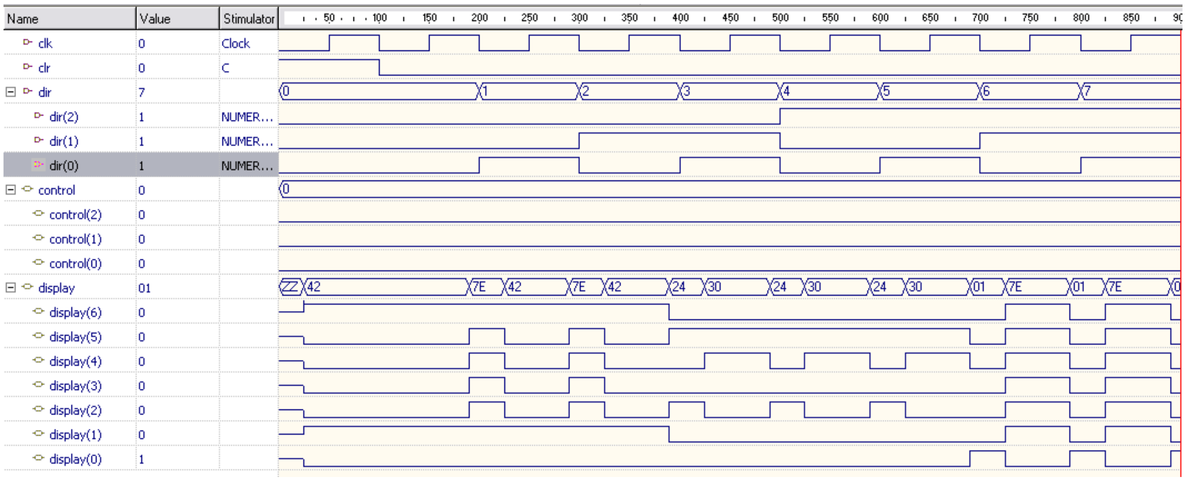
Córdova Pichardo Francisco Uziel

Diseño 1

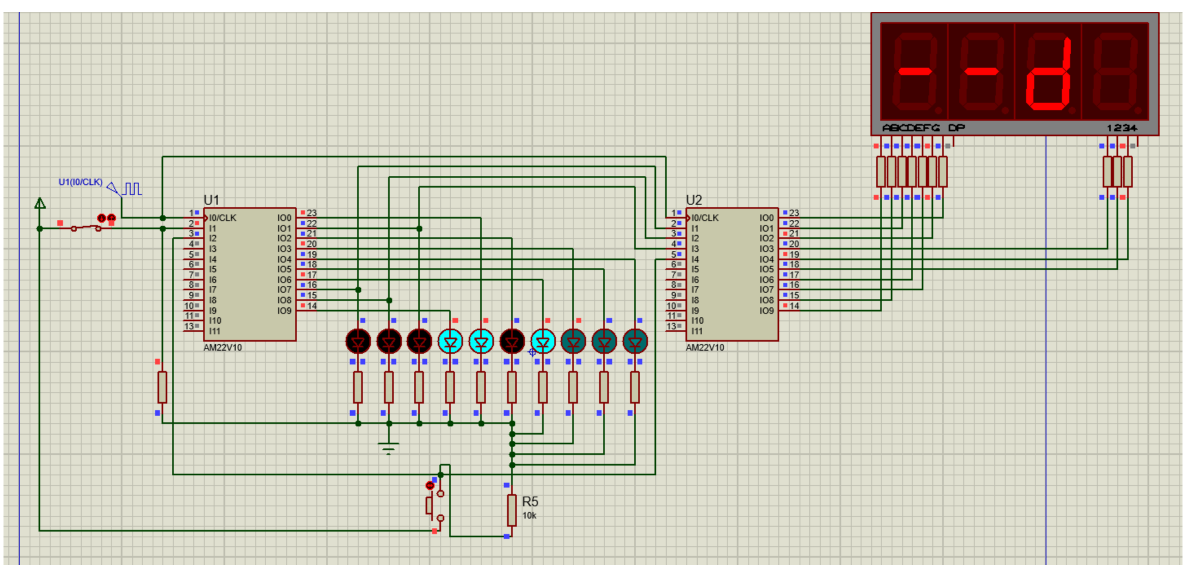
```
 1 library ieee;
 2 use ieee.std_logic_1164.all;
 3 use ieee.std_logic_arith.all;
 4 use ieee.std_logic_unsigned.all;
 5
 6 entity rom is port
 7 (
 8     clk,clr : in std_logic;
 9     dir : in std_logic_vector(2 downto 0);
10     control : out std_logic_vector(2 downto 0);
11     display : out std_logic_vector(6 downto 0)
12 );
13 end rom;
14
15 architecture Arom of rom is
16 signal dato0 : std_logic_vector(6 downto 0);
17 signal dato1 : std_logic_vector(6 downto 0);
18 signal dato2 : std_logic_vector(6 downto 0);
19 type memoria is array (0 to 7)of std_logic_vector(6 downto 0);
20 constant banco2 : memoria :=(
21     "1111110", --"-"
22     "1111110", --"-"
23     "1000010", --"d"
24     "1001111", --"I"
25     "0100100", --"S"
26     "0110000", --"E"
27     "0101010", --"ñ"
28     "0000001" --"O"
29 );
30 constant banco1 : memoria := (
31     "1111110", --"-"
32     "1000010", --"d"
33     "1001111", --"I"
34     "0100100", --"S"
35     "0110000", --"E"
36     "0101010", --"ñ"
37     "0000001", --"O"
38     "1111110" --"-"
39 );
```

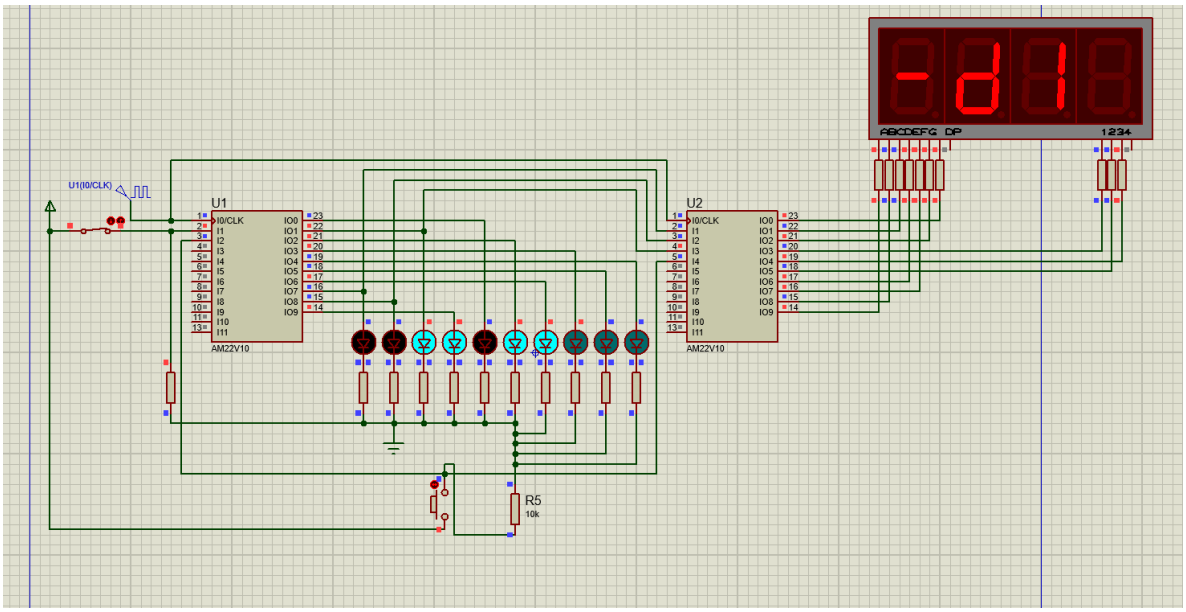
```
40 constant banco0 : memoria := (
41     "1000010", --"d"
42     "1001111", --"I"
43     "0100100", --"S"
44     "0110000", --"E"
45     "0101010", --"ñ"
46     "0000001", --"O"
47     "1111110", --"-"
48     "1111110" --"-"
49 );
50 begin
51 --contador de anillo
52     process(clr,clk)
53     begin
54         if(clr='1')then
55             control <= "001";
56             display <= "0000000";
57         elsif(rising_edge(clk))then
58             case control is
59                 when "001" => control <= "010";
60                 when "010" => control <= "100";
61                 when "100" => control <= "001";
62                 when others => control <= "---";
63             end case;
64         end if;
65     end process;
66 --fin del contador
67 --memoria
68     process(dir)
69         begin
70         dato0 <= banco0(conv_integer(dir));
71         dato1 <= banco1(conv_integer(dir));
72         dato2 <= banco2(conv_integer(dir));
73     end process;
74 --fin de la memoria
75
```

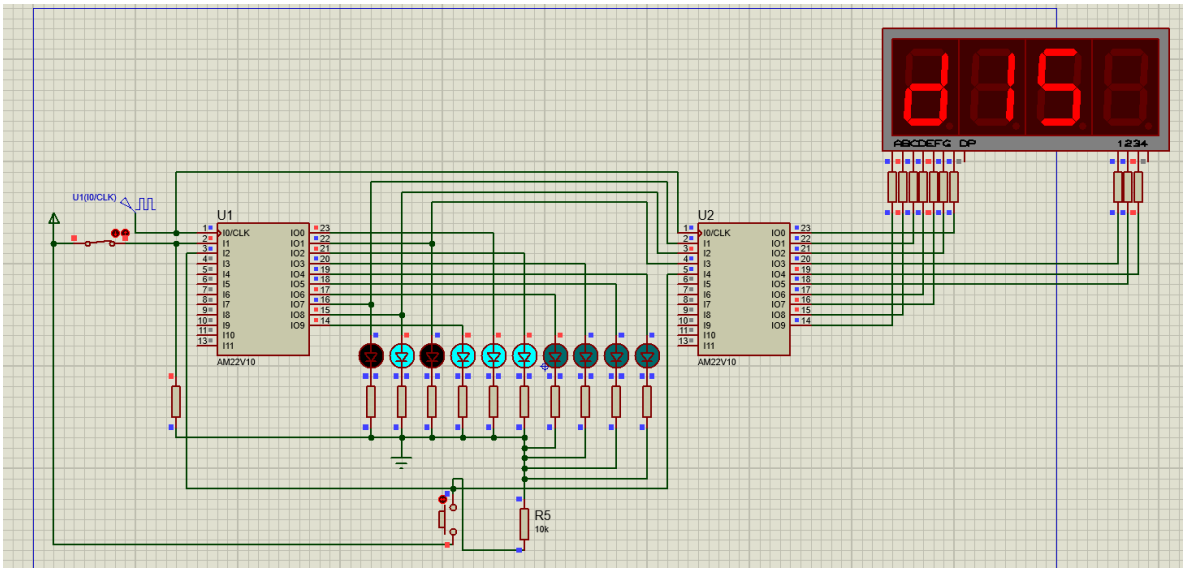
```
76 --Multiplexor--
77 process(control,dato0,dato1,dato2)
78     begin
79         if(control = "001") then
80             display <= dato0;
81         elsif(control = "010") then
82             display <= dato1;
83         else
84             display <= dato2;
85         end if;
86     end process;
87 end Arom;
```

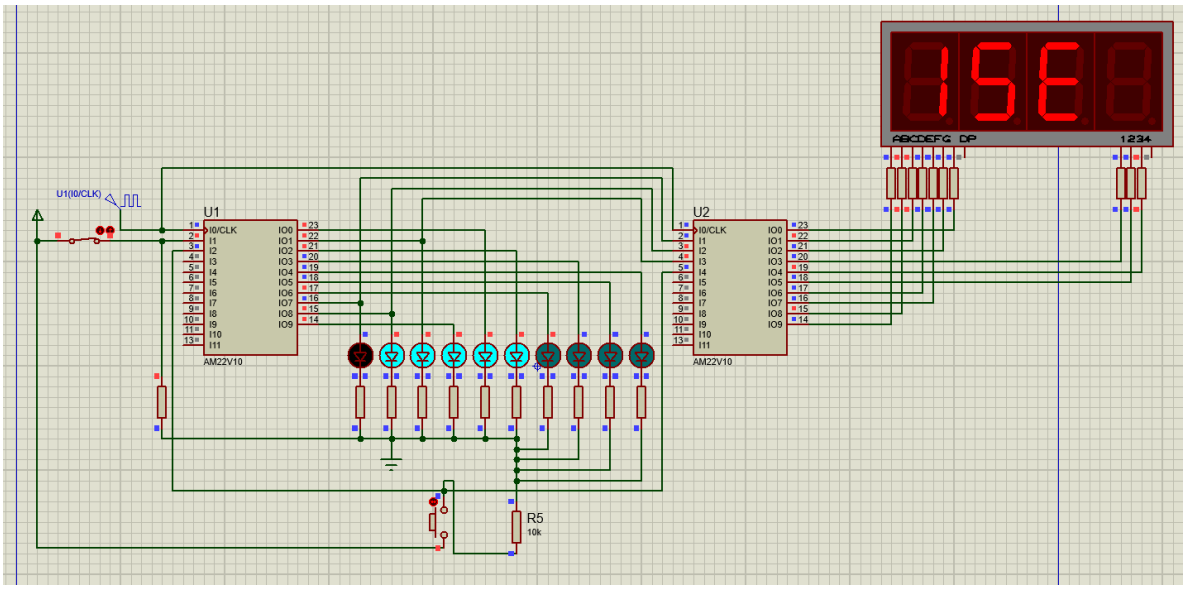


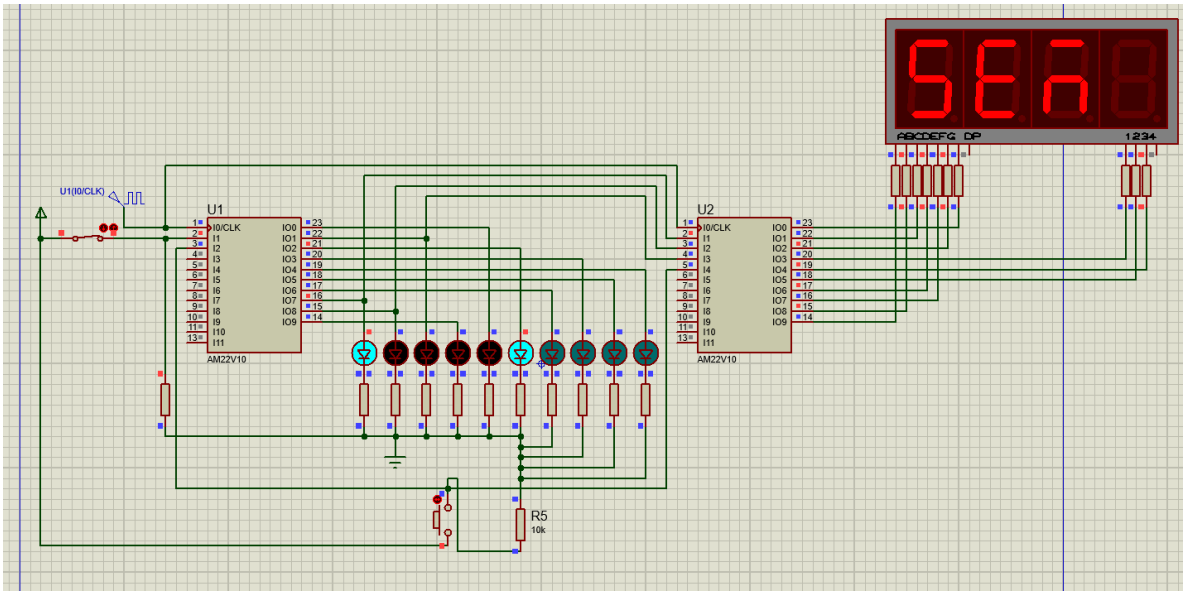

```
                         c22v10

              __________________________________________
        clk =| 1|                                    |24|* not used
     dir(2) =| 2|                                    |23|= display(0)
     dir(1) =| 3|                                    |22|= display(4)
     dir(0) =| 4|                                    |21|= display(1)
        clr =| 5|                                    |20|= control(2)
   not used *| 6|                                    |19|= control(0)
   not used *| 7|                                    |18|= control(1)
   not used *| 8|                                    |17|= display(3)
   not used *| 9|                                    |16|= display(2)
   not used *|10|                                    |15|= display(5)
   not used *|11|                                    |14|= display(6)
   not used *|12|                                    |13|* not used
              __________________________________________
```

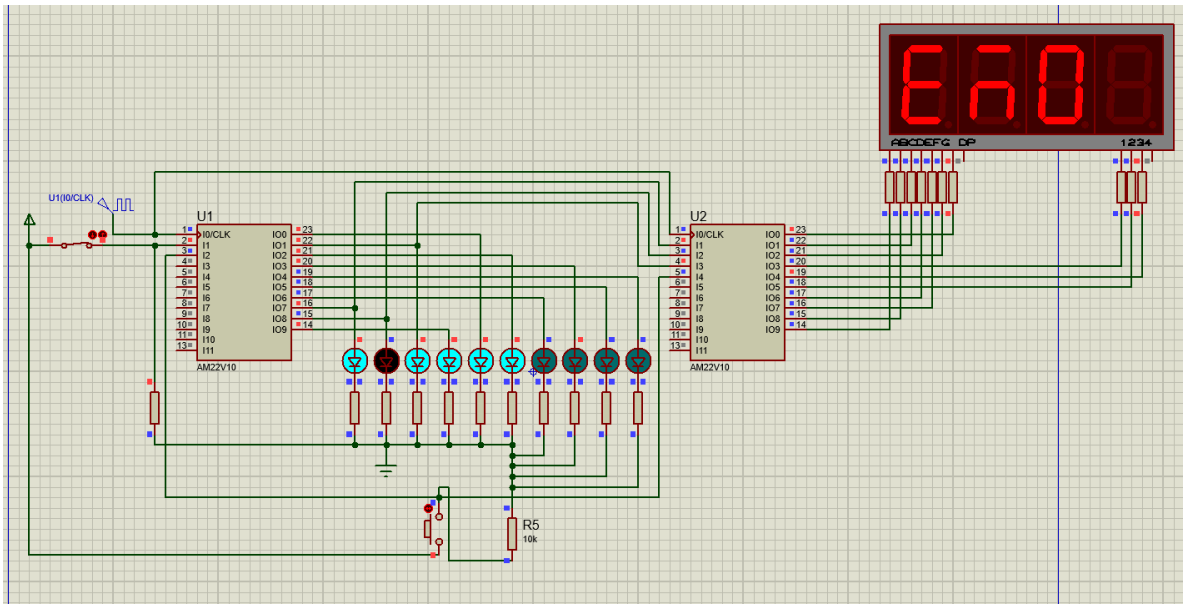


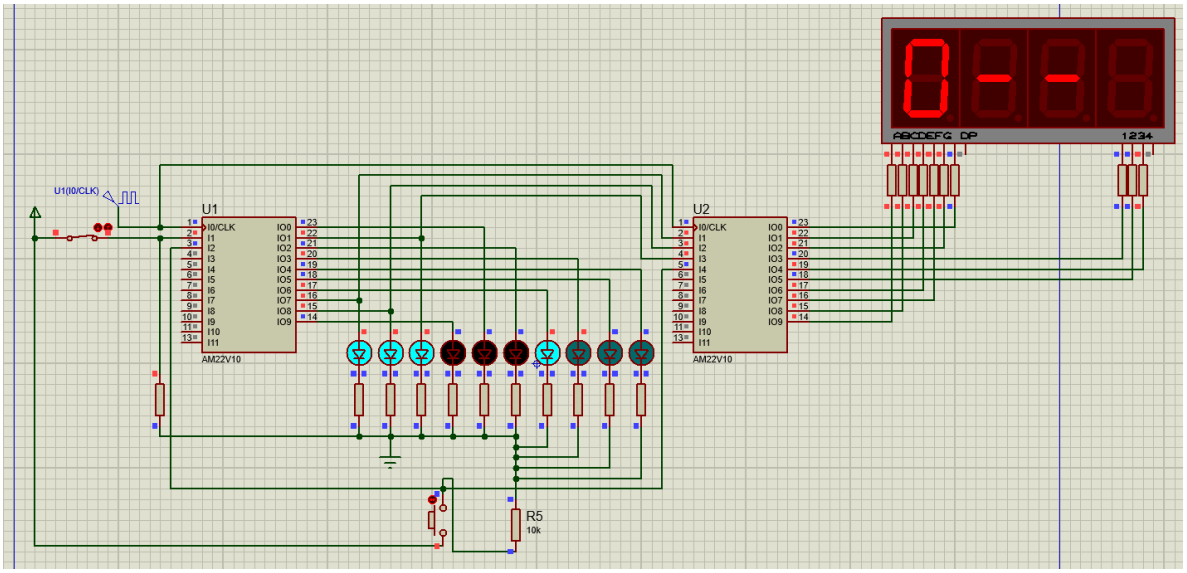

U1(I0/CLK)
U1
U2
AM22V10
R5
10k
ABCDEFG DP
1234


U1(I0/CLK)
U1
U2
AM22V10
R5
10k
ABCDEFG DP
1234


U1(I0/CLK)
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
U2
R5
10k
ABCDEFG DP
1234


U1(I0/CLK)
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
U2
R5
10k
ABCDEFG DP
1234


U1(I0/CLK)
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
U2
R5
10k
ABCDEFG DP
1234


U1(I0/CLK)
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
U2
R5
10k
ABCDEFG DP
1234

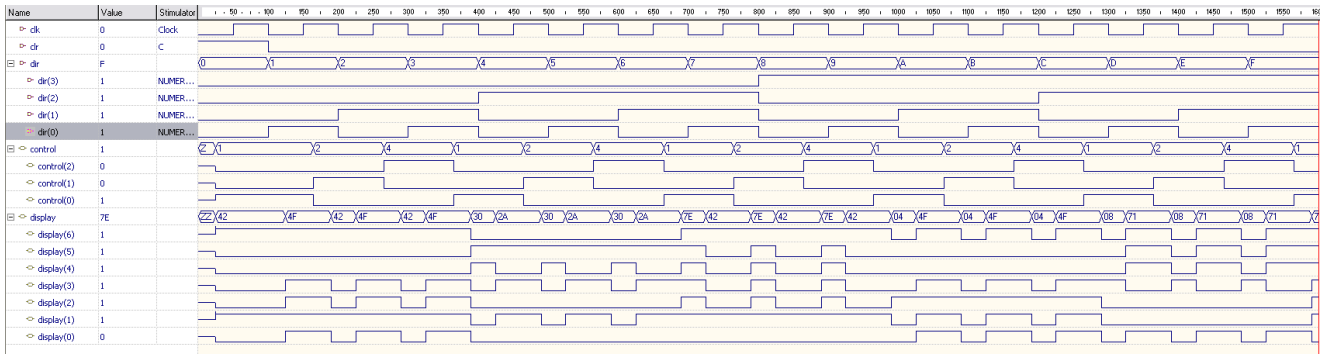
Diseño 2

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity rom is port
(
    clk,clr : in std_logic;
    dir : in std_logic_vector(3 downto 0);
    control : out std_logic_vector(2 downto 0);
    display : out std_logic_vector(6 downto 0)
);
end rom;

architecture Arom of rom is
signal dato0 : std_logic_vector(3 downto 0);
signal dato1 : std_logic_vector(3 downto 0);
signal dato2 : std_logic_vector(3 downto 0);
signal aux : std_logic_vector(3 downto 0);
constant guion : std_logic_vector(3 downto 0) := "0000";
constant d : std_logic_vector(3 downto 0) := "0001";
constant I : std_logic_vector(3 downto 0) := "0011";
constant S : std_logic_vector(3 downto 0) := "0010";
constant E : std_logic_vector(3 downto 0) := "0110";
constant n : std_logic_vector(3 downto 0) := "0111";
constant o : std_logic_vector(3 downto 0) := "0101";
constant g : std_logic_vector(3 downto 0) := "0100";
constant t : std_logic_vector(3 downto 0) := "1100";
constant A : std_logic_vector(3 downto 0) := "1101";
constant L : std_logic_vector(3 downto 0) := "1111";
constant X : std_logic_vector(3 downto 0) := "1110";

```

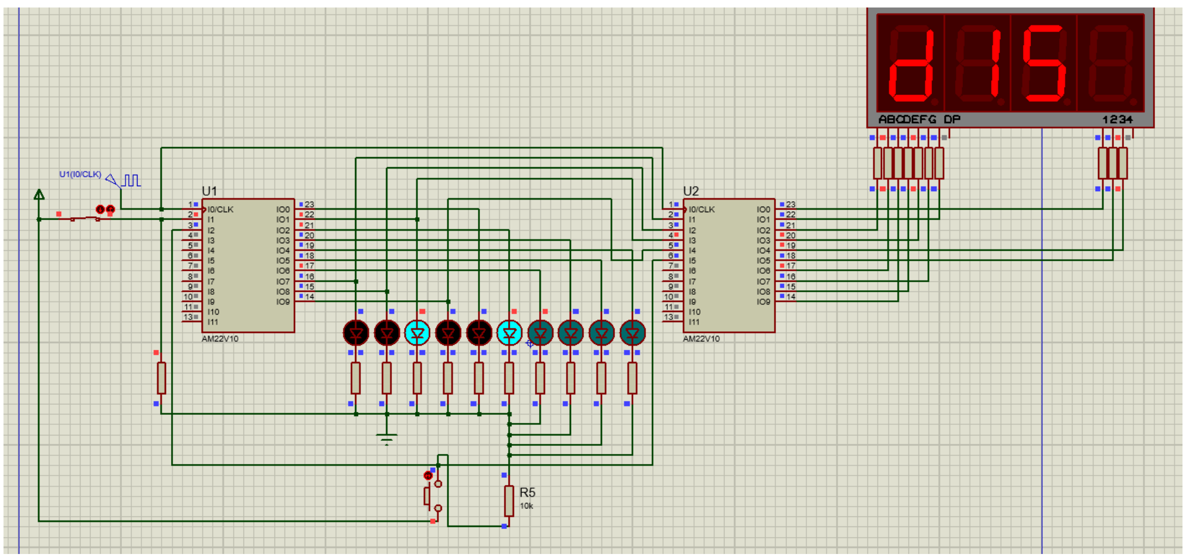
```
33 type memoria is array (0 to 15)of std_logic_vector(3 downto 0);
34 constant banco2 : memoria :=(
35     guion, --"1111110", --"-"
36     guion, --"1111110", --"-"
37     d,--"1000010", --"d"
38     I,--"1001111", --"I"
39     S,--"0100100", --"S"
40     E,--"0110000", --"E"
41     n,--"0101010", --"ñ"
42     O,--"0000001", --"O"
43     guion,--"1111111",
44     d,--"1000010", --"d"
45     I,--"1001111", --"I"
46     g,--"0000100", --"g"
47     I,--"1001111", --"I"
48     t,--"1110000", --"t"
49     A,--"0001000", --"A"
50     L --"1110001" --"L"
51 );
52 constant banco1 : memoria := (
53     guion, --"1111110", --"-"
54     d,--"1000010", --"d"
55     I,--"1001111", --"I"
56     S,--"0100100", --"S"
57     E,--"0110000", --"E"
58     n,--"0101010", --"ñ"
59     O,--"0000001", --"O"
60     guion,--"1111111",
61     d,--"1000010", --"d"
62     I,--"1001111", --"I"
63     g,--"0000100", --"g"
64     I,--"1001111", --"I"
65     t,--"1110000", --"t"
66     A,--"0001000", --"A"
67     L, --"1110001" --"L"
68     guion --"1111110", --"-"
69 );
```

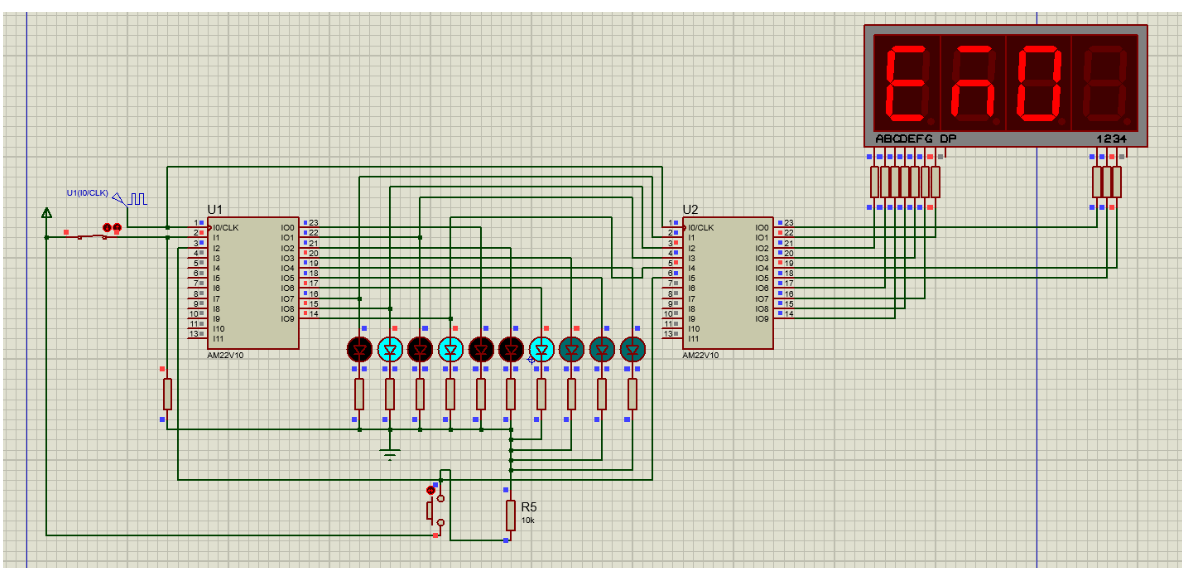
```
constant banco0 : memoria := (
    d,--"1000010", --"d"
    I,--"1001111", --"I"
    S,--"0100100", --"S"
    E,--"0110000", --"E"
    n,--"0101010", --"ñ"
    O,--"0000001", --"O"
    guion,--"1111111",
    d,--"1000010", --"d"
    I,--"1001111", --"I"
    g,--"0000100", --"g"
    I,--"1001111", --"I"
    t,--"1110000", --"t"
    A,--"0001000", --"A"
    L, --"1110001" --"L"
    guion, --"1111110", --"-"
    guion --"1111110", --"-"
);
begin
--contador de anillo
    process(clr,clk)
    begin
        if(clr='1')then
            control <= "001";
            display <= "0000000";
        elsif(rising_edge(clk))then
            case control is
                when "001" => control <= "010";
                when "010" => control <= "100";
                when "100" => control <= "001";
                when others => control <= "---";
            end case;
        end if;
    end process;
--fin del contador
--memoria
    process(dir)
        begin
        dato0 <= banco0(conv_integer(dir));
        dato1 <= banco1(conv_integer(dir));
        dato2 <= banco2(conv_integer(dir));
    end process;
--fin de la memoria
```

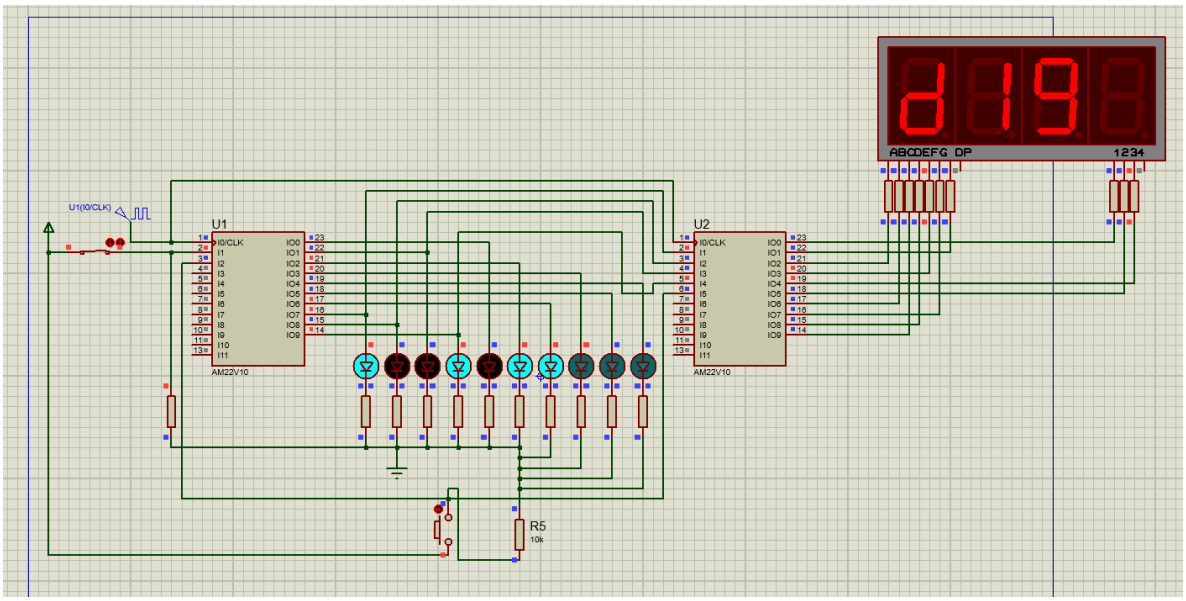
```
114 --Multiplexor--
115     process(control,dato0,dato1,dato2)
116         begin
117             if(control = "001") then
118                 aux <= dato0;
119             elsif(control = "010") then
120                 aux <= dato1;
121             elsif(control = "100") then
122                 aux <= dato2;
123             else
124                 aux <= "1110"; --no importa
125             end if;
126         end process;
127 --Decodificador
128     process(aux)
129         begin
130             case aux is
131                 when "0000" =>display <= "1111110"; --"-"
132                 when "0001" =>display <= "1000010"; --"d"
133                 when "0011" =>display <= "1001111"; --"I"
134                 when "0010" =>display <= "0100100"; --"S"
135                 when "0110" =>display <= "0110000"; --"E"
136                 when "0111" =>display <= "0101010"; --"ñ"
137                 when "0101" =>display <= "0000001"; --"o"
138                 when "0100" =>display <= "0000100"; --"g"
139                 when "1100" =>display <= "1110000"; --"t"
140                 when "1101" =>display <= "0001000"; --"A"
141                 when "1111" =>display <= "1110001"; --"L"
142                 -- when "1110" =>display <= "-------";
143                 -- when "1010" =>display <= "-------";
144                 -- when "1011" =>display <= "-------";
145                 when others => display <= "-------";
146 end case;
147 end process;
148 end Arom;
```

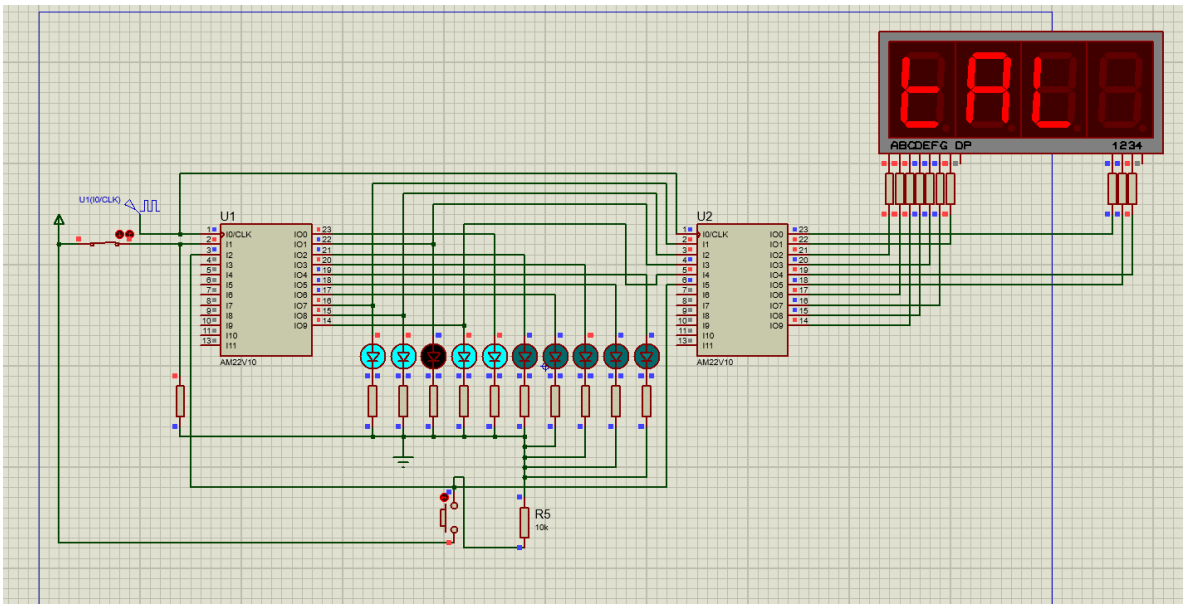


```
                              C22V10

                 __________________________________________
         clk =|  1|                                      |24|* not used
     dir(3)  =|  2|                                      |23|= control(2)
     dir(2)  =|  3|                                      |22|= display(0)
     dir(1)  =|  4|                                      |21|= display(6)
     dir(0)  =|  5|                                      |20|= display(2)
         clr =|  6|                                      |19|= control(0)
   not used *|  7|                                      |18|= control(1)
   not used *|  8|                                      |17|= display(5)
   not used *|  9|                                      |16|= display(1)
   not used *|10|                                      |15|= display(3)
   not used *|11|                                      |14|= display(4)
   not used *|12|                                      |13|* not used
                 __________________________________________
```






d19
ABCDEFG DP
1234
U1(I0/CLK)
U1
U2
AM22V10
R5
10k


tAL
ABCDEFG DP
1234
U1(I0/CLK)
U1
U2
AM22V10
R5
10k