

Informe Final de Proyecto

Sistemas Digitales 1



Estacionamiento Automatizado

Mauricio Alejandro Torres Benites

Cristhian Jose Tutiven Reyes

19-1-2024

1. Introducción

Como parte de la gestión eficiente de un parqueadero que enfrenta una saturación frecuente, surge la necesidad de implementar un sistema automatizado que permita monitorear y gestionar los espacios disponibles en tiempo real para que se obtenga un flujo de coches más constante y evitar la entrada de coches de manera innecesaria debido al desconocimiento de la cantidad de estacionamientos disponibles.

La solución propuesta involucra la instalación de sensores en cada estacionamiento que detecten la ocupación de este. Estos sensores transmitirán datos en tiempo real a un sistema centralizado el cual procesará estos cambios y determinará el número de espacios disponibles que será expuesto en una pantalla en los exteriores del parqueadero, facilitando el acceso al parqueadero debido y contribuyendo a una gestión más eficiente del flujo vehicular.

2. Antecedentes/Descripción del proyecto

Como medida para evitar un flujo de coches que sature el parqueadero, se plantea la automatización del control de accesos y disponibilidad de plazas, para ellos se busca implementar un sistema que utilice sensores de proximidad en las entradas, salidas y espacios donde se estacionarán los coches, que a su vez estarán siendo contabilizados aquellos espacios en tiempo real; por ello, con ayuda de un estudiante de Sistemas Digitales 1 plantea los requerimientos a continuación:

- Se utilizarán unos sensores de proximidad en la entrada y salida del parqueadero (SE y SS) que, al recibir la señal “1” se entiende que ha detectado un objeto y, al recibir la señal “0” se entiende que no hay ningún objeto en la proximidad.
- Para la detección de un objeto se elevará la pluma que permitirá la circulación de los coches dentro del parqueadero si hay espacio disponible, en cambio si la señal es 0, la pluma se mantendrá cerrada.
- Se utilizarán unos sensores de proximidad en los estacionamientos (Sn#) que, al recibir la señal “1” se entiende que ha detectado un objeto y, al recibir la señal “0” se entiende que no hay ningún objeto en la proximidad.
- Para la detección de un objeto se enviará una señal que reste una unidad en la cantidad de espacios disponibles, y cuando no se detecte nada, la cifra de espacios disponibles se mantendrá.

- Siempre que haya un espacio disponible el estacionamiento permitirá la entrada a cualquier coche para estacionarse, y la luz de un semáforo de colores permanecerá en verde.
Si no hay espacios el semáforo marcará rojo y no se abrirá la pluma.

3. Objetivo General

Desarrollar e implementar un proceso de automatización en sistemas de control de accesos y disponibilidad en un parqueo o plazas de estacionamiento, por medio del uso de sensores y componentes con lógica combinatoria, para optimizar la gestión del parqueo y el flujo de vehículos evitando una saturación de plazas de estacionamiento.

4. Objetivos Específicos

- Diseñar e implementar un sistema que contenga sensores en entradas y salidas para gestionar el ingreso y salida de vehículos para mantener registros que muestren información usada en el monitoreo de disponibilidad en plazas de parqueo.
- Realizar y Analizar un diagrama de bloques correspondiente a lo que se usará para minimizar el tiempo de desarrollo de la solución.
- Definir las variables de entradas y salidas que permitan desarrollar el sistema automatizado.
- Crear y Presentar la tabla de verdad del bloque para encontrar una función que de paso a la solución del problema.
- Desarrollar el sistema de gestión de disponibilidad de plazas con el uso de sensores, componentes electrónicos e implementarlo en un lenguaje de descripción de hardware, como VHDL; llegando así a visualizar una simulación en tiempo real de las plazas disponibles y la cuenta de estas mismas, en el parqueo.

5. Descripción de la solución

Mediante el uso de tablas de verdad con variables previamente definidas tanto de entradas y de salidas, se buscará la función de salida que presenta la solución para el problema, teniendo en cuenta que el parqueadero cuenta con n espacios, y ciertos componentes como un semáforo, contador, plumas de entrada y salida, y sensores; las variables de entradas y salidas serán nombradas en relación con el objetivo de cada componente y lo que hará. Con la variable se desarrolla la tabla de verdad y realizando ya sea con mapas de Karnaugh o formas canónicas y algebra de Boole, la función presenta estará lo más simplificada posible; esto ayudara a la creación de un diagrama con los componentes lógicos a usar, y con

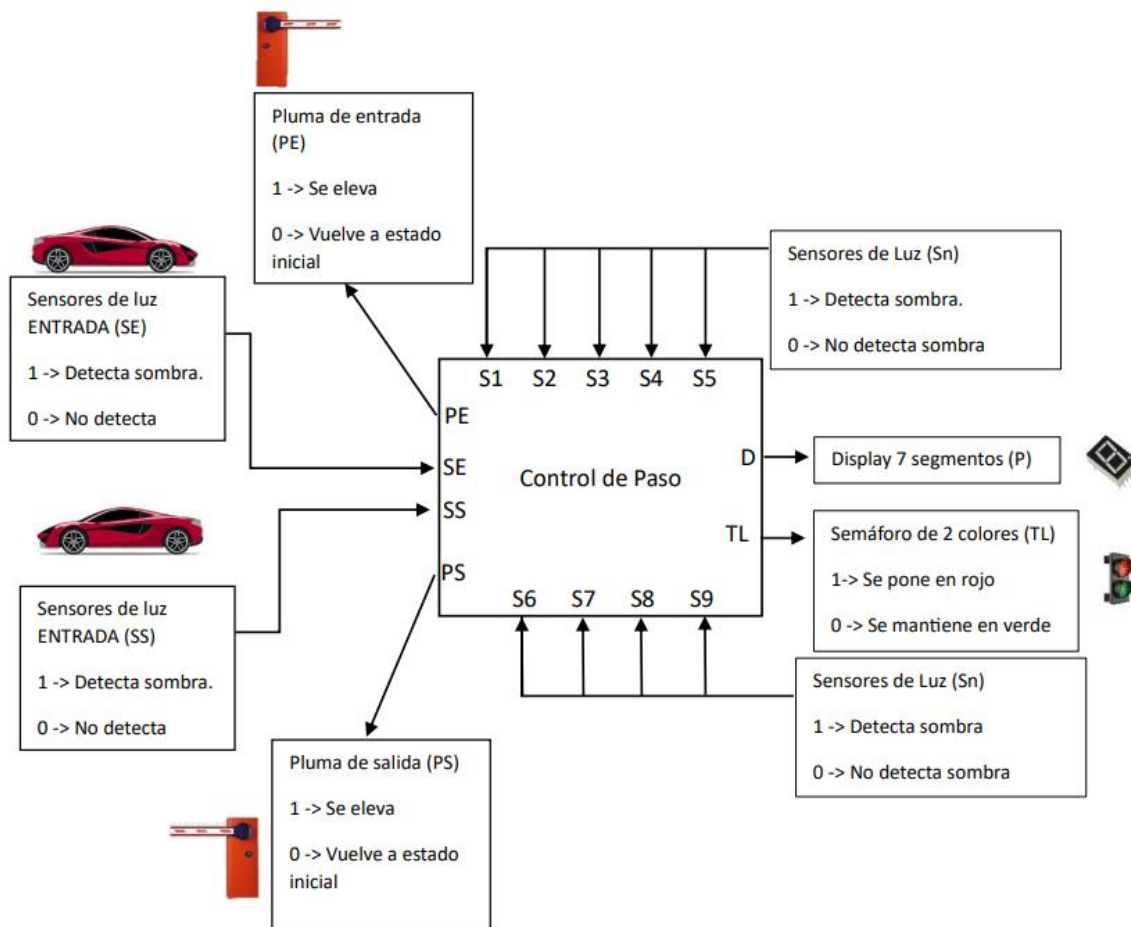
este se podrá realizar tanto la codificación en un lenguaje de descripción de hardware, usando para este caso VHDL, permitiendo implementar la solución con ayuda de varios elementos electrónicos como el FPGA.

1.1 Entradas y salidas

Entradas	
SE.H	Esta señal detecta la presencia de luz; carga el estado lógico "1" cuando el auto no permite el paso de luz y "0" cuando se detecta luz.
SS.H	Esta señal detecta la presencia de luz; carga el estado lógico "1" cuando el auto no permite el paso de luz y "0" cuando se detecta luz.
SN#.H	Esta señal detecta la presencia de luz vehículo; carga el estado lógico "1" cuando el auto sobre el sensor no permite el paso de luz y "0" cuando se detecta luz.

Salidas	
P.H	Esta señal detecta la presencia de luz; carga el estado lógico "1" cuando el auto no permite el paso de luz y "0" cuando se detecta luz.
PE.H	Esta señal detecta la presencia de luz; carga el estado lógico "1" cuando el auto no permite el paso de luz y "0" cuando se detecta luz.
PS.H	Esta señal detecta la presencia de luz vehículo; carga el estado lógico "1" cuando el auto sobre el sensor no permite el paso de luz y "0" cuando se detecta luz.
TL.H	La señal muestra el color rojo en el semáforo si todos los Sn tienen valor lógico "1", si no cumple la condición muestra el color verde.

1.2 Diagrama de bloques de la solución



1.3 Descripción de bloques

Pluma de entrada: Levanta o baja la pluma a unos 90 y 0 grados respectivamente para que el vehículo ingrese al estacionamiento o no lo haga dependiendo de la posición de la pluma.

Pluma de salida: Levanta o baja la pluma a unos 90 y 0 grados respectivamente, para que el vehículo salga o no lo haga dependiendo de la posición de la pluma en el estacionamiento.

Semáforo: Muestra dos colores que dan a conocer la disponibilidad de espacios, se coloca en color rojo para avisar que no existe alguna plaza disponible, al contrario de mostrar verde, dando a conocer si al menos una plaza se muestra disponible.

Contador: Mediante el uso de un display de 7 segmentos ayuda a visualizar la cantidad de espacios disponibles en el estacionamiento, el cual al mostrar “0” puede enviar la señal para que el semáforo cambie a color rojo.

Sensor de luz en entrada: Reconoce si existe presencia de vehículos fuera del estacionamiento en la entrada; al colocarse este sobre el sensor, envía una señal (digital) a la pluma de entrada, esto analógicamente significa que va a ingresar.

Sensor de luz en salida: Reconoce si existe presencia de vehículos dentro del estacionamiento en la salida; al colocarse este sobre el sensor, envía una señal (digital) a la pluma de salida, esto analógicamente significa que va a salir del parqueo.

Sensores de luz en plazas (Sn): Permite conocer si hay una plaza ocupada o desocupada, al momento de estar en el sitio el vehículo se encuentra sobre el sensor de luz que envía una señal, la cual analógicamente hace saber que la plaza está ocupada. Esto para cada plaza hasta la n-ésima plaza dentro del parqueo. A su vez estas señales en conjunto sirven para el contador y el semáforo.

1.3.1 Tabla de verdad del controlador

Plumas (entrada & salida).

SE	PE
0	0
1	1

SS	PS
0	0
1	1

- **Semáforo**

S1	S2	S3	S4	TLR	TLV
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

1.3.2 Expresión lógica del comportamiento del controlador

- $PE = SE$
- $PS = SS$
- $TLR = S1 * S2 * S3 * S4$
- $TLV = S1 + S2 + S3 + S4$

6. Ejemplos de funcionamiento

Ejemplo: En un estacionamiento de 4 plazas de estacionamiento, al momento de llegar un automóvil, se detendrá en la entrada, el cual, al estar sobre el sensor, permitirá que se levante la pluma dando paso a su ingreso, una vez que cruce la pluma de entrada, el vehículo se estacionara en una plaza confirmando el que esta está ocupada, haciendo que el contador muestre las plazas restantes que están desocupadas. Y así hará hasta que llegue el cuarto automóvil que, al estacionar, se mostrará el color rojo en el semáforo dando a conocer que no hay plazas disponibles; asimismo, cuando otro auto quiera ingresar la pluma no se levantará. Conforme los autos salgan del estacionamiento, se acercará a la pluma de salida, deteniéndose sobre el sensor en la salida, permitiendo levantar la pluma y además el contador sumara un espacio disponible, realizando el proceso hasta su capacidad máxima

Caso	Vehículos que ingresan	Sensores(sn) espacios ocupados	Espacios disponibles	Valores lógicos				
				Semáforo en rojo	Sensores(sn) (Plazas)			
					S4	S3	S2	S1
1	0	0	4	0	0	0	0	0
2	1	1	3	0	0	0	0	1
3	2	2	2	0	0	0	1	1
4	3	3	1	0	0	1	1	1
5	4	4	0 (full)	1	1	1	1	1

Caso 1: ingresan “0” vehículos, por lo que todos los espacios están disponibles, por ende, en el semáforo no se enciende el rojo (permanece verde), los sensores en las plazas presentaran un valor lógico de “0”.

Caso 2: ingresa “1” vehículo, por lo que de todos los espacios 3 están disponibles, por ende, en el semáforo no se enciende el rojo (permanece verde). También 1 de los sensores en las plazas presentaran un valor lógico de “1”.

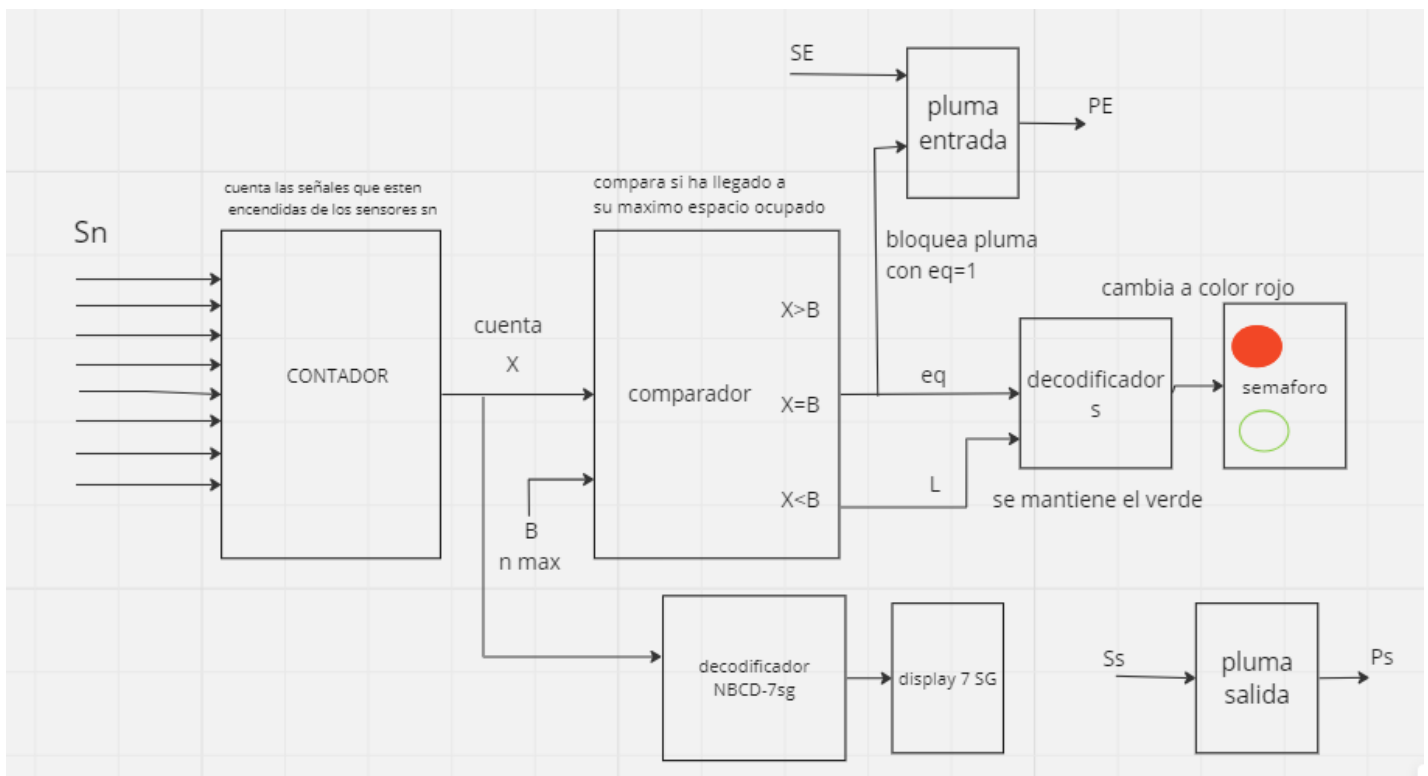
Caso 3: ingresan “2” vehículos, por ende, habrán 2 disponibles, así en el semáforo no se encenderá el rojo (permanece verde). También 2 de los sensores en las plazas presentaran un valor lógico de “1”.

Caso 4: ingresan “3” vehículos, por lo que existe 1 espacio disponible, así en el semáforo no se encenderá el rojo (permanece verde). También 3 de los sensores en las plazas presentaran un valor lógico de “1”.

Caso 5: ingresan “4” vehículos, por lo que todos los espacios están llenos, por ende, en el semáforo se enciende el rojo es decir se la salida ser un 1 lógico. También todos los sensores en las plazas presentaran un valor lógico de “1”.

En otros casos al ingreso, una vez lleno el parqueo en su totalidad, no se permitirá que la pluma de entrada se levante hasta que existe algún espacio desocupado, esto lo verificará cuando algún sensor en las plazas tenga valor lógico “0”.

7. Diagrama de Bloques



8. VHDL y Compilación.

- Contador, Decodificador, Display 7 segmentos

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use ieee.numeric_std.all;
4
5 entity contador_senales is
6   Port (
7     entradas : in  STD_LOGIC_VECTOR (14 downto 0);
8     salida  : out std_logic_vector(3 downto 0);
9     semaforo : out std_logic
10  );
11 end contador_senales;
12
13 architecture Behavioral of contador_senales is
14 begin
15   process(entradas)
16     variable temp_contador : integer := 0; -- usar variable local
17     begin
18       temp_contador := 0; -- inicializar el contador fuera del bucle
19
20       for i in 0 to 14 loop
21         if entradas(i) = '1' then
22           temp_contador := temp_contador + 1;
23         end if;
24       end loop;
25       if temp_contador = 15 then
26         semaforo <= '1';
27       end if;
28       salida <= std_logic_vector(to_unsigned(temp_contador, 4));
29     end process;
30 end Behavioral;
```

Ilustración 1: Contador de las señales de los sensores.

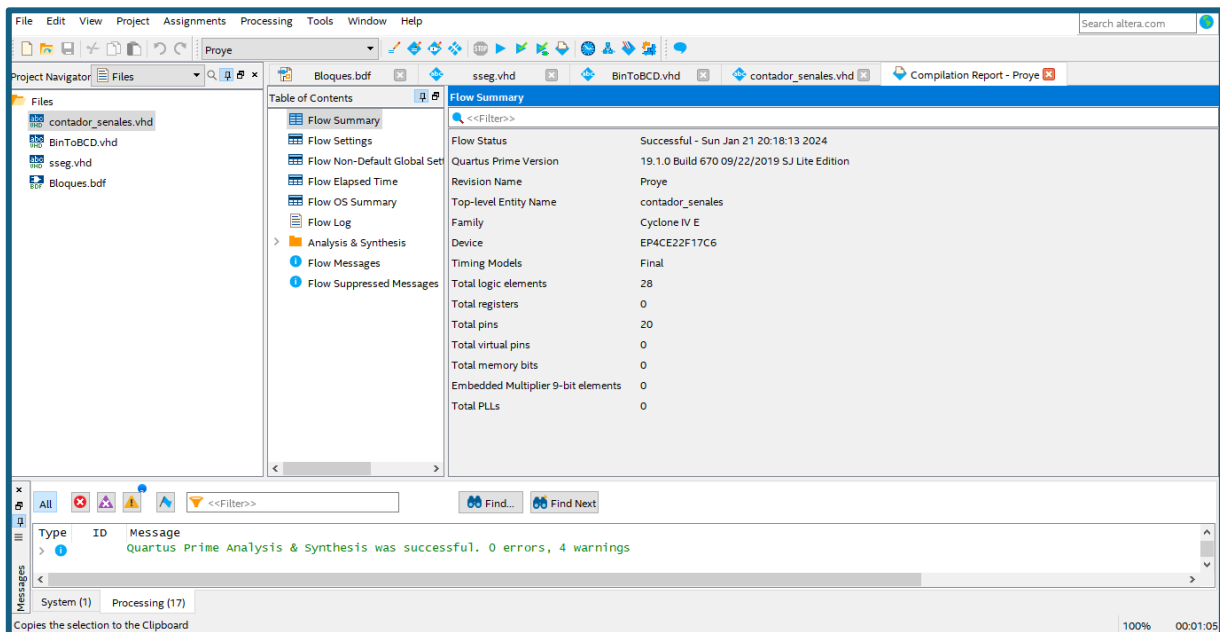


Ilustración 2: Compilación del contador.

- Binario a BCD

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.numeric_std.all;
4
5 entity BinToBCD is
6     port (
7         entrada : in  STD_LOGIC_VECTOR (3 downto 0);
8         decena   : out STD_LOGIC_VECTOR (3 downto 0);
9         unidad   : out STD_LOGIC_VECTOR (3 downto 0);
10    );
11 end BinToBCD;
12
13 architecture Behavioral of BinToBCD is
14     signal binario : STD_LOGIC_VECTOR (3 downto 0);
15     signal resta   : integer range 0 to 9;
16 begin
17     process(entrada)
18     begin
19         binario <= entrada;
20
21         if to_integer(unsigned(binario)) <= 9 then
22             decena <= "0000";
23             unidad <= std_logic_vector(to_unsigned(to_integer(unsigned(binario)), 4));
24         elsif to_integer(unsigned(binario)) <= 19 then
25             decena <= "0001";
26             resta <= to_integer(unsigned(binario)) - 10;
27             unidad <= std_logic_vector(to_unsigned(resta, 4));
28         else
29             decena <= "XXXX";
30             unidad <= "XXXX";
31         end if;
32     end process;
33 end Behavioral;

```

Ilustración 3: convertidor de Binario a BCD

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Set
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
 - Flow Messages
 - Flow Suppressed Messages

Flow Summary

Flow Status	
Flow Status	Successful - Sun Jan 21 20:23:33 2024
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	Proye
Top-level Entity Name	BinToBCD
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	11
Total registers	0
Total pins	12
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Messages

Type	ID	Message
Information		Quartus Prime Analysis & Synthesis was successful. 0 errors, 17 warnings

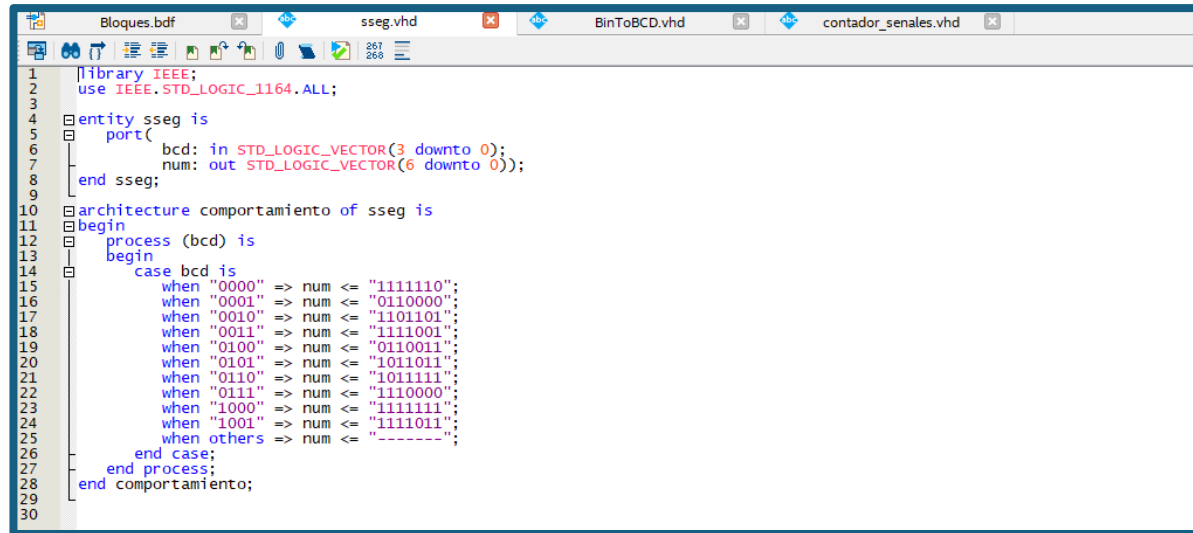
System (2) Processing (28)

Creates a new file

100% 00:00:56

Ilustración 4: Compilación de convertidor de Binario a BCD

- Decodificador BCD a 7 segmentos



```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity sseg is
5  port(
6      bcd: in STD_LOGIC_VECTOR(3 downto 0);
7      num: out STD_LOGIC_VECTOR(6 downto 0));
8  end sseg;
9
10 architecture comportamiento of sseg is
11 begin
12     process (bcd) is
13     begin
14         case bcd is
15             when "0000" => num <= "1111110";
16             when "0001" => num <= "0110000";
17             when "0010" => num <= "1101101";
18             when "0011" => num <= "1111001";
19             when "0100" => num <= "0110011";
20             when "0101" => num <= "1011011";
21             when "0110" => num <= "1011111";
22             when "0111" => num <= "1110000";
23             when "1000" => num <= "1111111";
24             when "1001" => num <= "1111011";
25             when others => num <= "-----";
26         end case;
27     end process;
28 end comportamiento;
29
30

```

Ilustración 5: Decodificador de BCD a 7 Segmentos para uso del display.

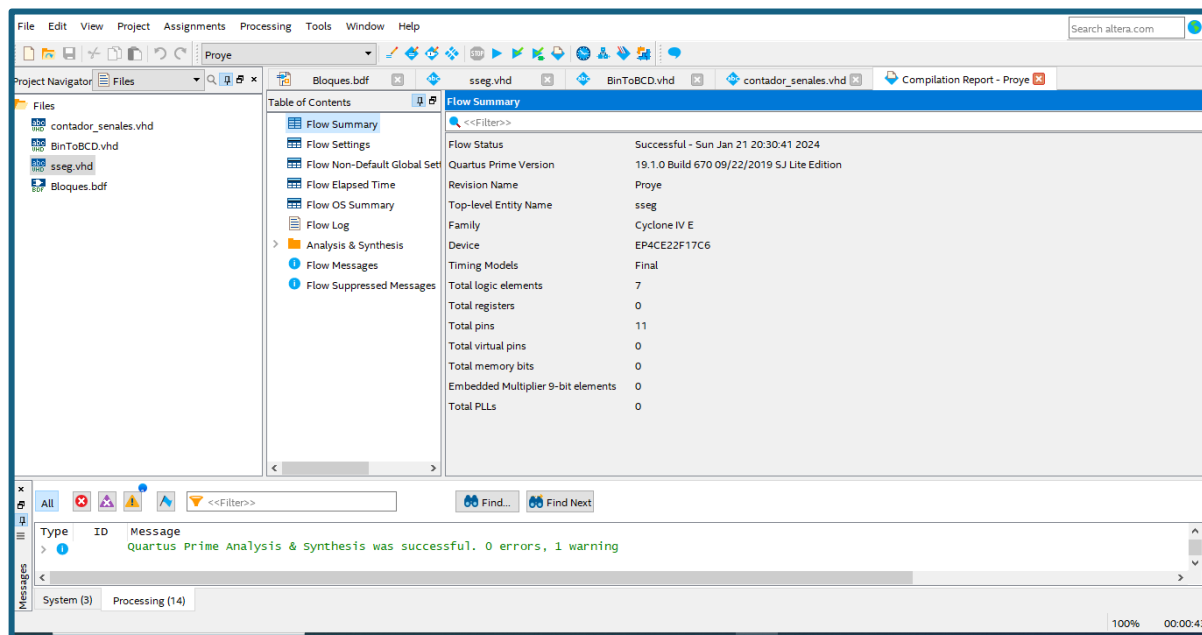


Ilustración 6: Compilación del decodificador BDC a 7segmentos.

- **Semáforo**

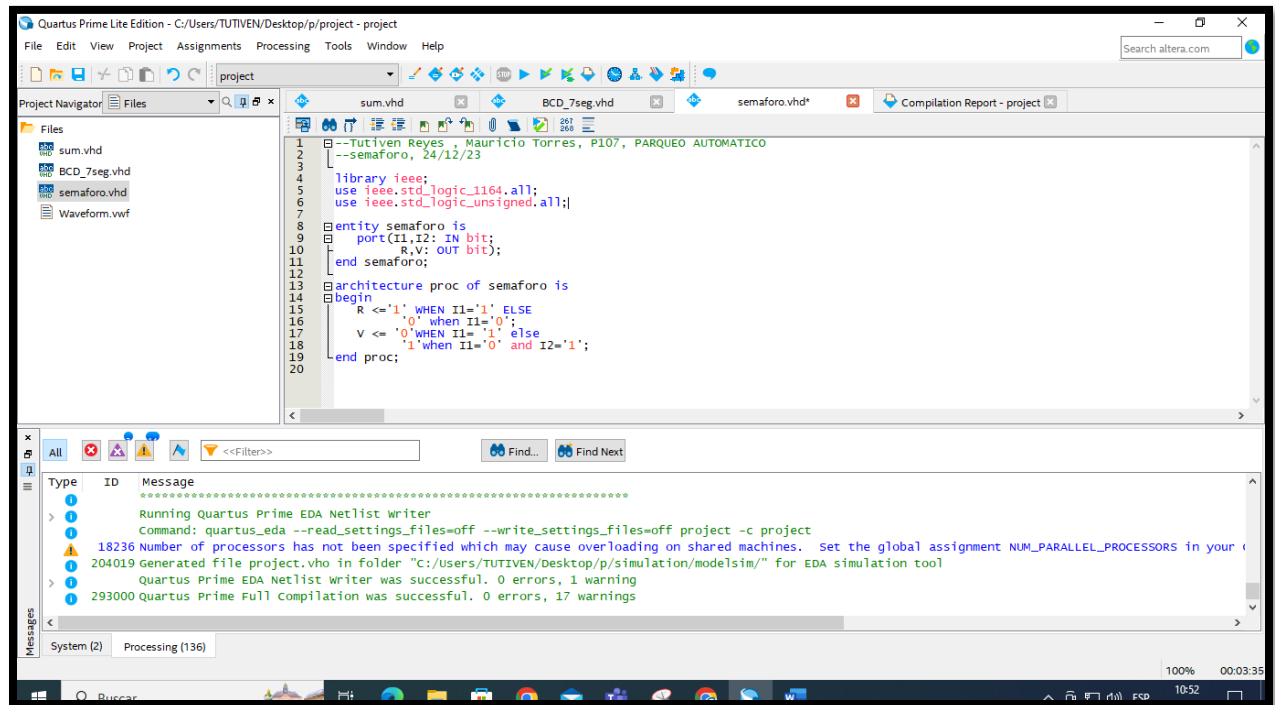
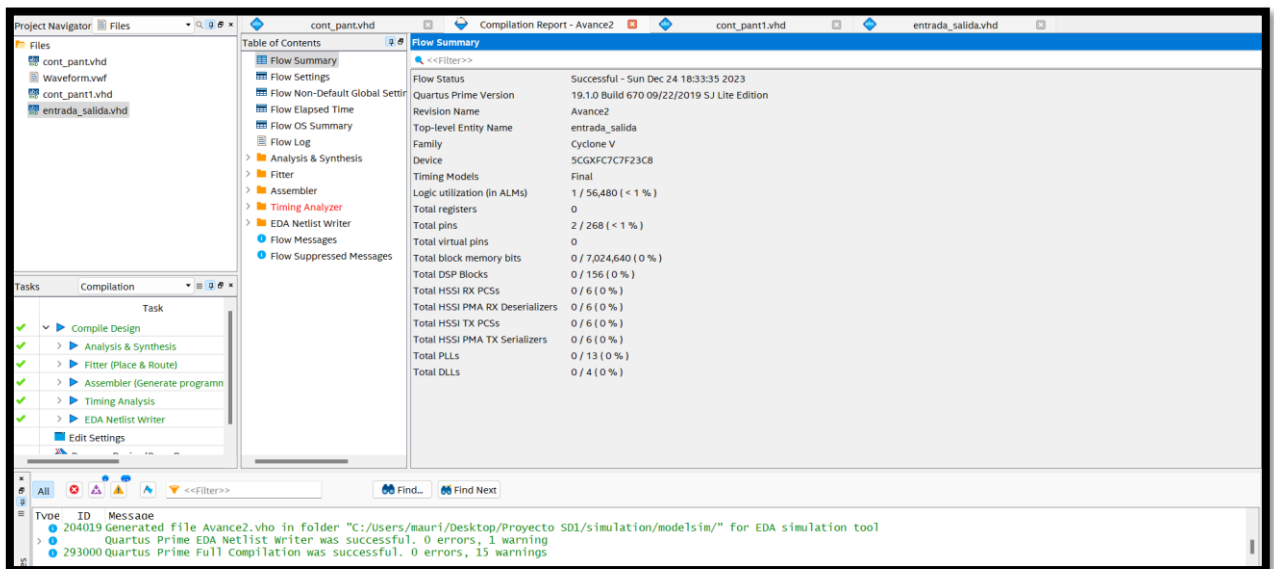
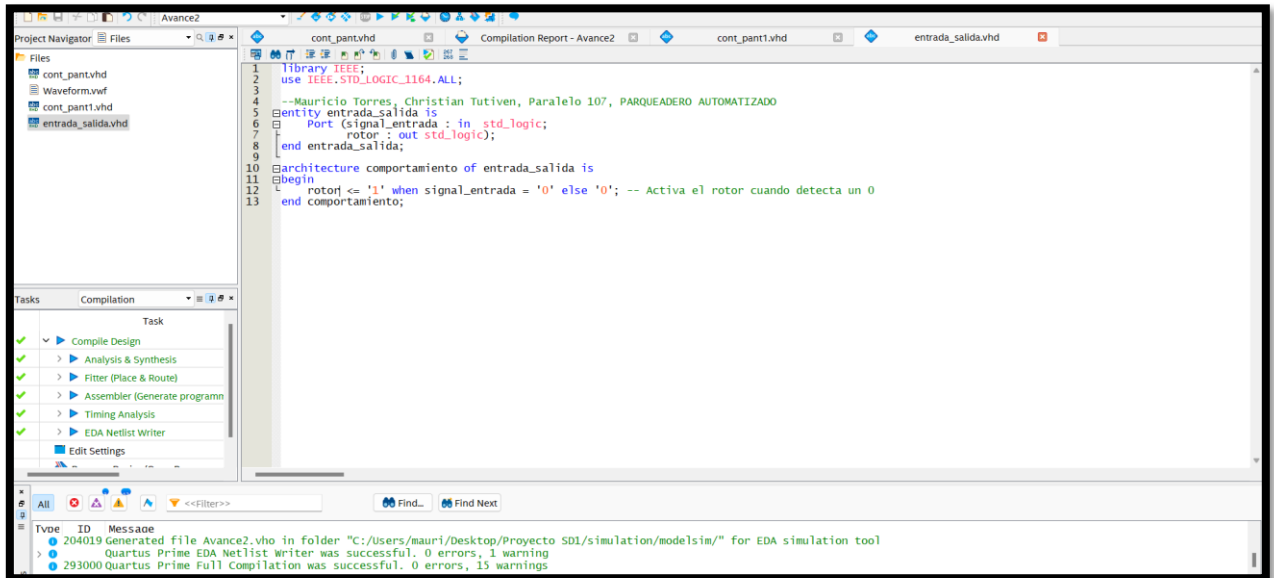
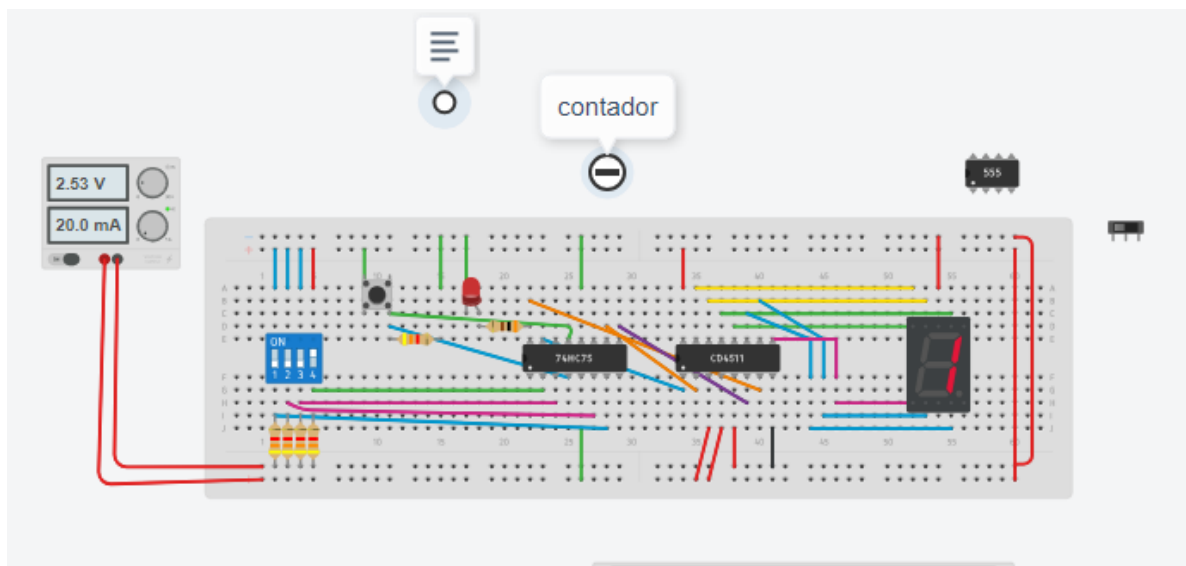
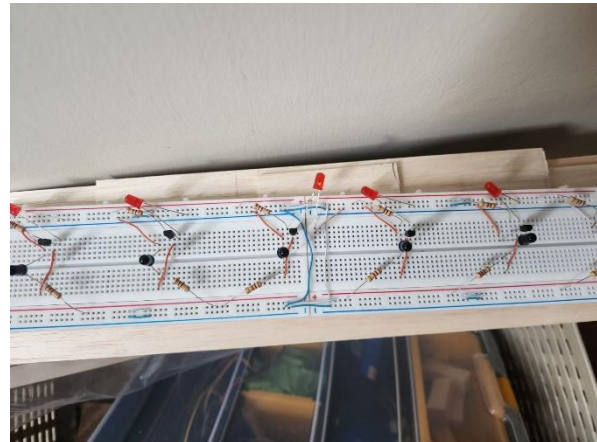
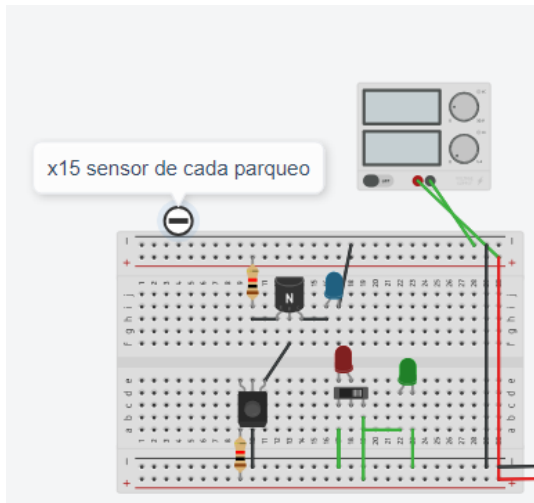


Ilustración 7: Semáforo, señalética para conocer datos de espacios de parqueo.

- **Pluma de Salida**



9. Esquema de conexión de periféricos a la FPGA



10. POSTER



Facultad de Ingeniería en
Electrónica y Computación

Cristhian José Talamán Bayes
cristi@espol.edu.ec



Mauro Alejandro Torres Rosales
mauro@espol.edu.ec





Estacionamiento Automatizado

PROBLEMA

La problemática presente, es la falta de un mejor control de sistemas de un parqueo disponible de una edificación, dentro de estos se puede llegar a pasar por momentos de tensión y estrés al buscar espacios disponibles, además del tiempo que conlleva hacerlo en casos de que este saturado y sea un área de grandes dimensiones. Asimismo, la falta de señales o indicadores que permitan conocer a los usuarios estos datos pueden llevar a una mala experiencia en el lugar.

OBJETIVO GENERAL

- Desarrollar e implementar un proceso de automatización en sistemas de control de accesos y disponibilidad en un parqueo o plazas de estacionamiento, por medio del uso de sensores y componentes con lógica combinatoria, para optimizar la gestión del parqueo y el flujo de vehículos evitando una saturación de plazas de estacionamiento.



METODOLOGÍA

Investigación y análisis previo

Diseño

Programación de los bloques

Implementación del VHDL

Pruebas y validación

Mejoras y optimización





RESULTADOS

- Control y visualización de espacios disponibles, mediante el uso del display de 7 segmentos, partiendo de las señales que envían los sensores de los puentes, mostrando la cantidad de estos espacios de parqueo.
- Envío de señal al led en representación del semáforo, verificando así la no disponibilidad de espacios de parqueo.
- Control en orden ascendente los puentes de parqueo y entrada del estacionamiento, así como la salida.



CONCLUSIONES

- En base a los resultados obtenidos, se ha realizado un exitoso desarrollo e implementación de un proceso de automatización en los sistemas de control de accesos y disponibilidad de plazas de parqueo, mediante el uso de componentes de lógica combinatoria, permitiendo así un monitoreo preciso y en tiempo real de la disponibilidad de dichas plazas de estacionamiento, lo cual contribuye al desarrollo de soluciones innovadoras, para evitar la congestión vehicular y facilitando el acceso equitativo a los usuarios.

REFERENCIAS

Escuela Superior Politécnica de Ingeniería de Esmeraldas (ESPOL). (2023). <https://www.espol.edu.ec/>

Universidad de Esmeraldas. (2023). <https://www.universidadezmeraldas.edu.ec/>