

PROYECTO FINAL:

ELEVADOR

Integrantes:

Francisco Antonio Palos Angulo 132759

Kaled Andrés Muhech Álvarez 137959

Jorge Luis Domínguez Trejo 132607

Manuel Aurelio Nozato Beltrán 129577

Cd. Obregón, Sonora., 27 de abril de 2017.

**ÍNDICE**

**1.- Introducción 01**

**2.- Antecedentes Teóricos 02**

**3.- Diseño del Proyecto**

**3.1 Enunciado del proyecto 03**

**3.2 Aspectos de intuición a considerar 03**

**3.3 Código VHDL 03**

**3.4 Diagrama de circuito impreso 07**

**4.-**  **Conclusiones, limitaciones y recomendaciones 08**

**5.- Anexos 09**

**1.- INTRODUCCIÓN**

En este reporte se encuentran todos los datos, teoría y pasos necesarios que se ocuparon para la construcción de nuestro elevador de 5 niveles hecho solo con la tecnología que vimos durante la clase de digitales I.

Para llevarlo a cabo, se analizara lo que sea más conveniente para desarrollar el proyecto.

Con la Gal y programación VHDL se puede desarrollar uno o varios programas que se pueden realizar con varios circuitos integrados en uno solo, considerando la cantidad de patas disponibles para realizarlo.

El proyecto está conformado por el sistema de una impresora, un motor, una fuente de voltaje, tres gales, dos display de cátodo común y una interface de potencia. Con la programación VHDL que pusimos en la Gales controlamos el elevador y las señales que mandamos hacia los displays.

**2.- ANTECEDENTES TEÓRICOS**

**GAL:**

GAL (Generic Array Logic), en español Arreglo Lógico Genérico, son un tipo de circuito integrado, de marca registrada por Lattice Semiconductor, que ha sido diseñados con el propósito de sustituir a la mayoría de las PAL, manteniendo la compatibilidad de sus terminales. Utiliza una matriz de memoria EEPROM en lugar por lo que se puede programar varias veces. Un GAL en su forma básica es un PLD con una matriz AND reprogramable, una matriz OR fija y una lógica de salida programable mediante una macrocelda. Esta estructura permite implementar cualquier función lógica como suma de productos con un número de términos definido.

**VHDL:**

VHDL. Lenguaje de descripción hardware estructurado para modelar sistemas digitales.  
**V**: VHSIC – Very High Speed Integrated Circuit  
**H**: Hardware  
**D**: Description  
**L**: Language

(Suponiendo que V2 es la tensión de referencia)

**Interfaz de Potencia:**

Las interfaces de potencia son dispositivos intermedios entre nuestro microcontrolador y aquellos aparatos que requieran cantidades de corriente mayores a los que pueden manejar el microcontrolador (por lo general estamos hablando de una corriente de unos 40 miliamperios como máximo por pin).  
Motores de paso, motores DC, servomotores, lámparas incandescentes, reflectores, grupos o tiras de leds, son ejemplos de dispositivos que podríamos llegar a controlar desde el microcontrolador a través de las interfaces de potencia. Es un grave error tratar de conectar este tipo de dispositivos directamente a los pines del microcontrolador. Nos ayudaremos de transistores, relés, puentes-H o interfaces electrónicas de control para construirlas.

**3.- DISEÑO DEL PROYECTO**

**3.1 Enunciado del proyecto.**

Realizar un sistema que permita mover lo que sería el cuarto del elevador en dos direcciones y además detenerlo en el momento en el que llegue a los pisos solicitados, esto lo lograremos mediante el uso de sistema de elevador con el que cuenta por dentro la impresora de tinta, para el proyecto solo nos fue necesario utilizar la tecnología VHDL y GAL.

**3.2 Aspectos de intuición a considerar**

La GAL que tiene el código del elevador recibirá valores de los sensores y de los push buttons, luego los comparará y una vez comparado realizará el movimiento que sea necesario para que las señales de ambos se igualen.

Una vez que las señales se igualen el motor se detendrá evitando así que cambie a un piso no buscado y si el botón del piso en el que ya se encuentra vuelve a ser presionado el elevador no deberá moverse.

**3.3 Código VHDL**

Primera gal Elevador:

library ieee;

use ieee.std\_logic\_1164.all;

entity controlador\_GAL is

port (

I\_Sensor: in std\_logic\_vector(4 downto 0);

I\_Teclado: in std\_logic\_vector(4 downto 0);

O\_Sensor: out std\_logic\_vector(2 downto 0);

O\_Teclado: out std\_logic\_vector(2 downto 0);

M: out std\_logic\_vector(1 downto 0));

end controlador\_GAL;

architecture funcion of controlador\_GAL is

signal Signal\_Sensor: std\_logic\_vector(2 downto 0):="000";

signal Signal\_Teclado: std\_logic\_vector(2 downto 0):="000";

begin

process (I\_Sensor)

begin

case I\_Sensor is

when "00001" => Signal\_Sensor <= "001";

when "00010" => Signal\_Sensor <= "010";

when "00100" => Signal\_Sensor <= "011";

when "01000" => Signal\_Sensor <= "100";

when "10000" => Signal\_Sensor <= "101";

when others => Signal\_Sensor <= Signal\_Sensor;

end case;

end process;

process (Signal\_Sensor)

begin

O\_Sensor <= Signal\_Sensor;

end process;

process (I\_Teclado)

begin

case I\_Teclado is

when "00001" => Signal\_Teclado <= "001";

when "00010" => Signal\_Teclado <= "010";

when "00100" => Signal\_Teclado <= "011";

when "01000" => Signal\_Teclado <= "100";

when "10000" => Signal\_Teclado <= "101";

when others => Signal\_Teclado <= Signal\_Teclado;

end case;

end process;

process (Signal\_Teclado)

begin

O\_Teclado <= Signal\_Teclado;

end process;

process (Signal\_Sensor, Signal\_Teclado) -- Comparador

begin

if Signal\_Sensor = Signal\_Teclado then

M <= "00";

elsif Signal\_Sensor < Signal\_Teclado then

M <= "01";

else

M <= "10";

end if;

end process;

end funcion;

Segunda gal SDA:

library ieee;

use ieee.std\_logic\_1164.all;

entity controlador\_GAL is

port (

I\_Sensor: in std\_logic\_vector(2 downto 0);

display\_acutal: out std\_logic\_vector(6 downto 0));

end controlador\_GAL;

architecture funcion of controlador\_GAL is

signal Signal\_Sensor: std\_logic\_vector(6 downto 0):="0000000";

begin

process (I\_Sensor)

begin

case I\_Sensor is

when "001" => Signal\_Sensor <= "0110000";

when "010" => Signal\_Sensor <= "1101101";

when "011" => Signal\_Sensor <= "1111001";

when "100" => Signal\_Sensor <= "0110011";

when "101" => Signal\_Sensor <= "1011011";

when others => Signal\_Sensor <= Signal\_Sensor;

end case;

end process;

process (Signal\_Sensor)

begin

display\_acutal <= Signal\_Sensor;

end process;

end funcion;

Tercera gal TDF:

library ieee;

use ieee.std\_logic\_1164.all;

entity controlador\_GAL is

port (

I\_Teclado: in std\_logic\_vector(2 downto 0);

display\_futuro: out std\_logic\_vector(6 downto 0));

end controlador\_GAL;

architecture funcion of controlador\_GAL is

signal Signal\_Teclado: std\_logic\_vector(6 downto 0):="0000000";

begin

process (I\_Teclado)

begin

case I\_Teclado is

when "001" => Signal\_Teclado <= "0110000";

when "010" => Signal\_Teclado <= "1101101";

when "011" => Signal\_Teclado <= "1111001";

when "100" => Signal\_Teclado <= "0110011";

when "101" => Signal\_Teclado <= "1011011";

when others => Signal\_Teclado <= Signal\_Teclado;

end case;

end process;

process (Signal\_Teclado)

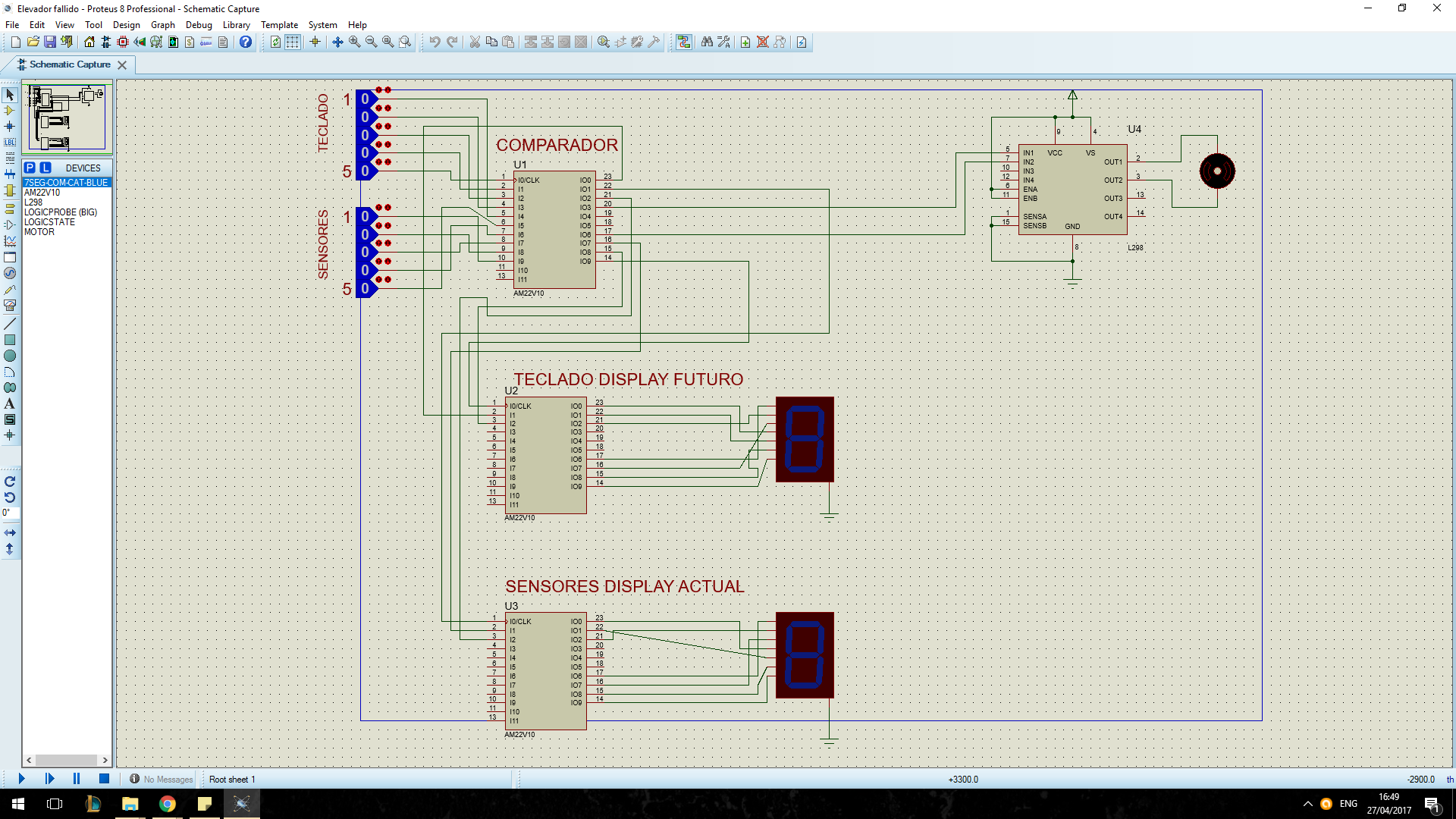
begin

display\_futuro <= Signal\_Teclado;

end process;

end funcion;

**3.4 Diagrama del circuito impreso.**



**4.- CONCLUSIONES, LIMITACIONES Y RECOMENDACIONES.**

Conclusiones

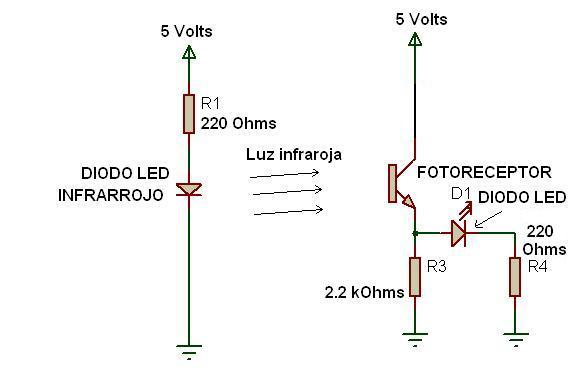
Este proyecto que realizamos para la clase de digitales I ha sido muy importante para nuestra formación como ingenieros en este tipo de áreas de ingeniería, ya que es nuestro primer proyecto de tipo Mecatrónico y nos ayudará mucho a futuro, ya que nos va dando una idea de cómo hacer proyectos más elaborados.

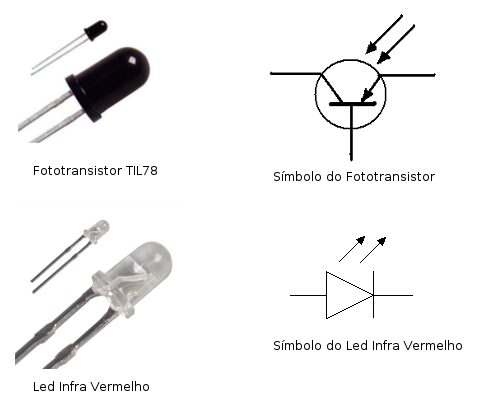
Este proyecto nos sirvió para en un futuro tener una mejor de como realizar algún proyecto, los pasos que se llevan a cabo y la forma correcta de para realizarlo, de esta manera tenemos un idea más formada del trabajo al cual se enfrenta un verdadero ingeniero.

La electrónica digital nos ofrece un vasto mundo de conocimiento sobre el mundo moderno, en el cual se pueden realizar muchísimas cosas que ya están descubiertas o que faltan por descubrir o mejorar, lo cual nos da muchas oportunidades de desarrollo a futuro.

**5.- ANEXOS.**

**Sensores**

****



GAL 22V10D



Display

