

# DESAFIO 1

**Juan Esteban Grajales Carmona**  
**Juan Manuel Sepúlveda Grajales**  
**Brayan Steven Avila Marin**

Departamento de Ingeniería Electrónica y  
Telecomunicaciones  
Universidad de Antioquia  
Medellín  
Febrero de 2022

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Resumen</b>	<b>2</b>
<b>3. Objetivos</b>	<b>2</b>
3.1. Objetivos generales . . . . .	2
3.2. Objetivos específicos . . . . .	2
<b>4. Marco teorico</b>	<b>2</b>
<b>5. Terminos claves</b>	<b>3</b>
<b>6. Recursos</b>	<b>6</b>
<b>7. Transmisión de datos</b>	<b>16</b>
<b>8. Decodificación</b>	<b>16</b>

## 1. Introducción

Esta es la primera sección, podemos agregar algunos elementos adicionales y todo será escrito correctamente. Más aún, si una palabra es demasiado larga y tiene que ser truncada, babel tratará de truncarla correctamente dependiendo del idioma.

## 2. Resumen

## 3. Objetivos

### 3.1. Objetivos generales

- llevar a cabalidad la implementación propuesta por los docentes además de complementar nuestros conocimientos sobre los proyectos mixtos entre hardware y software.
- Desarrollar un manejo y filtración adecuado de la información utilizando las bases vistas en clase.

### 3.2. Objetivos específicos

- Lograr comunicar dos sistemas como Arduino para enviar y recibir información de manera óptima y eficaz.
- Implementar el circuito integrado 74HC595 en el proyecto a desarrollar para encontrar solución a dicha circunstancia.
- Comprender como es el funcionamiento del proyecto además de cada uno de sus bloques dentro del esquema.

## 4. Marco teorico



Figura 1: Esquema del desafío 1

El propósito de este proyecto está basado en el trámite y recepción de información además de la correcta filtración de la información, así teniendo el

siguiente esquema (Foto del esquema). La información comprendida como grupos de números, será en primera instancia recibida por un Arduino número uno, posteriormente llevada a un bloque denominado codificación, luego al bloque de paralelización, siguiendo así al bloque de decodificación hasta finalmente dar salida al Arduino número dos donde se filtrará por medio de una clave y orden específica que información es verdadera y cuál no, para finalmente imprimir dicha información en una salida digital del Arduino.

## 5. Terminos claves

- Paralelización: El paralelismo de datos es un paradigma de la programación concurrente que consiste en subdividir el conjunto de datos de entrada a un programa, de manera que a cada procesador le corresponda un subconjunto de esos datos. Cada procesador efectuará la misma secuencia de operaciones que los otros procesadores sobre su subconjunto de datos asignado
- Comunicación serial: La comunicación serial es un protocolo de comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. El concepto de comunicación serial permite la transmisión-recepción bit a bit de un byte completo, este método de comunicación puede alcanzar mayores distancias.(aquí va una cita).

En la transmisión de datos en serie se utilizan pulsos binarios para transmitir los datos. El dígito binario uno está representado por cinco voltios o una lógica ALTA. Por el contrario, el cero binario se denota con una lógica BAJA o cero voltios. Para implementar la comunicación en serie, se requieren un origen y un destino. También se les conoce como emisor y receptor.(segunda cita de comunicación serial).

- Señal de reloj: (clock signal). En electrónica y especialmente en circuitos digitales síncronos, una señal de reloj es una señal usada para coordinar las acciones de dos o más circuitos. Una señal de reloj oscila entre estado alto o bajo, y gráficamente toma la forma de una onda cuadrada.

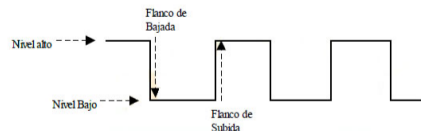


Figura 2: Grafico de una señal de reloj

Una señal de reloj es producida por un generador de reloj, como se dijo usualmente en forma de onda cuadrada y usualmente empleando una frecuencia fija constante.

Los circuitos que utilizan la señal de reloj para la sincronización pueden activarse en el flanco ascendente, flanco descendente o en ambos, por ejemplo, las memorias DDR SDRAM son activadas en ambos flancos.

La mayoría de los circuitos integrados complejos utilizan una señal de reloj para sincronizar sus diferentes partes y contar los tiempos de propagación.(agregar cita)

- latch: Se define como una de las funciones de algunos circuitos integrados, y este funciona para cargar los datos del pin de datos, al momento que se activa con el estado lógico “1”, es decir, su funcionamiento es sacar los datos recogidos en el integrado hacia sus respectivas salidas, si el latch no se activará dichos datos no serían enviados.
- bit: Se identifica Bit como el acrónimo de “binary digit”, que se traduce en español a “dígito binario”. Según esta definición, un bit es un dígito del sistema de numeración binario, que se representa con dos valores, el 0 y el 1.

En informática, bit es la unidad mínima de información. Se utiliza para representar la contraposición entre dos valores (apagado y encendido, falso y verdadero, abierto y cerrado).

En telecomunicaciones e informática, los bits son normalmente calculados en conjunto. Así tenemos 8 bits, 16 bits, 32 bits, etc.(Agregar cita)

- byte: Mientras el bit es la unidad mínima de información, byte es un conjunto fijo de bits. Aunque hay excepciones, en la mayoría de casos (sobre todo en relación al sector de la informática), un byte cuenta con 8 bits, de ahí su equiparación con el concepto de “octeto”.

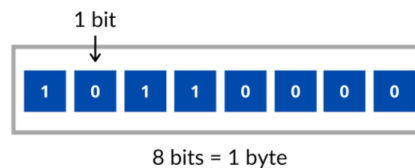


Figura 3: Representación binaria del bit y el byte

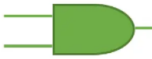
La partición estándar de bytes como grupos de 8 bits se basa a su vez en la codificación de caracteres ASCII (American Standard Code for Information Interchange). Por otra parte, la equivalencia gradual en informática se basa en bytes, para un cálculo más sencillo.

Por tanto, si 1 byte equivale a 8 bits, 1 kilobyte (kB / kbyte) equivale a 1024 bytes y así sucesivamente (megabytes, gigabytes, terabytes, petabytes, etc.)(agregar cita)

- Compuerta lógica: Las compuertas lógicas son circuitos electrónicos diseñados para obtener resultados booleanos (0,1), los cuales se obtienen de operaciones lógicas binarias (suma, multiplicación). Dichas compuertas son AND, OR, NOT, NAND, NOR, XOR, XNOR. Además se pueden conectar entre sí para obtener nuevas funciones. Este tipo de dispositivos lógicos se encuentran implementados con transistores y diodos en un semiconductor y actualmente podemos encontrarlas en formas de circuitos integrados lógicos. A continuación se definirá propiamente cada una de estas compuertas. A continuación se definirá propiamente cada una de estas compuertas:

- Compuerta AND: Para la compuerta AND, La salida estará en estado alto de tal manera que solo si las dos entradas se encuentran en estado alto. Por esta razón podemos considerar que es una multiplicación binaria.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1




Compuerta AND

Figura 4: Compuerta AND

- Compuerta NOT: En la compuerta NOT, el estado de la salida es inversa a la entrada. Evidentemente, una negación.

Q	Q'
0	1
1	0



Compuerta NOT

Figura 5: Compuerta NOT

- Compuerta XOR: La compuerta XOR Su salida estará en estado bajo cuando las dos entradas se encuentren en estado bajo o alto. Al mismo tiempo podemos observar que entradas iguales es cero y diferentes es uno.

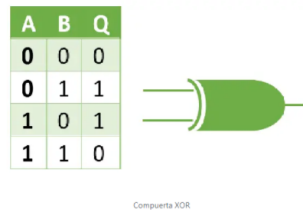


Figura 6: Compuerta XOR

- **Codificar: (Encoding).** En informática es un término muy extendido que se refiere, al proceso mediante el cual se produce una transformación de los datos, ya sea de analógico a digital (o viceversa) o de digital a digital. En el proceso de envío y recepción de la señal se llevan a cabo varios procesos de codificación/decodificación.(agregar cita)
- **Decodificar:** La decodificación es el proceso de convertir el código en texto plano o en cualquier formato que sea útil para procesos posteriores. Convierte las transmisiones de comunicación de datos codificados y los archivos a sus estados originales.

La mayoría de los ordenadores utilizan una metodología de codificación para transferir, guardar o utilizar datos. Los datos que deben codificarse se transforman mediante un mecanismo de codificación (por ejemplo, el American Standard Code for Information Interchange (ASCII) o BinHex) y se transmiten a través de un medio de comunicación.(agregar cita)

## 6. Recursos

- **circuito integrado 74HC595:** El 74HC595 es un CI bastante sencillo. Se trata de un registro de desplazamiento de 8-bit, es decir, tiene 8 biestables para almacenar 8 bits. El pin-out o patillas de este chip se puede ver en la imagen superior, con Vcc y GND para la alimentación, y luego las marcadas como Q que son las de datos. El resto corresponden a señales de reloj/control. La entrada la tiene en serie y la salida en paralelo. Por tanto, con una sola entrada, se pueden controlar a la vez esas 8 salidas. Solo necesitarás tres pines del microcontrolador usado (p.e.: Arduino) para manejarlo. Esas son Latch, Clock y Data.



Figura 7: Circuito integrado 74HC595

Descripción de los pines:

- Salida de datos paralela QA, QB, QC, QD, QE, QF, QG, QH (PIN 15, 1, 2, 3, 4, 5, 6, 7): Son todas las salidas de datos que se tiene.
- GND (PIN 8) Tierra.
- QH complementaria (PIN 9): Salida en serie Permite la conexión en cascada
- SCLR o Reset (PIN 10): Si se activa se borran los datos ya cargados anteriormente.
- SCK o Clock (PIN 11): Ayuda al desplazamiento de los datos.
- RCK o latch (PIN 12). Funciona para cargar los datos del pin 14 de datos, al momento que se activa con el estado lógico “1”
- OE o Output Enable (PIN 13): Se recomienda mandar a 0.
- SI o (PIN 14): Entrada de datos.
- VCC (PIN 16): Alimentación

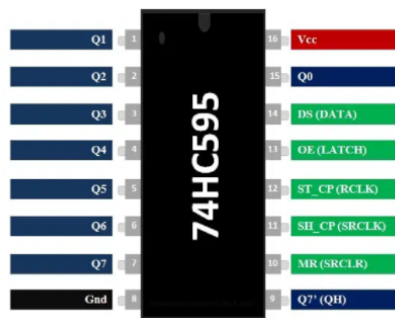


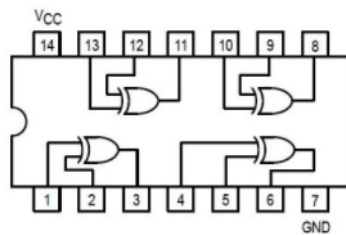
Figura 8: Pines del circuito integrado 74HC595

Añadido a esta información se deja la hoja de datos o datasheet del componente anteriormente mencionado: "<https://www.ti.com/lit/ds/symlink/cd74hc595.pdf>".



Dentro del desarrollo del proyecto este integrado se ve implementado dentro del circuito de paralelización, utilizado para recibir un byte desde el código fuente y dando como resultado ocho outputs cada una con los bits correspondientes al byte, esto con el fin de poder comparar esta información posteriormente.

- circuito integrado 74HC86: 4 compuertas XOR 74HC86 de dos entradas. CMOS Es una puerta lógica, o compuerta lógica, es un dispositivo electrónico con una función booleana. Suman, multiplican, niegan o afirman, incluyen o excluyen según sus propiedades lógicas. Se pueden aplicar a tecnología electrónica, eléctrica, mecánica, hidráulica y neumática. Son circuitos de conmutación integrados en un chip.



**TRUTH TABLE**

IN		OUT
A	B	Z
L	L	L
L	H	H
H	L	H
H	H	L

Figura 9: Circuito integrado 74HC86

Enfocado hacia el proyecto se utiliza en el circuito de comparación, siendo el primer filtro de información allí se comparan dos datos entregando salida low o cero si estas entradas son iguales, para posteriormente evaluar estos resultados. Adicionalmente se anexa el datasheet de dicho componente: <https://rocelec.widen.net/view/pdf/rrk3km7gxs/FAIRS32605-1.pdf>.

- Circuito integrado 74HC04: 6 compuertas inversoras (NOT). CMOS. Es la versión rápida del circuito integrado. HC del inglés “High commutation” alta conmutación. Se utiliza en equipos digitales que trabajan a altas frecuencias.

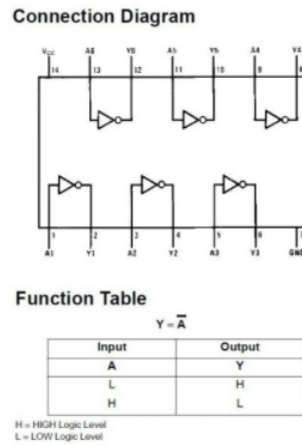


Figura 10: Circuito integrado 74HC04

- Circuito integrado 74HC08: Circuito integrado CMOS CD74HC08 consta de cuatro compuertas AND con dos entradas. El SN74HC08N es un circuito integrado que realiza la función booleana  $Y = A \bullet B$  o  $Y = A + B$  en lógica positiva. datasheet: <https://www.alldatasheet.com/datasheet-pdf/pdf/15524/PHILIPS/74HC08.html>

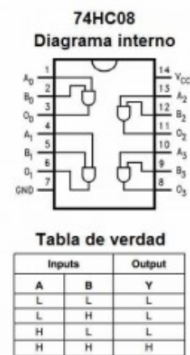


Figura 11: Circuito integrado 74HC08

- LCD16x2: El término LCD 16x2 se refiere a un pequeño dispositivo con pantalla de cristal líquido que cuenta con dos filas, de dieciséis caracteres cada una, que se utiliza para mostrar información, por lo general alfanumérica. Las capacidades de estos dispositivos son altas, pues se puede mostrar todo tipo de información sin importar qué tipo de símbolos o caracteres sean, el idioma o el lenguaje, pues el sistema puede mostrar cualquier carácter alfanumérico, símbolos y algunas figuras, el número de píxeles que tiene cada símbolo o carácter varía dependiendo del modelo del dispositivo y cada artefacto está controlado por un microcontrolador que está programado para dirigir el funcionamiento y la imagen mostrada en la pantalla.

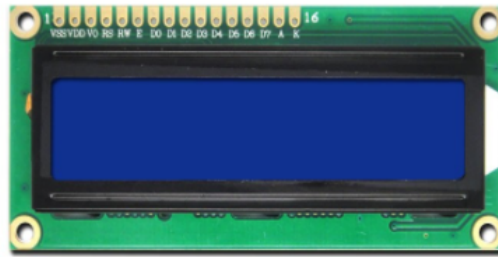


Figura 12: LCD16x2

- Arduino UNO: Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.



Figura 13: arduinoUNO

■ Circuito de paralelización:

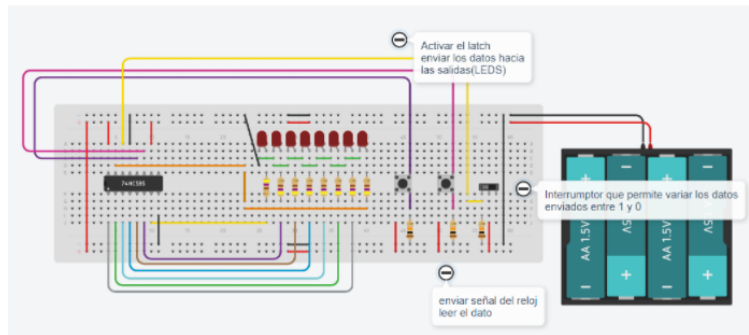


Figura 14: Circuito de paralelización

Este recurso se utiliza para el óptimo desarrollo de paralelizar los datos, es decir enviamos bit a bit por el pin de entrada del integrado (74HC595), este va llenando sus pines de salida siendo un máximo de ocho y para fines de funcionamiento en este esquema se saca dicha salida sea high o low hacia unos leds los cuales al tener una tierra común solo encienden con las señales de alto, se utiliza un interruptor para poder variar las entradas entre unos y ceros, posteriormente se envía la señal de reloj, la cual en este caso se hizo dependiente a un botón, pues al presionarlo hará un periodo completo de la señal, contando con un solo flanco de subida, es decir, al presionar el botón el integrado, captura la información de la entrada correspondiente y la lleva hacia una posición dentro de sí mismo siendo el primer caso a la salida número cero y cada vez que llega más información desplaza la existente una posición, así si el usuario ingresa un número binario por medio de la variación del suiche este número será paralelizado y entregado en señales unitarias para cada bit, luego de enviar un número binario completo (8 bits) se presiona el latch, el cual tiene la función de sacar los datos desde el integrado hacia las salidas, en este caso los leds, es decir, si no se hace uso del latch los datos no serán enviados y la información se perdería. Cabe destacar que en el momento en que se ingresa un número y este se paraleliza, no hay inconveniente en ingresar otro completo, pues al desplazar posiciones el primer número se perderá y no afectará al segundo en dicho proceso. Además se debe tener en cuenta que este esquema es únicamente presentado para fines de contextualización, pues en el esquema del proyecto final, este circuito no tendrá dentro de sus componentes los botones.

A continuación se expondrán algunas pruebas de la veracidad del funcionamiento del presente circuito.

- INPUT: 11100100
- OUTPUT: 11100100 (Paralizado)

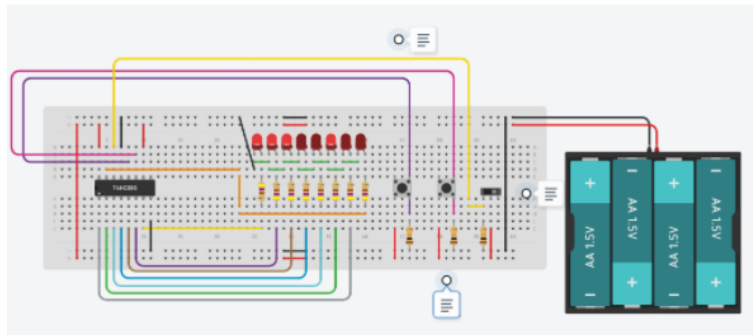


Figura 15: prueba paralelizacion

Podemos evidenciar que claramente el número ingresado ha sido correctamente paralelizado, pues podemos notar que aquellas posiciones donde el binario tiene un uno los leds correspondientes a esa posición están encendidos, y donde tiene un cero están apagados, haciendo así más fácil ingresar a la información del número bit a bit.

- INPUT: 11110001
- OUTPUT:11110001 (Paralelizado)

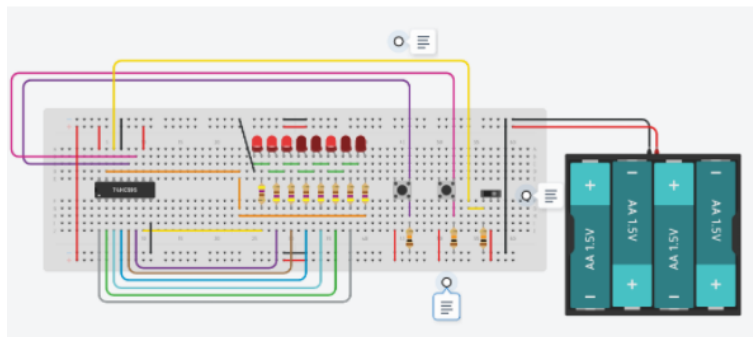


Figura 16: prueba paralelizacion 2

- INPUT:00011100
- OUTPUT:00011100(Paralelizado)

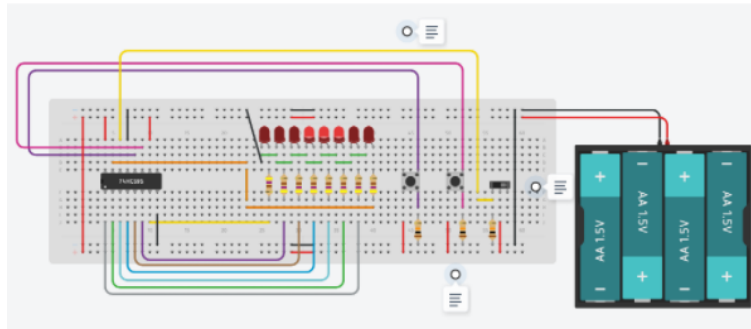


Figura 17: prueba paralelizacion 3

El circuito y montaje fue basado desde el siguiente link: "<https://uelectronics.com/producto/74hc595-registro-de-desplazamiento-sn74hc595n>"

Para profundizaciones y pruebas en el circuito se deja el link al proyecto: "<https://www.tinkercad.com/things/4zrdRUQidBi>"

■ Circuito de comparación:

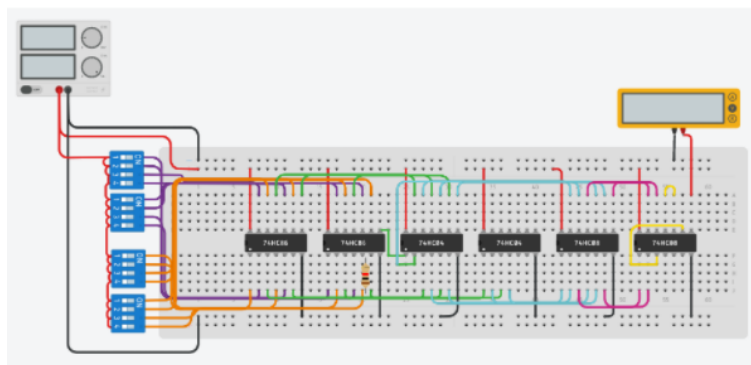


Figura 18: Circuito de comparación

Este montaje se realiza con la finalidad de poder comparar dos números de ocho bits, dando como resultado un valor booleano, cinco voltios si son iguales y cero voltios en caso contrario, se utilizan dos multiswitches, cada uno con cuatro salidas para representar un número es decir, para comparar dos se utilizaran cuatro de estos componentes, en primer momento se envía cada uno de los bits de la posición n de ambos números a dos entradas de la compuerta XOR (74HC06) allí son evaluados si los datos son iguales la

salida será igual a cero, este proceso se hace con las ocho parejas de bits, dejando como resultado ocho valores para cada pareja, luego esos datos se envían a la compuerta NOT (74HC04) donde simplemente los valores son evaluados e intercambiados por su contrario, es decir, si la entrada es un uno la compuerta entregará como salida un cero y viceversa, teniendo los ocho datos ya intercambiados por la compuerta NOT se llevan hacia una compuerta AND (74HC08) allí se forman parejas sin restricciones de orden, dando cuatro parejas que al ser evaluadas dejan como salida cuatro datos correspondientes a la comparación AND, es decir, si ambos datos son uno la salida será uno, de lo contrario será un cero, estos cuatro datos se llevan a otra compuerta AND entregando dos datos y para finalizar estos dos van a una última compuerta de tipo AND, finalmente quedamos con un solo dato y este es correspondiente a la comparación, por tanto si la salida es uno, los números ingresados son iguales y en caso de ser cero los números son distintos. Adicionalmente se debe mencionar que luego de pasar la compuerta XOR no hay orden en las parejas puesto que la única manera de que la salida final en AND sea un uno es el caso donde todas las salidas desde la compuerta XOR fueron uno respectivamente, así si solo hay una pareja de bits distinta la salida final será cero, dejando en evidencia que los números son diferentes. Asimismo se debe dejar claro que este montaje se hace para fines de contextualización, pues en el proyecto final, uno de los números será correspondiente a los bits paralelizados en el circuito de paralelización el cual entrega ocho salidas, y el número con el cual se compara será la bandera para poder saber qué información es verdadera, esta bandera es definida por los docentes y a su vez en el circuito es fijada en los switches. A continuación se mostrará algunas pruebas para este esquema:

- INPUT: 11001100 y 11001100
- OUTPUT: 5V (iguales)

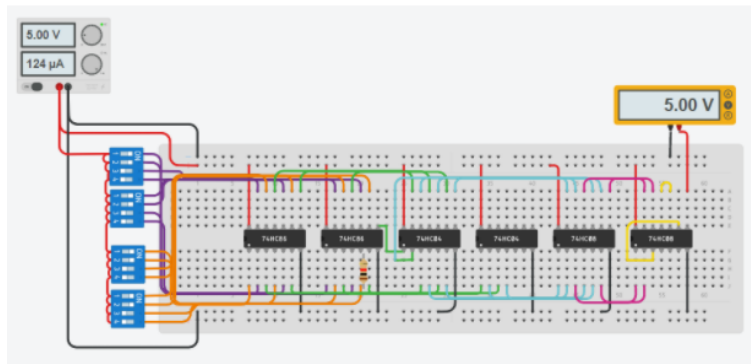


Figura 19: Prueba 1 de comparación

Claramente se evidencia que el comparador funciona, pues al tener las mismas entradas la salida es de 5V.

- INPUT: 00001111 y 00001101
- OUTPUT: 0V (diferentes)

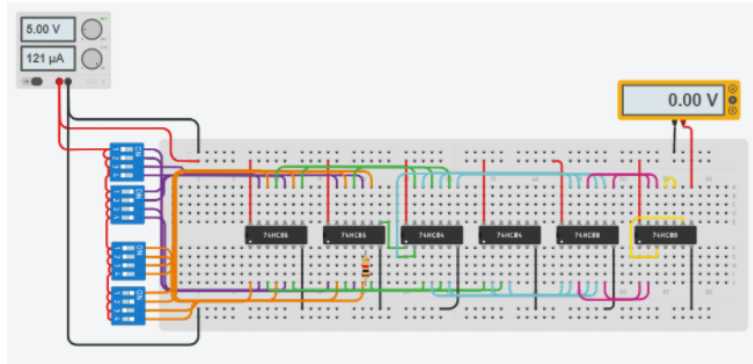


Figura 20: Prueba 2 de comparación

- INPUT: 00000001 y 1000000
- OUTPUT: 0V (diferentes)

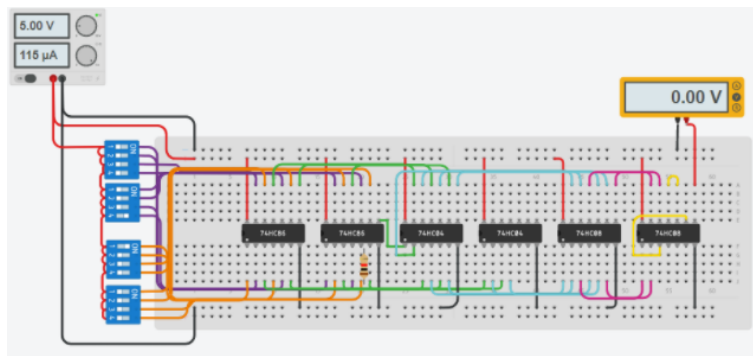


Figura 21: Prueba 3 de comparación

El circuito y montaje fue basado desde el siguiente link: <https://www.tinkercad.com/things/abmbYgjoszA/comparador-8-bits>

Para profundizaciones y pruebas en el circuito se deja el link al proyecto: <https://www.tinkercad.com/things/149SXbQUvTg>



## 7. Transmisión de datos

La transmisión de datos se realizará en primer momento ingresando el grupo de números a evaluar al Arduino número uno por medio del código fuente del proyecto, seleccionando el primer término de esta lista, dentro de su software lo convierte a su forma binario y este dato se envía a través del puerto serial hacia el circuito de paralelización donde el dato ingresado de ocho bits será fragmentado en cada bits asimilándose al funcionamiento de un demux, pues cuenta con una entrada comprendida como el dato enviado, y ocho salidas cada una con un bit del número binario anteriormente enviado, generando así ocho salidas nuevas a disposición del circuito.

## 8. Decodificación

El proceso para decodificar la información se ve abordado de dos posiciones una por medio de hardware y otra con software:

- en torno al hardware se realiza por medio de circuitos integrados (74HC86, 74HC04, 74HC08) siendo estas compuertas lógicas de tipo XOR, NOT y AND, el funcionamiento de dichos integrados es simple, teniendo en cuenta que tenemos dieciséis bits, es decir, ocho por cada número tomamos el bit de la misma posición para cada número, eso lo llevamos a la entrada XOR donde es evaluado, si ambos bits son iguales en la misma posición la salida de la compuerta será igual a cero, en caso contrario será uno, posteriormente cada señal es pasada a un negador el cual si recibe un cero entrega un uno y viceversa, así luego de este proceso nos quedarían ocho señales representadas por el número uno si ambos bits son iguales o cero si son distintos correspondientes a comparar dieciséis bits uno a uno, estas señales pasan al integrado por compuertas lógicas AND, las cuales solo entregarán un uno si ambas entradas son uno, por tanto al pasar el filtro de las compuertas AND quedan cuatro señales con las mismas condiciones en los valores, estas son trasladadas a otras compuertas AND y entregan dos señales resultantes, finalmente se vuelve a evaluar en una compuerta AND y esta salida se verá representada de la siguiente manera. 1 = cada uno de los bits de un número es igual a los bits del otro número en la misma posición, es decir, los números son iguales. 0 = hay bits distintos, es decir, los números son distintos. Teniendo esto claro uno de los números de ocho bits será correspondiente al paralelizado en binario y el número a comparar será la bandera brindada por el docente, así se logrará concluir si la bandera a llegado en la información o no.
- En torno al software la decodificación se hará por medio de funciones probadas previamente en el entorno de desarrollo QT con el objetivo de convertir el número binario a decimal y posteriormente llevarlo a una pantalla LCD en caso de verse requerido para tomar la información.

## Referencias

- Paralelizacioin: [https://es.wikipedia.org/wiki/Paralelismo\\_datos](https://es.wikipedia.org/wiki/Paralelismo_datos)
- Serial: [http://www.itq.edu.mx/carreras/IngElectronica/archivos\\_contenido/Apuntes](http://www.itq.edu.mx/carreras/IngElectronica/archivos_contenido/Apuntes)
- Señal de reloj: <https://es.wikipedia.org/wiki/Se>
- Bit y Byte: <https://www.jvs-informatica.com/blog/glosario/bit/>
- Compuertas logicas e imagenes : <https://hetpro-store.com/TUTORIALES/compuertas-logicas/>
- Codificar: <https://languages.oup.com/google-dictionary-es/>
- Decodificar: <https://languages.oup.com/google-dictionary-es/>
- circuito integrado 74HC595: <https://www.amazon.com/-/es/MCIGICM-74HC595-SN74HC595N-Registro-unidades/dp/B07HFWB9L9>  
[https://www.hwlibre.com/74hc595/?utm\\_source=feedburnerutm\\_medium=feedutm\\_campaign=Feed](https://www.hwlibre.com/74hc595/?utm_source=feedburnerutm_medium=feedutm_campaign=Feed)  
[https://www.hwlibre.com/74hc595/?utm\\_source=feedburnerutm\\_medium=feedutm\\_campaign=Feed](https://www.hwlibre.com/74hc595/?utm_source=feedburnerutm_medium=feedutm_campaign=Feed)  
<https://uelectronics.com/producto/74hc595-registro-de-desplazamiento-sn74hc595n/:text=>
- circuito integrado 74HC86: <https://geekbotelectronics.com/tienda/producto/74hc86-xor-2-entradas/>
- circuito integrado 74HC04: <https://geekbotelectronics.com/tienda/producto/74hc04-not-inversor/>
- Circuito integrado 74HC08: <https://www.carrod.mx/products/cmos-cuatro-compuertas-and-con-2-entradas-74hc08>
- LCD16x2: <https://hetpro-store.com/lcd-16x2-blog/>
- Arduino UNO: <https://arduino.cl/arduino-uno/>