



Universidad Nacional Autónoma de México
Facultad de Ingeniería
Ingeniería en Computación



Proyecto Final – Reporte Técnico

“ Sistema de venta de boletos del cine en una tarjeta FPGA”



Asignatura:	<u>Diseño Digital Moderno</u>
Clave asignatura:	<u>1645</u>
Profesor:	<u>M.I. Norma Elva Chávez Rodríguez</u>
Grupo:	<u>1</u>
Alumno:	<u>Ordiales Caballero, Iñaky</u>
Fecha de entrega:	<u>15 de diciembre del 2021.</u>

Semestre 2022-1

Introducción

Como parte de la evaluación de la asignatura de diseño digital moderno se tiene que realizar un proyecto final. Éste contará con una parte práctica entregada en formato de video y una parte escrita entregada como el documento actual. El proyecto para entregar debía ser una propuesta propia y creativa del alumno, la cual tendría que ser aprobada por el profesor. La propuesta debe ser implementada en una tarjeta FPGA, la cual ha sido utilizada a lo largo del laboratorio de la asignatura.

En este caso se decidió elaborar un sistema que simulara de manera sencilla una máquina vendedora de boletos para el cine. La idea general es que el usuario pueda manejarla siguiendo los mensajes que se le muestren. Su estructura y requerimientos se establecen a lo largo del escrito.

Objetivo

El alumno demostrará sus conocimientos obtenidos a lo largo del curso, tanto en laboratorio (lenguaje VHDL) como en teoría (diseño de sistemas), mediante la elaboración de un sistema en tarjeta FPGA.

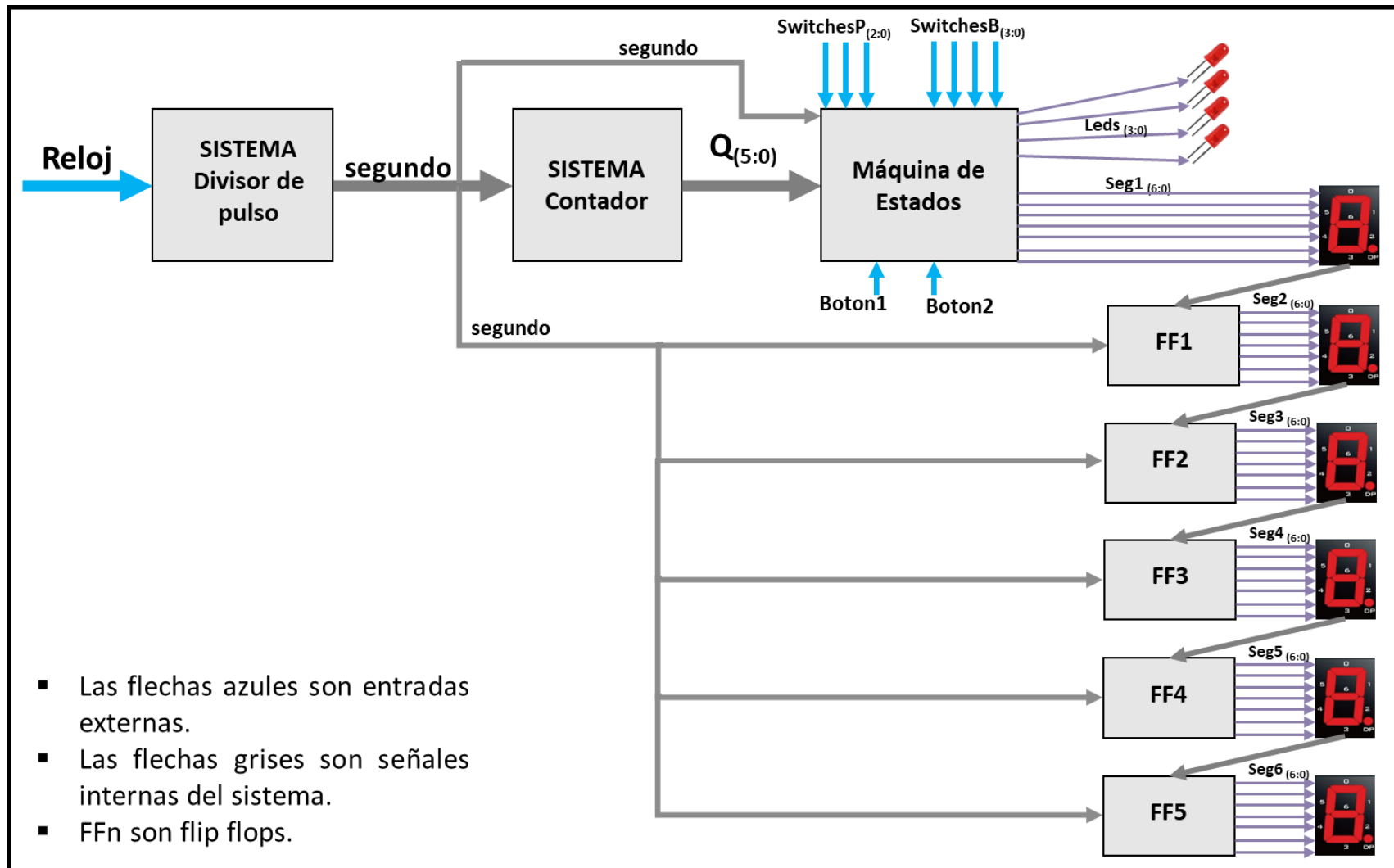
Material requerido

- Tarjeta electrónica: **DE10-Lite con Altera 10 MAX FPGA.**

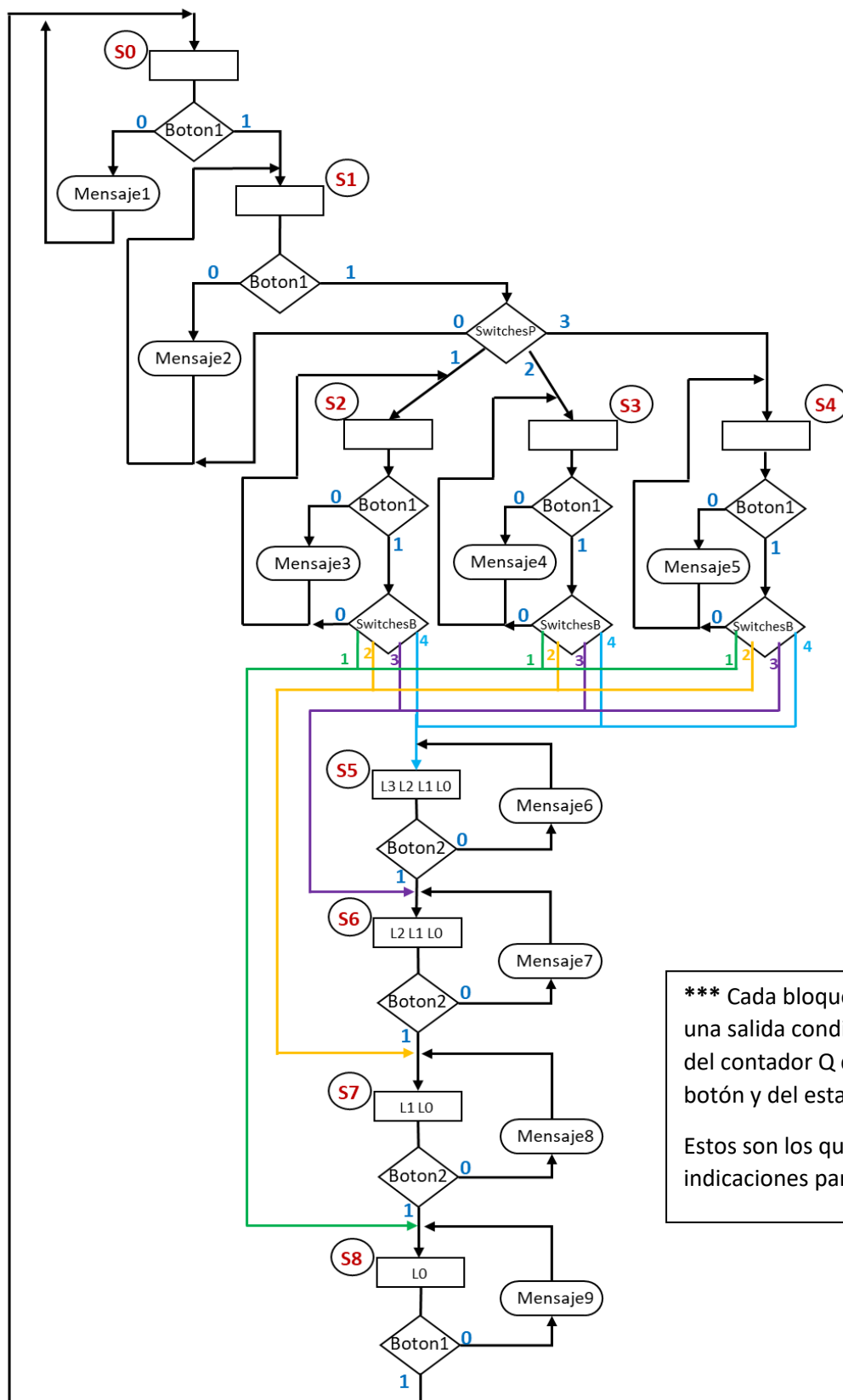
Especificaciones

Diseño e implementación en una tarjeta FPGA de un sistema de venta de boletos para el cine. El sistema indicará a través de los displays, mediante mensajes corredizos, la información e indicaciones para el usuario que desee adquirir los boletos. El usuario podrá elegir entre varias opciones de películas y podrá seleccionar hasta un máximo de 4 boletos mediante los switches de la tarjeta. Luego procederá a simular el pago a través de un botón. El sistema podrá usarse varias veces por diferentes usuarios.

Diagrama de bloques



Carta ASM (Algorithmic State Machine)



*** Cada bloque de **Mensaje#** es una salida condicional. Depende del contador Q del sistema, del botón y del estado actual.

Estos son los que muestran las indicaciones para el usuario.

Desarrollo

Para la creación del sistema lo primero que se tiene que llevar a cabo es la identificación de entradas y salidas requeridas. Éstas las podemos observar en el diagrama de bloques de la página 2. En nuestro caso son las siguientes.

Entradas de la tarjeta:

- ❖ Reloj interno
- ❖ Botón 0
- ❖ Botón 1
- ❖ Switches 0, 1 y 2
- ❖ Switches 6, 7, 8 y 9.

Salidas mostradas en la tarjeta:

- ❖ Leds 0, 1, 2 y 3
- ❖ 6 Displays.

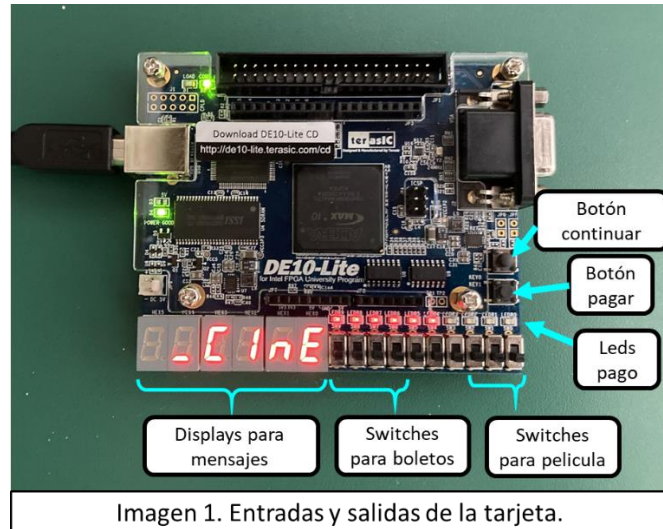


Imagen 1. Entradas y salidas de la tarjeta.

Ya que tenemos las entradas y salidas identificadas, hay que plantear bien el sistema. En este caso como deseamos trabajar de manera secuencial y no sólo combinacional se estableció el siguiente funcionamiento del sistema mediante estados:

En el estado inicial del sistema los leds estarán apagados y en los displays se muestra el mensaje 1: “**_Cine Fi_ Pica botón**”, el usuario deberá apretar el botón primer botón como primera interacción. De ahí pasará a otro estado donde aparecen las opciones de películas, el mensaje 2: “**Hola_ ¿Qué película quieres? = 1_Cars 2_Eso 3_UP**”. Aquí el usuario deberá indicar mediante los switches 0, 1 y 2 cuál es la opción que escoge. La elección de la película será un sistema de prioridad por lo que el switch más hacia la izquierda que esté activado será la película correspondiente. Dependiendo de la elección de película se pasará a alguno de los siguientes tres estados, cada uno con su mensaje específico dependiendo de la película. Mensaje 3: “**1_Cars ¿cuántos boletos?**”, mensaje 4: “**2_Eso ¿cuántos boletos?**” y mensaje 5: “**3_Up ¿cuántos boletos?**”. Para cualquiera de los posibles estados, el usuario deberá seleccionar cuántos boletos quiere comprar, de 1 a 4, mediante los switches 6, 7, 8 y 9 de la tarjeta. Dependiendo la entrada seleccionada pasará al pago de los boletos, donde son 4 estados, uno por cada posibilidad. Los mensajes de los estados serían el Mensaje 6, 7, 8 y 9: “**Pagar n pesos**”. Además, en cada uno de estos estados se mostrará a través de los leds cuánto hay que pagar. Para pagar el usuario deberá picar el segundo botón el número de veces necesario. Finalmente, al finalizar el pago se agradecerá con un mensaje y se debe picar el primer botón para regresar al inicio.

Como es un sistema secuencial, el funcionamiento descrito anteriormente se puede representar en una carta ASM. Ésta se puede observar en la página 3 del documento y representa el flujo que sigue el sistema a través de estados (con sus salidas), decisiones y salidas condicionales. El pulso que obtengamos al dividir el reloj interno de la tarjeta será lo que nos indique un cambio de estado (el cambio puede ser volver al mismo estado).

A continuación, se agrega la implementación en lenguaje VHDL del sistema.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity final is
    port( reloj: in std_logic;
          boton1, boton2: in std_logic;
          switchesB: in std_logic_vector(3 downto 0);
          switchesP: in std_logic_vector(2 downto 0);
          display1, display2, display3, display4, display5,
          display6: inout std_logic_vector(6 downto 0);
          leds: out std_logic_vector(3 downto 0));
end final;
```

architecture Behavioral of final is

```
    type estados is (s0, s1, s2, s3, s4, s5, s6, s7, s8, s9);
    signal epresente: estados := s0;
    signal esiguiente: estados;

    signal segundo: std_logic;
    signal Q: std_logic_vector(5 downto 0);
    signal limiteMensaje: std_logic_vector(5 downto 0) := "011000";
    signal reinicio: std_logic := '0';
```

begin

```
    divisor: process(reloj)
        variable cuenta: std_logic_vector(27 downto 0) := X"00000000";
        begin if rising_edge(reloj) then
            if cuenta = X"48009E0" then
                cuenta := X"00000000";
            else
                cuenta := cuenta+1;
            end if;
        end if;
        segundo <= cuenta(23);
    end process;
```

```
    MdE1: process(segundo) -- transición
        variable cuenta: std_logic_vector(5 downto 0) := "000000";
        begin if rising_edge(segundo) then
            if esiguiente = epresente then
                epresente <= esiguiente;
```

```

else
    if esiguiente = s0 then
        limiteMensaje <= "011000";
    elsif esiguiente = s1 then
        limiteMensaje <= "110101";
    elsif esiguiente = s2 then
        limiteMensaje <= "011010";
    elsif esiguiente = s3 then
        limiteMensaje <= "011001";
    elsif esiguiente = s4 then
        limiteMensaje <= "011000";
    elsif esiguiente = s5 then
        limiteMensaje <= "010000";
    elsif esiguiente = s6 then
        limiteMensaje <= "010000";
    elsif esiguiente = s7 then
        limiteMensaje <= "010000";
    elsif esiguiente = s8 then
        limiteMensaje <= "010000";
    else -- s9
        limiteMensaje <= "001101";
    end if;
    reinicio <= '1';
    epresente <= esiguiente;
end if;
if cuenta = limiteMensaje or reinicio = '1' then
    cuenta := "000000";
    reinicio <= '0';
else
    cuenta := cuenta+1;
end if;
Q <= cuenta;
end if;
end process;

```

```

MdE2: process(epresente, boton1, boton2, Q, switchesP, switchesB)

```

```

begin case epresente is

```

```

    when s0 => -- inicio cine
        if Q = "000000" then
            leds <= "0000";
            display1 <= "1110111";
        elsif Q = "000001" then
            display1 <= "1000110";
        elsif Q = "000010" then
            display1 <= "1111001";
        elsif Q = "000011" then
            display1 <= "0101011";
        elsif Q = "000100" then
            display1 <= "0000110";
        elsif Q = "000110" then
            display1 <= "0001110";
        elsif Q = "000111" then
            display1 <= "1111001";
        elsif Q = "001000" then
            display1 <= "1110111";
        elsif Q = "001011" then
            display1 <= "0001100";
        elsif Q = "001100" then
            display1 <= "1111001";
        elsif Q = "001101" then

```

```

        display1 <= "1000110";
    elsif Q = "001110" then
        display1 <= "0001000";
    elsif Q = "010001" then
        display1 <= "0000011";
    elsif Q = "010010" then
        display1 <= "0100011";
    elsif Q = "010011" then
        display1 <= "1001110";
    elsif Q = "010100" then
        display1 <= "0100011";
    elsif Q = "010101" then
        display1 <= "0101011";
    else
        display1 <= "1111111";
    end if;

    if boton1 = '0' then
        esiguiente <= s1;
    else
        esiguiente <= s0;
    end if;

when s1 => -- eleccion pelicula
    if Q = "000000" then
        display1 <= "0001001";
    elsif Q = "000001" then
        display1 <= "1000000";
    elsif Q = "000010" then
        display1 <= "1000111";
    elsif Q = "000011" then
        display1 <= "0001000";
    elsif Q = "000100" then
        display1 <= "1110111";
    elsif Q = "000110" then
        display1 <= "0100101";
    elsif Q = "000111" then
        display1 <= "0011000";
    elsif Q = "001000" then
        display1 <= "1000001";
    elsif Q = "001001" then
        display1 <= "0000110";
    elsif Q = "001011" then
        display1 <= "0001100";
    elsif Q = "001100" then
        display1 <= "0000110";
    elsif Q = "001101" then
        display1 <= "1000111";
    elsif Q = "001110" then
        display1 <= "1111001";
    elsif Q = "001111" then
        display1 <= "1000110";
    elsif Q = "010000" then
        display1 <= "1000001";
    elsif Q = "010001" then
        display1 <= "1000111";
    elsif Q = "010010" then
        display1 <= "0001000";
    elsif Q = "010100" then
        display1 <= "0011000";
    elsif Q = "010101" then
        display1 <= "1000001";

```



```

elsif Q = "010110" then
    display1 <= "1111001";
elsif Q = "010111" then
    display1 <= "0000110";
elsif Q = "011000" then
    display1 <= "0101111";
elsif Q = "011001" then
    display1 <= "0000110";
elsif Q = "011010" then
    display1 <= "0010010";
elsif Q = "011011" then
    display1 <= "0101100";
elsif Q = "011101" then
    display1 <= "0110111";
elsif Q = "011111" then
    display1 <= "1111001";
elsif Q = "100000" then
    display1 <= "1110111";
elsif Q = "100001" then
    display1 <= "1000110";
elsif Q = "100010" then
    display1 <= "0001000";
elsif Q = "100011" then
    display1 <= "0101111";
elsif Q = "100100" then
    display1 <= "0010010";
elsif Q = "100111" then
    display1 <= "0100100";
elsif Q = "101000" then
    display1 <= "1110111";
elsif Q = "101001" then
    display1 <= "0000110";
elsif Q = "101010" then
    display1 <= "0010010";
elsif Q = "101011" then
    display1 <= "1000000";
elsif Q = "101110" then
    display1 <= "0110000";
elsif Q = "101111" then
    display1 <= "1110111";
elsif Q = "110000" then
    display1 <= "1000001";
elsif Q = "110001" then
    display1 <= "0001100";
else
    display1 <= "1111111";
end if;

if boton1 = '0' then
    if switchesP = "001" then
        esiguiente <= s2;
    elsif switchesP = "010" then
        esiguiente <= s3;
    elsif switchesP = "011" then
        esiguiente <= s3;
    elsif switchesP = "100" then
        esiguiente <= s4;
    elsif switchesP = "101" then
        esiguiente <= s4;
    elsif switchesP = "110" then
        esiguiente <= s4;
    elsif switchesP = "111" then
        esiguiente <= s4;

```

```

        else
            esiguiente <= s1;
        end if;
    else
        esiguiente <= s1;
    end if;

when s2 => -- 1_Cars boletos
    if Q = "000000" then
        display1 <= "1111001";
    elsif Q = "000001" then
        display1 <= "1110111";
    elsif Q = "000010" then
        display1 <= "1000110";
    elsif Q = "000011" then
        display1 <= "0001000";
    elsif Q = "000100" then
        display1 <= "0101111";
    elsif Q = "000101" then
        display1 <= "0010010";
    elsif Q = "000111" then
        display1 <= "0100101";
    elsif Q = "001000" then
        display1 <= "1000110";
    elsif Q = "001001" then
        display1 <= "1000001";
    elsif Q = "001010" then
        display1 <= "0001000";
    elsif Q = "001011" then
        display1 <= "0101011";
    elsif Q = "001100" then
        display1 <= "1001110";
    elsif Q = "001101" then
        display1 <= "0100011";
    elsif Q = "001110" then
        display1 <= "0010010";
    elsif Q = "010000" then
        display1 <= "0000011";
    elsif Q = "010001" then
        display1 <= "0100011";
    elsif Q = "010010" then
        display1 <= "1000111";
    elsif Q = "010011" then
        display1 <= "0000110";
    elsif Q = "010100" then
        display1 <= "1001110";
    elsif Q = "010101" then
        display1 <= "0100011";
    elsif Q = "010110" then
        display1 <= "0010010";
    elsif Q = "010111" then
        display1 <= "0101100";
    else
        display1 <= "1111111";
    end if;

    if boton1 = '0' then
        if switchesB = "0000" then
            esiguiente <= s2;
        else
            if switchesB = "0001" then
                esiguiente <= s8;
            end if;
        end if;
    end if;
end when;

```

```

        elsif switchesB = "0010" then
            esiguiente <= s7;
        elsif switchesB = "0011" then
            esiguiente <= s7;
        elsif switchesB = "0100" then
            esiguiente <= s6;
        elsif switchesB = "0101" then
            esiguiente <= s6;
        elsif switchesB = "0110" then
            esiguiente <= s6;
        elsif switchesB = "0111" then
            esiguiente <= s6;
        else
            esiguiente <= s5;
        end if;
    end if;
else
    esiguiente <= s2;
end if;

when s3 => -- 2_Eso boletos
    if Q = "000000" then
        display1 <= "0100100";
    elsif Q = "000001" then
        display1 <= "1110111";
    elsif Q = "000010" then
        display1 <= "0000110";
    elsif Q = "000011" then
        display1 <= "0010010";
    elsif Q = "000100" then
        display1 <= "1000000";
    elsif Q = "000110" then
        display1 <= "0100101";
    elsif Q = "000111" then
        display1 <= "1000110";
    elsif Q = "001000" then
        display1 <= "1000001";
    elsif Q = "001001" then
        display1 <= "0001000";
    elsif Q = "001010" then
        display1 <= "0101011";
    elsif Q = "001011" then
        display1 <= "1001110";
    elsif Q = "001100" then
        display1 <= "0100011";
    elsif Q = "001101" then
        display1 <= "0010010";
    elsif Q = "001111" then
        display1 <= "0000011";
    elsif Q = "010000" then
        display1 <= "0100011";
    elsif Q = "010001" then
        display1 <= "1000111";
    elsif Q = "010010" then
        display1 <= "0000110";
    elsif Q = "010011" then
        display1 <= "1001110";
    elsif Q = "010100" then
        display1 <= "0100011";
    elsif Q = "010101" then
        display1 <= "0010010";
    elsif Q = "010110" then

```

```

        display1 <= "0101100";
    else
        display1 <= "1111111";
    end if;

    if boton1 = '0' then
        if switchesB = "0000" then
            esiguiente <= s3;
        else
            if switchesB = "0001" then
                esiguiente <= s8;
            elsif switchesB = "0010" then
                esiguiente <= s7;
            elsif switchesB = "0011" then
                esiguiente <= s7;
            elsif switchesB = "0100" then
                esiguiente <= s6;
            elsif switchesB = "0101" then
                esiguiente <= s6;
            elsif switchesB = "0110" then
                esiguiente <= s6;
            elsif switchesB = "0111" then
                esiguiente <= s6;
            else
                esiguiente <= s5;
            end if;
        end if;
    else
        esiguiente <= s3;
    end if;

when s4 => -- 3_UP boletos
    if Q = "000000" then
        display1 <= "0110000";
    elsif Q = "000001" then
        display1 <= "1110111";
    elsif Q = "000010" then
        display1 <= "1000001";
    elsif Q = "000011" then
        display1 <= "0001100";
    elsif Q = "000101" then
        display1 <= "0100101";
    elsif Q = "000110" then
        display1 <= "1000110";
    elsif Q = "000111" then
        display1 <= "1000001";
    elsif Q = "001000" then
        display1 <= "0001000";
    elsif Q = "001001" then
        display1 <= "0101011";
    elsif Q = "001010" then
        display1 <= "1001110";
    elsif Q = "001011" then
        display1 <= "0100011";
    elsif Q = "001100" then
        display1 <= "0010010";
    elsif Q = "001110" then
        display1 <= "0000011";
    elsif Q = "001111" then
        display1 <= "0100011";
    elsif Q = "010000" then
        display1 <= "1000111";

```

```

elseif Q = "010001" then
    display1 <= "0000110";
elseif Q = "010010" then
    display1 <= "1001110";
elseif Q = "010011" then
    display1 <= "0100011";
elseif Q = "010100" then
    display1 <= "0010010";
elseif Q = "010101" then
    display1 <= "0101100";
else
    display1 <= "1111111";
end if;

if boton1 = '0' then
    if switchesB = "0000" then
        esiguiente <= s4;
    else
        if switchesB = "0001" then
            esiguiente <= s8;
        elseif switchesB = "0010" then
            esiguiente <= s7;
        elseif switchesB = "0011" then
            esiguiente <= s7;
        elseif switchesB = "0100" then
            esiguiente <= s6;
        elseif switchesB = "0101" then
            esiguiente <= s6;
        elseif switchesB = "0110" then
            esiguiente <= s6;
        elseif switchesB = "0111" then
            esiguiente <= s6;
        else
            esiguiente <= s5;
        end if;
    end if;
else
    esiguiente <= s4;
end if;

when s5 => -- Pagar 4
    leds <= "1111";
    if Q = "000000" then
        display1 <= "0001100";
    elseif Q = "000001" then
        display1 <= "0001000";
    elseif Q = "000010" then
        display1 <= "1000010";
    elseif Q = "000011" then
        display1 <= "0001000";
    elseif Q = "000100" then
        display1 <= "0101111";
    elseif Q = "000110" then
        display1 <= "0011001"; -- Número 4
    elseif Q = "001000" then
        display1 <= "0001100";
    elseif Q = "001001" then
        display1 <= "0000110";
    elseif Q = "001010" then
        display1 <= "0010010";
    elseif Q = "001011" then
        display1 <= "1000000";

```

```

elseif Q = "001100" then
    display1 <= "0010010";
else
    display1 <= "1111111";
end if;

if boton2 = '0' then
    esiguiente <= s6;
else
    esiguiente <= s5;
end if;

when s6 => -- Pagar 3
    leds <= "0111";
    if Q = "000000" then
        display1 <= "0001100";
    elseif Q = "000001" then
        display1 <= "0001000";
    elseif Q = "000010" then
        display1 <= "1000010";
    elseif Q = "000011" then
        display1 <= "0001000";
    elseif Q = "000100" then
        display1 <= "0101111";
    elseif Q = "000110" then
        display1 <= "0110000"; -- Número 3
    elseif Q = "001000" then
        display1 <= "0001100";
    elseif Q = "001001" then
        display1 <= "0000110";
    elseif Q = "001010" then
        display1 <= "0010010";
    elseif Q = "001011" then
        display1 <= "1000000";
    elseif Q = "001100" then
        display1 <= "0010010";
    else
        display1 <= "1111111";
    end if;

    if boton2 = '0' then
        esiguiente <= s7;
    else
        esiguiente <= s6;
    end if;

when s7 => -- Pagar 2
    leds <= "0011";
    if Q = "000000" then
        display1 <= "0001100";
    elseif Q = "000001" then
        display1 <= "0001000";
    elseif Q = "000010" then
        display1 <= "1000010";
    elseif Q = "000011" then
        display1 <= "0001000";
    elseif Q = "000100" then
        display1 <= "0101111";
    elseif Q = "000110" then
        display1 <= "0100100"; -- Número 2
    elseif Q = "001000" then

```

```

        display1 <= "0001100";
    elsif Q = "001001" then
        display1 <= "0000110";
    elsif Q = "001010" then
        display1 <= "0010010";
    elsif Q = "001011" then
        display1 <= "1000000";
    elsif Q = "001100" then
        display1 <= "0010010";
    else
        display1 <= "1111111";
    end if;

    if boton2 = '0' then
        esiguiente <= s8;
    else
        esiguiente <= s7;
    end if;

when s8 => -- Pagar 1
    leds <= "0001";
    if Q = "000000" then
        display1 <= "0001100";
    elsif Q = "000001" then
        display1 <= "0001000";
    elsif Q = "000010" then
        display1 <= "1000010";
    elsif Q = "000011" then
        display1 <= "0001000";
    elsif Q = "000100" then
        display1 <= "0101111";
    elsif Q = "000110" then
        display1 <= "1111001"; -- Número 1
    elsif Q = "001000" then
        display1 <= "0001100";
    elsif Q = "001001" then
        display1 <= "0000110";
    elsif Q = "001010" then
        display1 <= "0010010";
    elsif Q = "001011" then
        display1 <= "1000000";
    else
        display1 <= "1111111";
    end if;

    if boton2 = '0' then
        esiguiente <= s9;
    else
        esiguiente <= s8;
    end if;

when s9 => -- Gracias
    leds <= "0000";
    if Q = "000000" then
        display1 <= "1000010";
    elsif Q = "000001" then
        display1 <= "0101111";
    elsif Q = "000010" then
        display1 <= "0001000";
    elsif Q = "000011" then
        display1 <= "1000110";

```

```

        elsif Q = "000100" then
            display1 <= "1111001";
        elsif Q = "000101" then
            display1 <= "0001000";
        elsif Q = "000110" then
            display1 <= "0010010";
        elsif Q = "001000" then
            display1 <= "1110111";
        elsif Q = "001001" then
            display1 <= "1110111";
        elsif Q = "001010" then
            display1 <= "1110111";
        else
            display1 <= "1111111";
        end if;

        if boton1 = '0' then
            esiguiente <= s0;
        else
            esiguiente <= s9;
        end if;

    end case;
end process;

```

```

-- Cambio de corrimiento de registros.
FF1: process(segundo)
begin
    if rising_edge(segundo) then
        display2 <= display1;
    end if;
end process;

FF2: process(segundo)
begin
    if rising_edge(segundo) then
        display3 <= display2;
    end if;
end process;

FF3: process(segundo)
begin
    if rising_edge(segundo) then
        display4 <= display3;
    end if;
end process;

FF4: process(segundo)
begin
    if rising_edge(segundo) then
        display5 <= display4;
    end if;
end process;

FF5: process(segundo)
begin
    if rising_edge(segundo) then
        display6 <= display5;
    end if;
end process;

```

```
end Behavioral;
```


Métodos utilizados

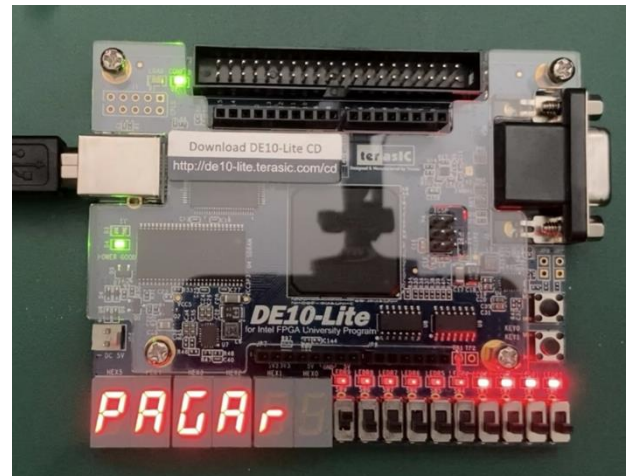
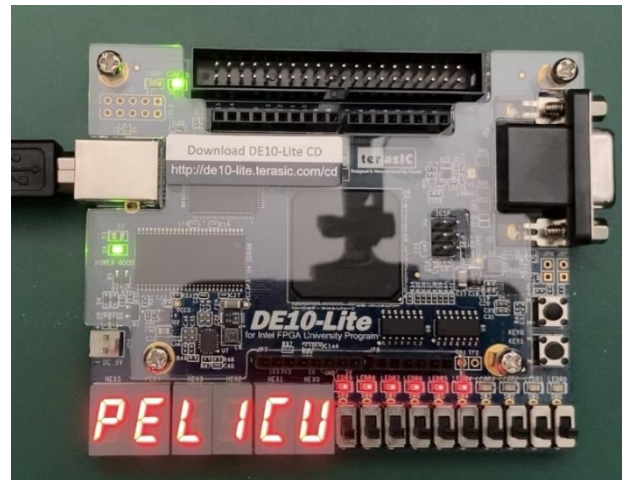
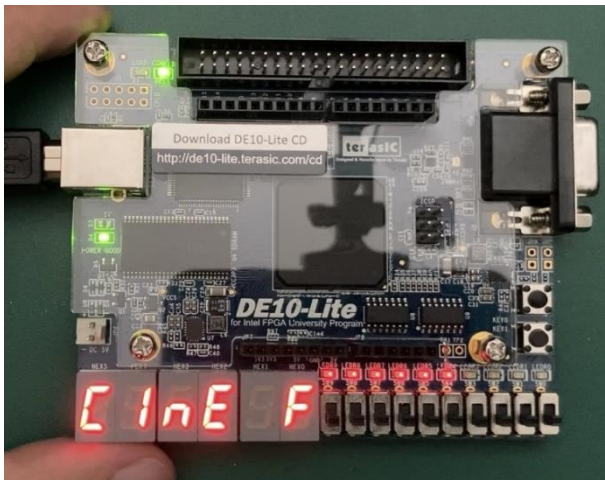
Para la elaboración del sistema (tanto diseño como implementación) se utilizaron diversos métodos vistos en la asignatura. Algunos de los más relevantes fueron los siguientes:

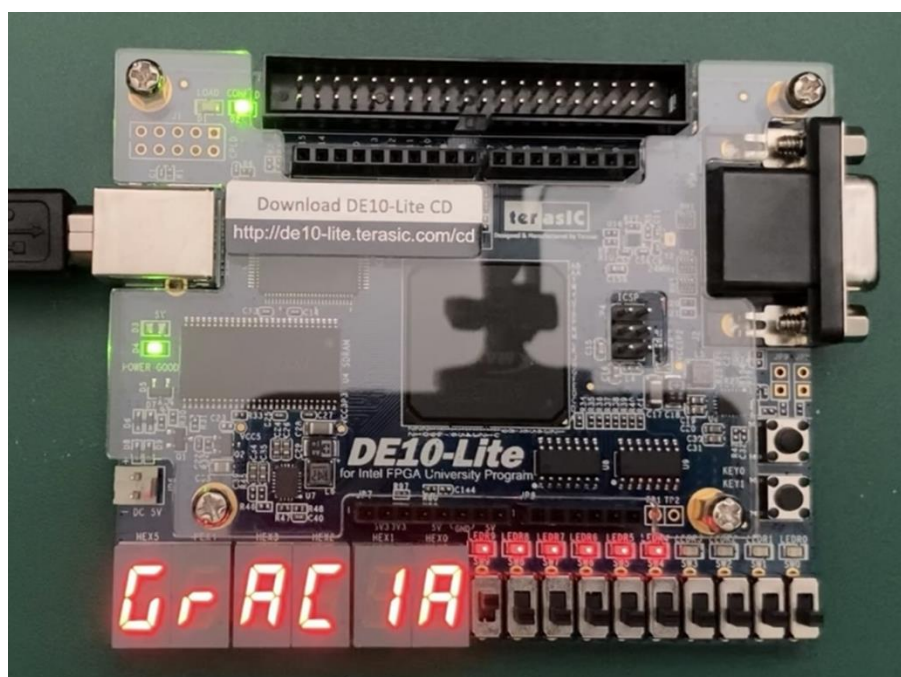
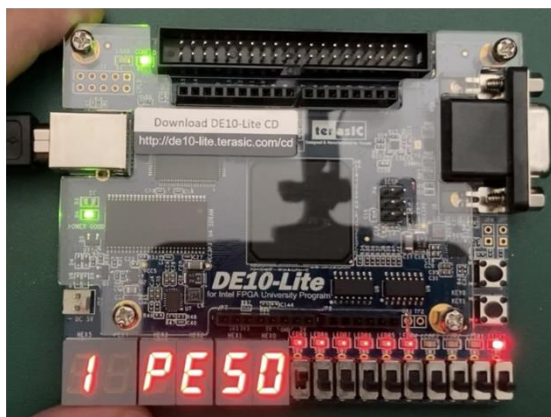
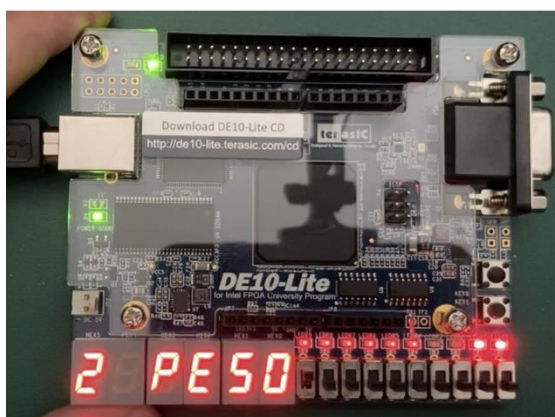
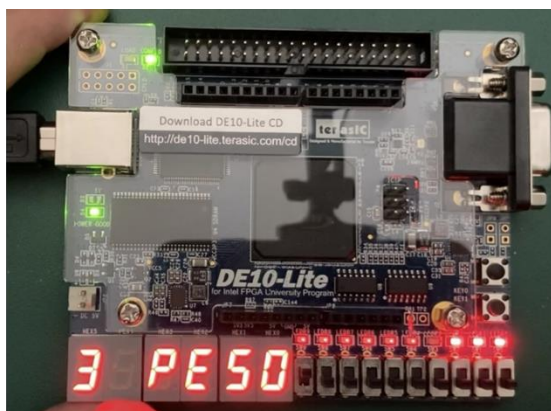
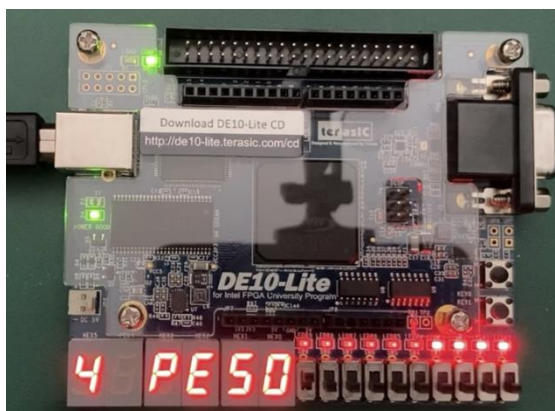
- **Sistemas de prioridad:** Para la selección de la película y del número de boletos, se utilizaron sistemas de prioridad, esto quiere decir que la salida sería el binario más significativo que esté activado. Por lo que sin importar si la entrada en switches era “100” ó “111” de todos modos se activaría la opción 3 en este caso.
- **Comparadores:** En muchas secciones tuvimos que comparar si un botón había sido activado (mediante lógica negada) para saber que se debía realizar un cambio de estado. Por lo que se necesitaba saber si una señal era igual a ‘0’ o no.
- **Memoria:** Para la salida del mensaje en los displays se utilizó memoria para poder asignar los valores que estarían prendidos entre los 7 segmentos. Según la entrada de una señal de cuenta se accedía a la memoria y asignaba los valores correspondientes.
- **Relojes (contadores):** Como nuestro sistema es secuencial y no sólo combinacional, lo más importante que se debe tener es un reloj para avanzar la secuencia. En el caso de la tarjeta el reloj que contiene es de 50 MHz por lo que tuvimos que ir dividiendo la señal (entre 2 – lógica binaria) para obtener una frecuencia aceptable. Con esta se utilizaron variables contadoras para asignarle a una señal su valor correspondiente según fuera necesario.
- **Flip flops:** Uno de los temas que más se me complicó en el curso fueron los flip flops, pero el verlo de manera práctica ayudó mucho a comprenderlos mejor. En el sistema se utilizaron flip flops para los registros de corrimiento, estos al recibir un pulso del contador del reloj leían los datos del display anterior y los desplegaban en el que les correspondía.
- **Cartas ASM:** Como se mostró anteriormente la carta ASM es una parte esencial del diseño de los sistemas secuenciales. En ella pudimos establecer los estados necesarios, así como sus salidas normales y salidas condicionales. Esto nos permitió establecer el flujo deseado para el sistema y sirve de base al momento de codificarlo.
- **Bibliotecas propias en VHDL:** Para poder hacer una máquina de estados en el lenguaje VHDL tuvimos que crear nuestra propia biblioteca de estados. Los demás procesos podían usarla para identificar los estados posibles, el actual y el siguiente.

Siempre es interesante el poder ver cómo todo lo revisado en teoría se complementa para dar como resultado un sistema más completo.

Resultados obtenidos

La implementación del sistema en la tarjeta FPGA fue llevada a cabo exitosamente. El sistema funcionó correctamente según las especificaciones dadas, podía utilizarse varias veces por diferentes usuarios y no dejaba de funcionar. En las siguientes imágenes se pueden observar algunos de las palabras claves de los mensajes de cada estado del sistema. Además, se pueden observar la salida de los leds según sea la suma que se debe pagar. Para poder apreciar mejor los resultados y el sistema obtenido se puede ver el video que se entregó junto con este reporte. En él se explica más tanto la parte teórica (diagrama de bloques, carta ASM y código VHDL) como la parte práctica (implementación y demostración en tarjeta FPGA). Las imágenes que aquí se muestran fueron obtenidas del material expuesto en dicho video.





Trabajo a futuro y conclusiones

Como trabajo futuro para el sistema presentado se me ocurren algunas ideas de mejoras. La primera es agregarle un cupo máximo tanto a cada función independiente, como al cine en general. Si ya estuviera una función llena a la hora de seleccionarla en lugar de preguntarte cuántos boletos deseas comprar, mostraría un mensaje de “función agotada”. Igualmente, si ya todas las funciones están agotadas, el sistema debería pasarte un mensaje de “cupos máximos alcanzados” al tratar de avanzar de la portada (estado 0) y se limitaría a los estados de la portada y del cupo máximo (nuevo estado 10).

Otra área donde se puede trabajar a futuro es la del cambio de mensajes en los displays cuando el sistema cambia de estado. Como se puede observar en el video, al cambiar de estado el primer carácter que se muestra en los displays corresponde al del nuevo mensaje, pero en la cuenta del estado anterior. Ya el segundo carácter sí corresponde al inicio del mensaje correspondiente del estado. Lo ideal sería poder cambiar esto para que en lugar de que se viera un carácter equivocado se agregara un espacio para siempre separar los mensajes anteriores y los que apenas están iniciando. Además, otro aspecto de los mensajes que se podría agregar sería que los mensajes informativos donde no se tiene que seleccionar nada (“función agotada”, “cupos máximos”, “gracias”), se reprodujeran tres veces y luego pasaran al siguiente estado. En lugar de tener que apretar un botón para continuar.

En un principio traté de añadir algunos de estos trabajos a futuro al sistema, pero me empezaron a causar muchas dificultades, por lo que finalmente desistí y lo desarrollé en esta sección como áreas de oportunidad. Como conclusión general, creo que el proyecto nos permitió demostrar de forma creativa que sí aprendimos los temas de la asignatura. Además, siempre es útil pasar de la teoría a la práctica y darnos cuenta que lo que se enseña realmente se puede aprovechar en la vida real. Como la profesora nos ha dicho muchas veces, con lo que aprendimos en esta clase (junto los conocimientos acumulados de la carrera) ya podríamos hacer sistemas sencillos pero útiles e inclusive comercializarlos. Al final yo quedé satisfecho con mis resultados y creo que se cumplieron tanto los objetivos como las especificaciones planteadas en un inicio.