Código VHDL

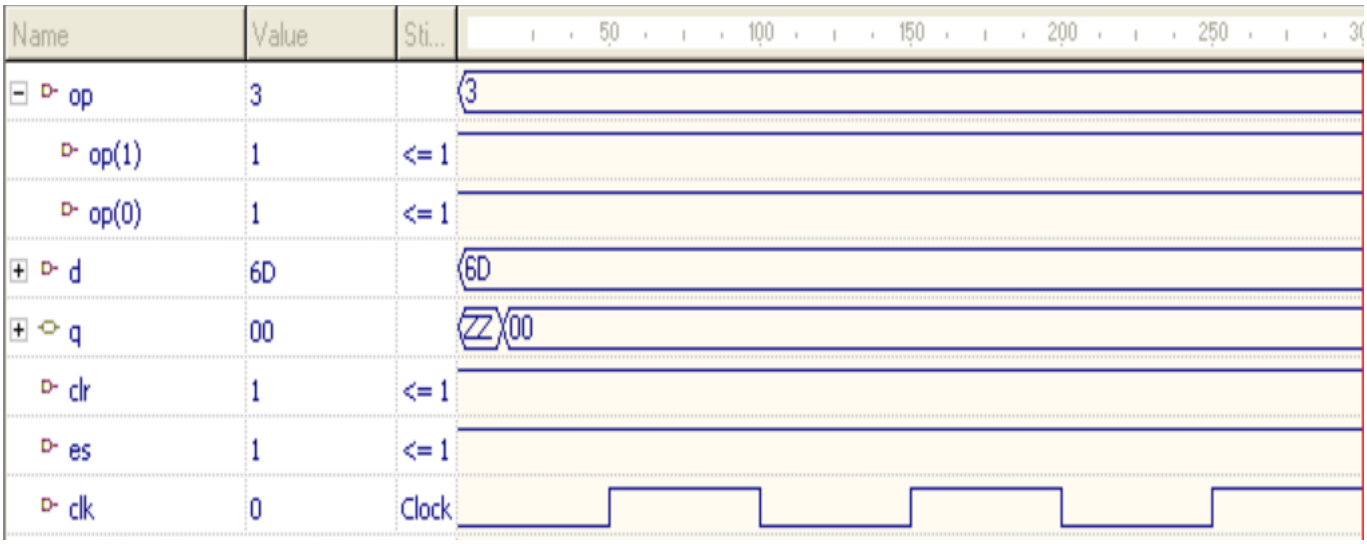
```
 1 --Ulises Jesùs Santos Mèndez
 2 --Pràctica 3 Registros
 3 --2CV8
 4 library ieee;
 5 use ieee.std_logic_1164.all;
 6 use work.std_arith.all;
 7
 8 entity registro is
 9 port(es,clk,clr: in std_logic;
10      d: in std_logic_vector(6 downto 0);
11      op: in std_logic_vector(1 downto 0);
12      q: out std_logic_vector(6 downto 0));
13
14     attribute pin_numbers of registro: entity is
15     "clk:1 es:2 d(6):3 d(5):4 d(4):5 d(3):6 d(2):7 "
16 &   "d(1):8 d(0):9 op(0):10 op(1):11 clr:13 q(6):20 "
17 &   "q(5):19 q(4):18 q(3):17 q(2):16 q(1):15 q(0):14 ";
18
19 end registro;
20
21 architecture arqreg of registro is
22     signal auxD,auxQ: std_logic_vector (6 downto 0);
23 begin
24 --Ciclo concurrente del multiplexor
```

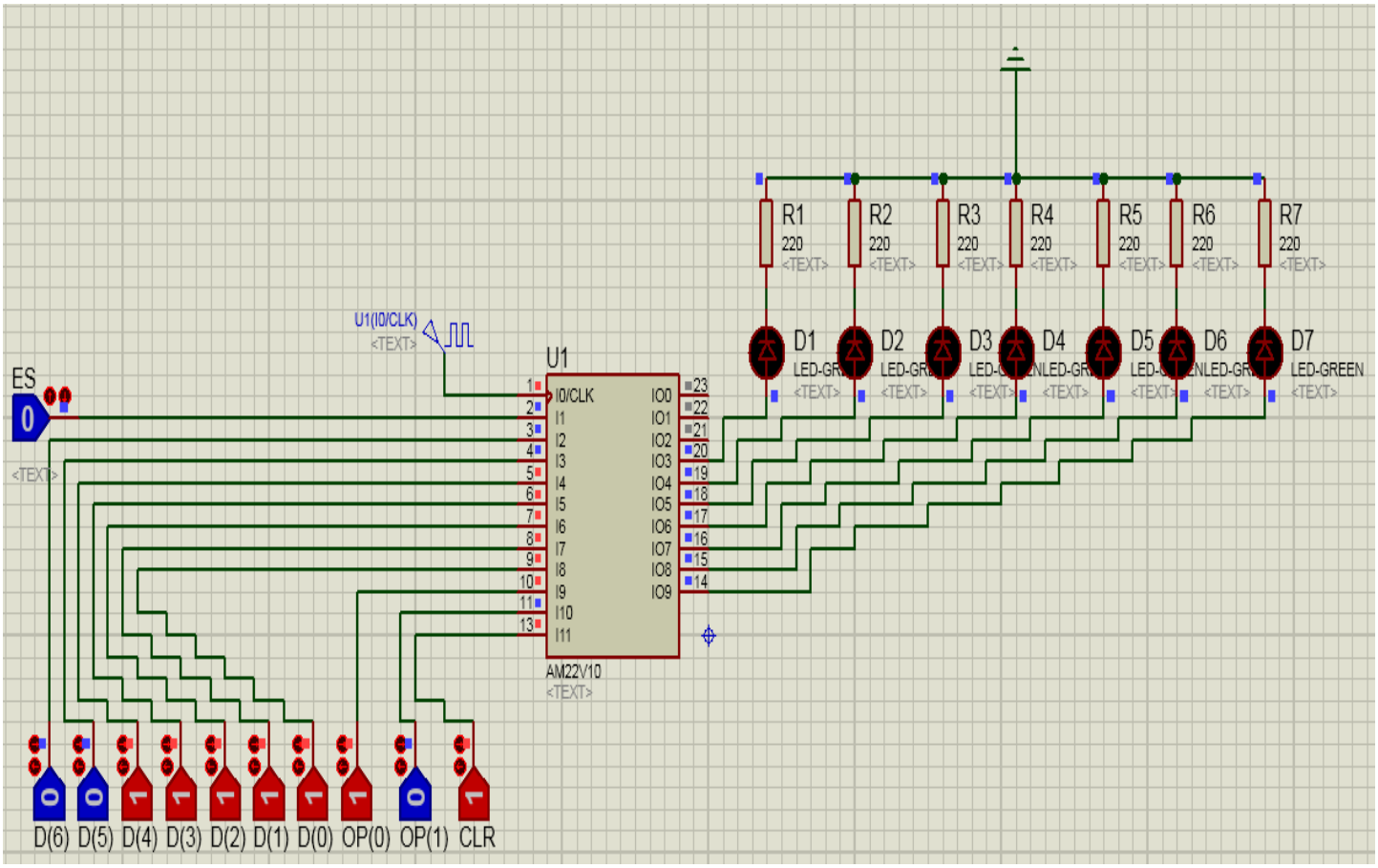
```
25 -- 00 -> retencion
26 -- 01 -> carga
27 -- 10 -> corrimiento izquierda
28 -- 11 -> corrimiento derecha
29     process(op,d,auxQ,es, auxD)
30     begin
31         case op is
32             when "00" =>
33                 auxD <= auxQ;
34             when "01" =>
35                 auxD <= d;
36             when "10" =>
37                 for i in 0 to 6 loop
38                     if(i>0) then
39                     auxD(i)<=auxQ(i-1);
40                     else
41                     auxD(i)<=es;
42                     end if;
43                 end loop;
44             when "11" =>
45                 for i in 0 to 6 loop
46                     if(i<6) then
47                     auxD(i)<=auxQ(i+1);
48                     else
```

```
49                      auxD(i)<=es;
50                      end if;
51                  end loop;
52              when others =>
53                  auxD<="0000000";
54              end case;
55               end process;
56
57
58  --Ciclo secuencial para la serie de flip flops
59      process(clk, clr)
60      begin
61          if(clr = '1')then
62              for i in 0 to 6 loop
63              auxQ(i)<='0';
64          end loop;
65          elsif(rising_edge(clk))then
66              for i in 0 to 6 loop
67              auxQ <= auxD;
68              end loop;
69          end if;
70      end process;
71      q<= auxQ;
72  end arqreg;
```

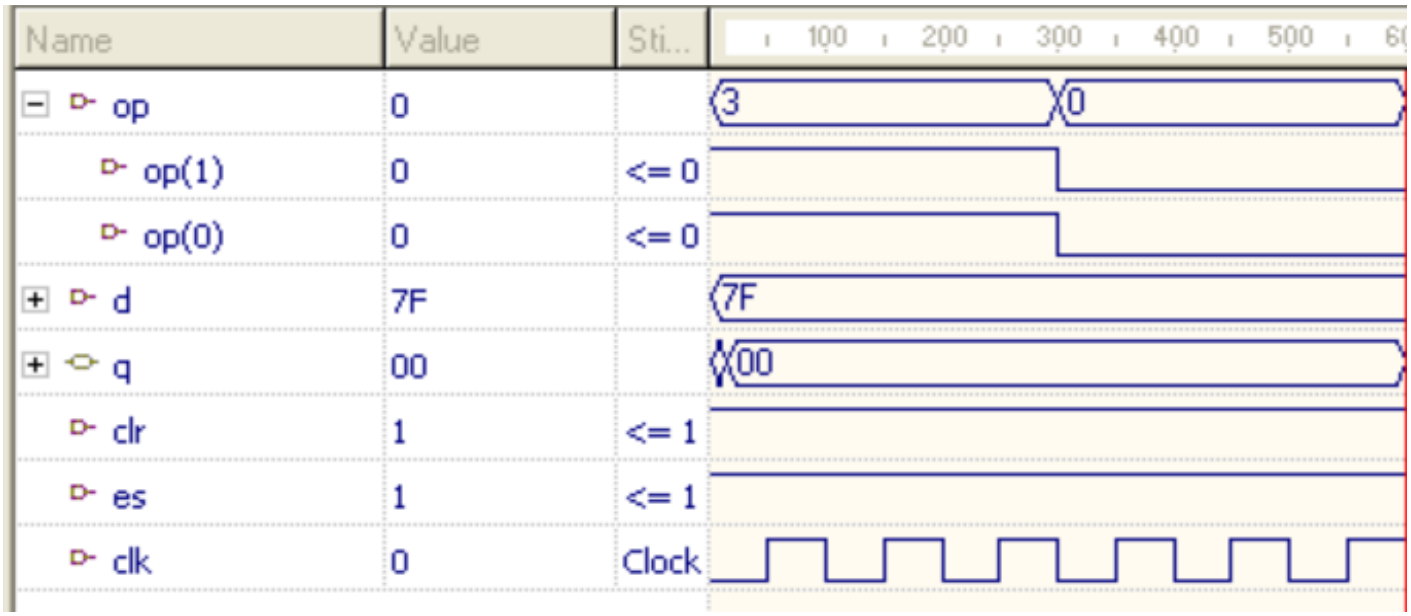
Simulación en Active HDL-Sim y en Proteus

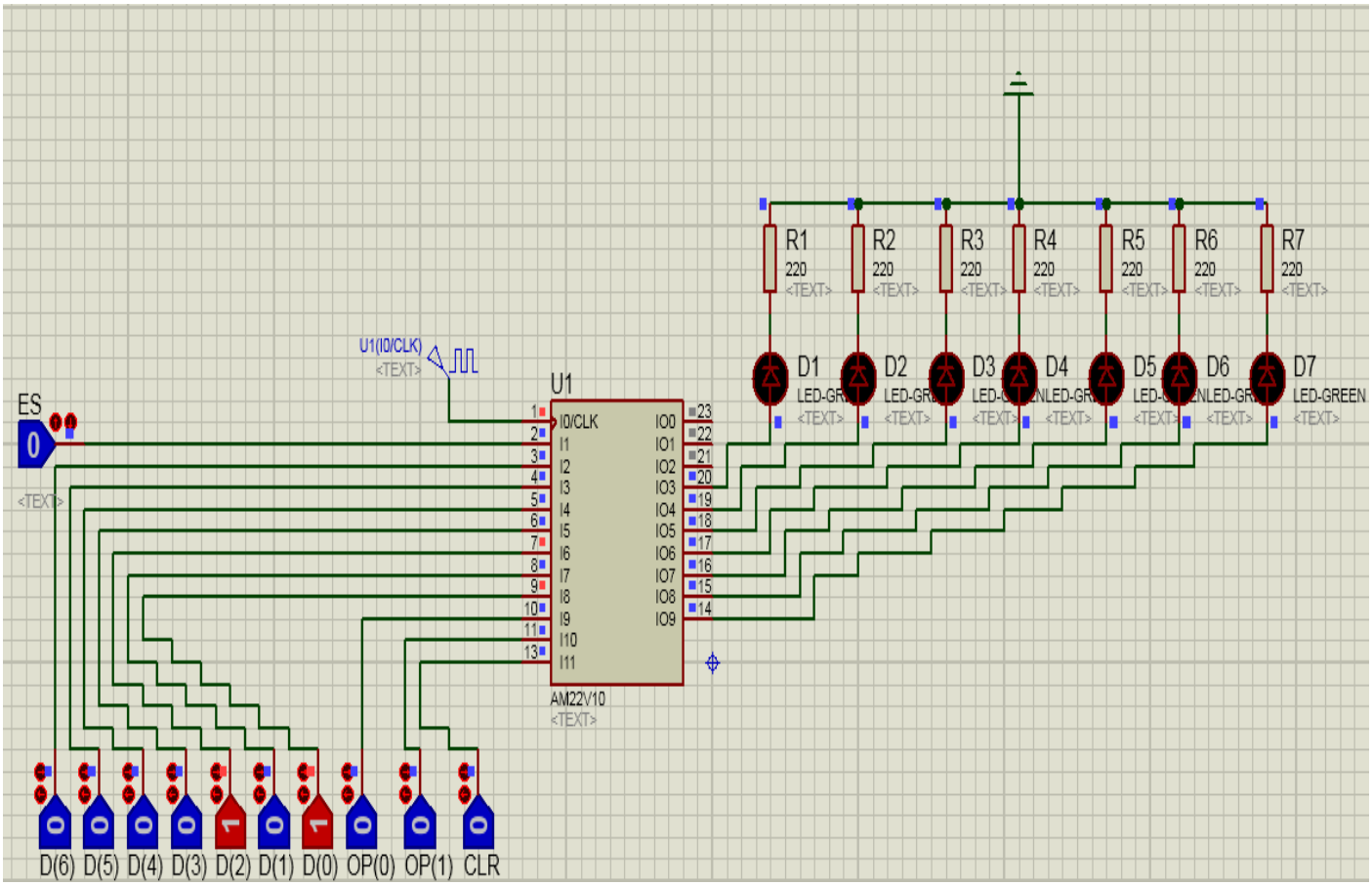
a) Reset



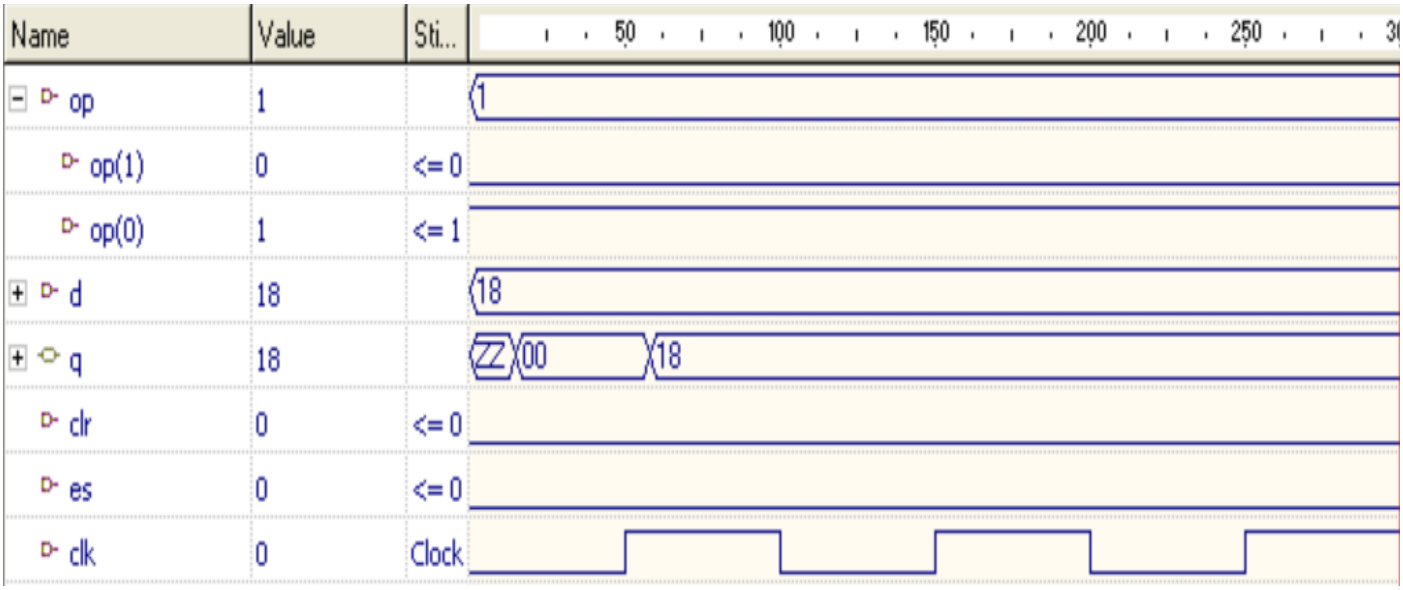


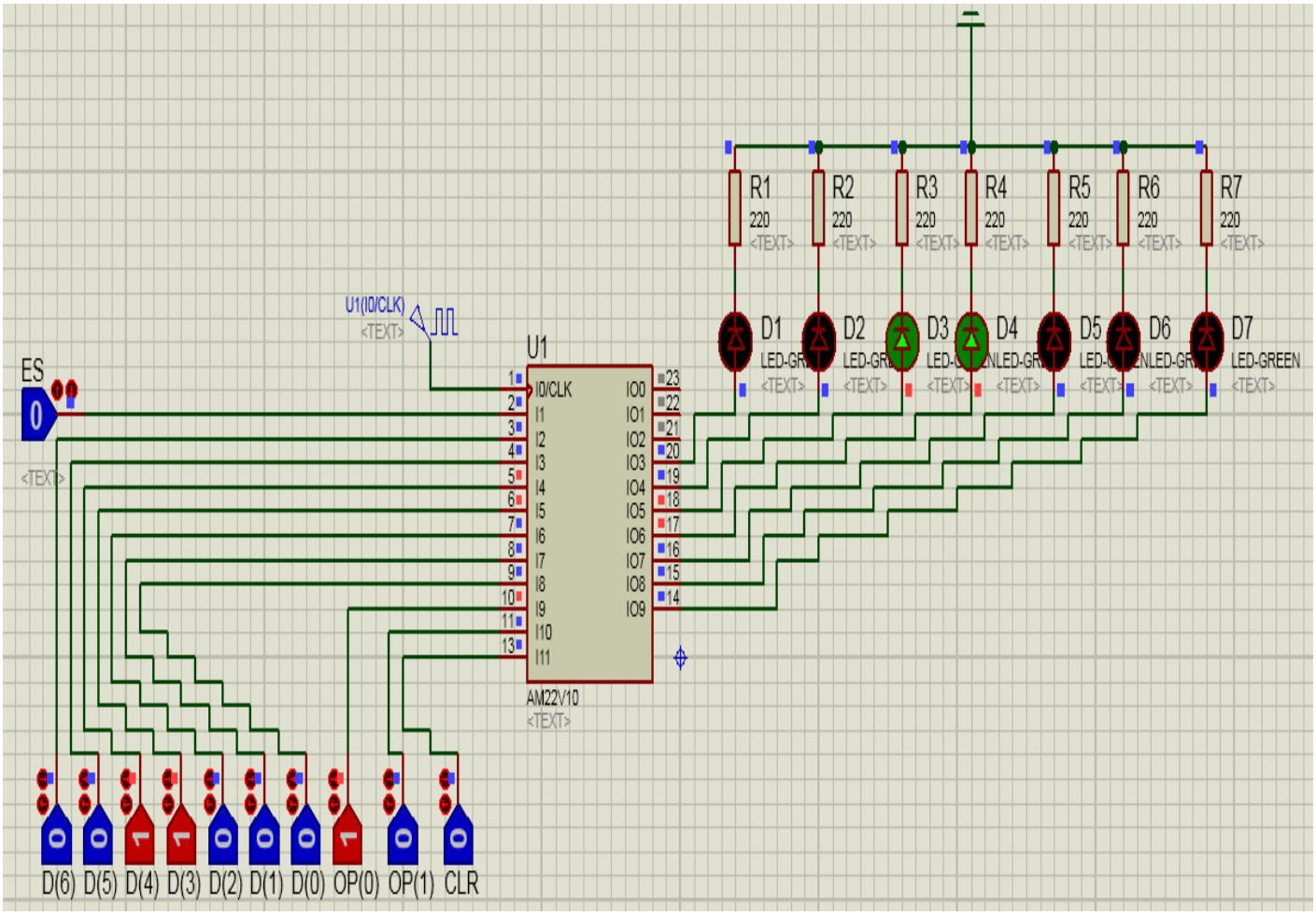
b) Retención durante dos ciclos de reloj



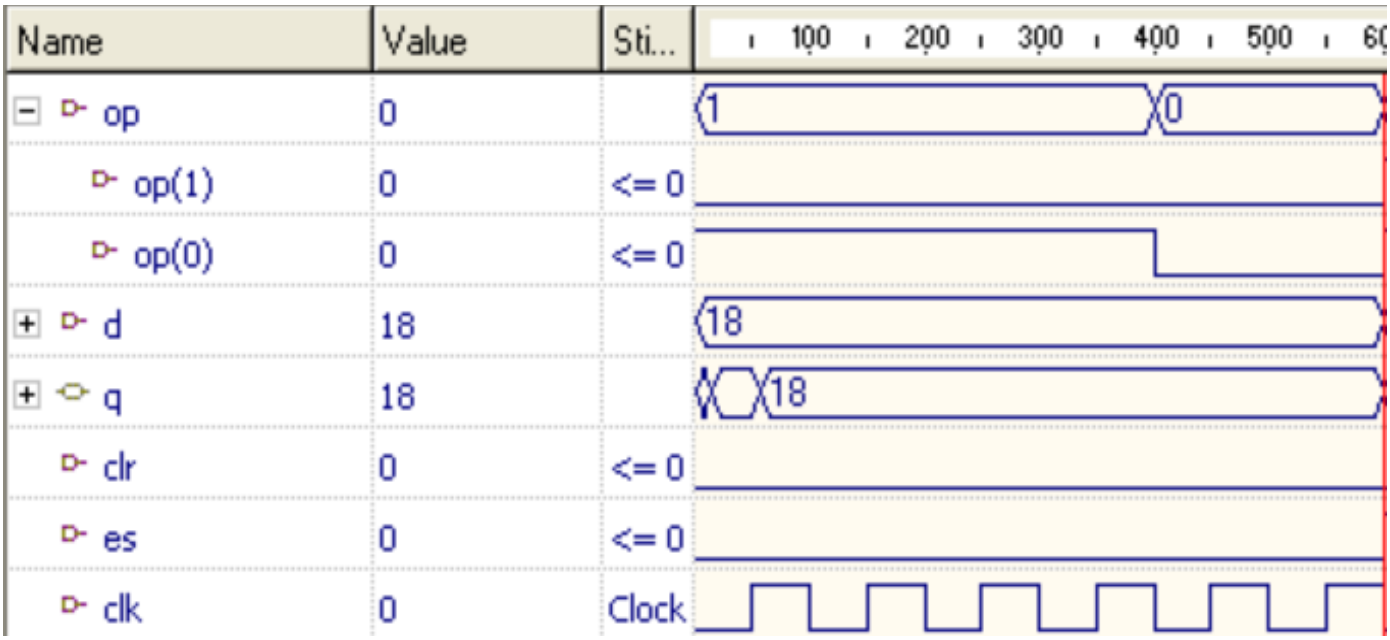


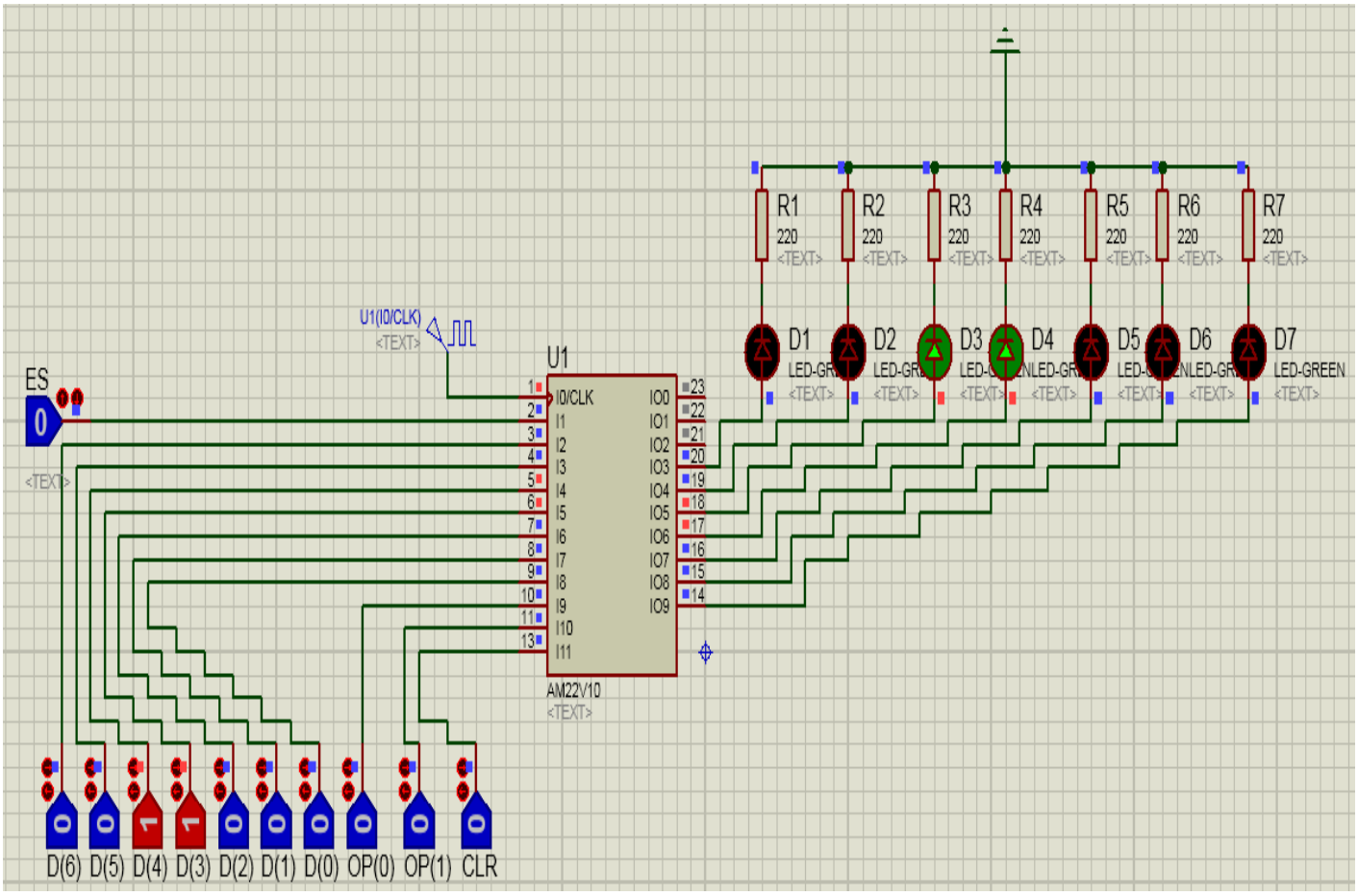
c) Cargar el valor 0x18



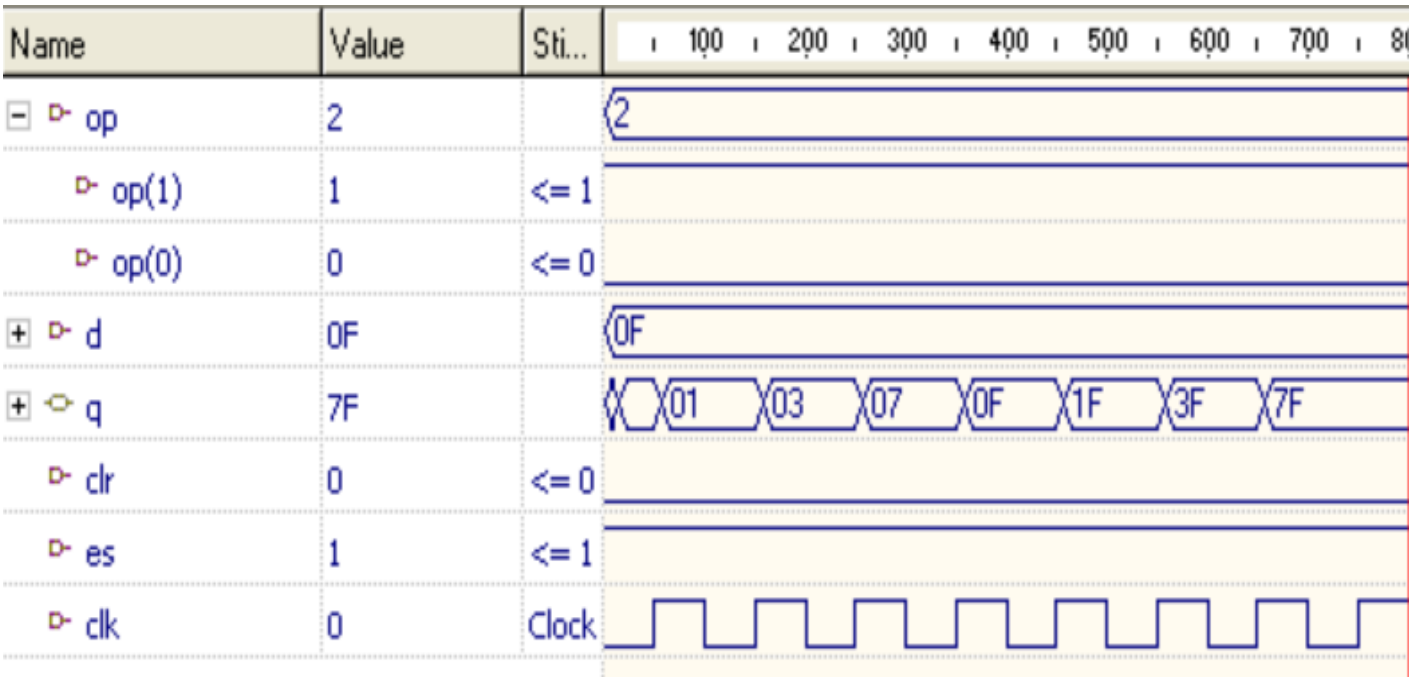


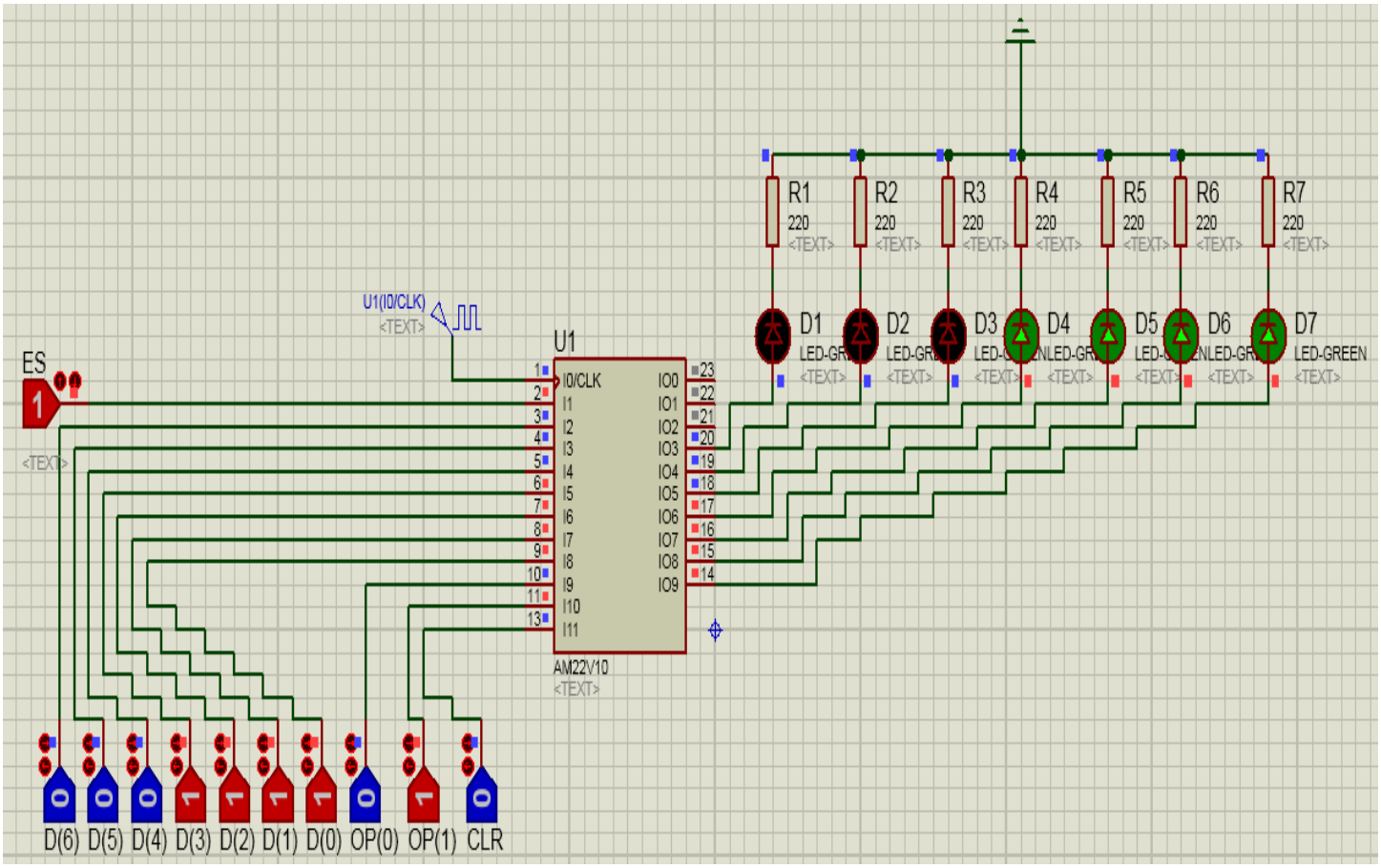
d) Retener 2 ciclos de reloj



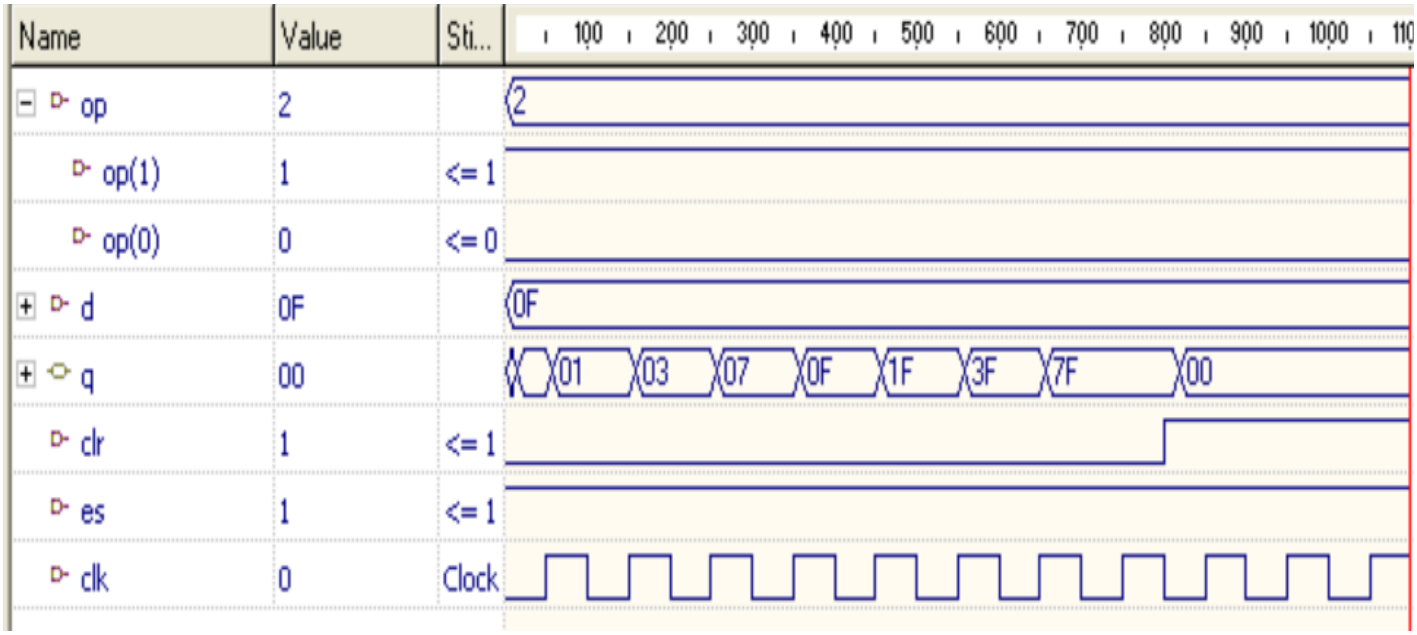


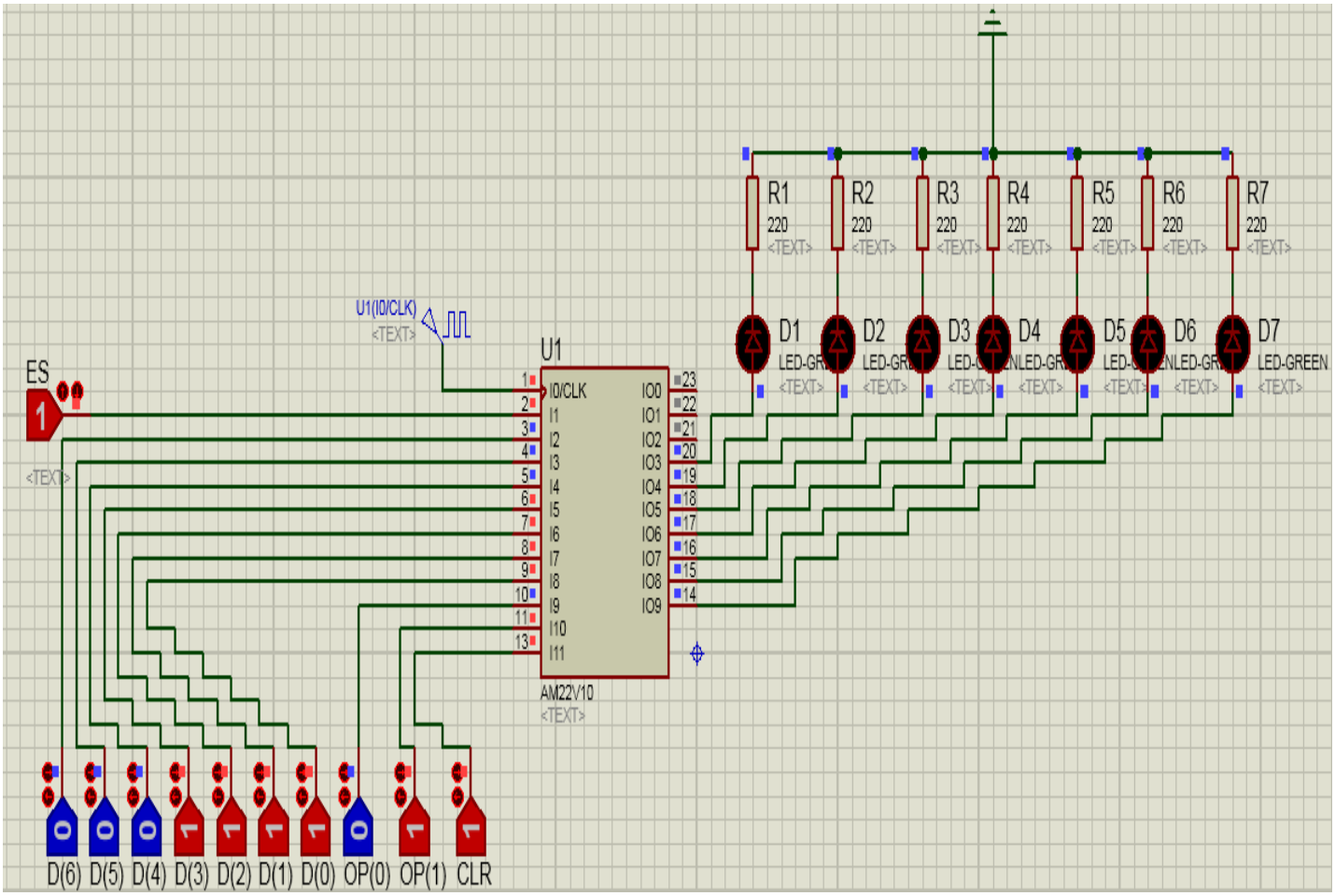
e) Hacer corrimiento a la izquierda de 4 bits con ES=1



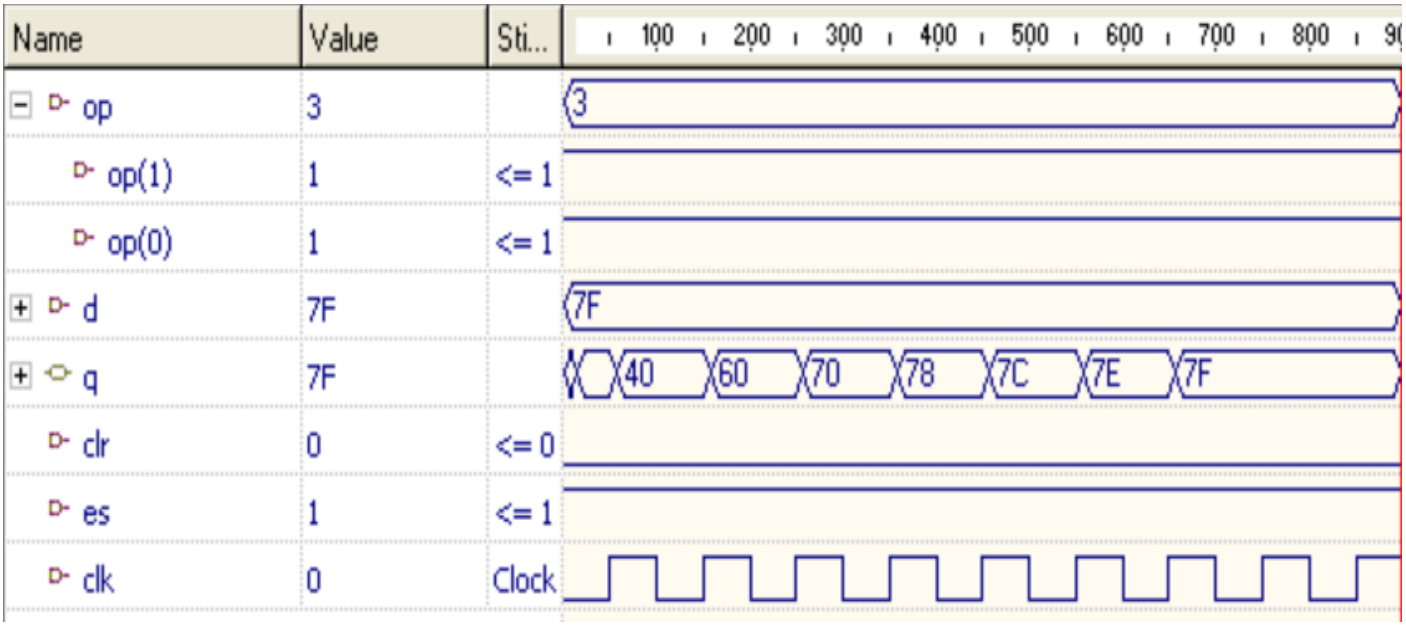


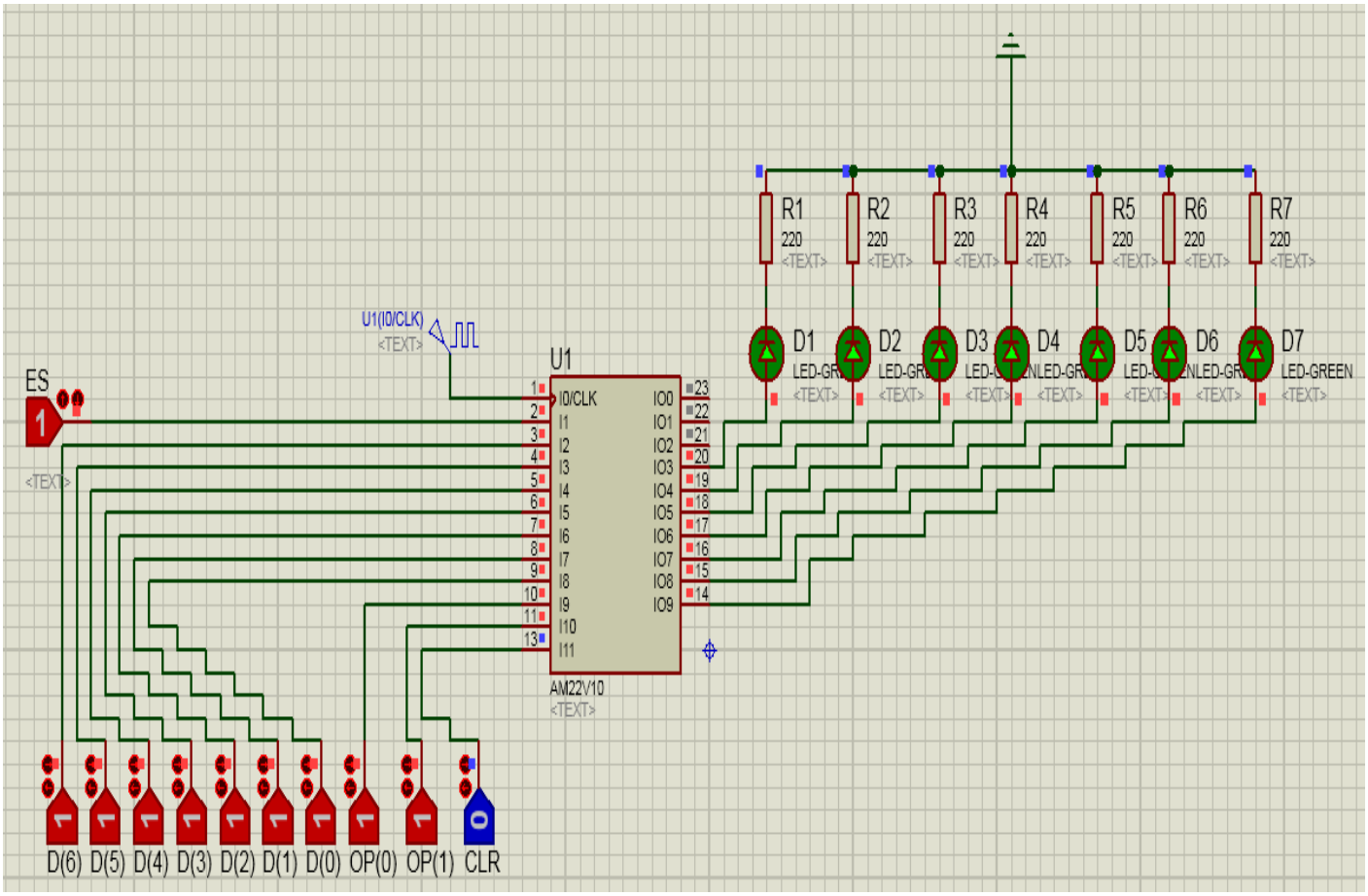
f) Reset


Name
Value
Sti...
100
200
300
400
500
600
700
800
900
1000
op
2
op(1)
1
<= 1
op(0)
0
<= 0
d
0F
q
00
01
03
07
0F
1F
3F
7F
00
clr
1
<= 1
es
1
<= 1
clk
0
Clock


R1
R2
R3
R4
R5
R6
R7
220
D1
D2
D3
D4
D5
D6
D7
LED-GREEN
U1(ID/CLK)
U1
ID/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
ES
D(6)
D(5)
D(4)
D(3)
D(2)
D(1)
D(0)
OP(0)
OP(1)
CLR

g) Hacer corrimiento a la derecha hasta que el registro quede lleno de 1’s





CUESTIONARIO

1) ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?
   R= 1 dispositivo PLD 22V10
2) ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?
   R= 2 Flip Flop 4013 y 4 multiplexores 74ls139
3) ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?
   R=12 de entrada y 7 de salida
4) ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?
5) R= 4 términos producto y se ocupó el 86% del PLD 22V10.
6) ¿Cuáles son tus observaciones con respecto al funcionamiento del registro?
   R= todos los registros tienen una gran relación de entradas y salidas de los flip flop de manera que pueden ejecutar las operaciones debido a entradas y salidas y a los flancos del reloj.
7) ¿Cuáles son las señales que funcionan de manera síncrona y cuáles de manera asíncrona?
   R= El reloj funciona de manera síncrona y los registros de corrimiento a derecha e izquierda, los asíncronos son los registros de carga y retención.
8) ¿Qué puedes concluir de esta práctica?
   R= La práctica nos mostró las formas en que se comunican los flip flops para hacer operaciones y actuar como memorias.