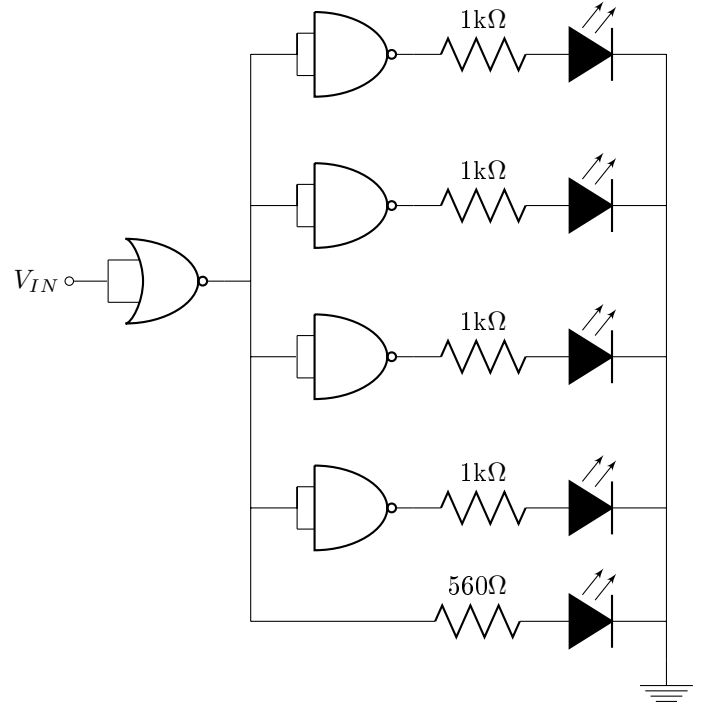


Figura 8: Esquema del circuito

Las compuertas *nand* utilizadas provinieron del integrado 74LS00, es decir que son de tecnología TTL. Si bien estas compuertas no son compatibles con las CMOS en principio, los rangos de tensiones que representan ceros o unos lógicos coinciden en esta configuración (como se discutió en el ejercicio 2).

Los tiempos característicos resultaron ser los siguientes:

	En vacío	Con carga
Propagación H-L	12	10
Propagación L-H	15	12
Rise time	85	180
Fall time	93	93

Tabla 3: Tiempos medidos (en nanosegundos)

Se observa que al estar cargado el circuito, los tiempos de propagación se reducen (aunque muy levemente), mientras que el *rise time* sube y el *fall time* se mantiene igual. Esto puede deberse a que alguna componente capacitiva parásita de la carga no permite que la señal varíe tan rápidamente.

A frecuencias bajas (de  $1\text{Hz}$ ), observando los LEDs se determinó que los niveles lógicos a las salidas eran los correctos, puesto que conmutaban aproximadamente dos veces por segundo, con la salida de las *nand* siendo la opuesta de la de las *nor*, y midiendo la tensión de salida de la com-

<sup>6</sup>Los *hazards* no se observaron todas las veces que se realizaron estas transiciones, sino esporádicamente.

puerta *nor* se verificó que la misma era la opuesta de la de la entrada.

Sin embargo, cabe destacar que el LED conectado directamente a la salida del 74HC02 iluminaba menos que los demás, y mucho menos que si el otro integrado no se conectaba. Esto indicaría que se está superando el *fanout* del integrado. Dado que las compuertas del 74LS00 son de tecnología TTL, el *switching* se hace con la corriente de base. Como la impedancia de entrada de un transistor BJT es mucho menor que la de un CMOS, esta corriente será mucho mayor que la tensión de *gate* de este último. Como las compuertas HC no están diseñadas para ser compatibles con compuertas TTL, esto puede estar generando problemas al trabajar en estas condiciones.

Al aumentar la frecuencia a  $100kHz$ , se observó que el 74HC02 levantaba temperatura, si bien no llegó a quemar al tacto incluso después de pasado más de un minuto. Midiendo la tensión de alimentación, se determinó que la misma seguía conservando su valor de 5V proporcionados por la fuente.

Al agregar un capacitor entre  $V_{CC}$  y tierra, se logró que el *ripple* de la fuente se reduzca: la tensión pico a pico de la alimentación descendió de 132mV a 80mV, es decir de un 2.64% a un 1.6% del valor de  $V_{CC}$ . Sin embargo, la temperatura del integrado no se vio afectada de manera apreciable.

El capacitor de desacople utilizado fue de 100nF, de acuerdo a lo recomendado por la *data sheet* del integrado de Texas Instruments<sup>7</sup>. El mismo se colocó lo más cerca posible del *pin* correspondiente a  $V_{CC}$  del integrado, con el objetivo de contrarrestar los efectos inductivos que puedan tener los cables, de forma tal que si el integrado pide un pico de corriente el mismo sea otorgado por el capacitor sin inducir una tensión en las bobinas parásitas.

## Ejercicio 5

En esta sección, el objetivo es estudiar el comportamiento de una compuerta *or* CMOS y de una *and* TTL, así como la interacción entre ambas. En particular haremos énfasis en qué ocurre cuando una de las entradas está flotante.

### Compuerta *and* TTL

Se utilizó una de las 4 compuertas *and* presentes en el integrado 74LS08, alimentándolo con  $V_{CC} = 5V$ . Para que

esta alimentación fuese más estable que la que proporcionan las fuentes del laboratorio de la universidad, se incluyó un capacitor de desacople de 100nF, conectado entre  $V_{CC}$  y tierra, y posicionado lo más cercano al integrado posible. Otra consideración que se tuvo para reducir el ruido fue conectar las entradas de las *gates* no utilizadas a  $V_{CC}$  a través de una resistencia de  $1k\Omega$ , a fin de evitar que la salida de las mismas oscile<sup>8</sup>.

La compuerta de interés se conectó con una entrada al nivel alto, es decir a  $V_{CC}$ , también pasando por una resistencia de  $1k\Omega$ .

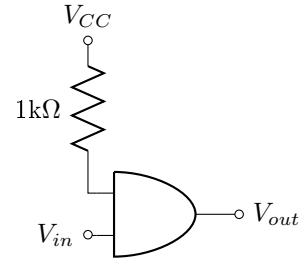


Figura 9: Conexión de la compuerta TTL

De acuerdo a las propiedades del álgebra de Boole,  $x \cdot 1 = x$ , y por lo tanto la salida debería estar en el mismo nivel lógico que la entrada (lo cual se verificó experimentalmente). Se observó que al dejar la entrada flotante, la salida se fijaba en el valor alto, midiéndose a la salida aproximadamente 4.4V. Realizando un DC *sweep*, se determinó que esto implica que la tensión flotante fue consistentemente mayor a 1.3V, puesto que para tensiones menores la salida sería 0.

### Compuerta *or* CMOS

El integrado utilizado fue el 74HC32. Al igual que para el LS, se alimentó con  $V_{CC} = 5V$ , utilizando un capacitor de desacople de 100nF. Las entradas de las compuertas no utilizadas, en cambio, se conectaron a tierra.

En este caso, se conectó una entrada a tierra y la otra se dejó flotante. Análogamente al caso anterior, como  $x + 0 = x$ , la salida debería ser igual a la entrada. Esto se pudo verificar cuando la entrada estaba fija en un valor, pero no así para el caso que se quiere estudiar.

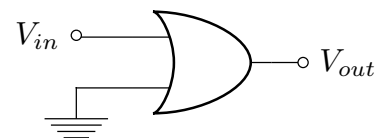


Figura 10: Conexión de la compuerta CMOS

<sup>7</sup>Fuente: <http://www.ti.com/lit/ds/symlink/sn74hc02.pdf> (consultado: 16/10/18)

<sup>8</sup>Esta práctica se recomienda en la nota de aplicación *Designing with TTL* de Fairchild Semiconductors (consultado: 05/10/18).

Cuando la entrada no se fijaba en ningún valor, lo observado era lo siguiente:

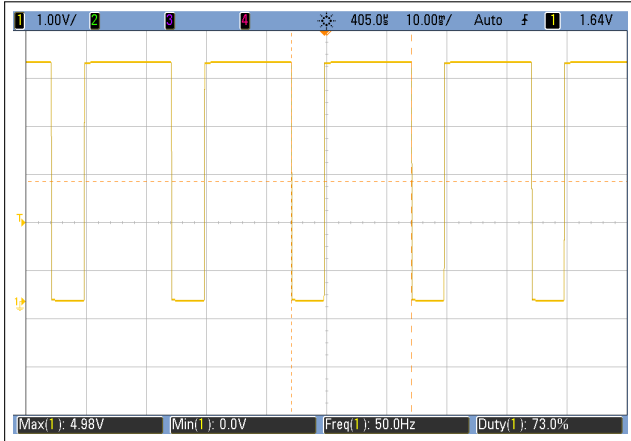


Figura 11: Salida del *or* CMOS con una entrada flotante

Esta señal cuadrada se debe a que la alta impedancia de la compuerta provoca que sea muy susceptible al ruido, y por lo tanto oscila con los 50Hz de la línea. Cabe aclarar que esta oscilación no se observaba siempre: moviendo los cables y tocando las conexiones se podían obtener tanto ceros como unos lógicos.

En cuanto al DC *sweep*, en este caso indicó que la tensión de entrada estaba oscilando entre tensiones menores y mayores a 2.2V, puesto que a partir de la misma la salida resultaba ser 1 y viceversa.

## Compuerta CMOS cargando a la compuerta TTL

Habiendo estudiado el comportamiento de cada compuerta por separado, se procedió a observar la interacción entre ambas, conectando la salida del *and* a la entrada del *or*.

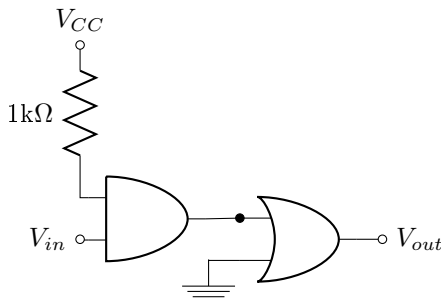


Figura 12: Conexión entre las compuertas TTL y CMOS

Por lo discutido anteriormente, idealmente la salida de este circuito tiene el mismo valor lógico que la entrada. Esto se verificó al fijar la tensión de entrada en un valor determinado, consistentemente observando el mismo valor

a la salida. Al dejar la entrada flotante, a la salida se observaba un 1 lógico. Esto es consistente con lo obtenido para la TTL aislada, puesto que si su salida cuando la entrada es flotante es 1, también lo será la salida de todo el circuito.

El DC *sweep* de esta configuración está también en línea con lo analizado hasta el momento. Si bien el valor de  $V_{in}$  para el cual la salida transiciona de 0 a 1 no es exactamente igual a los 1.3V que se obtuvieron para la TTL, si no que fue de 1.4V, la diferencia entre ambos es de un 7%, lo cual está dentro de un rango razonable.

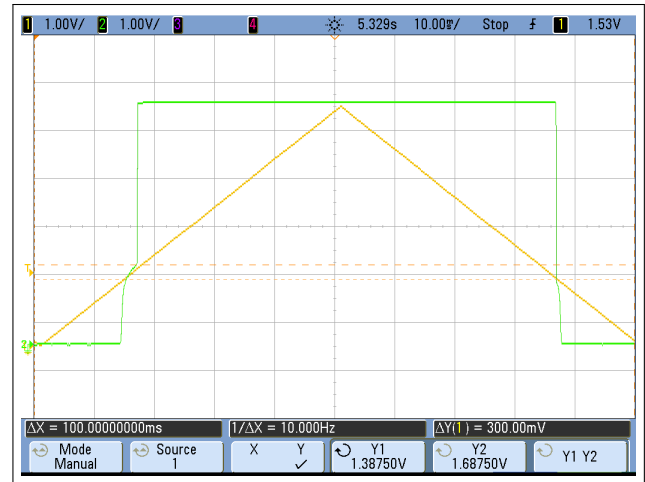


Figura 13: DC *sweep* del circuito TTL-CMOS

## Conclusiones

De acuerdo a los resultados obtenidos, las compuertas CMOS son más susceptibles al ruido que las TTL si se las deja flotantes. De no fijar la entrada de una CMOS, es probable que se induzca ruido proveniente de la línea. Si bien esto no se observó en la TTL, es una buena práctica fijar entradas no utilizadas a un valor lógico determinado, de acuerdo a la bibliografía consultada.

## Ejercicio 6

Se implementaron un latch SR y un flip flop D a partir de compuertas lógicas discretas. En ambos casos, se midieron sus parámetros característicos, y se compararon los resultados obtenidos con de sus equivalentes comerciales.



## Flip flop D

### Funcionamiento

El flip flop tipo D transfiere la entrada a la salida en cada ciclo de *clock*. Por ende, se lo puede utilizar como unidad básica de memoria.

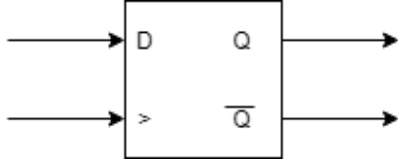


Figura 14: Representación del flip flop

clk	D	Q
↑	X	X
↓	X	$Q_{n-1}$

Tabla 4: Tabla de verdad del flip flop D

### Circuito lógico

Una posible implementación del flip flop D es la siguiente:

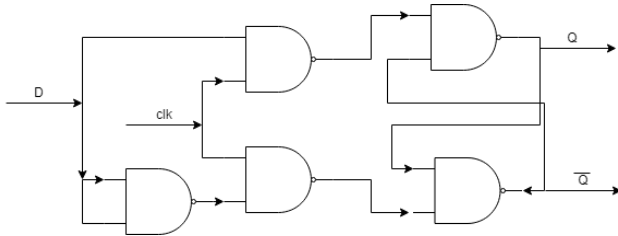


Figura 15: Circuito digital del flip flop D

### Implementación y medición

Se implementó el circuito indicado en la sección anterior con compuertas lógicas *nand*. Para ello, se utilizó el integrado 74HC00.

Del circuito previamente mencionado, se midió el tiempo de establecimiento, el *rise time* y el *fall time*.

clk	D	$Q_{n-1}$	$Q_n$	Tiempo de establecimiento
↑	0	1	0	25.2ns
↑	1	0	1	29ns

Tabla 5: Tiempo de establecimiento del flip flop D medido

Rise time	Fall time
45ns	49.6ns

Tabla 6: Rise time y fall time medidos para el flip flop D.

Para realizar la comparación se eligió el integrado comercial 74HC74. De su hoja de datos<sup>9</sup>, se obtuvieron los siguientes tiempos:

clk	D	$Q_{n-1}$	$Q_n$	Tiempo de establecimiento
↑	0	1	0	13ns
↑	1	0	1	18ns

Tabla 7: Tiempo de establecimiento del 74HC74 según su *data sheet*

Rise time	Fall time
< 400ns	< 400ns

Tabla 8: Rise time y fall time del 74HC74 según su *data sheet*

Como se puede observar, los tiempos de propagación del flip flop comercial son menores a los del que se implementó. Esto se puede deber a que el comercial hace un uso más eficiente de los transistores.

## Latch SR

### Funcionamiento

El latch SR posee tres entradas: una de *clock*, para poder sincronizar, otra de set (pone la salida a *high*) y de reset (salida en *low*).

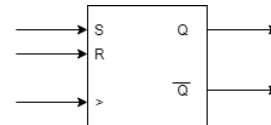


Figura 16: Representación de un latch

clk	S	R	Q
↑	0	0	$Q_{n-1}$
↑	0	1	0
↑	1	0	1
↑	1	1	Indeterminado

Tabla 9: Tabla de verdad latch SR

<sup>9</sup>Disponible en: <http://www.mouser.com/ds/2/308/74HC74-108792.pdf> (consultado: 14/10/2018).

## Circuito lógico

Una posible implementación del latch SR es la siguiente:

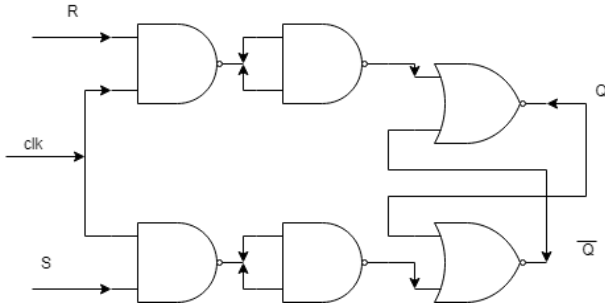


Figura 17: Circuito digital del latch SR

## Implementación y medición

Se implementó el circuito indicado en la sección anterior exclusivamente con compuertas lógicas *nand*, reemplazando las *nor* por un equivalente lógico con este tipo de compuerta. Se volvió a hacer uso, entonces, del integrado 74HC00.

Nuevamente, se midieron el tiempo de establecimiento, el rise time y el fall time.

clk	R	S	$Q_{n-1}$	$Q_n$	Tiempo de establecimiento
↑	0	1	0	1	21.6ns
↑	1	0	1	0	14ns

Tabla 10: Tiempo de establecimiento medido del latch SR

Rise time	Fall time
21.6ns	14ns

Tabla 11: Rise time y fall time medidos del latch SR

Para realizar la comparación se eligió el integrado comercial 74HC279. De su hoja de datos<sup>10</sup> se obtuvieron los siguientes tiempos:

clk	D	$Q_{n-1}$	$Q_n$	Tiempo de establecimiento
↑	0	1	0	20ns
↑	1	0	1	13ns

Tabla 12: Tiempo de establecimiento del 74HC279 según su *data sheet*

<sup>10</sup>Disponible en: <http://noel.feld.cvut.cz/hw/st/1937.pdf> (consultado: 14/10/2018).

Rise time	Fall time
< 400ns	< 400ns

Tabla 13: Rise time y fall time del 74HC279 según su *data sheet*

## Ejercicio 7

Se implementaron un contador síncronico y otro asíncronico con compuertas lógicas. Se compararon sus funcionamientos y se halló la máxima velocidad de operación de cada uno.

### Contador síncronico

Los contadores síncronicos son aquellos en los cuales todos sus flip flop cambian de estado al mismo tiempo. Esto se debe a que cada flip flop comparte el mismo *clock*.

#### Funcionamiento

Tal como se mencionó anteriormente, un contador asíncronico basa su funcionamiento en flip flops, particularmente de tipo T.

El circuito lógico correspondiente al contador síncronico de 3 bits es el siguiente:

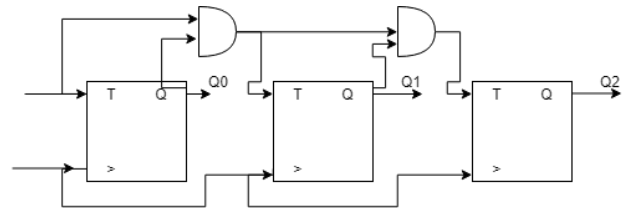


Figura 18: Circuito digital contador síncronico

Este contador es síncronico de tres bits ascendente (desde cero), donde Q0, Q1 y Q2 son las salidas del contador. La entrada del circuito es un pin de *enable* y otro de *clock*.

### Implementación y Medicion

Se implementó el circuito previamente mencionado con compuertas *and* (74HC00) y flip flops SR (CD4027). Junutando los terminales del SR se obtuvo el flip flop T.

En la siguiente imagen se observa el funcionamiento del contador:

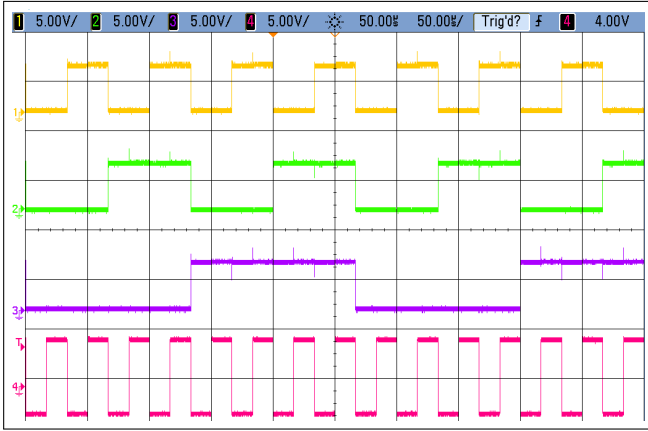


Figura 19: Funcionamiento del contador (*clock* en rosa, Q0 amarillo, Q1 verde, Q2 violeta)

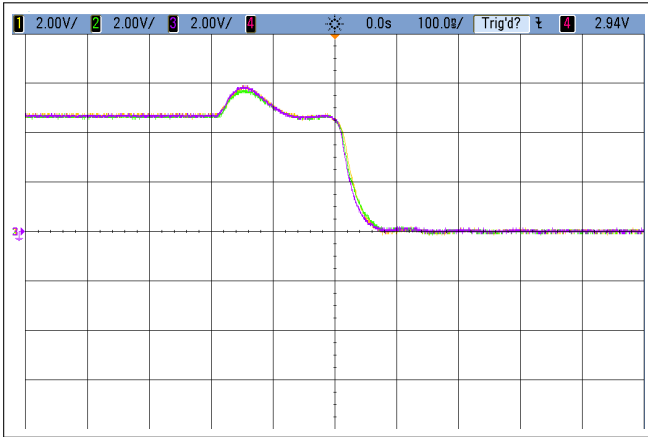


Figura 20: Contador pasando de 7 a 0 (*clock* en rosa, Q0 amarillo, Q1 verde, Q2 violeta)

En la figura 20 se observa cómo el cambio de estado en los tres flip flops ocurre de forma prácticamente instantánea.

### Máxima velocidad de conteo

La máxima velocidad a la que puede contar el circuito propuesto, depende de tres factores: el tiempo de establecimiento del flip flop, el de la compuerta *and*, y el tiempo de *set up* (es decir, el que se requiere para que el sistema se estabilice).

De esta manera queda definida la máxima frecuencia del clock:

$$f_{C_{MAX}} = \frac{1}{t_{eAND} + t_{eFlipflop} + t_{setup}}$$

## Contador asincrónico

Los contadores asincrónicos son aquellos en los cuales la señal de *clock* ingresa por uno de los flip flops y se propaga por el resto.

### Funcionamiento

El contador asincrónico basa su funcionamiento en flip flops T. A diferencia del contador síncrono, los flip flops no comparten *clock*, sino que se conectan en cascada.

El circuito lógico correspondiente es el siguiente:

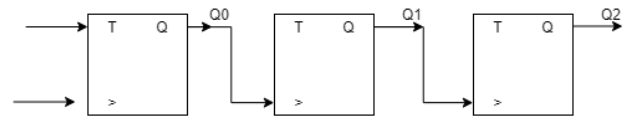


Figura 21: Circuito digital contador asincrónico

Este contador es asincrónico de tres bits ascendente (desde cero), donde Q0 a Q2 son las salidas del contador. La entrada del circuito es un pin de *enable* y otro de *clock*.

### Implementación

Se implementó el circuito anteriormente indicado, con flip flops SR (CD4027). Juntando los terminales del SR se obtuvo un flip flop T.

En la siguiente imagen se observa el funcionamiento del contador:

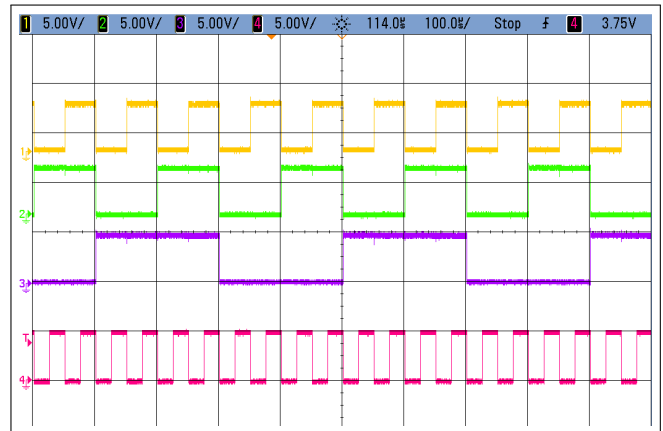


Figura 22: Funcionamiento del contador (*clock* en rosa, Q0 amarillo, Q1 verde y Q2 violeta)

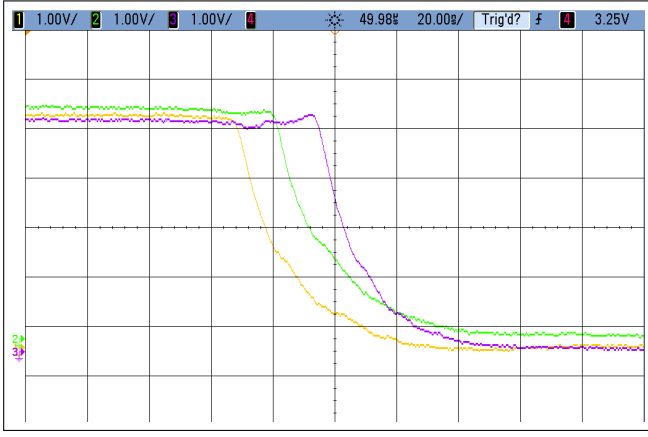


Figura 23: Contador pasando de 7 a cero (*clock* en rosa, Q0 amarillo, Q1 verde y Q2 violeta)

En la figura 23 se observa que los tiempos entre cada salida son mayores que los que se obtuvieron para el contador sincrónico, lo cual era lo esperado dado que el *clock* se debe propagar por cada flip flop.

### Máxima velocidad de conteo

A diferencia del contador sincrónico, en el asincrónico la velocidad máxima de operación depende del tiempo de establecimiento de cada flip flop. Por ende la máxima frecuencia de clock soportada es la siguiente (donde N es el número de flip flops utilizados):

$$f_{MAX} = \frac{1}{N \cdot t_{e_{flipflop}}}$$

## Ejercicio 8

Se propuso el diseño de un circuito que permita medir distancias utilizando un sensor ultrasónico de distancia (HC-SR04).

El circuito debe cumplir con los siguientes requerimientos:

- *trigger*: pin de entrada que dispara la medición en el flanco positivo.
- *trigger enable*: pin de entrada que habilita la señal de disparo.
- *meas*: 8 bits de salida que indican el tiempo medido en unidades de 100µs.

- *meas ready*: pin de salida que indica que la medición ha finalizado.

### Sensor HC-SR04

El sensor de distancia HC-SR04<sup>11</sup> es un sensor de distancia ultrasónico. Posee cuatro terminales: dos de alimentación (*VCC* y *gnd*), el *trigger* y *echo*.

El terminal de *trigger* acciona la medición. Para ello se debe cambiar el estado de este terminal a *high* por más de 10µs y, de esta manera, el sensor comienza a medir. Luego por el pin de *echo* se devuelve un pulso de ancho T µs. La duración de este pulso es proporcional a la distancia medida: Distancia =  $\frac{1\mu s}{58} [cm]$ .

El rango de medición del sensor es de 2cm a 4 metros. Por ende, el ancho del pulso devuelto por *echo* es de entre 116µs y 23200µs.



Figura 24: Sensor de distancia

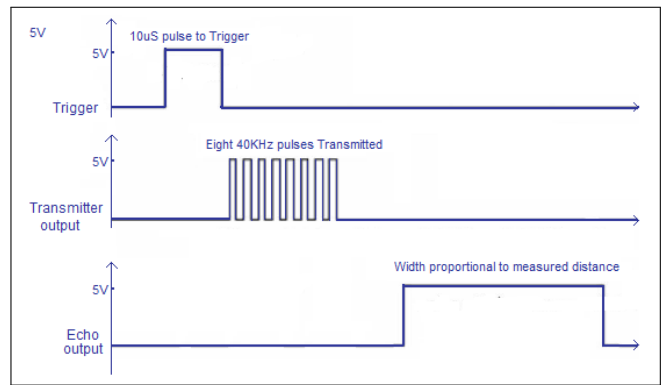


Figura 25: Diagrama temporal

### Circuito Implementado

Se modularizó el circuito de la siguiente manera:

<sup>11</sup>Datasheet del sensor: <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf> (consultado: 13/10/18).

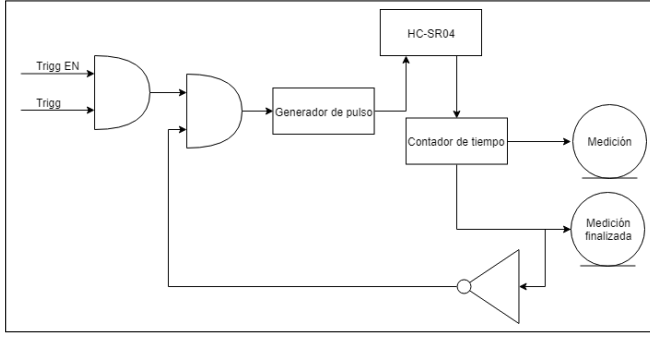


Figura 26: Diagrama de bloques

Los dos módulos principales, tal como se muestra en la imagen 26, son el generador de pulso y el contador de tiempo.

El generador de pulso, cuando recibe un flanco positivo, genera un pulso mayor a  $10\mu s$  para que el sensor comience a medir.

El contador de tiempo, por otra parte, mide el ancho del pulso devuelto por *echo* y lo devuelve en 8 bits. También se encarga de indicar que la medición finalizó.

### Generador de pulso

El generador de pulso se implementó con un LM555. El circuito utilizado fue el de la figura 27. Este circuito genera un pulso de ancho  $t = R_a C \cdot 1.1s$ . Para generar el disparo se debe generar un pulso menor que el configurado.

Como el sensor requiere un ancho de pulso mayor de  $10\mu s$ , se eligieron los componentes para un pulso de  $50\mu s$ , por ende el capacitor  $C = 10nF$  y  $R_a = 5k\Omega$

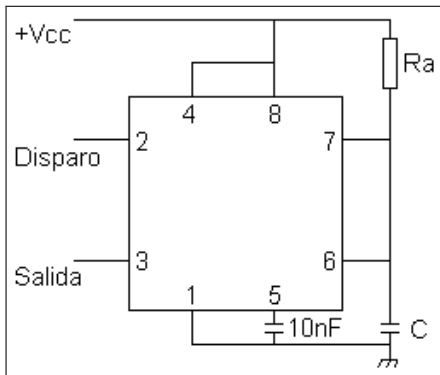


Figura 27: Circuito del generador de pulsos

Para evitar que al generador de pulso le ingrese una señal mayor a  $50\mu s$  se le colocó a la entrada (disparo) un RC con tiempo característico de  $10\mu s$ .

### Contador de tiempo

Para medir cuánto tiempo el pulso de *echo* está en *high*, se utilizó un contador de 8 bits y una señal cuadrada de 50% de *duty cycle* como *clock*.

El contador utilizado cuenta flancos positivos de *clock*. Por ende, para contar cuánto tiempo la señal de *echo* estuvo en *high*, se conectó la entrada de clock del contador, a la salida de una *and*. Las entradas de la *and* son la señal de *echo* y una señal cuadrada de  $10kHz$ . De esta manera, cada vez que ocurre un flanco de *clock* y la señal *echo* está en *high* el contador incrementa en una unidad. Debido a que la frecuencia de la señal cuadrada es de  $10kHz$ , cada unidad en el contador representa  $100\mu s$ .

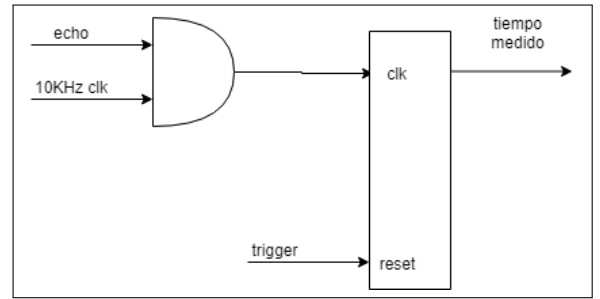


Figura 28: Contador de tiempo

En cuanto al reset del contador, se lo conectó al *trigger* del sensor. De esta manera, cada vez que se dispara una medición el contador vuelve a cero.

### Meas ready

En cuanto a la salida de *meas ready*, bastó con negar la señal de *echo*. Mientras no haya medición, la señal de *echo* se mantiene en *low* y por ende *meas ready* se mantiene en *high*. Cuando *echo* está en *high*, en cambio, quiere decir que se está midiendo, y *meas ready* se encuentra entonces en *low*.

### Prototipo

Previo al diseño final del circuito, se decidió construir un prototipo del mismo en una *protoboard*, para así probar el correcto funcionamiento de cada etapa y del conjunto.

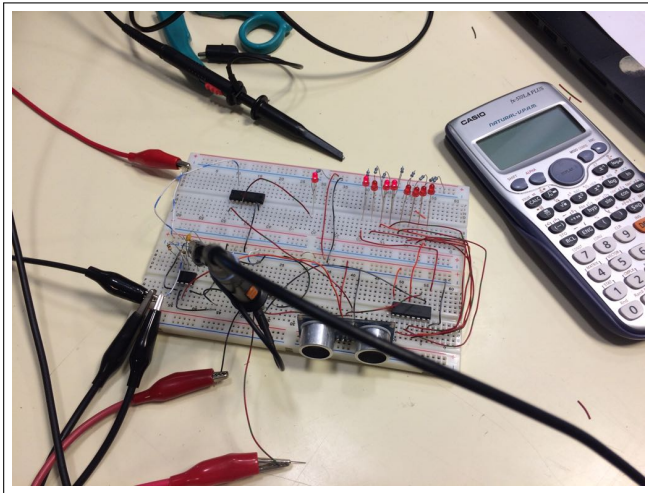


Figura 29: Prototipo del circuito

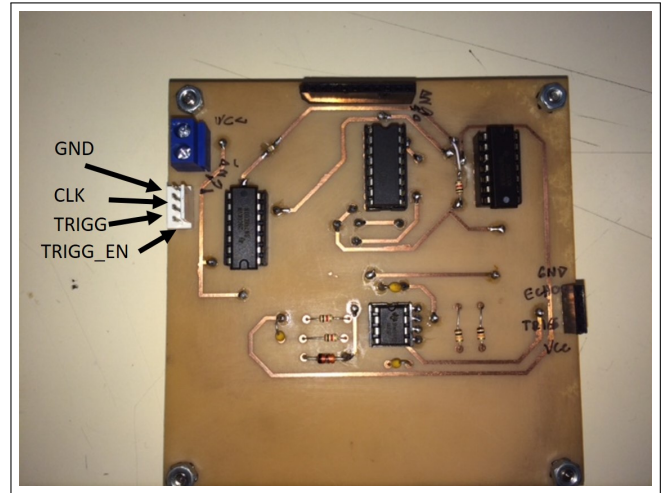


Figura 31: PCB

El prototipo funcionó correctamente, tal como se puede apreciar en [este video](#). Se procedió, pues, al armado de la placa.

## Circuito

El circuito final construido posee las siguientes características:

- Entrada de *trigger*, *trigger enable* y *clock*. El *clock* al que se lo debe conectar es de  $10kHz$  (señal cuadrada, 5V de amplitud y 50% de *duty cycle*).
- Salida 8 bits que indican el tiempo medido, y 1 bit que indica que la medición finalizó.

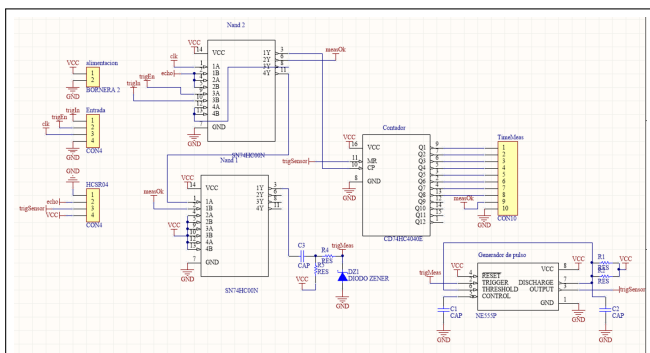


Figura 30: Esquemático

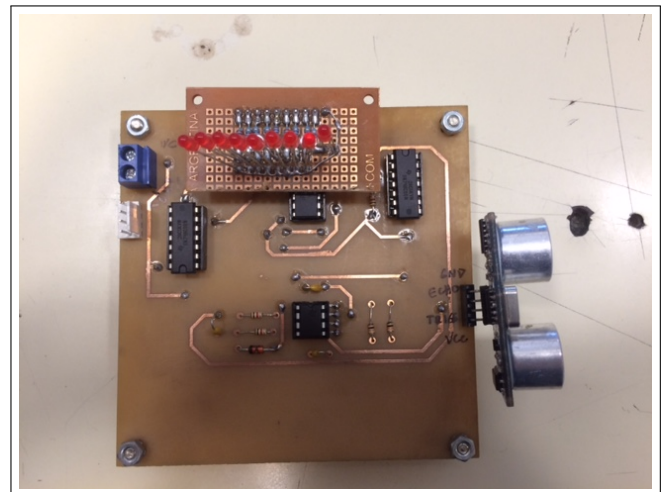


Figura 32: PCB con sensor y led para indicar la salida

## Simulación

Se simuló en Iverilog el circuito, y con GTKWave se visualizó su comportamiento en el tiempo.

Los módulos utilizados en la simulación son los mismos que los anteriormente descriptos, es decir hay un módulo encargado de generar un pulso y otro de contar el tiempo.

La figura 33 es la simulación temporal del circuito. "meas" es el resultado de la medición, "sEcho" es el pulso devuelto por el sensor, "sTrigger" es el pulso que el circuito le envía al sensor para comenzar la medición, "trigger" es el pulso que recibe el circuito para comenzar a medir, y "meas ready" indica que la medición finalizó.

Tal como se observa en la imagen, primero se dispara el *trigger*, posteriormente el circuito genera en *sTrigger* un pulso de ancho  $50\mu s$ . Después *sEcho* pasa a *high* (comienza a medir el sensor) y *meas ready* pasa a *low*. El contador

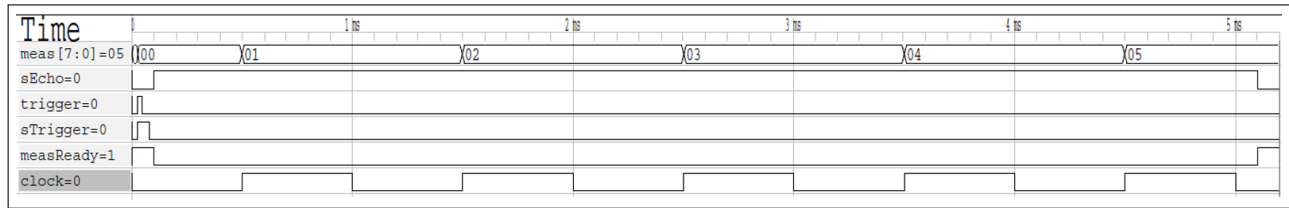


Figura 33: Simulación

comienza a contar cuántos pulsos de *clock* la señal de echo estuvo en *high*. Cuando sEcho pasa a *low* (la medición fi-

nalizó) y *meas ready* pasa a *high*. Como el ancho del echo es de 500µs, el contador llegó a 5.