

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.13 ELECTRONICA III

TRABAJO PRÁCTICO 2

Tecnología de circuitos lógicos
Multivibradores
Contadores secuenciales
Almacenamiento digital

Grupo 4:

Lisandro ALVAREZ 57.771

Milton DELGADO 56.451

Paulo NAVARRO 57.775

Matias FOGG 56.252

Profesores:

Kevin DEWALD

Pablo WUNDES

Realizado: 10/10/2018

Presentado: 17/10/2018

Corrección:

Ejercicio 2

Procedimos a alimentar una compuerta del integrado LS02 con la salida de una misma compuerta pero del integrado HC02, y luego alimentamos de la misma manera pero en sentido inverso. Pudimos notar que hay zonas donde el circuito armado no debería andar de forma óptima por el margen de ruido que manejan las distintas compuertas pero funciona igual ya que la caída de tensión en el LS02 no es tan grande y alcanza a caer cerca del límite del HC02 con 4,2V aproximadamente que es el mínimo del estado HIGH para el HC02. Si hubiera sido menor el valor de la tensión no hubiéramos obtenido alguna salida por lo marcado en las hojas de datos.

El **fan-out** está determinado por la cantidad de corriente que puede aceptar en la entrada cada CI y es la cantidad de pines que puede alimentar un CI con alguna de sus salidas. En la tecnología CMOS(HC) según su hoja de dato acepta 20mA como máximo, mientras que la tecnología TTL(LS) acepta 0,4mA como máximo en la entrada y 8mA en la salida. Haciendo las cuentas directas de estos casos, con la salida de un integrado LS puedo alimentar hasta 20 entradas LS, mientras que con un HC puedo alimentar 50 entradas LS. Cabe destacar que en la hoja de datos que brinda el fabricante solo asegura el funcionamiento de hasta 10 pines LS-TTL con una salida del HC, el cual debe ser para el peor caso que puede surgir para este integrado.

Al hacer las mediciones alimentando con el HCT y notamos un comportamiento mejor en cuanto la alimentación del LS02, ya que la tecnología HCT es a base de CMOS pero tiene una gran tolerancia con la tecnología TTL en cuanto a los valores de tensión.

Ejercicio 3

Se nos solicitó simplificar e implementar la siguiente tabla de verdad, y analizar las consecuencias de utilizar una metodología de menor costo.

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Se simplificó a través de Diagramas de Karnaugh:

a \ bc	00	01	11	10
	0	1	1	1
0	0	1	1	1
1	0	1	0	0

$$Y = \bar{B}C + \bar{A}B$$

$$Y = \overline{\overline{\bar{B}C + \bar{A}B}}$$

$$Y = (\overline{\bar{B}C}).(\overline{\bar{A}B})$$

Expresada la función lógica en esta forma, se puede realizar una implementación de menor costo utilizando solamente dos integrados, un integrado con compuertas NAND y uno con compuertas NOT. El diagrama del circuito logico resultante se muestra en la Figura 1

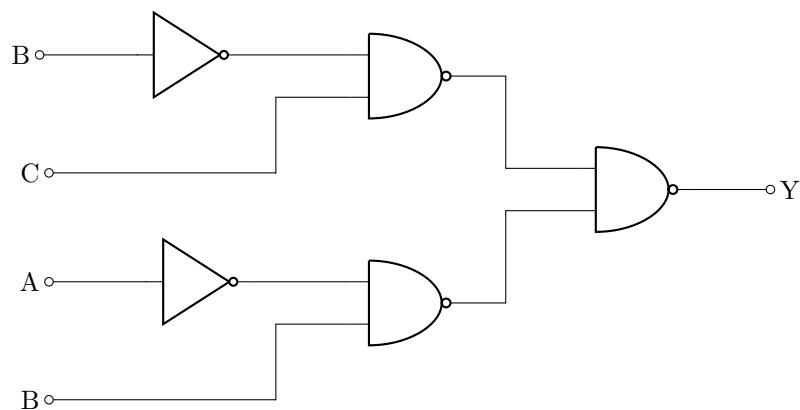


Figura 1: Diagrama del circuito implementado

El circuito implementado presenta una dificultad al transicionar entre miniterminos no adyacentes, pues no se pueden conmutar las entradas simultáneamente, entonces cuando uno esperaría ver en la salida un valor alto constante, observa una transición alto-bajo-alto. Si bien este pulso bajo que se observa es de muy corta duración, es suficiente para producir comportamientos inesperados en un eventual circuito lógico conectado a la salida de nuestra implementación.

Las señales de alimentación se togglearon entre el mintermino 3 y 5 con el siguiente código utilizando una placa experimental Arduino UNO;

```
int A = 5;
int B = 6;
int C = 7;
void setup() {
  // put your setup code here, to run once:
  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(C, OUTPUT);

  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);

  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);

  delay(1000);
}
```

Como se comentó anteriormente en cada transición del mintermino 3 al 5, se pasa por el mintermino 7, al cual le corresponde un 0 en la tabla de verdad, mientras que en el pasaje del mintermino 5 al 3, se transiciona a través del mintermino 1, al cual le corresponde un 1 en la tabla de verdad. El pulso observado en la Figura 2 corresponde al instante en que las entradas pasan por el mintermino 7.

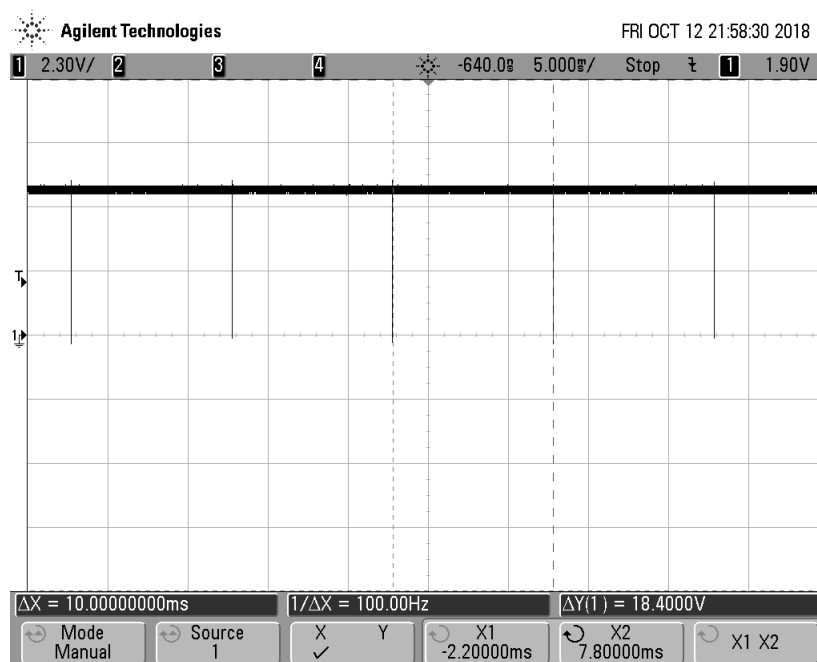


Figura 2: Respuesta a las transiciones mostradas

En la Figura 3 se muestra una vista ampliada del pulso negativo que se mostro en la Figura 2. El ancho del pulso es de $10\mu\text{seg}$, tiempo considerablemente largo como para producir cuambios inesperados en circuitos posteriores, como ya se explicó.

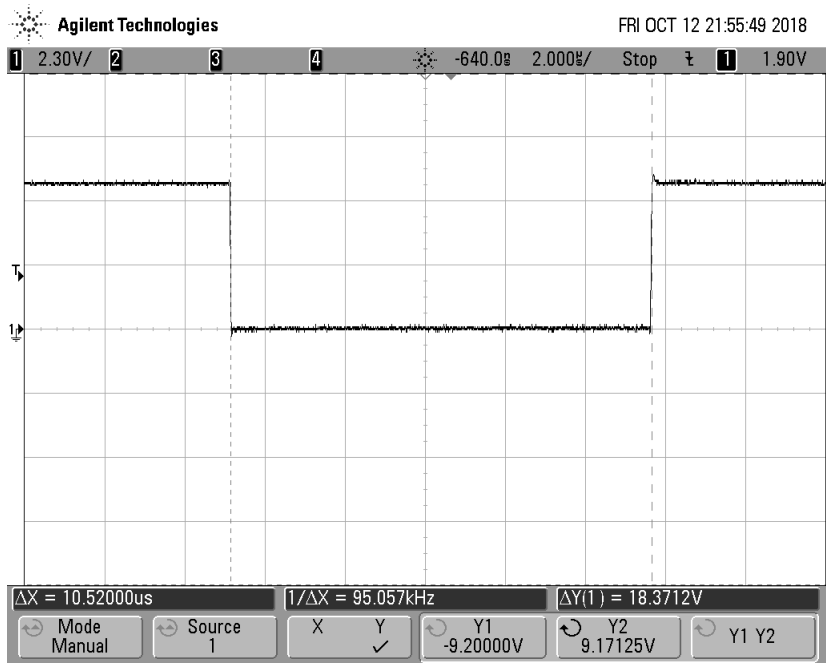


Figura 3: Ancho del pulso producido a la salida

Ejercicio 4

Se nos solicitó medir los tiempos de propagación, rise time y fall time de una compuerta 74HC02 en vacío, y luego repetir el experimento implementando el circuito mostrado en la Figura 4. La compuerta NOR es una 74HC02 y las NAND son 74HC00

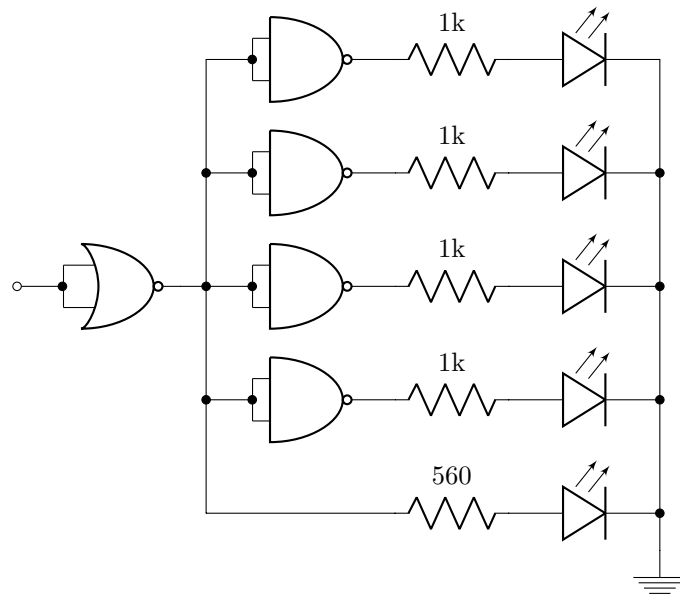


Figura 4: Circuito a implementar

A continuación se nos solicitó aumentar la frecuencia del generador de señales a 100kHz y medir la tensión de alimentación, y repetir la experiencia conectando capacitores de desacople en los terminales de alimentación de los integrados. Los resultados obtenidos para los tiempos de propagación, rise time y fall time de la compuerta 74HC02 en vacío fueron:

Tiempo de propagación	9.39 nseg
Rise time	9.93 nseg
Fall time	35.5 nseg

Cuando se implementó el circuito mostrado en la Figura 4, los valores obtenidos fueron:

Tiempo de propagación	12.7 nseg
Rise time	38.8 nseg
Fall time	32.9 nseg

Se observa que el tiempo de propagación y el fall time se ven ligeramente afectados. No así el rise time, el cuál es aproximadamente 4 veces mayor cuando se carga la salida de la compuerta NOR con el circuito mostrado.

Seguidamente se aumentó la frecuencia del generador de señales a 100kHz y se midió la señal de alimentación. La Figura 5 muestra unos pequeños sobrepicos que coinciden con los flancos de la señal de entrada. Estos sobrepicos se producen porque cuando las compuertas conmutan producen una demanda de corriente significativa, la cual es provista por la fuente de alimentación. La línea de alimentación de los integrados posee una impedancia despreciable cuando no hay variaciones de corriente en el circuito, sin embargo cuando conmutan las compuertas y las fluctuaciones en la corriente son repentinas, los efectos de la impedancia se manifiestan, y aparecen sobrepicos de tensión en la línea.

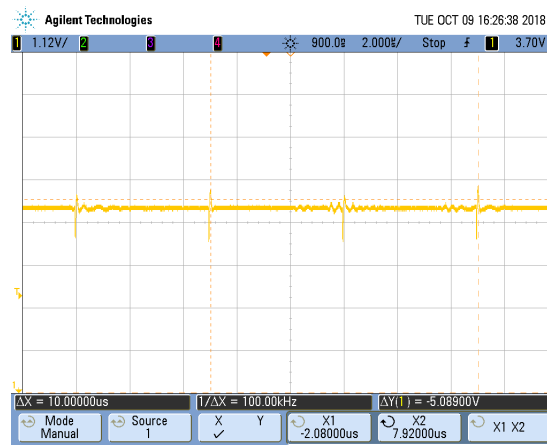


Figura 5: Tensión en el terminal de alimentacion del 74HC02 sin capacitor

Para suavizar este efecto, se colocaron, como indica la consigna, capacitores de desacople en los terminales de alimentación de los circuitos integrados. Como muestra la Figura 6, el efecto es contrarrestado de forma significativa. De esta forma, cuando el circuito demanda un flujo de corriente repentinamente, el capacitor provee la corriente al integrado y no se producen cambios repentinos de corriente en la línea de alimentación.

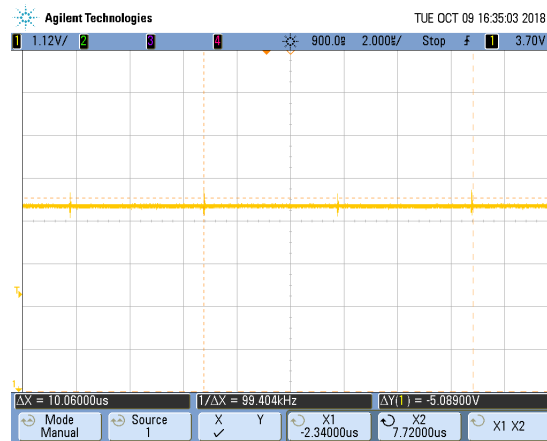


Figura 6: Tensión en el terminal de alimentacion del 74HC02 con capacitor

Ejercicio 5

Alimentamos una compuerta AND de tecnología TTL, integrado HD74LS08, conectando una entrada a la tensión V_{cc} y la otra al generador de funciones. También alimentamos una compuerta OR de tecnología CMOS, integrado SN74HC32N, conectando una entrada a tierra y la otra al generador de funciones. La alimentación de ambos integrados es de 6V, el máximo sugerido por ambos fabricantes. Analizando la salida podemos notar que la compuerta OR devuelve un pico de tensión como ya notamos en la tecnología CMOS y la compuerta AND devuelve solo 3,5V a la salida, ambos parámetros están dentro de lo normal según las hojas de datos de cada una. Luego procedimos a conectar las compuertas de la siguiente forma:

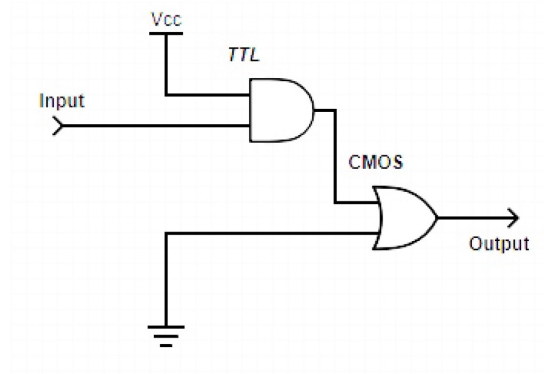


Figura 7: Circuito ejercicio 5

Midiendo en la salida de la OR esperaríamos que no hubiera salida o si la hubiera tuviera problemas, ya que la entrada mínima que acepta la OR es de 4,2V con la alimentación elegida y la AND otorga como ya mencionamos 3,5V. En nuestro caso el circuito lógico funcionó como si no hubiera ningún tipo de problema, creemos que esto puede deberse a que las hojas de datos otorgan datos donde el fabricante se asegura que sus componentes funcionen pero pueden haber algunos que funcionen fuera del alcance determinado por el fabricante, esto quiere decir que puede funcionar pero no está garantizado el buen funcionamiento. En caso de haber tenido algún error hubiéramos optado por utilizar dos transistores para poner un pull-up que se active con la baja salida de la AND y alimente con $V_{dd} = 5V$ la entrada de la OR como en el siguiente esquema.

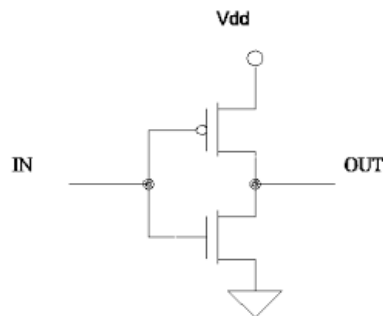


Figura 8: Circuito de Pull-Up

Ejercicio 6

Se nos pidió implementar un Flip-Flop D y un Latch SR a partir de compuertas lógicas discretas, medir sus parámetros que consideramos importantes para la caracterización de su funcionamiento y comparar con equivalentes comerciales. Para verificar las tablas características de funcionamiento de los dispositivos se utilizó una placa experimental Arduino UNO para programar las distintas configuraciones de señales de entrada, y se midieron señales de entrada y salida en un osciloscopio digital.

Latch SR

Para la implementación del Latch SR se siguió el diseño del *Gated SR Latch*, encontrado en la sección 5.2 de *Fundamentals of Digital Logic with Verilog Design*, el mismo se muestra en la Figura 9. El dispositivo se implementó utilizando compuertas NAND 74HC00, de tecnología CMOS.

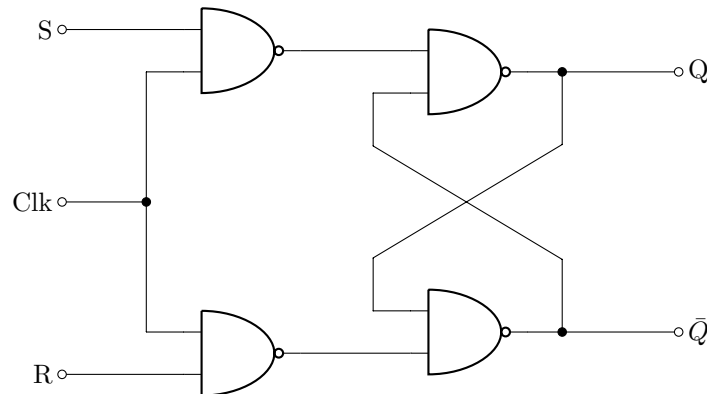


Figura 9: Latch SR implementado

Se analizó su correcto funcionamiento para las distintas configuraciones de señales de entrada, y se obtuvo la siguiente tabla característica:

Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$
1	0	0	$Q(t)$
1	1	0	1
1	0	1	0
1	1	1	?

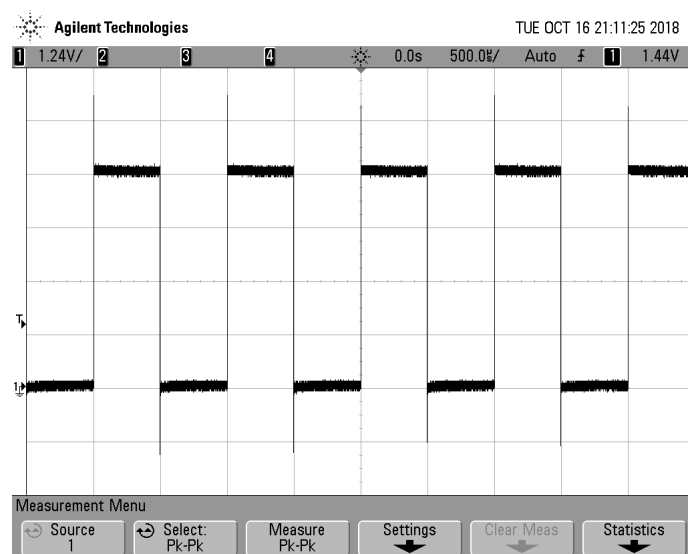


Figura 10: Estado inestable del Latch SR

Las respuestas obtenidas para las distintas configuraciones coinciden con los valores teóricos esperados. Sin embargo, como lo indica el análisis del circuito, la configuración de señales de entrada $Clk = 1$, $S = 1$ y $R = 1$ presenta una inestabilidad, en la cual el valor de salida $Q(t = 1)$ no se establece en ningún valor. Para esta configuración, se registró en la Figura 10 la forma de la salida.

En cuanto a los tiempos de respuesta de la compuerta, se midieron los tiempos de rise, fall y propagación, obteniéndose los siguientes valores:

Rise Time	27.6 nseg
Fall Time	26.8 nseg
Propagación	12.7 nseg

Se buscaron integrados comerciales de tecnología CMOS que implementen un Latch SR para comparar los parámetros medidos. El integrado CD4043 hallado es un integrado 3-state, motivo por el cual, quizás, difieran sus parámetros de los nuestros. Los tiempos de respuesta presentes en la hoja de datos del CD4043 son:

Rise Time	100 nseg
Fall Time	100 nseg
Propagación	150 nseg

Flip-Flop D

Para la implementación del Flip-Flop D, se siguió al igual que para el Latch SR el diseño presentado en la sección 5.3 y 5.4 de *Fundamentals of Digital Logic with Verilog Design*. Se utilizó la configuración de dos Latch D master-slave, para lograr un Flip-Flop D activado por flanco ascendente.

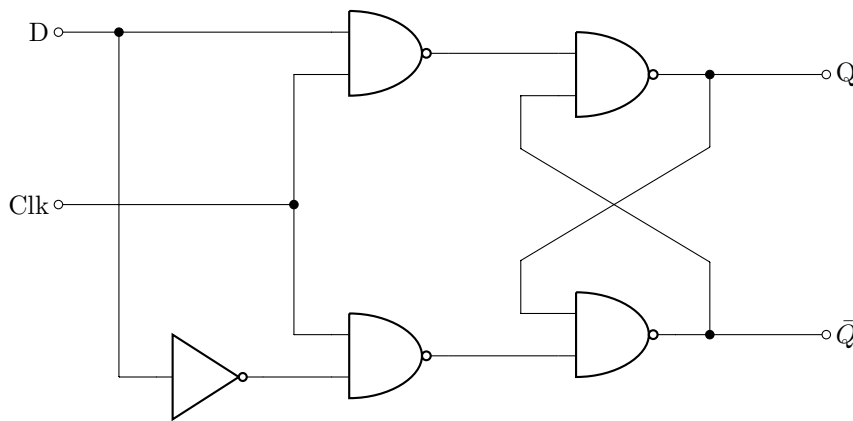


Figura 11: Modulo Latch D

En la Figura 11 se muestra la implementación de un módulo Latch D, el cual se utilizará, como se explicó para implementar el Flip-Flop D, ilustrado en la Figura 12

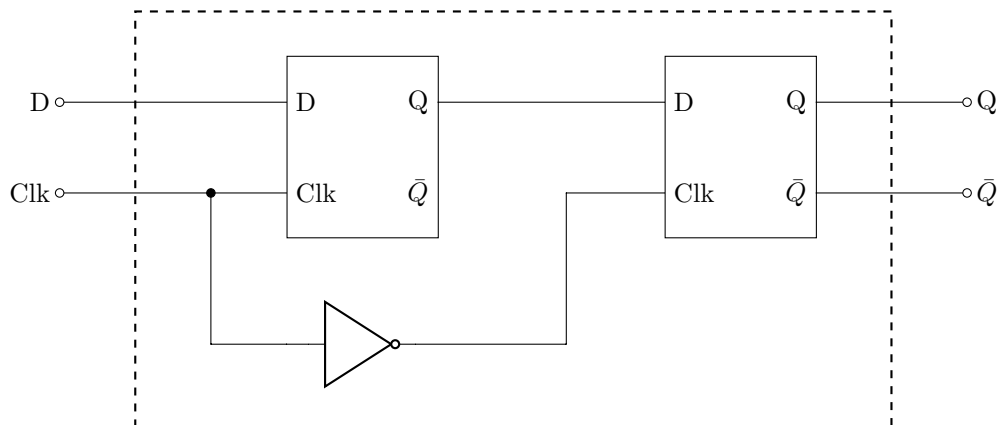





Figura 12: Diagrama Flip-Flop D

La respuesta de las salidas en función a las entradas fue la esperada, cumpliéndose la siguiente tabla característica de un Flip-Flop D de flanco ascendente:

Clk	D	Q(t+1)
	0	0
	1	1
	x	Q(t)

Se midieron los tiempos de rise, fall y de propagación. Se obtuvieron los siguientes resultados:

Tiempo de Propagación	18.8 nseg
Rise Time	27.8 nseg
Fall Time	27.6 nseg

Los resultados obtenidos se contrastan con los datos provistos por la hoja de datos del integrado comercial 74HC74, que implementa un Flip-Flop D. Los tiempos de propagación y de transición que se especifican son:

Tiempo de Propagación	17 nseg
Tiempo de Transición	7 nseg

La diferencia apreciada entre los tiempos de transición medidos y los especificados en la hoja de datos del integrado comercial, puede deberse a que los tiempos de transición de las señales provistas por el Arduino UNO son significativamente mayores a los tiempos de respuesta de las compuertas utilizadas para implementar el Flip Flop, por lo tanto esto induce un error en la medición.

Ejercicio 8

Trigger HC-SR04

El primer paso en el diseño del circuito fue implementar un bloque circuital que proporcione el pulso de trigger al sensor HC-SR04, el cuál, como indica en las especificaciones, debe ser de $10\mu\text{seg}$, sin embargo, habiéndose realizado pruebas utilizando una placa de experimentación Arduino UNO, se determinó que se puede disparar el sensor con pulsos de ordenes superiores de extensión. Esta observación tiene relevancia, ya que generar un pulso tan breve mediante un pulsador mecánico trae ciertas complejidades. Dicho esto, a continuación se explica el circuito utilizado para generar el pulso.

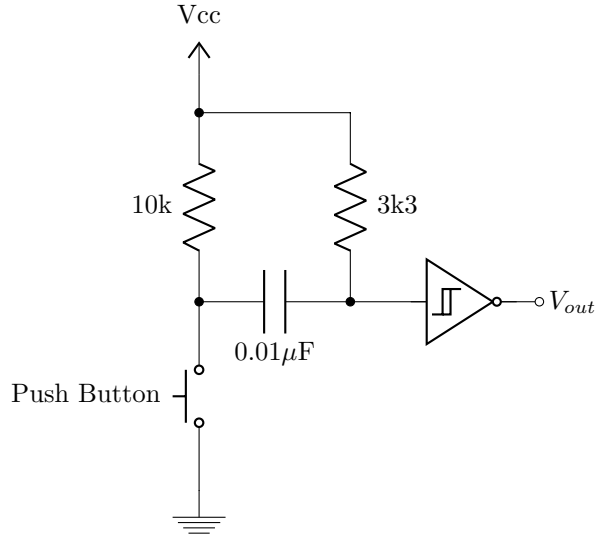


Figura 13: Circuito para generar pulso de 30mseg

Los valores del capacitor de $0,01\mu\text{F}$ y la resistencia de $3\text{k}3$ se determinaron al resolver el transitorio del circuito, y considerando los valores $V_{T_{MAX}}^+$ (Máximo valor en la entrada para el cuál tengo HIGH en la salida del inversor Schmitt trigger 74HC14) y el valor de V_{cc} para obtener un pulso de aproximadamente $30\mu\text{seg}$. La respuesta teórica aproximada que se obtiene en la entrada y la salida del inversor que se muestra en la Figura 13 se muestra en la Figura 14.

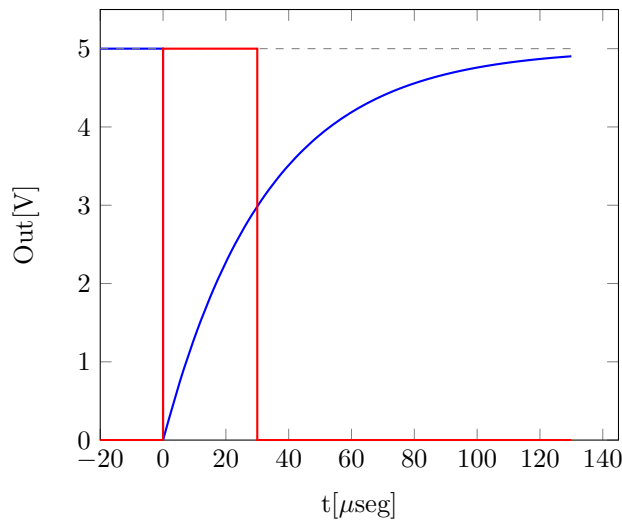


Figura 14: Respuesta del transitorio y la compuerta inversora

Por supuesto no debe enviarse la 'instrucción' de triggerear el HC-SR04 si el circuito no habilita a hacerlo, es decir, si el circuito esta procesando una medición anterior. Para esto, se utiliza una compuerta AND con una

entrada linkeada al registro que va a determinar si el circuito esta ocupado o listo para usarse y otra al pulso generado, explicado anteriormente. La lógica que determine si el circuito está o no ocupado se explicará mas adelante.

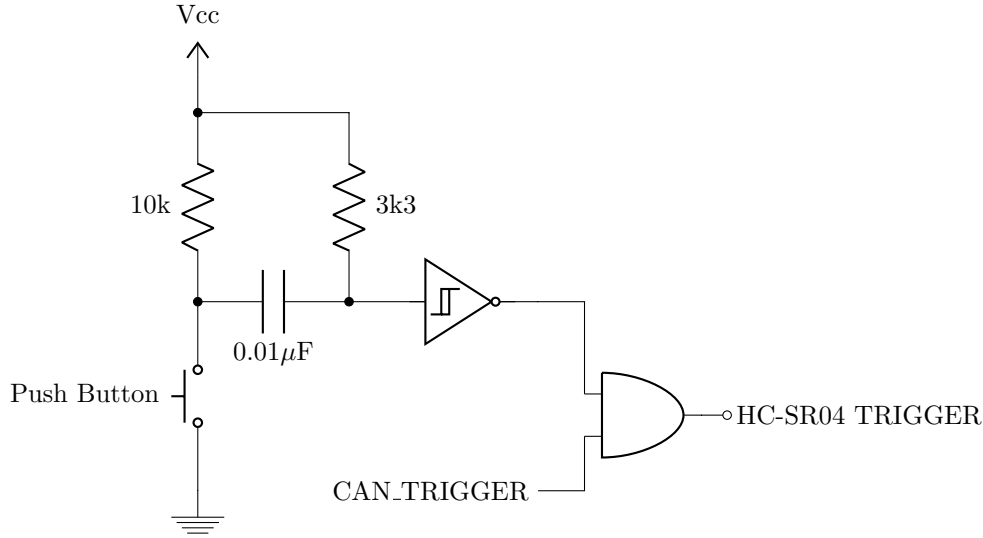


Figura 15: Circuito para triggerear HC-SR04

La variable CAN TRIGGER determina si se puede o no triggerear el sensor. Esta variable es función de dos variables, una es CIRCUIT READY e indica si el circuito está listo para utilizarse y la otra es TRIGGER_ENABLE, y habilita o deshabilita el trigger manualmente. El diagrama lógico que las vincula se muestra en la Figura 18:

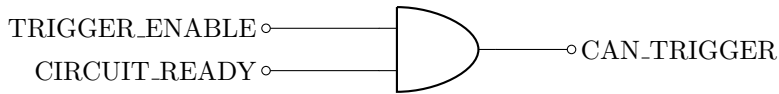


Figura 16: Logica a la entrada del contador

Diseño del Clock

Uno de los parámetros mas importantes en el diseño del circuito es elegir una apropiada frecuencia de clock, ya que esta nos va a determinar la medición máxima y mínima que nuestro circuito será capaz de realizar. Debido a que contamos con un contador de 8 bits, el numero máximo de 'clicks' que podremos contar está dado por $n_{max} = 2^8 - 1 = 255$. El fabricante del sensor HC-SR04 nos indinca que el *echo* generado, el cual nos indica la distancia medida por el sensor, está dado por:

$$\frac{\mu seg}{58} = cm$$

Si definimos:

$$\begin{cases} d : \text{distancia medida en cm} \\ n : \text{'ticks' de reloj contados} \\ T_{\mu} : \text{Período del clock en } \mu seg \end{cases}$$

Entonces podemos detrmnar los parametos a partir de la siguiente expresión:

$$\frac{nT_{\mu}}{58} = d$$

Se determinó como condición la distancia mínima a medir, la cual según el fabricante del sensor, es 2cm, y de esta forma se obtuvo que el período del clock T debe ser de $116\mu seg$. La distancia máxima que se podrá medir será de 255 cm. La frecuencia del clock será entonces $f \approx 8,6kHz$.

Para la implementación del clock se decidió utilizar un integrado 555 en modo astable. La configuración del mismo en el modo mencionado se muestra en la Figura 17. Las expresión, según la hoja de datos de los

integrados 555, para determinar el comportamiento del dispositivo en función de los componentes R_1 , R_2 y C son:

$$\begin{cases} t_H = \ln(2)(R_1 + R_2)C \\ t_L = \ln(2)R_2C \\ T = t_L + t_H \end{cases}$$

Trabajando con estas expresiones, se calcularon los valores de los componentes de forma que se obtenga un clock con los parámetros anteriormente mencionados, y se determinaron:

$$\begin{cases} R_1 = 3,35k\Omega \\ R_2 = 6,694k\Omega \\ C = 0,01\mu F \end{cases}$$

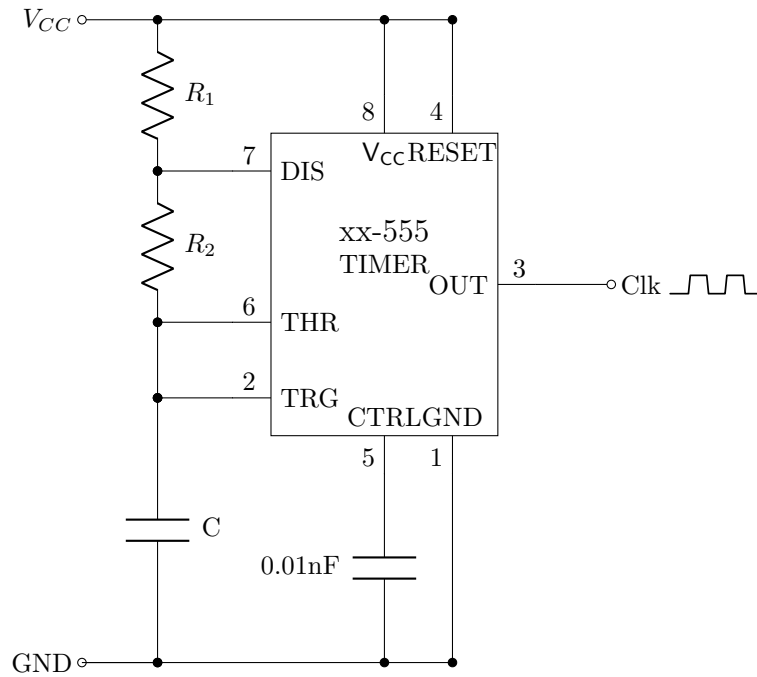


Figura 17: xx555 configurado en modo astable

Implementacion del contador

El integrado elegido para contar la extensión del pulso *ECHO* del sensor HC-SR04 fue un *74HC4040*, el cuál cuenta con 3 modos de funcionamiento, en función de los estados de sus tres entradas *CP*, *CLR* y , los mismos se detallan en la siguiente tabla:

CP	MR	MODO
$\overline{\text{L}}$	LOW	Hold
$\overline{\text{L}}$	LOW	Count
X	HIGH	Reset

Se pretende que en tanto el sensor devuelva un *ECHO*, el contador este activo, y mientras el sensor no devuelva *ECHO*, el contador no cuente. Dicho esto, la entrada *CP* será función del *Clk* y del pulso *ECHO*, como se muestra en la Figura 18

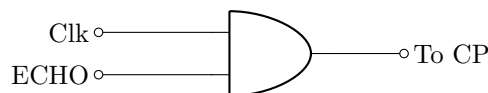


Figura 18: Logica a la entrada del contador

La entrada *MR* es controlada por un pulsador físico, de forma que esta ponga todas las salidas del contador en LOW, y prepare el circuito para una eventual nueva medición.

Reset Button

El usuario al pulsar el boton de reset, además de resetear el contador, toggleará un Flip-Flop que almacenará el estado de la variable CIRCUIT READY, ilustrada en la Figura 15, para dejar el circuito habilitado para una nueva solicitud de medición.