



Proyecto Final.

Diseño Digital Moderno.

Grupo 1.

M.I. Chávez Rodríguez Norma Elva.

Equipo 22.

Arriaga Mejía José Carlos.

316017862.

Objetivo

El alumno creara un sistema funcional que se ocupe en el día a día.

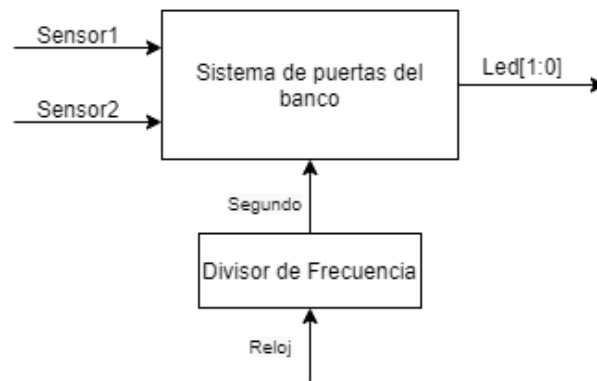
Planteamiento del problema.

Un banco requiere un sistema para regular la entrada y salida de su establecimiento. El sistema contará con dos puertas conectadas, cuando una de estas este abierta la otra no se podrá abrir hasta 3 segundos después de haberse cerrado. Se tendrán un sensor, para cada puerta, que detectará el estado de la puerta y con un led se mostrará si la puerta se puede o no abrir, apagado y prendido, respectivamente.

Comenzamos identificando nuestras entradas y salidas de nuestro sistema.

Como entrada tenemos que cada puerta tiene un sensor para saber si la puerta está o no abierta, y un led que indica si se puede o no abrir la puerta. Además, se tiene un reloj para que cuando se abra una puerta la otra no se puede abrir hasta 3 segundos después de que se cierre la puerta.

Como salida tenemos los leds que indican si la puerta se puede abrir o no.

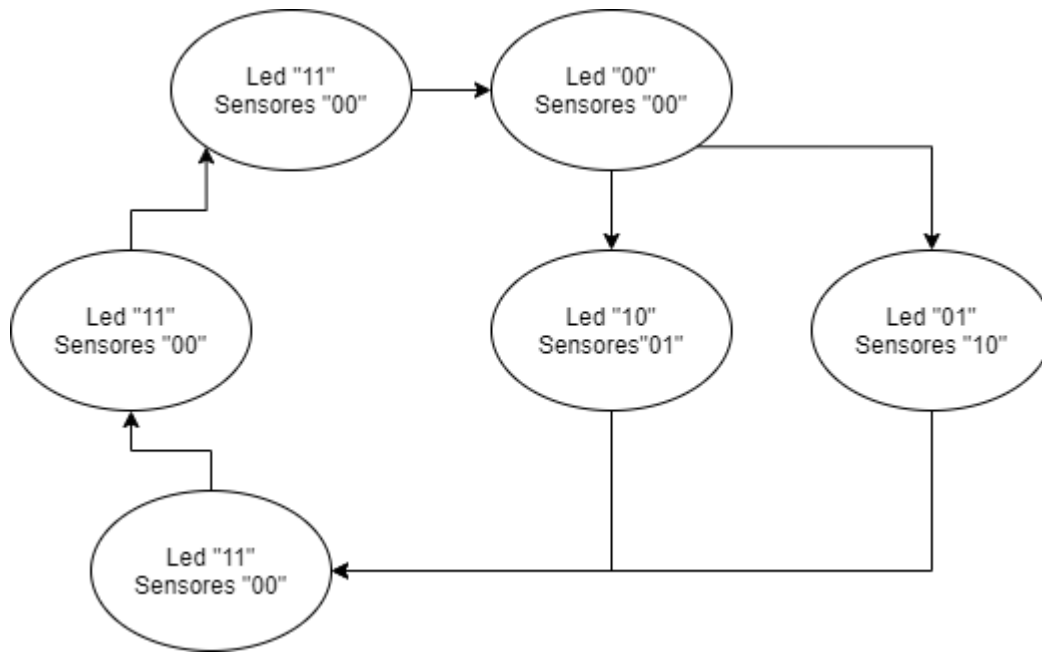


Identificamos los estados del sistema.

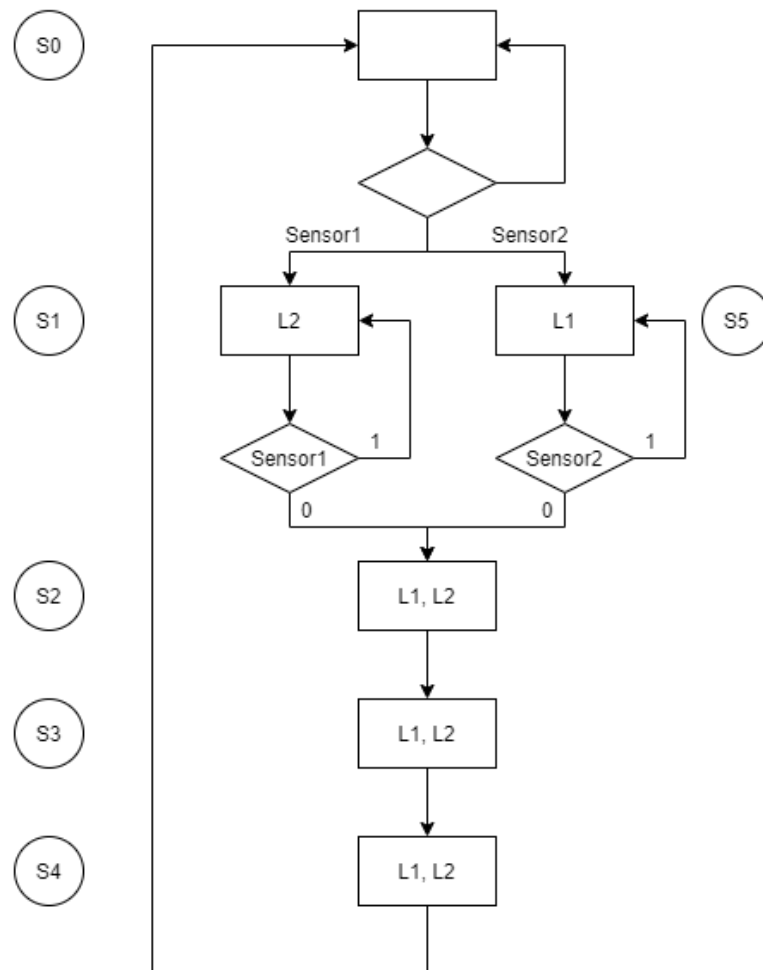
En el primer estado tenemos que cuando las dos puertas estén cerradas y ya hayan pasado 3 segundos ambos leds estarán apagados y las puertas estarán cerradas.

Dos estados son cuando se abre una puerta, un estado para cuando la puerta 1 se abre el led de la puerta 2 se encenderá, notificando que no se puede abrir. El led de la puerta 1 se mantendrá apagada ya que la puerta estará abierta. El siguiente estado es igual, pero con la puerta 2.

Los siguientes tres estados son cuando las dos puertas se cerraron y ambos leds están prendidos, cada uno de estos estados representan a un segundo.



Teniendo nuestro diagrama de estados realizamos nuestra carta ASM.



Con nuestra carta ASM, realizamos nuestro código del sistema.

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_arith.all;
4  use IEEE.std_logic_unsigned.all;
5
6  entity SdP is
7  port (reloj: in std_logic;
8        Sensor1: in std_logic;
9        Sensor2: in std_logic;
10       Led: out std_logic_vector (1 downto 0));
11 end SdP;
12
13 architecture Behavioral of SdP is
14 type estados is (s0,s1,s2,s3,s4,s5);
15 signal Ep, es: estados;
16 signal seg: std_logic;
17
18 begin
19 divisor: process (reloj)
20 variable cuenta: std_logic_vector (27 downto 0):=x"0000000";
21 begin
22     if rising_edge (reloj) then
23         if cuenta=x"48009E0" then
24             cuenta := x"0000000";
25         else
26             cuenta:=cuenta+1;
27         end if;
28     end if;
29     seg<=cuenta(24);
30 end process;
31
32 MdE1:process(seg)
33 begin
34     if rising_edge (seg) then
35         ep<=es;
36     end if;
37 end process;
38
39 MdE2:process(ep)
40 begin
41     case ep is
42     when s0=>
43         Led <="00";
44         if Sensor1='1' and Sensor2='0' then
45             es<=s1;
46         else
47             if (Sensor2='1' and Sensor1='0') then
48                 es<=s5;
49             else
50                 es<=s0;
51             end if;
52         end if;
53     when s1=>
54         Led <="10";
55         if Sensor1='0' then
56             es<=s2;
57         else
58             es<=s1;
59         end if;
```

```

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

```

```

when s2=>
    Led <="11";
    es<=s3;

when s3=>
    Led <="11";
    es<=s4;

when s4=>
    Led <="11";
    es<=s0;
when s5=>
    Led <="01";
    if Sensor2='0' then
        es<=s2;
    else
        es<=s5;
    end if;
end case;
end process;
end Behavioral;

```

