

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

Se tomaron los datos de máximos y mínimos de las entradas y salidas para los distintos estados de las hojas de datos de los circuitos en cuestión, y se exhiben en el cuadro 1.

	Vcc	Voltage (25 C)	Vcc	Voltage (25 C)	Vcc	Voltage (25 C)
	74HC02		74HCT02		74LS02	
Minimum HIGH Level Input Voltage	2.0V 4.5V 6.0V	1.5V 3.15V 4.2V	4.5V a 5.5V	2V	4.75V	2V
Maximum LOW Level Input Voltage	2.0V 4.5V 6.0V	0.5V 1.35V 1.8V	4.5V a 5.5V	0.8V	4.75V	0.8V
Minimum HIGH Level Output Voltage	2.0V 4.5V 6.0V	1.9V 4.4V 5.9V	4.5V a 5.5V	4.4V	4.75V	2.7V
Maximum LOW Level Output Voltage	2.0V 4.5V 6.0V	0.1V 0.1V 0.1V	4.5V a 5.5V	0.1V	4.75V	0.5V

Cuadro 1: Tabla de información obtenida de las hojas de datos

Como se ilustra en las figuras para los cuatro casos planteados, sólo habría problemas si se intenta conectar un transistor HC a la salida de un LS: la salida alta del LS puede caer en la región indeterminada de la entrada del HC, y sin que falle ningún componente fallaría el circuito.

En todos los demás casos se genera un margen de error para las entradas de los transistores.

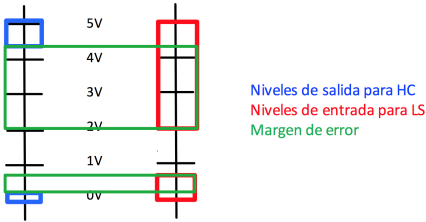


Figura 1: Niveles de tensión para caso HC alimenta a LS.

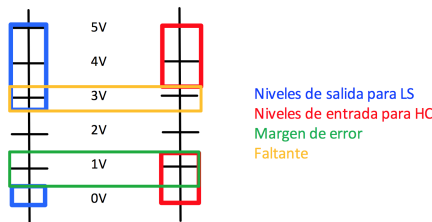


Figura 3: Niveles de tensión para caso LS alimenta a HC.

Figura 2: Circuito utilizado

Figura 4: Implementación con 74HC112

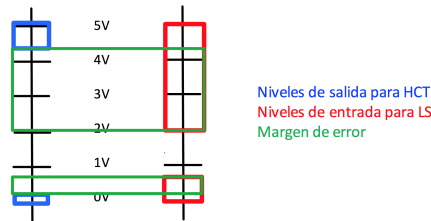


Figura 5: Niveles de tensión para caso HCT alimenta a LS.

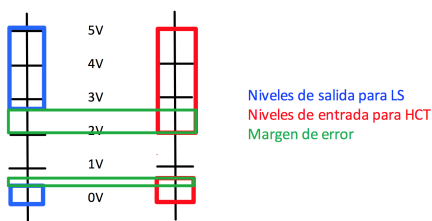


Figura 7: Niveles de tensión para caso LS alimenta a HCT.

Figura 6: Circuito utilizado

Figura 8: Implementación con 74HC112

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 habríamos 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.

OBSERVACIÓN: Haciendo zoom en las señales se puede observar que la tecnología TTL siempre otorga una tensión más baja de lo esperado, esto se debe a que internamente contiene una resistencia que limita la corriente que circulará por el transistor pero provocará una caída de tensión la cual se ve reflejada a la salida. Por otra parte la tecnología CMOS tiene un efecto capacitivo interno que provoca una pequeña oscilación cuando responde al escalón. Aquí dejamos unas imágenes mostrando los efectos mencionados.

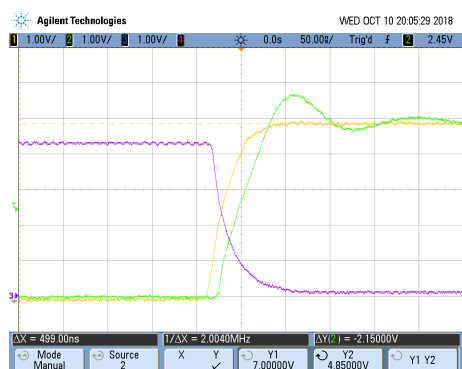


Figura 9: LS alimentando al HC

Figura 10: Circuito utilizado

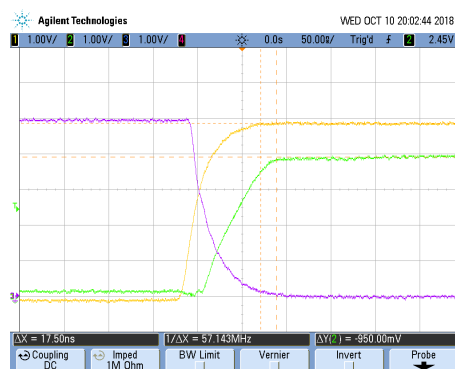


Figura 11: HC alimentando al LS

Figura 12: Implementación con 74HC112

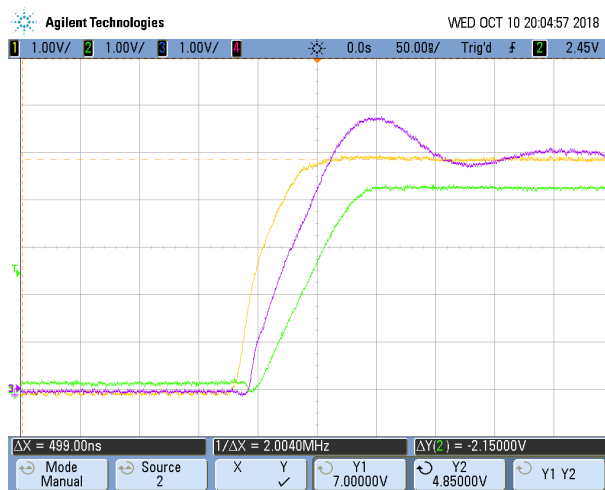


Figura 13: Ambas señales sobrepuestas

En las imágenes anteriores la señal amarilla es la de entrada, la cual corresponde a una onda cuadrada que va de 0V a 5V. La señal verde es la de salida y la violeta es la señal intermedia (la señal que esta entre compuerta y compuerta). Mientras que en la tercer imagen están sobrepuestas ambas señales de salida, donde la amarilla corresponde a la de entrada, la señal verde a la salida del 74LS02 y la señal violeta es la salida del 74HC02.

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

Figura 14: Tabla de verdad dada

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

Figura 15: Mapa de Karnaugh

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 17

El circuito implementado presenta una dificultad al transicionar entre minterminos 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);
}
```

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

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

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

Figura 16: Función Lógica

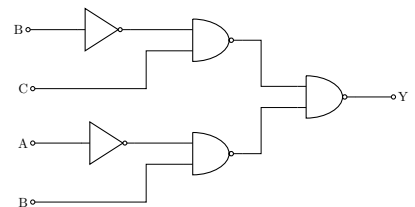


Figura 17: Diagrama del circuito implementado

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 18 corresponde al instante en que las entradas pasan por el mintermino 7.

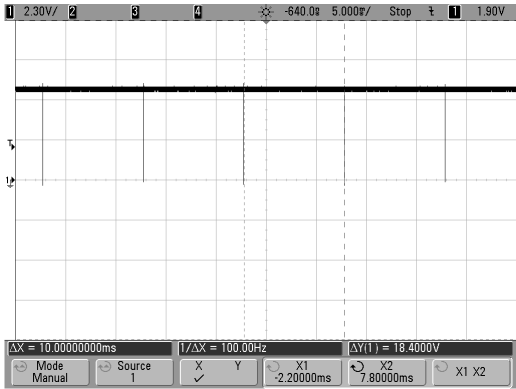


Figura 18: Respuesta a las transiciones mostradas

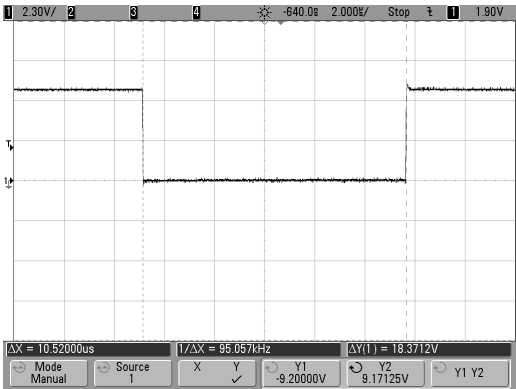


Figura 19: Ancho del pulso producido a la salida

En la Figura 19 se muestra una vista ampliada del pulso negativo que se mostró en la Figura 18. El ancho del pulso es de $10\mu\text{seg}$, tiempo considerablemente largo como para producir cambios inesperados en circuitos posteriores, como ya se explicó.

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 20. La compuerta NOR es una 74HC02 y las NAND son 74HC00.

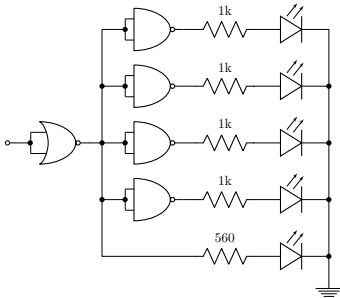


Figura 20: Circuito a implementar

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 20, 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 21 muestra unos pequeños sobrepicos que coinciden con los flancos de la señal de entrada. Estos sobrepicos se poducen 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.

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 22, 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.

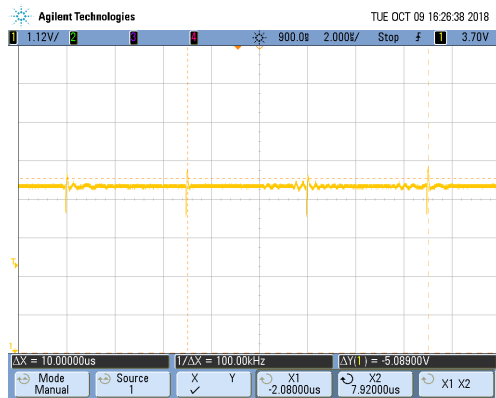


Figura 21: Tensión en el terminal de alimentación del 74HC02 sin capacitor

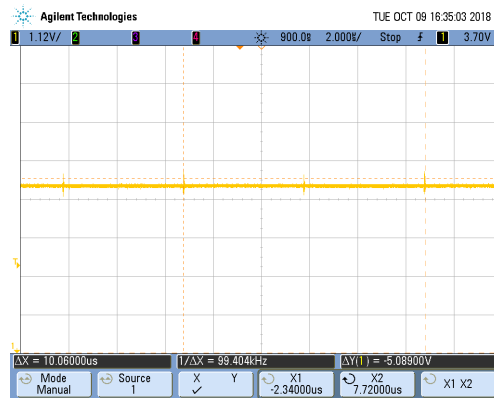


Figura 22: Tensión en el terminal de alimentación del 74HC02 con capacitor

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:

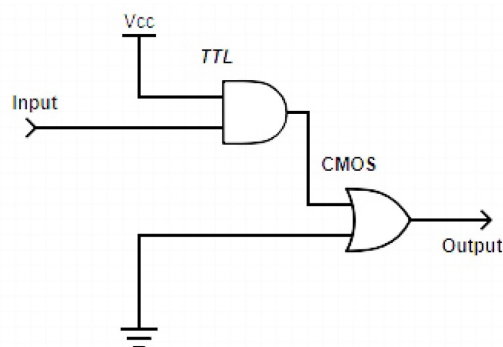


Figura 23: Circuito ejercicio 5

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.

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

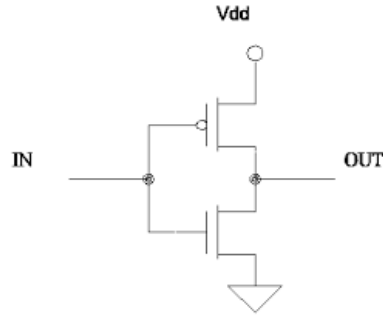


Figura 24: Circuito de Pull-Up

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 25. El dispositivo se implementó utilizando compuertas NAND 74HC00, de tecnología CMOS.

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	?

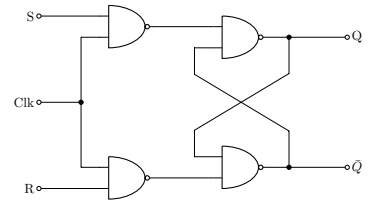


Figura 25: Latch SR implementado

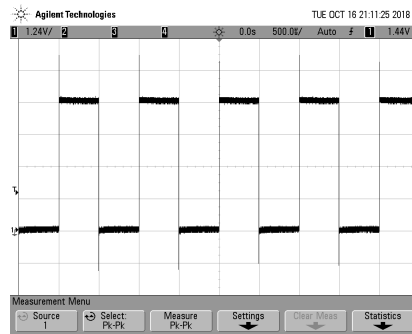


Figura 26: Estado inestable del Latch SR

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 26 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.

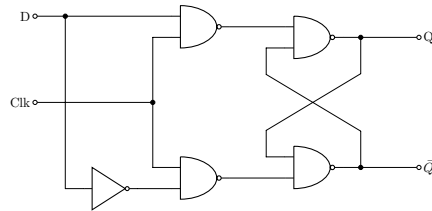


Figura 27: Módulo Latch D

En la Figura 27 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 28

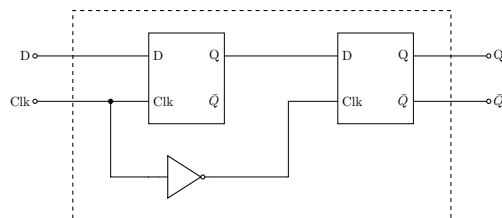


Figura 28: Diagrama Flip-Flop D

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 7

Contador Asincrónico

Para el contador asincrónico se siguió el esquema mostrado en la Figura 29. Se implementó utilizando Flip-Flops JK mediante integrados 74HC112, como puede observarse en la Figura 30. Utilizamos una señal cuadrada

otorgada por el generador de señales, de 5V con un ciclo de trabajo del 50 %. En particular nuestro contador va de 7 a 0, cuenta hacia atrás, trabaja con flancos descendentes del Clock para el bit menos significativo y con flancos ascendentes para los últimos 2 bits.

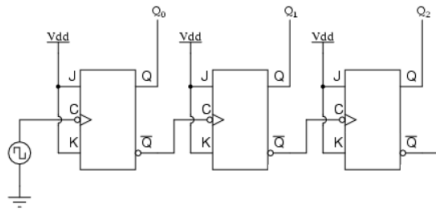


Figura 29: Circuito utilizado

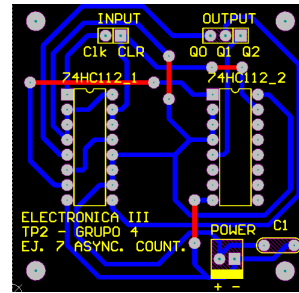


Figura 30: Implementación con 74HC112

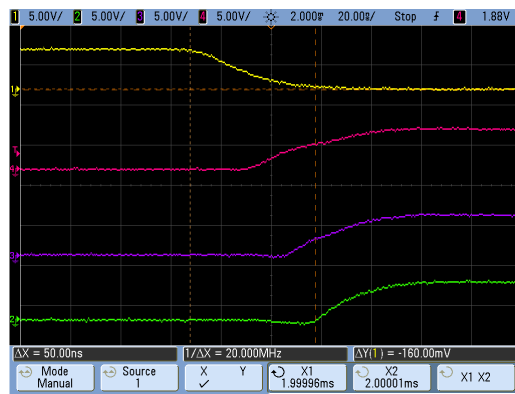


Figura 31: Medicion del delay acumulado

Las salidas no cambian al mismo tiempo, hay un delay entre que la señal conectada a la entrada clock del flip-flop cambia hasta que la salida correspondiente también lo hace, entonces tenemos un delay total más grande a mayor cantidad de flip-flops conectados en cascada, habrá un mayor retraso entre que la señal del clock cambia y lo hace el bit más significativo. Esto es una desventaja de estos contadores porque hay que tener cuidado con la velocidad con la que trabaja el circuito ya que esta no puede superar la frecuencia que se obtiene luego de medir el máximo delay acumulado por todas las compuertas, también sucede con estos contadores que pasamos por estados falsos hasta llegar a cambiar el bit más significativo pero cambia tan rápido que esto no suele ser un problema grave para muchas aplicaciones.

Medimos el tiempo de propagación total desde que la señal del Clock comienza a descender hasta que el bit más significativo (Q_2) comienza a elevarse y obtuvimos un valor de 50 ns (ver Figura 31), observamos el datasheet y el fabricante nos dice que trabajando a

5V y a 25 °C el tiempo de propagación de cada flip-flop es de 17 ns, con las 3 compuertas tendríamos 51 ns por lo que comprobamos que lo medido es aproximado al valor teórico esperado. En las Figuras 32,33,34,35 podemos ver el comportamiento del circuito a medida que aumentamos la frecuencia del Clock, a grandes frecuencias podemos ver como las señales medidas empiezan a tener un comportamiento extraño que puede deberse a que el ancho de banda del osciloscopio que utilizamos no era el suficiente y que este está interviniendo en las mediciones, pero a grandes rasgos podemos notar que a pesar de todo se mantiene entre los valores que el integrado detecta como 1 y como 0 según sea el caso, pero al llegar a 23.8 MHz podemos ver que los valores de las salidas tooglean, varían entre 0 y 1 y esto se debe a que ya mi tiempo de delay es mayor al periodo de la señal con la que estoy trabajando, al mirar el datasheet observamos que el fabricante nos dice que en las condiciones que trabajamos no debemos de pasar los 70 MHz, y esto dividido entre los 3 flip-flops me da 23.33 MHz.

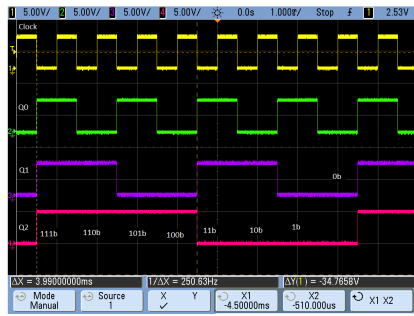


Figura 32: Comportamiento a bajas frec



Figura 33: Comportamiento a 11.9 MHz

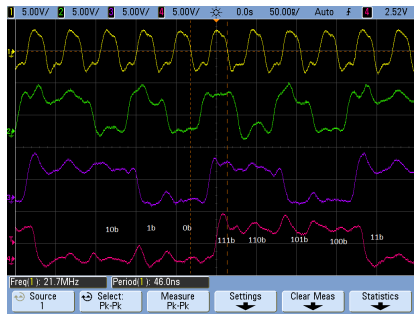


Figura 34: A 21.7 MHz

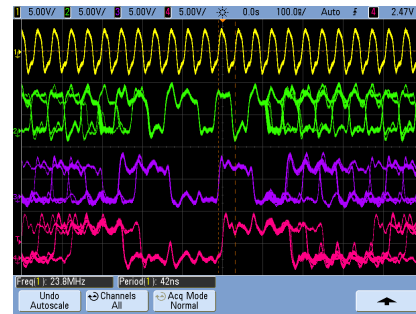


Figura 35: A 23.8 MHz

Contador Sincrónico

Este contador que avanza de forma ascendente. La diferencia con el anterior es que los clock de cada Flip-Flop estarán conectados directamente a la misma señal en lugar de usar la señal del flip-flop anterior, se puede observar en la Figura 36. Este cambio hará que cada parte se active con el flanco descendente de la misma señal, reaccionando todas las partes del contador a un mismo tiempo determinado, siempre y cuando se trate del mismo componente o tengan las mismas características temporales, y no obtener un arrastre en el delay final.

Al mirar el tiempo de propagación de este circuito (Figura 37) obtenemos que solo hay 24ns de delay, 17ns correspondiente al flip-flop (ahora no se suman sus delay porque son independientes) y 7ns correspondientes a la compuerta AND del circuito, esto quiere decir que recién a los 42,66MHz el circuito empezará a togglear porque el tiempo de delay superará la frecuencia de trabajo al igual que en el caso anterior. No pudimos obtener imágenes de este caso porque se necesita tener un osciloscopio de más de 200MHz debido a que si el instrumento no tiene un ancho de banda 5 veces mayor a la frecuencia de trabajo se perderán detalles de la señal que impiden tener una medición correcta.

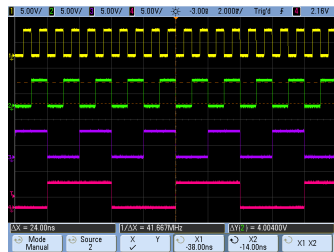


Figura 36: Comportamiento a bajas frec

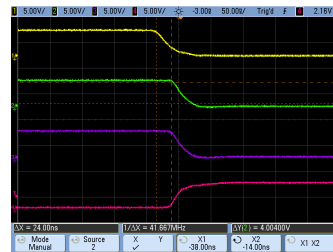


Figura 37: Medición del tiempo de propagación

Ejercicio 8

Se nos pidió diseñar e implementar un circuito capaz de medir una distancia utilizando el sensor ultrasonico HC-SR04. En las secciones posteriores se detallan los aspectos que hemos considerado mas relevantes en el proceso de diseño del dispositivo.

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.

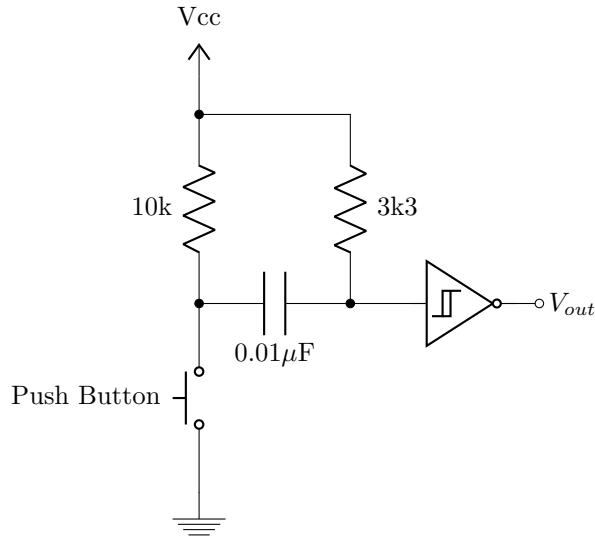


Figura 38: Circuito para generar pulso de 30mseg

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_{TMAX}^+ (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 38 se muestra en la Figura 39.

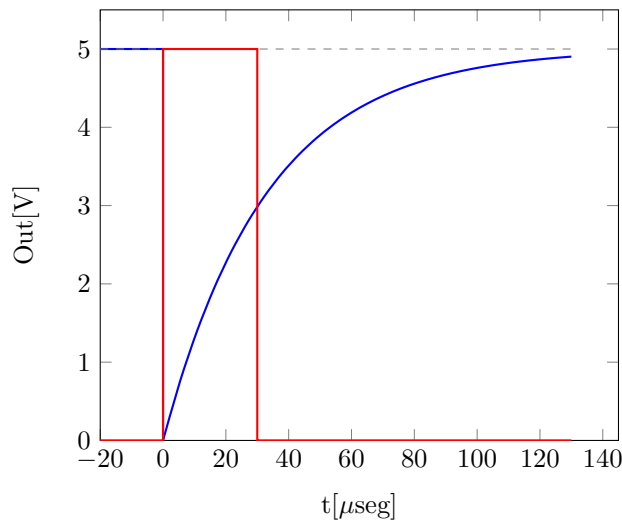


Figura 39: Respuesta del transitorio y la compuerta inversora

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 linkada 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.

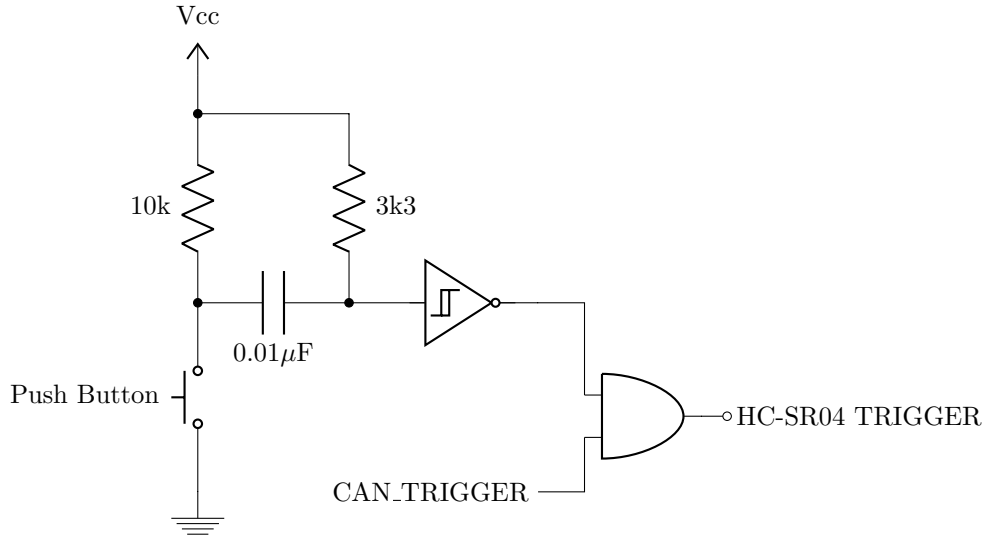


Figura 40: Circuito para triggerear HC-SR04

La variable CAN TRIGGER determina si se puede o no triggerear el sensor. Esta variable es función de dos variables, una es READY e indica si el circuito está listo para utilizarse y la otra es TRIGENABLE, y habilita o deshabilita el trigger manualmente. El diagrama lógico que las vincula se muestra en la Figura 43:

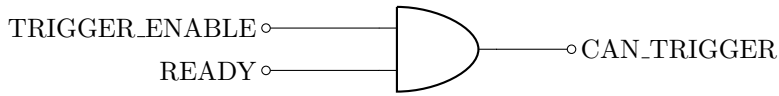


Figura 41: Logica a la entrada del contador

Diseño del Clock

Uno de los parámetros mas importantes en el diseño del circuito es elegir una apropiada frecuencia de clock, ya que esta será la que controle al contador. Implelentar un buen diseño de clock se traducirá en una medición más exacta.

La consigna determina que la salida debe expresar la extensión del pulso ECHO en unidades de $100\mu seg$, por lo tanto, el contador debe contar un click, por cada $100\mu seg$ de pulso. Por lo tanto se determiina que el período de la señal de clock debe ser $T = 100/\mu seg$

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 42. 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:

$$\begin{cases} R_1 = 2,885k\Omega \\ R_2 = 5,771k\Omega \\ C = 0,01\mu F \end{cases}$$

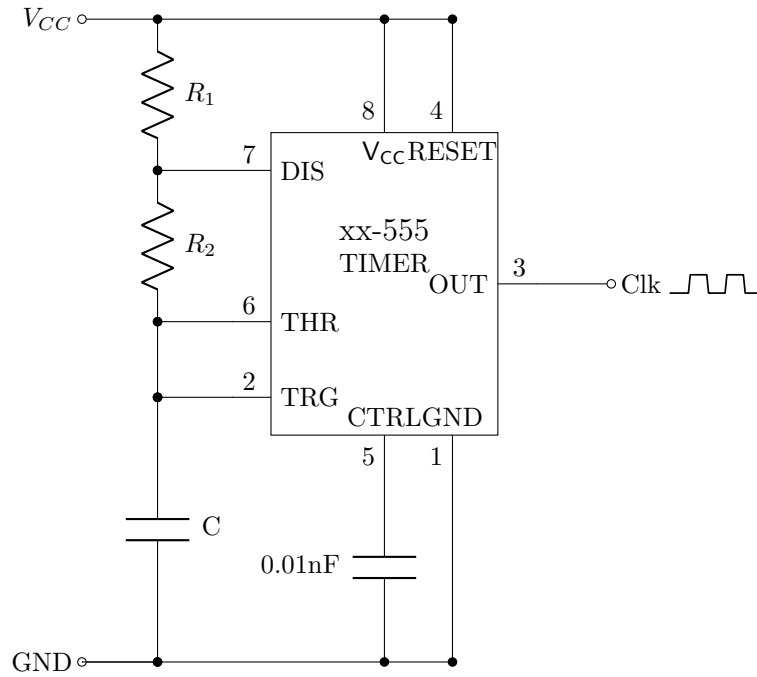


Figura 42: xx555 configurado en modo astable

Debido a que los valores calculados de las resistencias no son valores comerciales, se eligieron los dos valores comerciales mas cercanos para simplificar la implementación. Elegir valores distintos a los analíticos modificará el período del clock. Los valores de los componentes elegidos para la implementación son:

$$\begin{cases} R_1 = 2,7k\Omega \\ R_2 = 5,6k\Omega \\ C = 0,01\mu F \end{cases}$$

Estos valores nos dan un período $T = 96,347\mu\text{seg}$. Esta claro que se caerá en un error al utilizar un clock cuyo período no sea exactamente $100\mu\text{seg}$. Este error se hace mayor en módulo a medida que la extensión del pulso medido crece, sin embargo, el error relativo se achica a medida que el pulso crece.

Implementacion del contador

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

CP	MR	MODO
$\overline{\text{F}}$	LOW	Hold
$\overline{\text{F}}$	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 *CLK* será función del *Clk* y del pulso *ECHO*, como se muestra en la Figura 43

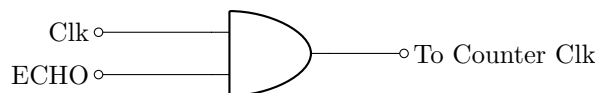


Figura 43: Logica a la entrada del contador

La entrada *RST* 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 - Trigger Enable Button

El usuario al pulsar el boton RESET, además de resetear el contador, toggleará un Flip-Flop JK configurado en modo Toggle que almacena en su salida el estado de la variable CONTANDO. Dado que $READY = \overline{CONTANDO}$, cuando CONTANDO=1, el circuito impedirá que el usuario dispare una solicitud al sensor, tal como ilustra la Figura 40.

Al pulsar el botón TRIG EN (tal como figura en la placa), se toggleará un Flip-Flop que almacena en su salida una variable que, tal como indica la consigna, habilita o deshabilita la solicitud de disparo al HC-SR04. A difetencia de la variable READY, esta es una variable que el usuario puede manipular para habilitar y deshabilitar manualmente.

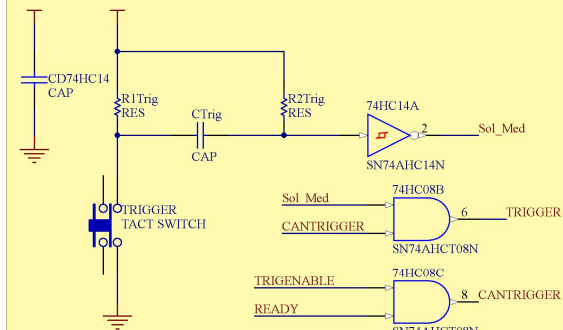
Output

La salida se muestra en 7 LED's que indican el estado del contador en codigo binario. Sobre la placa se indica la ponderación de cada LED, es decir a que bit del código binario corresponde.

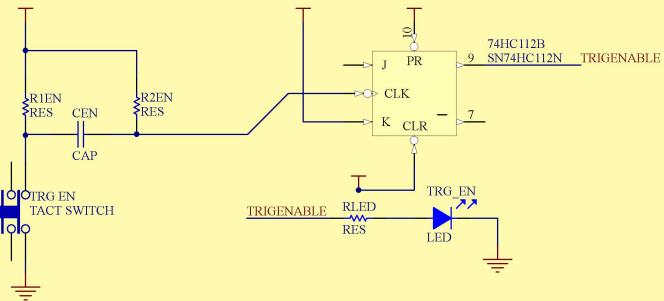
El esquemático del diseño y el circuito impreso implementado se detallan en dos hojas separadas a continuación.

Esquemático

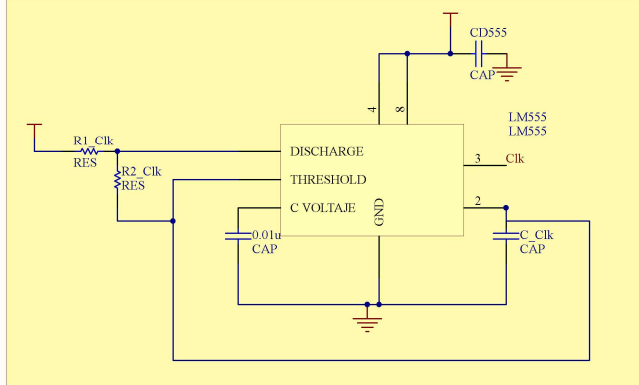
Trigger



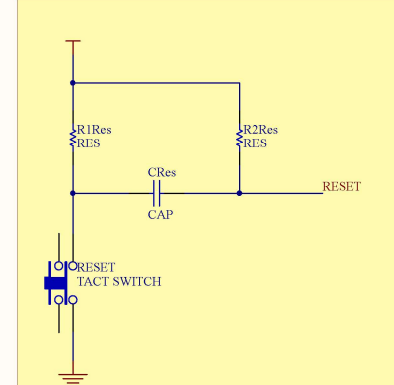
Trigger Enable



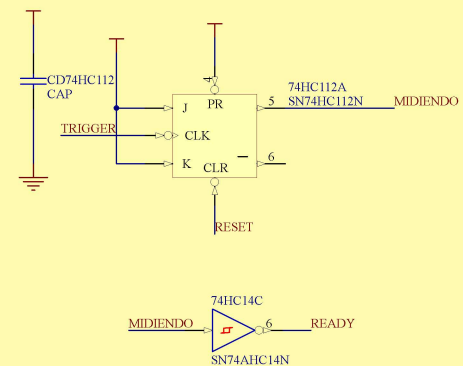
Clock



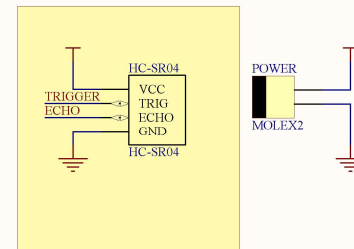
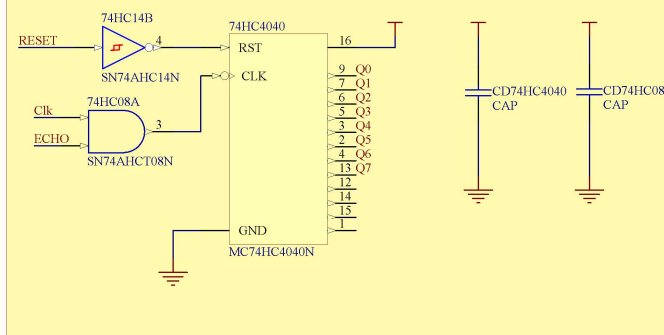
Reset



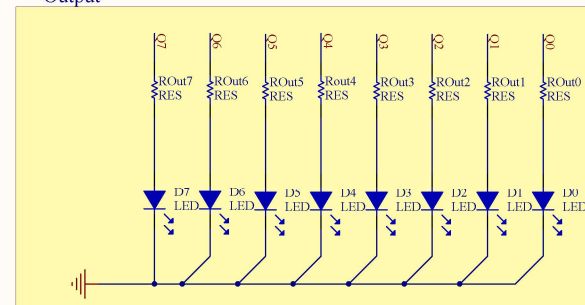
Circuit Ready

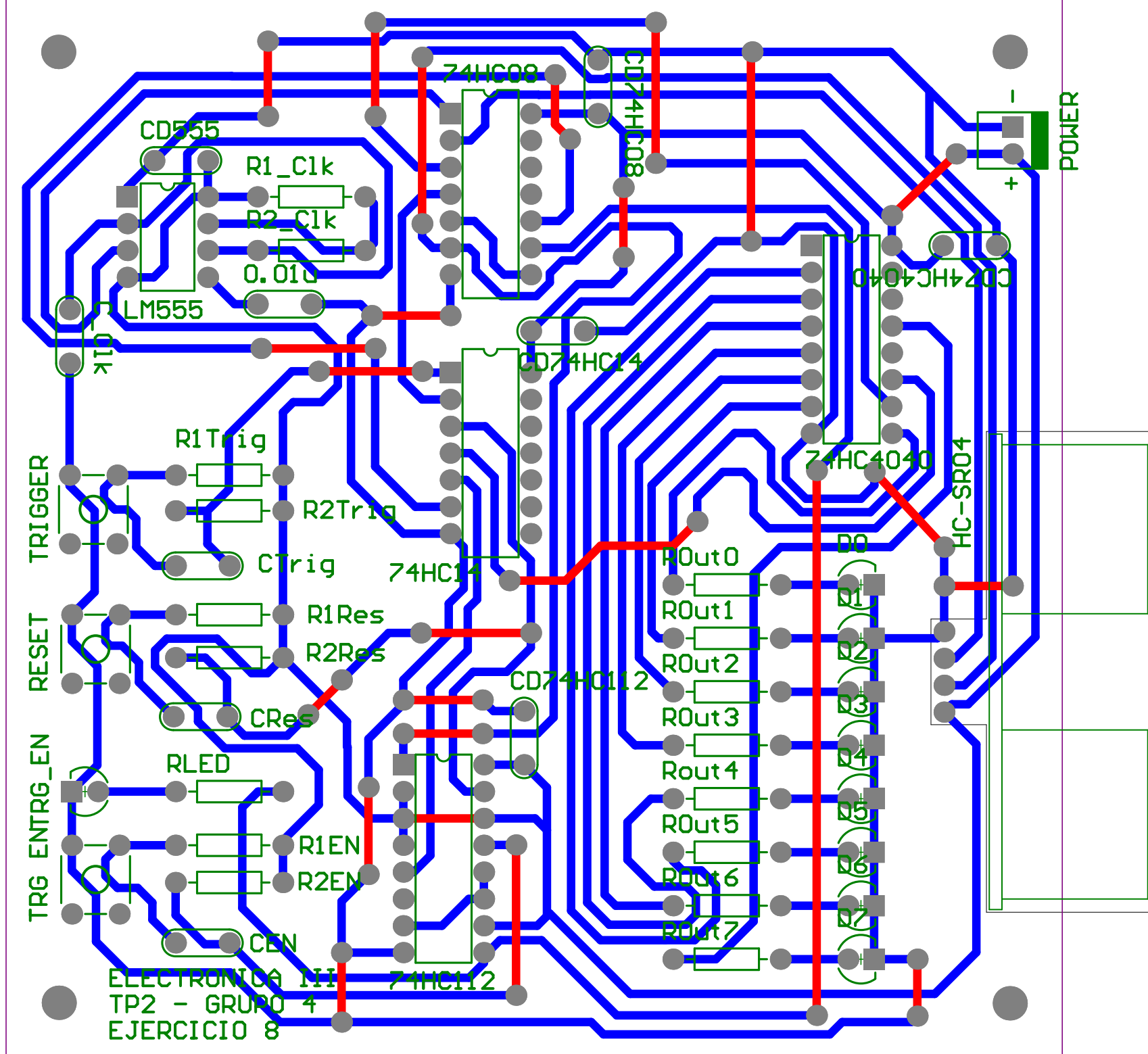


Contador



Output





ELECTRONICA III
TP2 - GRUPO 4
EJERCICIO 8