

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

Sección B

Ingeniero Otto Leiva

Aux Christian Real

22 de Noviembre de 2019, Guatemala



Manual Tecnico

No.	Carnet	Nombre	Ponderación
1	201700308	Adrián Byron Ernesto Alvarado Alfaro	100
2	201700317	Carlos Alejandro Tenes Mejía	100
3	201700521	Ludwing Gabriel Paz Hernández	100
4	201700733	Byron Antonio Orellana Alburez	100
5	201709166	Jackeline Alexandra Benitez Benitez	100

Introducción

A lo largo de los años la tecnología ha ido avanzando a pasos agigantados, hemos llegado a tal punto que todos los dispositivos que están siendo creados tienen una tendencia a ser cada vez más pequeños con cada año que pasa. Un circuito lógico secuencial es aquel cuyas salidas no solo dependen de sus entradas actuales, sino también de su posición o estado actual, almacenada en elementos de memoria.

Es importante conocer las funcionalidades de un nuevo programa, sobre todo si no conocemos nada de el. En este caso este programa es destinado para jugadores de Snake, el famoso juego de la culebrita en donde mientras la culebra come, esta incrementa su tamaño. La diferencia de este snake con los convencionales es que puede jugarse desde la computadora, y además mostrarse en una matriz de LED.

Para esto, se utiliza un puerto paralelo simulando un puerto serial. Un puerto paralelo es una interfaz entre un computador y un periférico, cuya principal característica es que los bits de datos viajan juntos, enviando un paquete de byte a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus. Con los cual se puede realizar la comunicación serial, La comunicación serie o comunicación secuencial, en telecomunicaciones e informática, es el proceso de envío de datos de un bit a la vez, de forma secuencial, sobre un canal de comunicación o un bus.

Objetivos y Alcances del Sistema:

Objetivos:

- Realizar un juego cuyo funcionamiento sea correcto.
- Realizar un correcto movimiento de la serpiente tanto en la aplicación como en la matriz LED.
- Crear una interfaz de usuario atractiva con un buen rendimiento y funcionamiento.

Alcances de Sistema:

- Registrar las coordenadas de la serpiente a medida que se juega.
- Hacer un correcto movimiento de la serpiente a través de la aplicación.
- Implementar correctamente los botones de movimiento.

Especificaciones Tecnicas

Lenguaje de Programación: Visual Express 2010, en C#

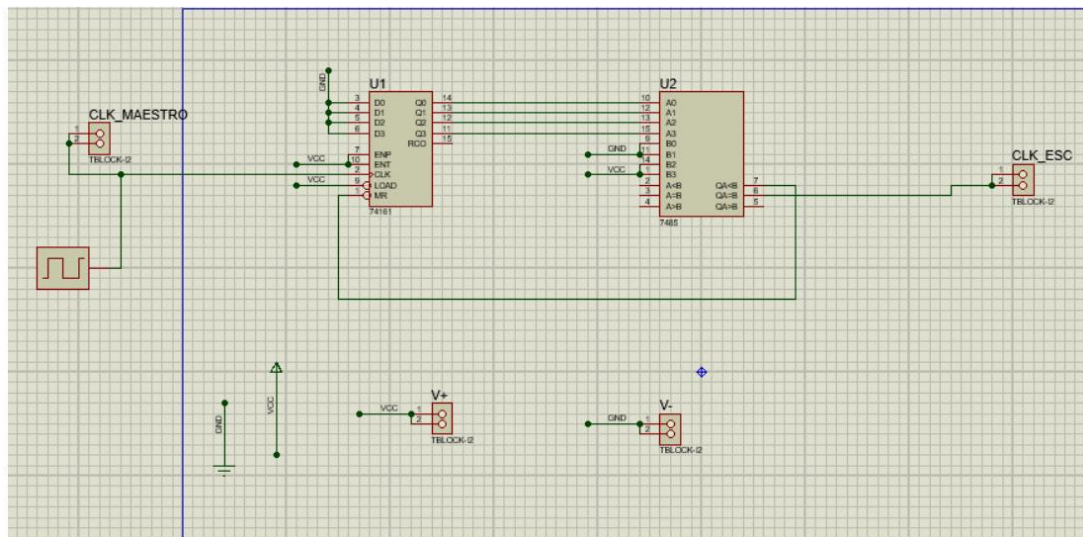
Sistema Operativo: Microsoft Windows XP Profesional, Version 2002,
Service pack 3

Funciones Booleanas y Diagramas:

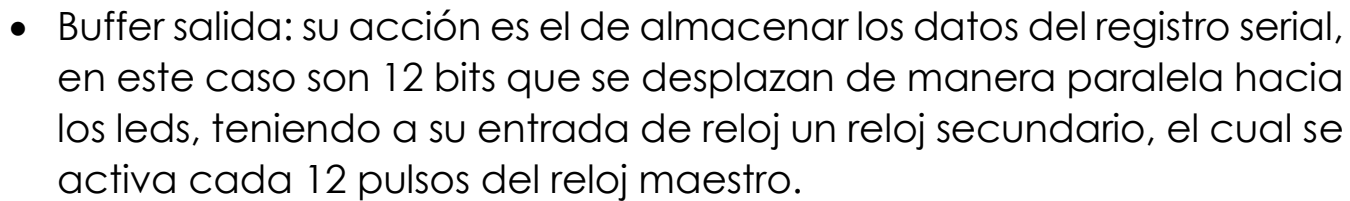
Para el presente proyecto, como lo especifica el enunciado, se requiere de un tipo de monitor realizado con leds representando una matriz de 12x12, la computadora, manda de manera continua los pulsos por columna a modo de pila, teniendo registro serial y un buffer de datos que permiten un flujo continuo de datos, a su vez una demultiplexación para que la cadena de bits se dirija a su respectiva columna.

Partes de los circuitos y Diagramas con Explicación

- Contador maestro: es el encargado de contar los pulsos de entrada, este cuando llega a 12 se reinicia mandando un pulso de reloj secundario.

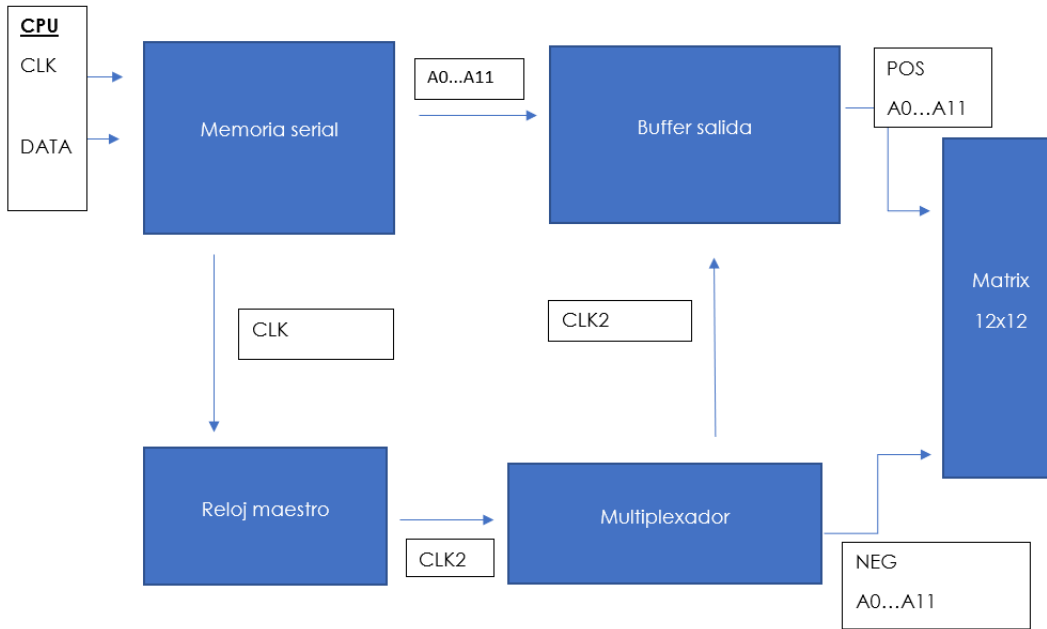


- Memoria serial: es la encargada de almacenar los datos que llegan desde el computador estos solo se desplazan sobre los 12 registros existentes recibiendo los pulsos de reloj que provienen del computador.



-
- The diagram illustrates a 16-bit parallel adder circuit. It features two 74181 ALUs (DM1 and DM2) and two 74148 decoders (U1 and U2). The ALUs are configured to perform addition, with their carry-in inputs connected to a common carry chain. The decoders are used to generate the carry signals for the ALUs. The circuit is powered by a 5V supply and includes a 10kΩ pull-up resistor for the carry-in of the first ALU. The output of the adder is a 16-bit bus, which is connected to a 16-bit parallel-to-serial converter (P2S) and a 16-bit serial-to-parallel converter (S2P). The P2S and S2P are connected to a 16-bit parallel-to-serial converter (P2S) and a 16-bit serial-to-parallel converter (S2P) respectively. The circuit is tested using a logic analyzer (LA) and a logic converter (LC).

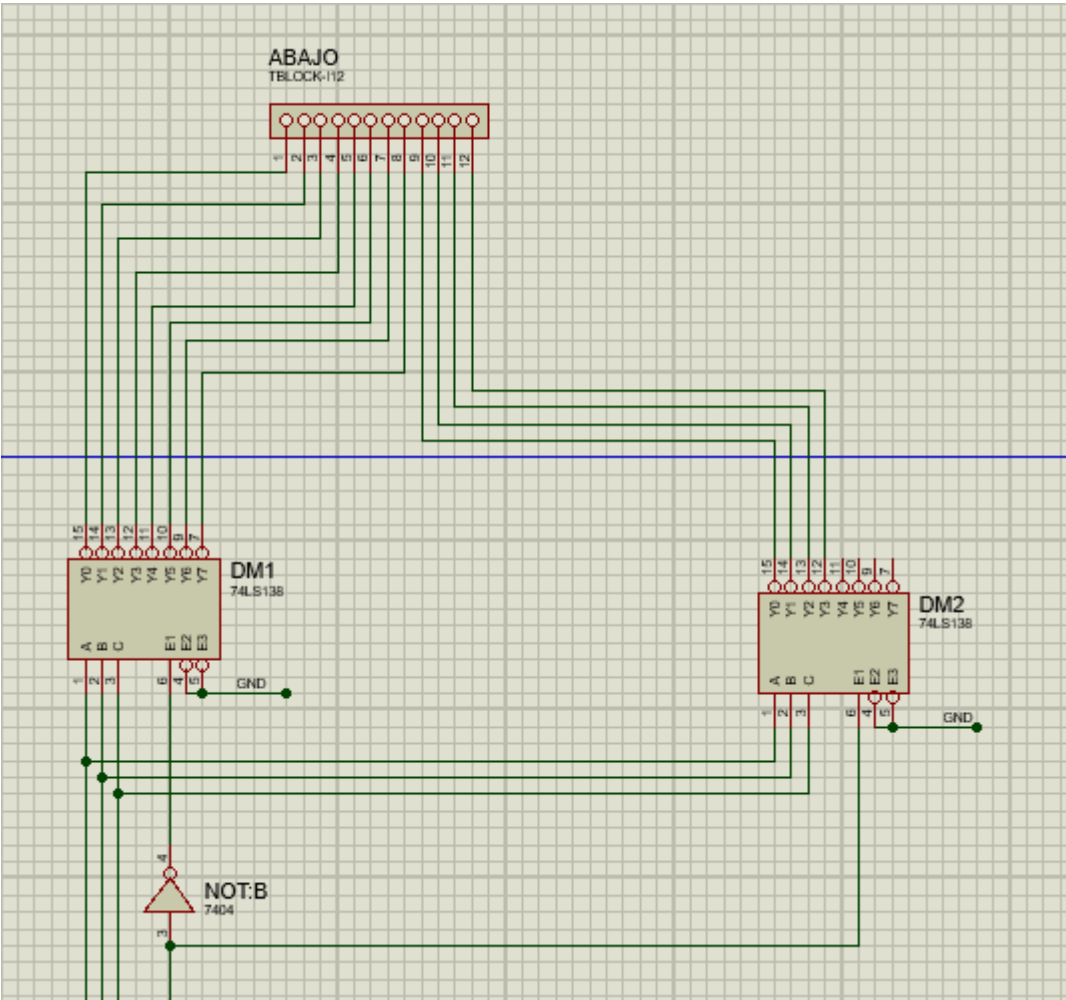
Vista a Bloque



Funcionamiento de la multiplexación

Para que la matriz funcione de una manera optima es necesario que exista un selector, que cada vez que una línea se cargue por completo exista un salto en la selección, sin embargo, los demultiplexores existentes solo tienen 8 salidas, entonces, por medio de una tabla de verdad se logró obtener un decoder de 4x16, como solo se necesitan 12 líneas, se ignoran las ultimas 4 líneas.

[illegible]



Listado de Métodos y Funciones Implementadas:

- **Inpout32.dll:** Utilizado para obtener el acceso a los puertos.
- **With:** Para establecer un ancho en determinado objeto
- **Height:** Para establecer una altura en determinado objeto.
- **BackColor:** Para establecer un fondo a una figura, label o form.
- **Location:** Para obtener las coordenadas.
- **Manejador:** clase que contiene la lista de Usuarios en el sistema y maneja el usuario que inicia sesión.
- **Usuario:** clase que contiene los atributos de un usuario, tales como usuario, contraseña, puntos, tiempo y nivel.
- **List:** clase que permite el manejo de Listas para el almacenamiento de usuarios.
- **agregarUsuario(Usuario):** método que maneja el metodo nativo de la clase List que agrega a la lista.
- **buscarUsuario(String):** método que devuelve un usuario en específico, dependiendo del parámetro recibido.
- **ActionListener():** Permite tener llamar una accion
- **ActionPerformed():** Hace la accion
- **getContentPane():** Para añadirlo a Panel
- **SetBounds:** Cambiar tamaño y coordenadas
- **SetSize:** Cambiar Tamaño
- **SetVisible:** Para Ocultar o Aparecer
- **JOptionPane:** para mensajes pequeños
- **Button, Label, Combos, JtextField:** para implemetar en el panel
- **Jframe:** Ventana que se usa
- **DefaultTableModel:** Parar crear tablas
- **Jpanel:** para crear nuevas laminas o espacios en el frame
- **Defaultsetoperation:** para cerrarlo automaticamente

- **Array:** Para guardar los objetos creados
- **If y Switch:** para el proceso de ocultar y aparecer los labels en la carga
- **Jtable:** Para crear la tabla
- **Toolkit:** para obtener tamaño de la pantalla
- **AddItem:** para añadir items al combo
- **Get y set:** para obtener los campos de un jtext field
- **Hide:** para ocultar un form
- **Show:** método para mostrar un objeto.
- **FileReader:** para leer el archivo.
- **DataGridView:** objeto que sirve para mostrar datos en forma de tabla.

- **Dudas o Consultas**

- Adrián Alvarado
Software Developer, byronalvarado9090@gmail.com
- Byron Orellana
Software Developer, orellanab80@gmail.com
- Carlos Tenes
Software Developer drometex98@gmail.com
- Gabriel Paz
Software Developer ludwing.paz@gmail.com
- Jackeline Benitez
Software Developer jackelinebenitez112@gmail.com