

Laboratorio de Electrónica 3: Fase 2

Control de tráfico y paso peatonal salida de la U*

Cindy Melissa Gatica Arriola, 201709692,^{1,**} Alejandro José Luncey Contreras, 202112396,^{1,***} and Angel Efrén de León Orozco, 202107891^{1,****}

¹Facultad de Ingeniería, Universidad de San Carlos,
Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.

El proyecto consiste en la realización de un controlador de semáforo para la salida de la universidad dirigido mediante la tarjeta FPGA Elbert Spartan 3A.

I. OBJETIVOS

A. Generales

- Desarrollar habilidades y destrezas de desarrollo de código fuente.

B. Específicos

- * Que el estudiante aplique los conocimientos adquiridos en favor de la problemática nacional y mundial.
- * Desarrolle un proyecto de laboratorio con aplicación a optimización de procesos.

II. CIRCUITOS PROPUESTOS

A. LEDS como semáforos

Se planea utilizar resistores de 330Ω y led de color: rojo, verde y amarillo para simular el semáforo de la entrada y salida, tanto para los carros, como para los peatones.



Figura 1: Semáforo de entrada, salida y de peatones

B. Circuito de decisión

Para reconocer si se trata de día, de tarde/ noche, se necesita un circuito que active si se trata de uno de estos casos. Para ello, se utilizará dos push buttons, uno para la mañana y otro para la tarde.

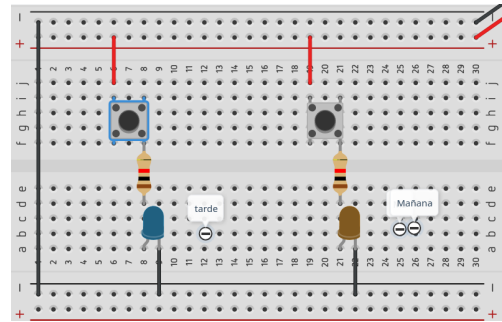


Figura 2: Push buttons para la decisión de mañana o tarde

III. LISTA DE IDEAS

Los materiales que se utilizarán son los siguientes

A. Materiales

- * Tarjeta FPGA Elbert Spartan 3A.
- * Protoboard.
- * Leds de colores para simular los semáforos.
- * Resistencias de 330Ω

B. Funciones recicladas

En el diagrama de flujo realizado, se puede notar que se pueden crear tres principales funciones:

- Función de luz verde
- Función de luz amarilla
- Función de luz roja

* Laboratorios de Electrónica

** e-mail: 2787947930101@ingenieria.usac.edu.gt

*** e-mail: 2978391790101@ingenieria.usac.edu.gt

**** e-mail: 3300648001201@ingenieria.usac.edu.gt

En cada una de ellas, el patrón se repite, indistintamente si es entrada o salida, si es mañana o noche. Por lo tanto, lo único que varía en algunos casos es el tiempo. Para abordar este tema, se utilizarán tres funciones que tendrán como parámetro el "tiempo", el cual variará según las siguientes tablas dadas por la ingeniera a cargo

Mañana		
Color	Entrada	Salida
Verde	20s	10s
Amarillo	5s	5s
Rojo	20s	20s

Figura 3: Tiempo ingresado a las funciones para la mañana

Tarde-Noche		
Color	Entrada	Salida
Verde	15s	20s
Amarillo	5s	5s
Rojo	20s	20s

Figura 4: Tiempo ingresado a las funciones para la tarde

C. Uso de señal de reloj

Se utilizará una señal de reloj con un ciclo de trabajo del 50 % para poder simular los segundos transcurridos desde el momento del cambio en la luz del semáforo.

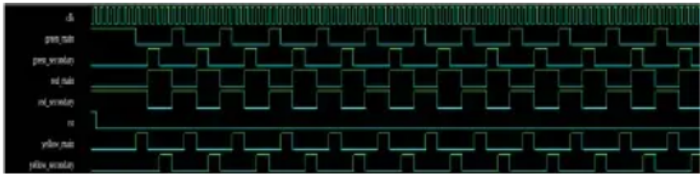


Figura 5: señal de reloj

IV. DUDAS

- ¿El buffer debe sonar a partir de 9s que le quedan de los 15 s o de los 20 s?
- ¿Salida y entrada deben estar sincronizados?
- ¿Para simular tarde o día debemos utilizar algún circuito accionador, como push buttoms o LDR?
- En caso de que no se puedna utilizar push buttoms como actuadores del proyecto, es permitido utilizar cualquier tipo de sensor? por ejemplo, una ldr?

- La alarma se simulará como una señal de reloj o se puede utilizar una alarma externa?

V. CRONOGRAMA

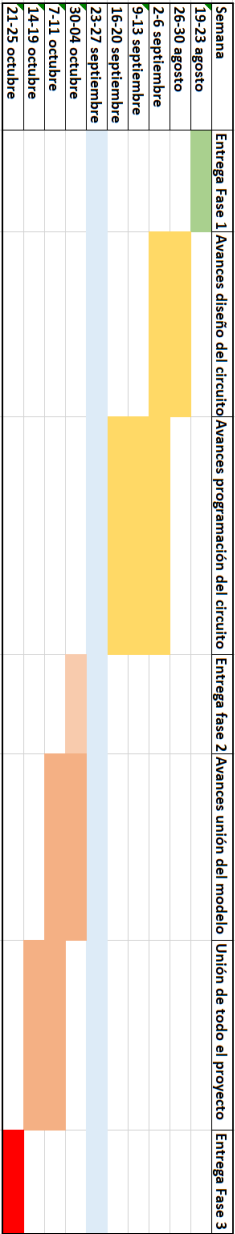


Figura 6: Cronograma de actividades. Elaboración propia, 2024.

VI. DIAGRAMA DE BLOQUES

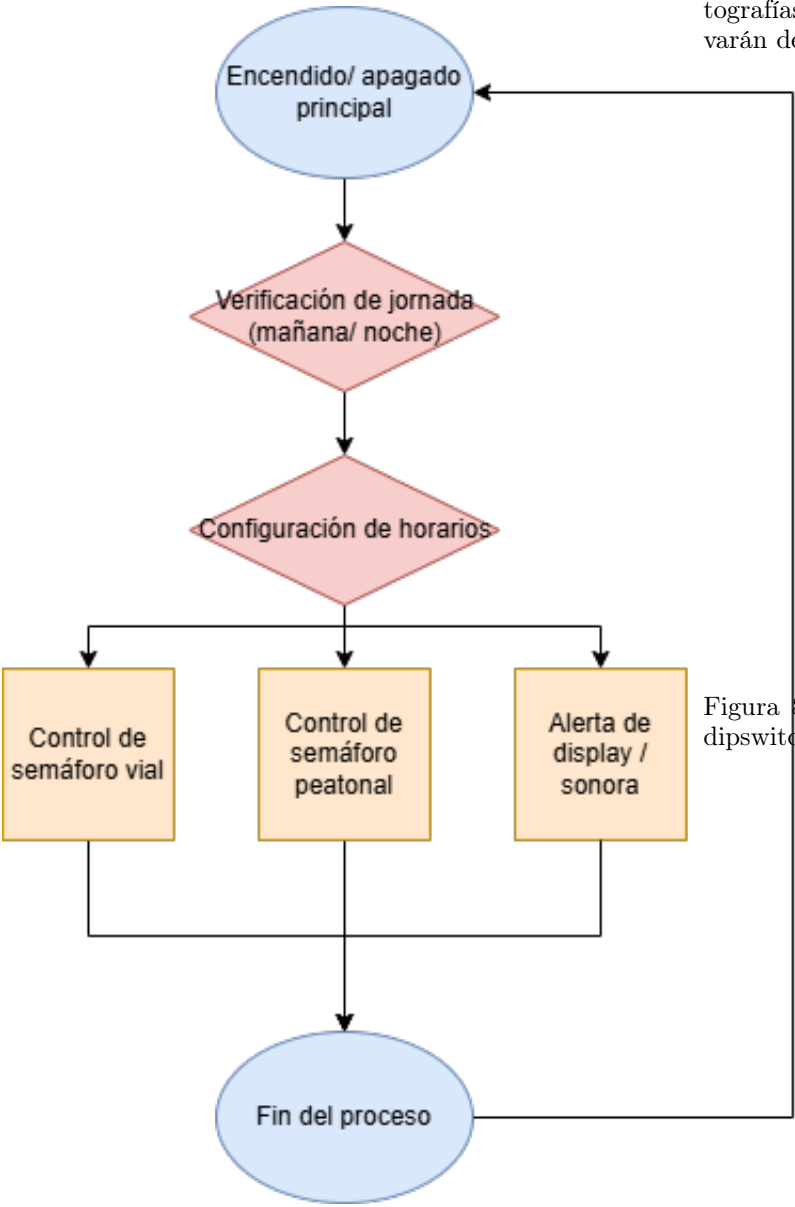


Figura 7: Cronograma de actividades. Elaboración propia, 2024.

Se implementó un esquema equivalente, el cual se muestra a continuación. También disponible en el siguiente link de drive o en el siguiente url https://drive.google.com/file/d/1MerjsydelneNC1iF1-szF0GCotFSZ3n4/view?usp=drive_link

VII. FOTOGRAFÍAS DE LOS AVANCES

Se muestran los avances obtenidos en las siguientes fotografías, en las cuales se puede observar cómo los leds varán de acuerdo al estado que les corresponde.

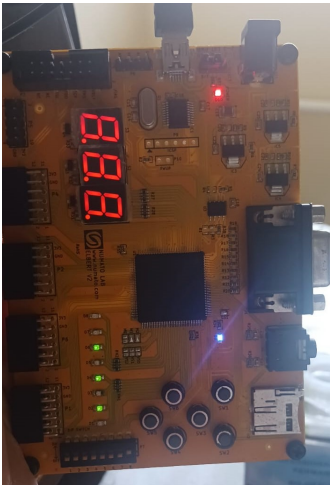


Figura 8: LEDs encendidos cuando el estado es alto y dipswitch presionado para mañana.
Fuente: Elaboración propia, 2024

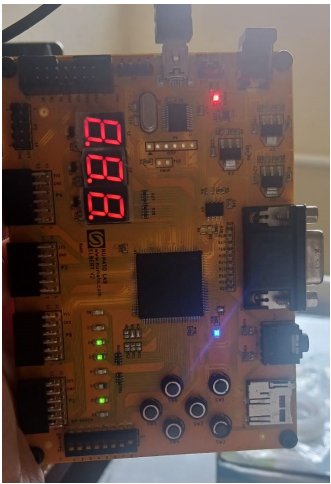


Figura 9: LEDs encendidos cuando el estado es alto y dipswitch presionado para mañana.
Fuente: Elaboración propia, 2024

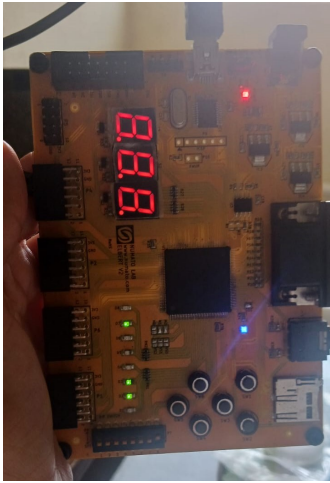


Figura 10: LEDs encendidos cuando el estado es alto y dipswitch presionado para mañana.

Fuente: Elaboración propia, 2024

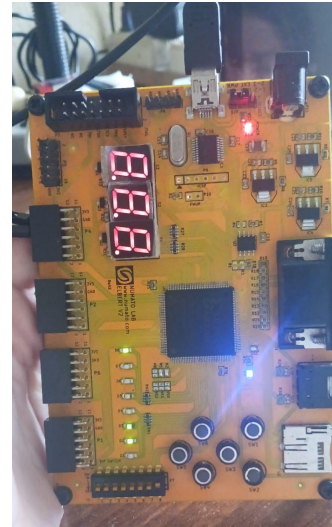


Figura 11: LEDs encendidos cuando el estado es alto y dipswitch presionado para mañana.

Fuente: Elaboración propia, 2024

VIII. CÓDIGO IMPLEMENTADO

[1] Grossman, S. (Segunda edición). (1987). *Álgebra lineal*. México: Grupo Editorial Iberoamericana.

[2] Anónimo. *I-V Characteristic Curves* [En línea][25 de octubre de 2012]. Disponible en: <http://www.electronics-tutorials.ws/blog/i-v-characteristic-curves.html>

VIII. CÓDIGO IMPLEMENTADO

```
--Este codigo simula los semaforos de entrada y salida de los carros. Tanto para mañana  
como para noche  
--Se intento agregar el semaforo de la salida de la mañana para los peatones, aun no se  
ha implementado para la entrada y en la tarde.  
-- Falta agregar buzzer, display  
-- El unico error es que el tiempo en rojo de los peatones y el tiempo en rojo de los carros  
es el mismo, 20 s  
-- para mañana dip switch 8 y para la tarde dip switch 7  
-- led de entrada D8 verde, D7 amarillo, D6 rojo CARROS  
-- led de salida D5 verde, D4 amarillo, D3 rojo CARROS  
-- led de salida peatones D2 verde, D1 rojo
```

```
library IEEE;  
use IEEE.std_logic_1164.all;  
use IEEE.numeric_std.all;
```

```
entity Semaforo is
```

```
    Port(  
        Clk: in std_logic;  
        Selector_manana, Selector_tarde: in std_logic;  
        Verde_carros_entrada, Amarillo_carros_entrada, Rojo_carros_entrada: out std_logic;  
        Verde_carros_salida, Amarillo_carros_salida, Rojo_carros_salida: out std_logic;  
        Verde_peatones_salida, Rojo_peatones_salida: out std_logic  
    );
```

```
end Semaforo;
```

```
architecture Behavioral of Semaforo is
```

```
    -- Definir los estados del semáforo  
    type FSM is (EstadoVerde_carros_entrada, EstadoAmarillo_carros_entrada,  
EstadoRojo_carros_entrada);  
    type FSM_salida is (EstadoVerde_carros_salida, EstadoAmarillo_carros_salida,  
EstadoRojo_carros_salida);  
    type FSM_peatones is (EstadoVerde_peatones_salida,  
EstadoRojo_peatones_salida);
```

```
    -- Señales para almacenar el estado actual del semáforo  
    signal Estado_Semaforo_Entrada_Carros: FSM := EstadoVerde_carros_entrada; --  
Inicializamos en EstadoVerde  
    signal Estado_Semaforo_Salida_Carros: FSM_salida := EstadoVerde_carros_salida; --  
Inicializamos en EstadoVerde  
    signal Estado_Semaforo_Peatones_Salida: FSM_peatones :=  
EstadoRojo_peatones_salida; -- Inicializamos peatones en rojo
```

```
    -- Contadores para medir los tiempos en cada estado  
    signal Conteo_salida: integer range 0 to 240000000 := 0; -- Contador con un rango  
apropiado Salida  
    signal Conteo_entrada: integer range 0 to 240000000 := 0; -- Contador con un rango
```

```

apropiado Entrada
signal Conteo_peatones: integer range 0 to 240000000 := 0; -- Contador para
peatones salida

-- Señal para indicar si el semáforo de salida de carros está en rojo
signal Salida_Carros_Rojo: std_logic := '0';

begin

-- Proceso para controlar el semáforo
process(Clk)
begin
    if rising_edge(Clk) then
        -- Si ambos switches están en la posición incorrecta, desactivar semáforos
        if (Selector_manana = '1' and Selector_tarde = '1') or (Selector_manana = '0' and
Selector_tarde = '0') then
            Verde_carros_entrada <= '0';
            Amarillo_carros_entrada <= '0';
            Rojo_carros_entrada <= '0';
            Verde_carros_salida <= '0';
            Amarillo_carros_salida <= '0';
            Rojo_carros_salida <= '0';
            Verde_peatones_salida <= '0';
            Rojo_peatones_salida <= '0';

        else
            -- Modo mañana (Selector_manana activado, tarde desactivado)
            if Selector_manana = '0' and Selector_tarde = '1' then -- Inversión lógica si es
activo en bajo
                -- Semáforo de entrada
                case Estado_Semaforo_Entrada_Carros is
                    when EstadoVerde_carros_entrada =>
                        if (Conteo_entrada < 2*120000000) then
                            Conteo_entrada <= Conteo_entrada + 1;
                            Verde_carros_entrada <= '1';
                            Amarillo_carros_entrada <= '0';
                            Rojo_carros_entrada <= '0';
                        else
                            Conteo_entrada <= 0;
                            Verde_carros_entrada <= '0';
                            Estado_Semaforo_Entrada_Carros <=
EstadoAmarillo_carros_entrada;
                        end if;

                    when EstadoAmarillo_carros_entrada =>
                        if (Conteo_entrada < 60000000) then
                            Conteo_entrada <= Conteo_entrada + 1;
                            Verde_carros_entrada <= '0';
                            Amarillo_carros_entrada <= '1';
                            Rojo_carros_entrada <= '0';
                        else
                            Conteo_entrada <= 0;
                            Amarillo_carros_entrada <= '0';
                            Estado_Semaforo_Entrada_Carros <= EstadoRojo_carros_entrada;

```

```

end if;

when EstadoRojo_carros_entrada =>
  if (Conteo_entrada < 2*120000000) then
    Conteo_entrada <= Conteo_entrada + 1;
    Verde_carros_entrada <= '0';
    Amarillo_carros_entrada <= '0';
    Rojo_carros_entrada <= '1';
  else
    Conteo_entrada <= 0;
    Rojo_carros_entrada <= '0';
    Estado_Semaforo_Entrada_Carros <= EstadoVerde_carros_entrada;
  end if;

when others =>
  Estado_Semaforo_Entrada_Carros <= EstadoVerde_carros_entrada;
end case;

-- Semáforo de salida
case Estado_Semaforo_Salida_Carros is
when EstadoVerde_carros_salida =>
  if (Conteo_salida < 2*60000000) then
    Conteo_salida <= Conteo_salida + 1;
    Verde_carros_salida <= '1';
    Amarillo_carros_salida <= '0';
    Rojo_carros_salida <= '0';
  else
    Conteo_salida <= 0;
    Verde_carros_salida <= '0';
    Estado_Semaforo_Salida_Carros <= EstadoAmarillo_carros_salida;
  end if;

when EstadoAmarillo_carros_salida =>
  if (Conteo_salida < 60000000) then
    Conteo_salida <= Conteo_salida + 1;
    Verde_carros_salida <= '0';
    Amarillo_carros_salida <= '1';
    Rojo_carros_salida <= '0';
  else
    Conteo_salida <= 0;
    Amarillo_carros_salida <= '0';
    Estado_Semaforo_Salida_Carros <= EstadoRojo_carros_salida;
  end if;

when EstadoRojo_carros_salida =>
  if (Conteo_salida < 2*120000000) then
    Conteo_salida <= Conteo_salida + 1;
    Verde_carros_salida <= '0';
    Amarillo_carros_salida <= '0';
    Rojo_carros_salida <= '1';

```

```

Salida_Carros_Rojo <= '1';

```

```

else

```

```

        Conteo_salida <= 0;
        Rojo_carros_salida <= '0';

Salida_Carros_Rojo <= '0';
        Estado_Semaforo_Salida_Carros <= EstadoVerde_carros_salida;
    end if;

                                when others =>
        Estado_Semaforo_Salida_Carros <= EstadoVerde_carros_salida;

    end case;

----- parte de los peatones -----
                                case
Estado_Semaforo_Peatones_Salida is
                                when
EstadoRojo_peatones_salida =>
        --if (Salida_Carros_Rojo = '1') and (Conteo_peatones < 2*60000000)
then -- Si el semáforo de carros está en rojo
                                if (Salida_Carros_Rojo
= '1') then -- Si el semáforo de carros está en rojo
        Conteo_peatones <= 0; -- Reiniciar el contador
        Estado_Semaforo_Peatones_Salida <=
EstadoVerde_peatones_salida;
        else
            -- Mantenerse en rojo
            Verde_peatones_salida <= '0';
            Rojo_peatones_salida <= '1';
        end if;

        when EstadoVerde_peatones_salida =>
            if (Conteo_peatones < 2*90000000) then -- 15 segundos
                Conteo_peatones <= Conteo_peatones + 1;
                Verde_peatones_salida <= '1';
                Rojo_peatones_salida <= '0';
            else
                Conteo_peatones <= 0;
                Verde_peatones_salida <= '0';
                Estado_Semaforo_Peatones_Salida <= EstadoRojo_peatones_salida;
            end if;

                                when others =>
        Estado_Semaforo_Peatones_Salida <= EstadoRojo_peatones_salida;

                                end case;

-----
-- Modo tarde (Selector_tarde activado, mañana desactivado)
elsif Selector_manana = '1' and Selector_tarde = '0' then
    case Estado_Semaforo_Entrada_Carros is
        when EstadoVerde_carros_entrada =>
            if (Conteo_entrada < 2*90000000) then
                Conteo_entrada <= Conteo_entrada + 1;
                Verde_carros_entrada <= '1';

```



```

        Amarillo_carros_entrada <= '0';
        Rojo_carros_entrada <= '0';
    else
        Conteo_entrada <= 0;
        Verde_carros_entrada <= '0';
        Estado_Semaforo_Entrada_Carros <=
EstadoAmarillo_carros_entrada;
    end if;

    when EstadoAmarillo_carros_entrada =>
        if (Conteo_entrada < 60000000) then
            Conteo_entrada <= Conteo_entrada + 1;
            Verde_carros_entrada <= '0';
            Amarillo_carros_entrada <= '1';
            Rojo_carros_entrada <= '0';
        else
            Conteo_entrada <= 0;
            Amarillo_carros_entrada <= '0';
            Estado_Semaforo_Entrada_Carros <= EstadoRojo_carros_entrada;
        end if;

    when EstadoRojo_carros_entrada =>
        if (Conteo_entrada < 2*120000000) then
            Conteo_entrada <= Conteo_entrada + 1;
            Verde_carros_entrada <= '0';
            Amarillo_carros_entrada <= '0';
            Rojo_carros_entrada <= '1';
        else
            Conteo_entrada <= 0;
            Rojo_carros_entrada <= '0';
            Estado_Semaforo_Entrada_Carros <= EstadoVerde_carros_entrada;
        end if;

    when others =>
        Estado_Semaforo_Entrada_Carros <= EstadoVerde_carros_entrada;
end case;

-- Semáforo de salida
case Estado_Semaforo_Salida_Carros is
    when EstadoVerde_carros_salida =>
        if (Conteo_salida < 2*120000000) then
            Conteo_salida <= Conteo_salida + 1;
            Verde_carros_salida <= '1';
            Amarillo_carros_salida <= '0';
            Rojo_carros_salida <= '0';
        else
            Conteo_salida <= 0;
            Verde_carros_salida <= '0';
            Estado_Semaforo_Salida_Carros <= EstadoAmarillo_carros_salida;
        end if;

    when EstadoAmarillo_carros_salida =>
        if (Conteo_salida < 60000000) then
            Conteo_salida <= Conteo_salida + 1;

```

```

        Verde_carros_salida <= '0';
        Amarillo_carros_salida <= '1';
        Rojo_carros_salida <= '0';
    else
        Conteo_salida <= 0;
        Amarillo_carros_salida <= '0';
        Estado_Semaforo_Salida_Carros <= EstadoRojo_carros_salida;
    end if;

    when EstadoRojo_carros_salida =>
        if (Conteo_salida < 2*120000000) then
            Conteo_salida <= Conteo_salida + 1;
            Verde_carros_salida <= '0';
            Amarillo_carros_salida <= '0';
            Rojo_carros_salida <= '1';
        else
            Conteo_salida <= 0;
            Rojo_carros_salida <= '0';
            Estado_Semaforo_Salida_Carros <= EstadoVerde_carros_salida;
        end if;

        when others =>
            Estado_Semaforo_Salida_Carros <= EstadoVerde_carros_salida;
        end case;
    end if;
end if;
end process;
end Behavioral;

```