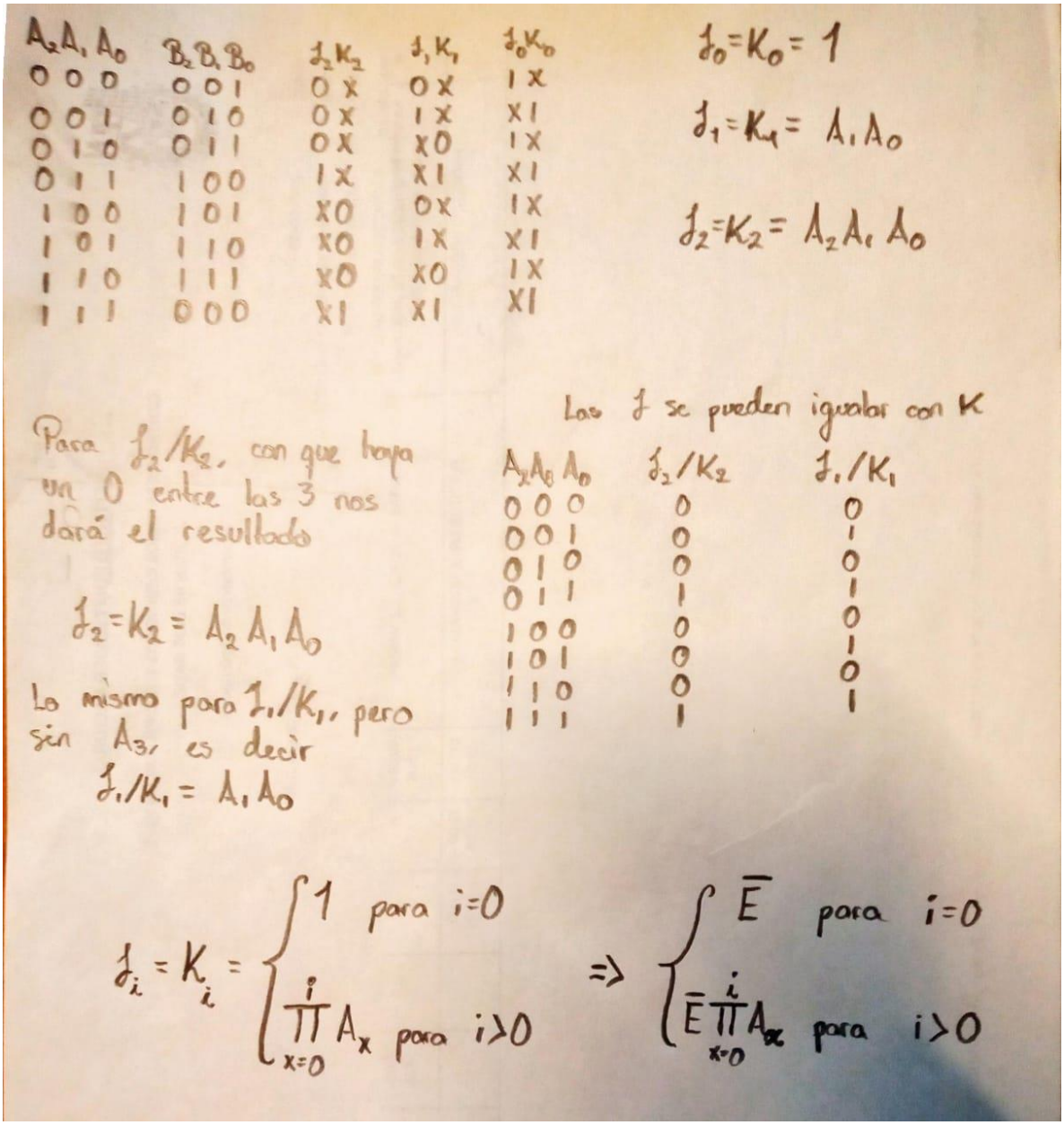
Martínez Coronel Brayan Yosafat

ANÁLISIS PARA FLIP FLOP JK



$A_2A_1A_0$	$B_2B_1B_0$	J_2K_2	J_1K_1	J_0K_0
000	001	0X	0X	1X
001	010	0X	1X	X1
010	011	0X	X0	1X
011	100	1X	X1	X1
100	101	X0	0X	1X
101	110	X0	1X	X1
110	111	X0	X0	1X
111	000	X1	X1	X1

$J_0 = K_0 = 1$

$J_1 = K_1 = A_1 A_0$

$J_2 = K_2 = A_2 A_1 A_0$

Para J_2/K_2, con que haya un 0 entre las 3 nos dará el resultado

$J_2 = K_2 = A_2 A_1 A_0$

Lo mismo para J_1/K_1, pero sin A_3, es decir

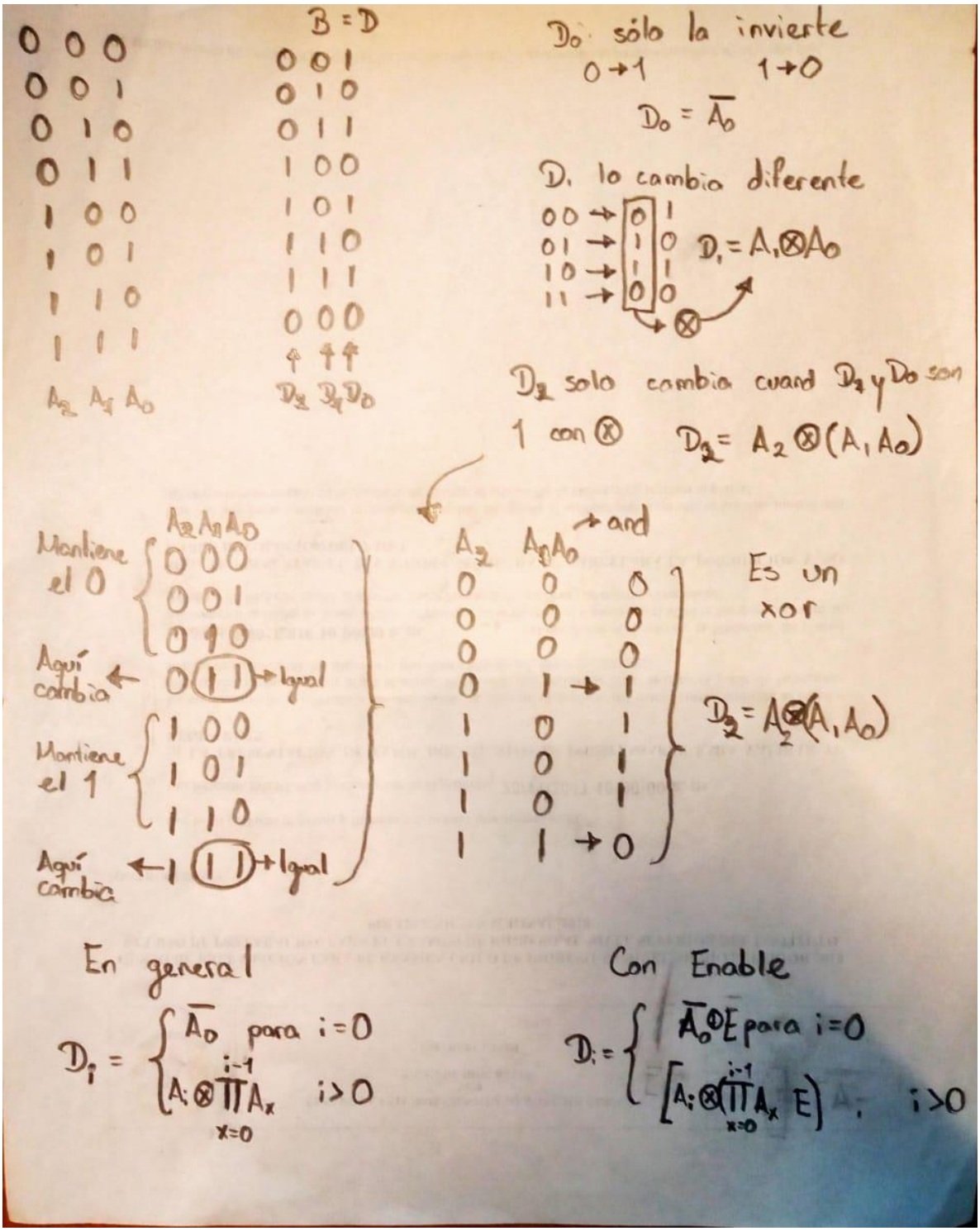
$J_1/K_1 = A_1 A_0$

Las J se pueden igualar con K

$A_2A_1A_0$	J_2/K_2	J_1/K_1
000	0	0
001	0	1
010	0	0
011	1	1
100	0	0
101	0	1
110	0	0
111	1	1

$$J_i = K_i = \begin{cases} 1 \text{ para } i=0 \\ \prod_{x=0}^{i} A_x \text{ para } i>0 \end{cases} \Rightarrow \begin{cases} \bar{E} \text{ para } i=0 \\ \bar{E} \prod_{x=0}^{i} A_x \text{ para } i>0 \end{cases}$$

ANÁLISIS PARA FLIP FLOP D



CÓDIGO FUENTE

CONTADOR DE 3 BITS

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity C3 is port(
5      EN, CLK, CLR : in std_logic;
6      S : inout std_logic_vector (2 downto 0)
7  );
8  end entity;

10 architecture Contador3 of C3 is
11 begin
12      process (CLR, CLK)
13      variable X : std_logic;
14      begin
15          if (CLR = '1') then
16              S <= "000";
17          elsif (rising_edge(CLK)) then
18              X := '1';
19              for j in 0 to 2 loop
20                  if (j-1 >= 0) then
21                      for k in 0 to j-1 loop
22                          X := X and S(k);
23                      end loop;
24                  end if;
25
26                  X := X and EN;
27                  S(j) <= S(j) xor X;
28              end loop;
29          end if;
30      end process;
31 end architecture;
```

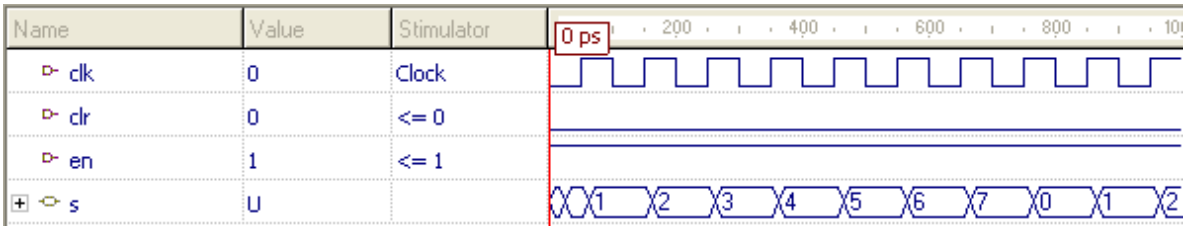
CONTADOR DE 7 BITS

```
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 C7 is port(
7      CLK, CLR, EN, L, UD : in std_logic;
8      D : in std_logic_vector (6 downto 0);
9      C : out std_logic;
10     S : inout std_logic_vector (7 downto 0)
11 );
12 end entity;
```

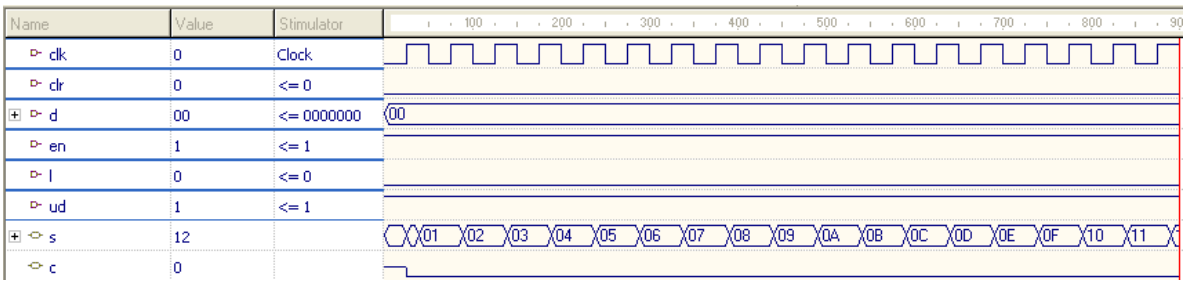
```
14 architecture Contador of C7 is
15 begin
16     process (CLK, CLR)
17     variable x : std_logic := '1';
18     begin
19         if (CLR = '1') then
20             S <= "00000000";
21         elsif (rising_edge(CLK)) then
22             if (EN = '0') then --retiene
23                 S <= S;
24             else
25                 if (L = '1') then --carga
26                     S(6 downto 0) <= D;
27                 else --conteo
28                     if (UD = '1') then --ascendente
29                         S <= S + 1;
30                     else --descendente
31                         S <= S - 1;
32                     end if;
33                 end if;
34             end if;
35         end if;
36
37         C <= S(7);
38     end process;
39 end architecture;
```

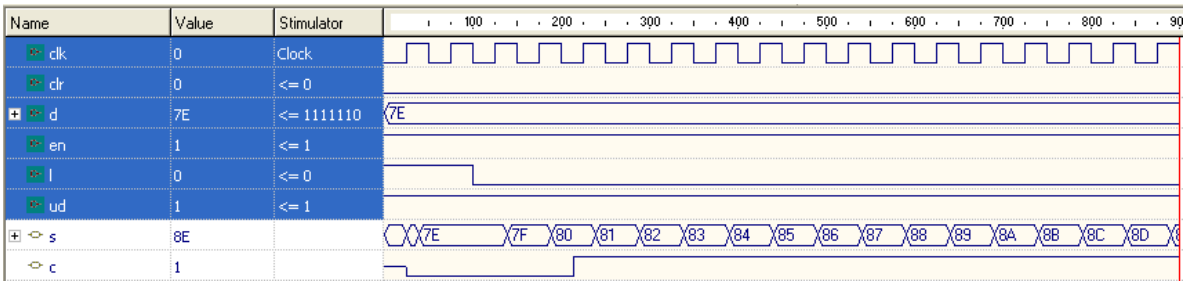
SIMULACIONES EN GALAXY

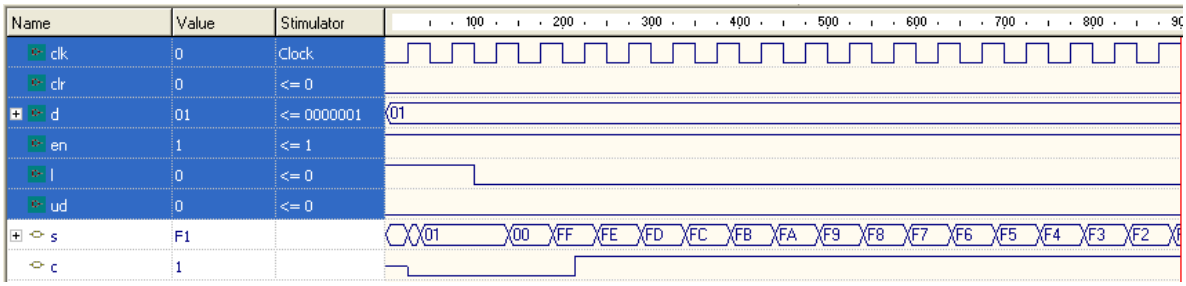
CONTADOR DE 3 BITS



CONTADOR DE 7 BITS





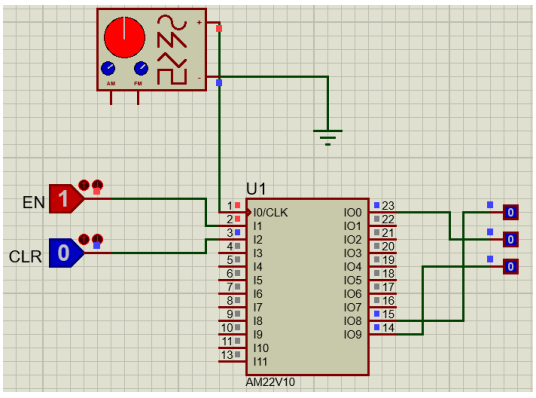


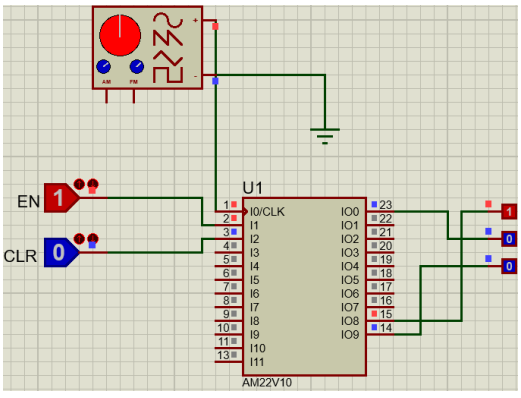
SIMULACIÓN EN PROTEUS

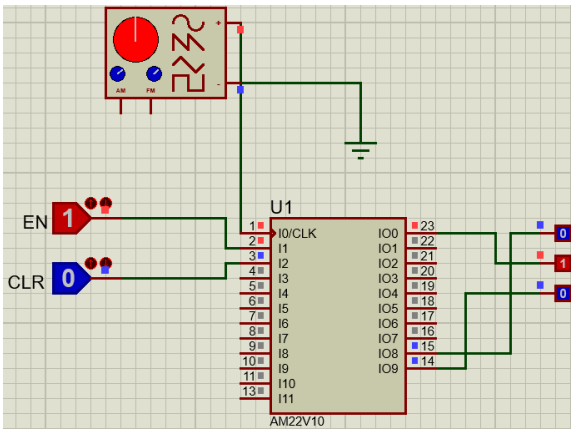
CONTADOR DE 3 BITS

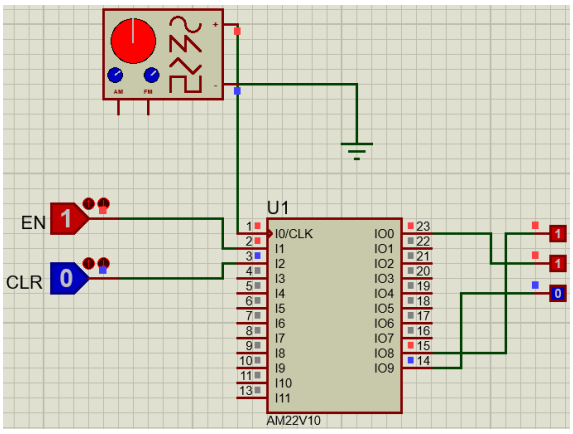
C22V10

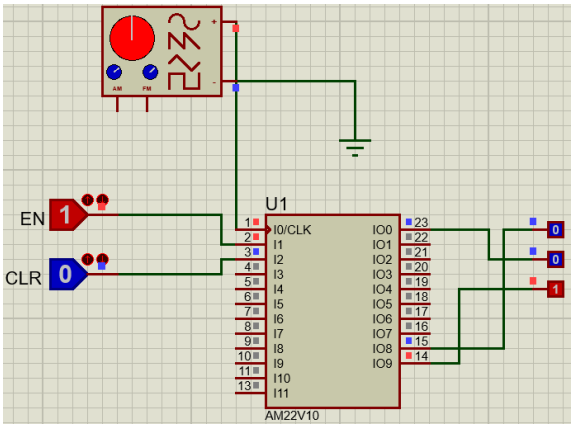
```
      clk =|  1|                              |24|* not used
       en =|  2|                              |23|= s(1)
      clr =|  3|                              |22|* not used
not used *|  4|                              |21|* not used
not used *|  5|                              |20|* not used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

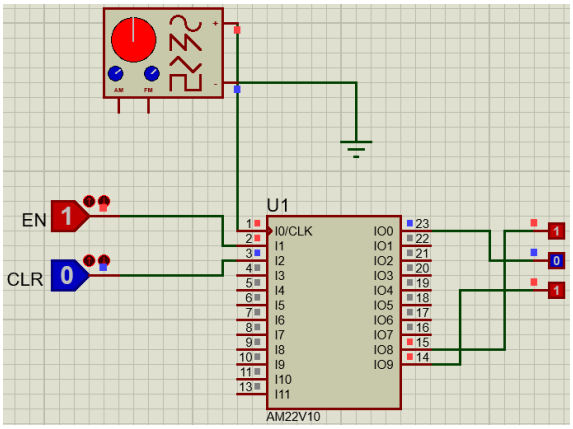


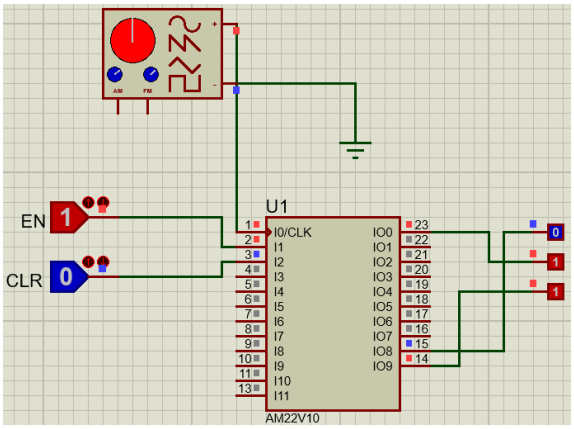


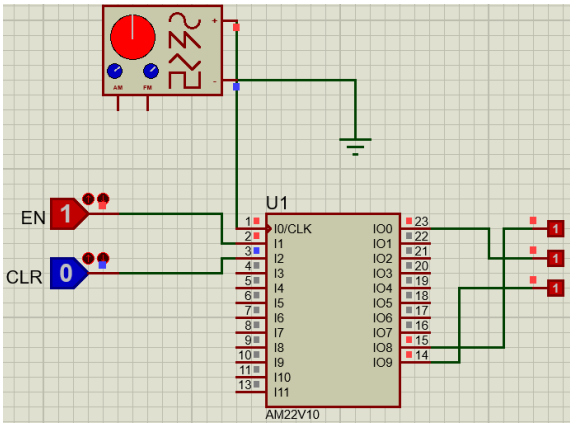

EN
1
CLR
0
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
0
1
0


EN
1
CLR
0
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
1
1
0


EN
1
CLR
0
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
0
0
1


EN
1
CLR
0
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
1
0
1

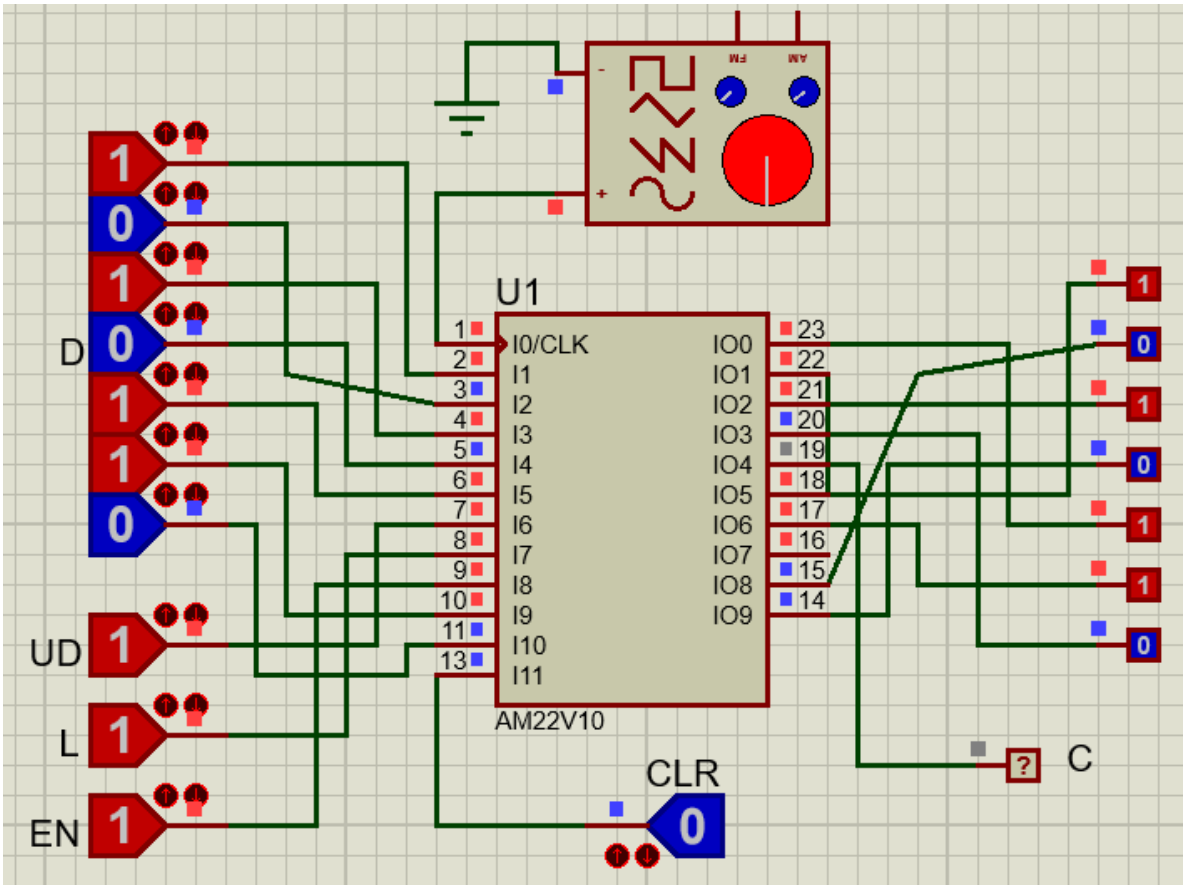

EN
1
CLR
0
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
0
1
1


EN
1
CLR
0
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
1
1
1

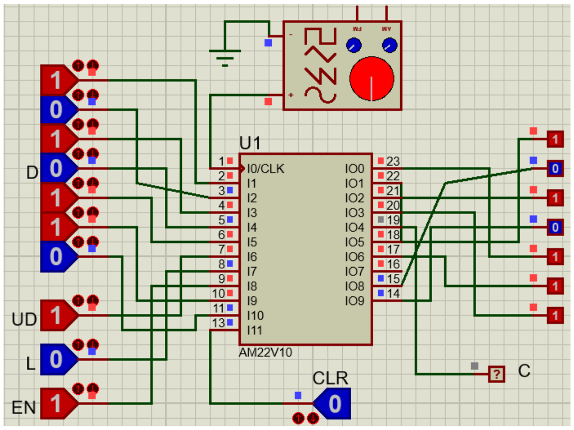
CONTADOR DE 7 BITS

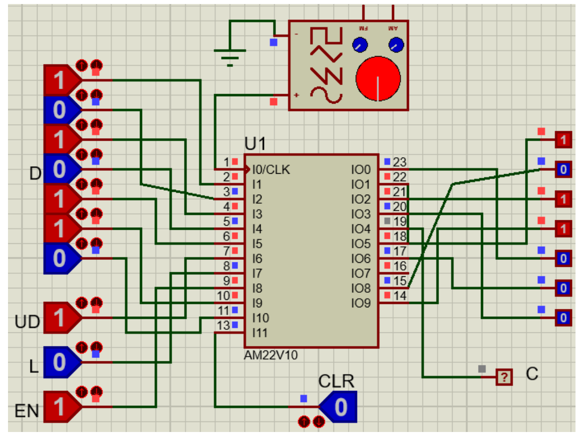
```
                 ____________________________________________
       clk =|  1|                                  |24|* not used
    d(6) =|  2|                                  |23|= s(2)
    d(5) =|  3|                                  |22|= s(4)
    d(4) =|  4|                                  |21|= s(6)
    d(3) =|  5|                                  |20|= s(0)
    d(2) =|  6|                                  |19|* not used
      ud =|  7|                                  |18|= c
       l =|  8|                                  |17|= s(1)
      en =|  9|                                  |16|= s(7)
    d(1) =|10|                                   |15|= s(5)
    d(0) =|11|                                   |14|= s(3)
not used *|12|                                   |13|= clr
                 ____________________________________________
```

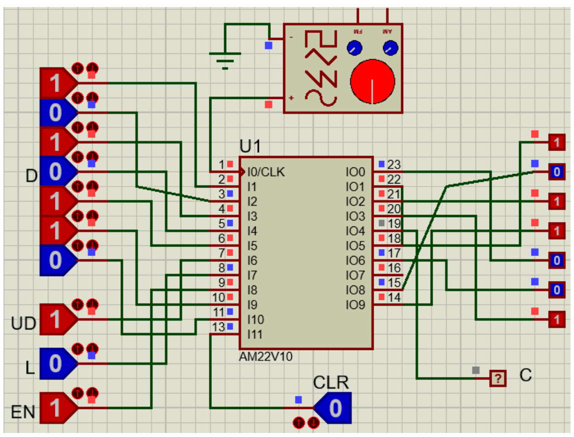
Cargar 87 (En binario: 1010111)

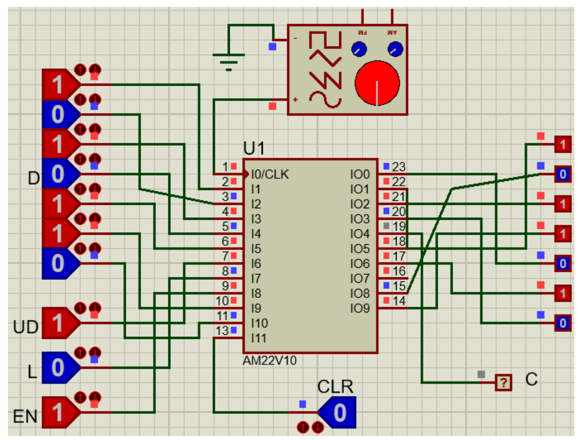


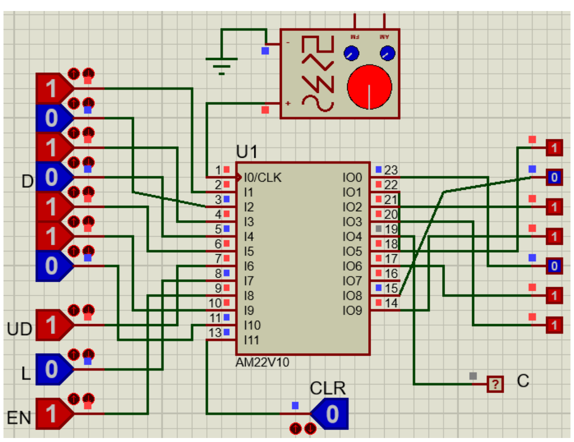
Contar hasta 95 (En binario: 1011111)

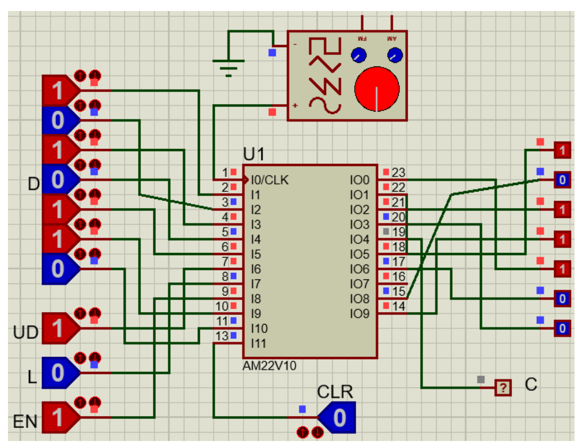


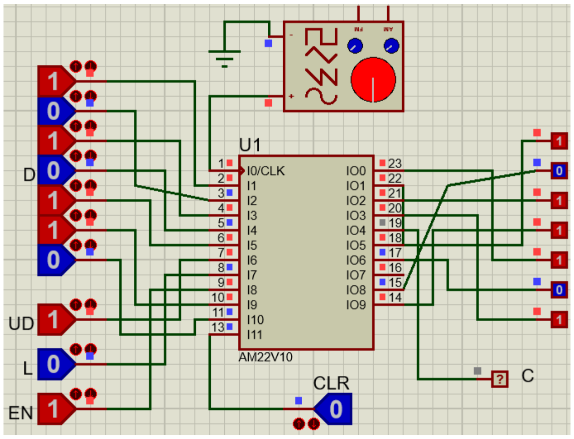


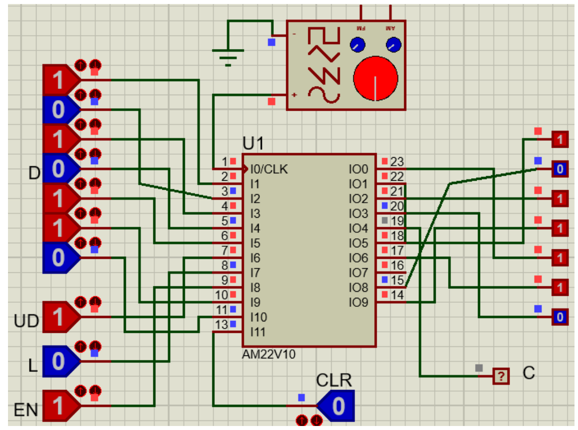


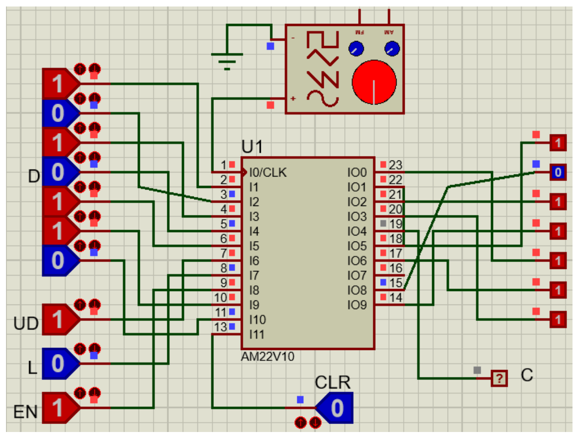




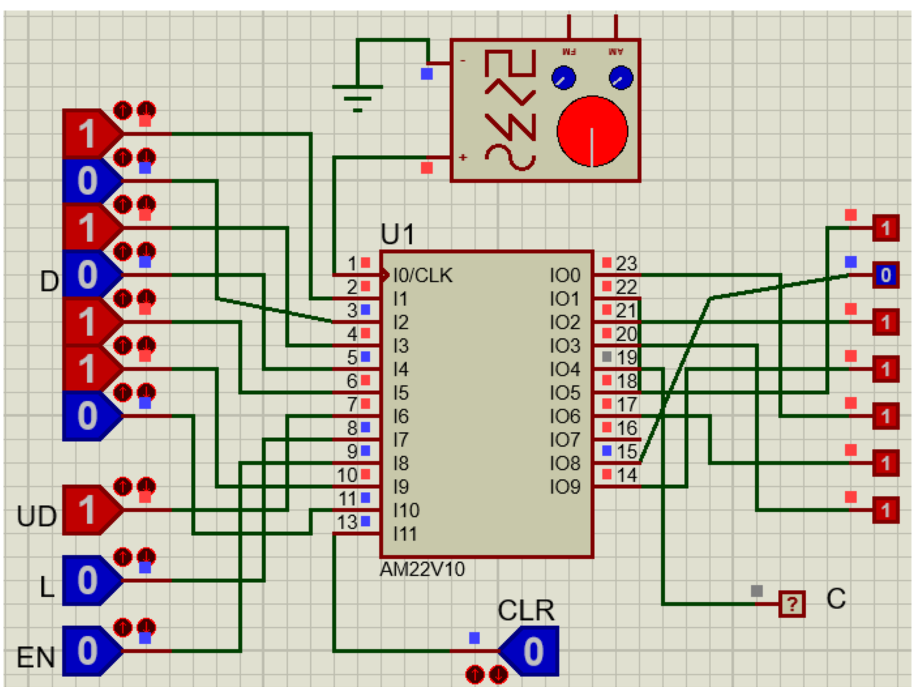




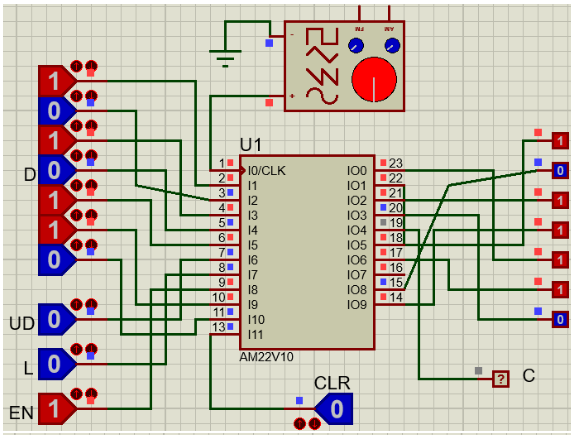


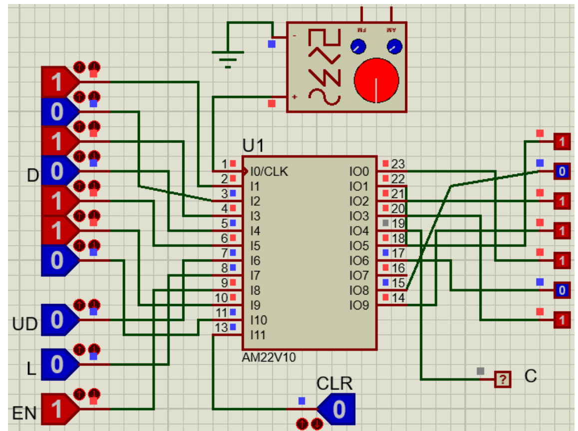


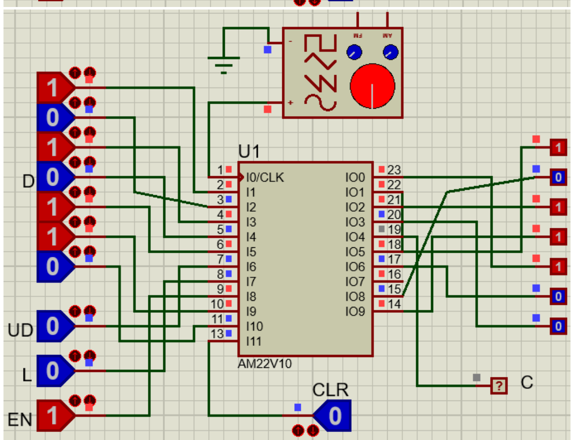
Retención en 3 ciclos (el enable está apagado)

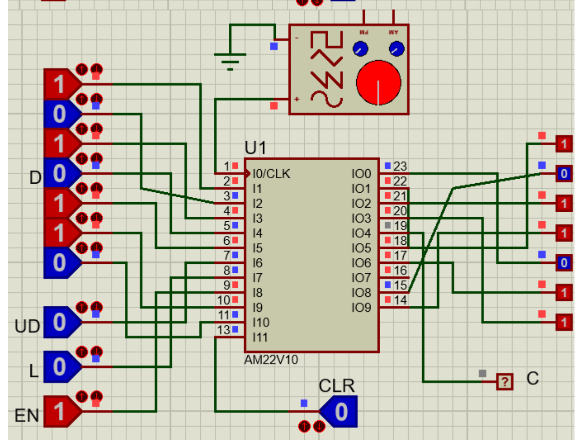


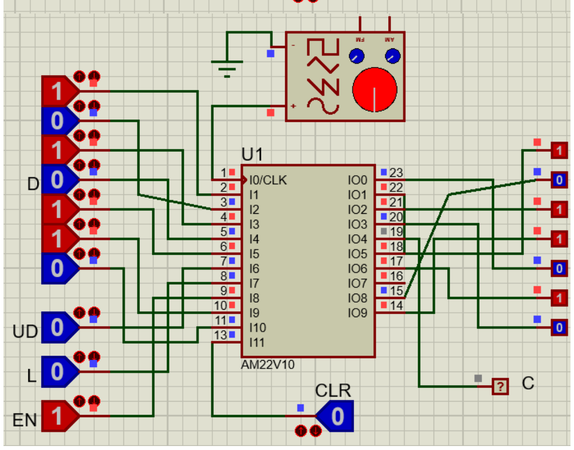
Contar hasta el 78 (En binario: 1001110)

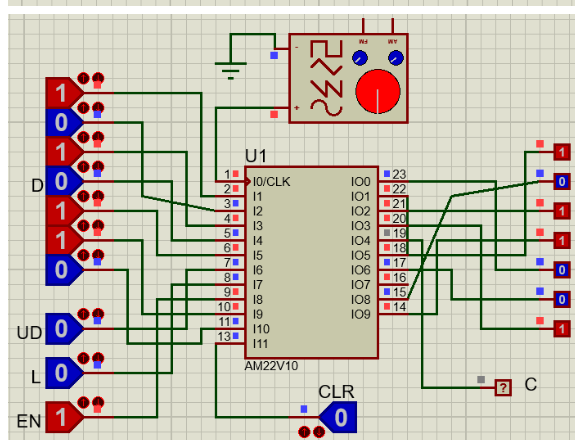


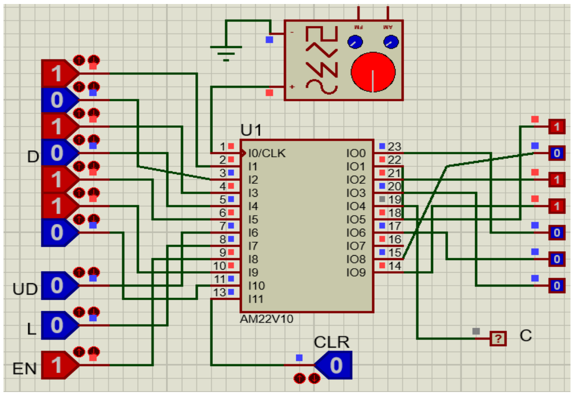


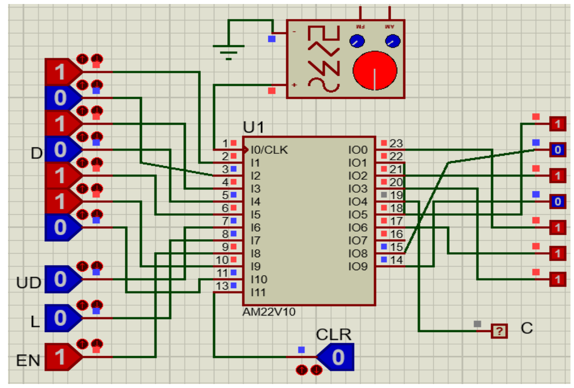


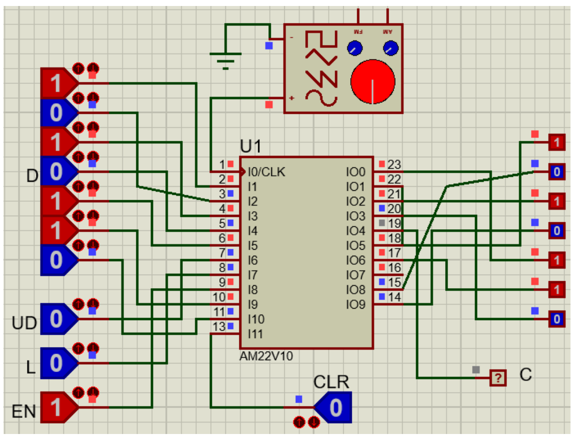


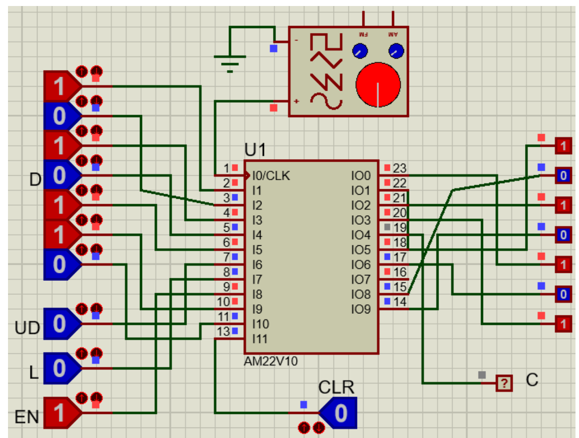


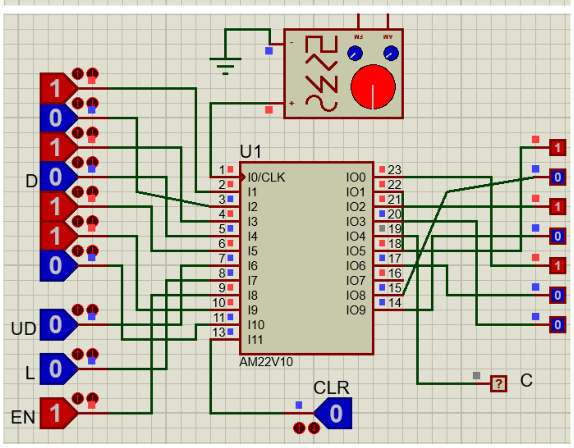


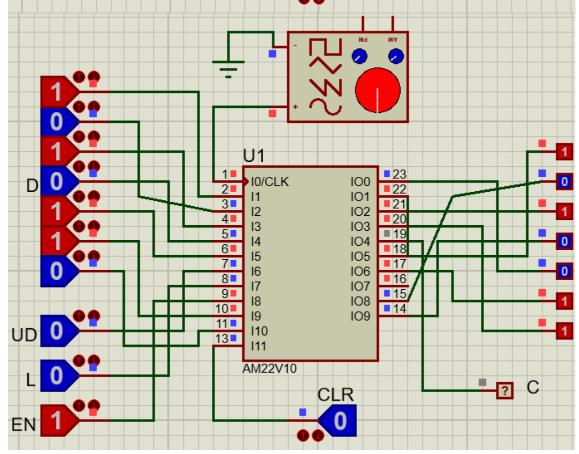

U1
AM22V10
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
D
UD
L
EN
CLR
C

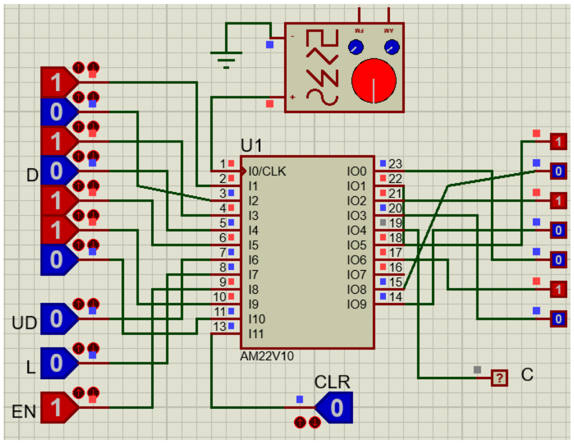

U1
AM22V10
D
UD
L
EN
CLR
C

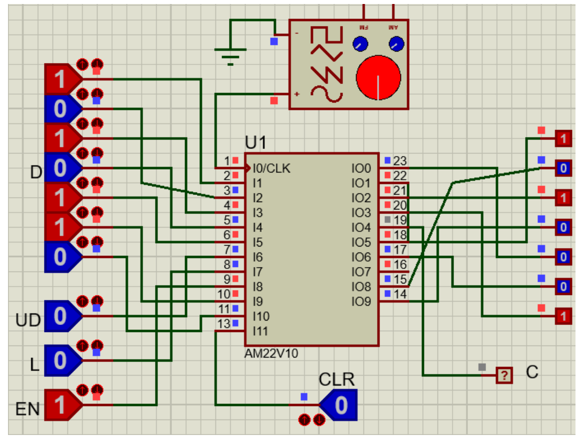

U1
AM22V10
D
UD
L
EN
CLR
C

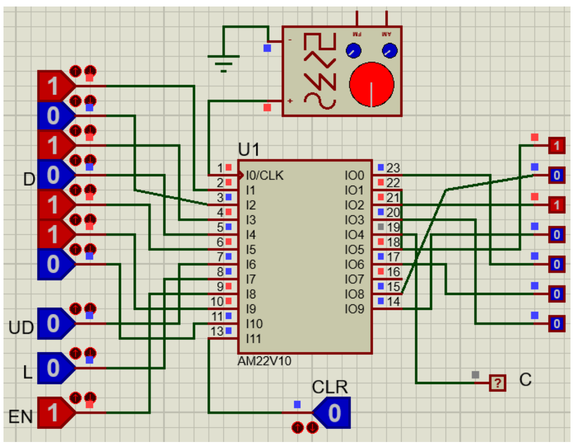

U1
AM22V10
D
UD
L
EN
CLR
C

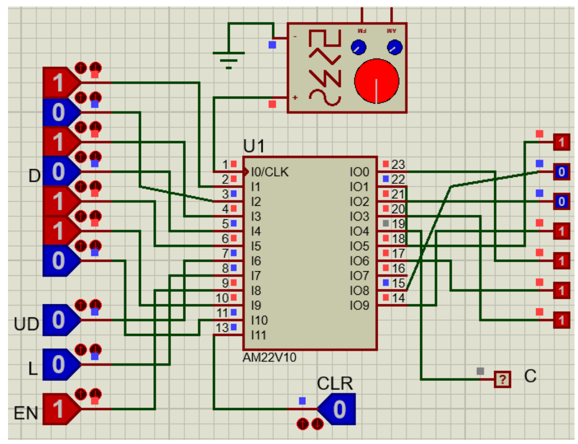

U1
AM22V10
D
UD
L
EN
CLR
C

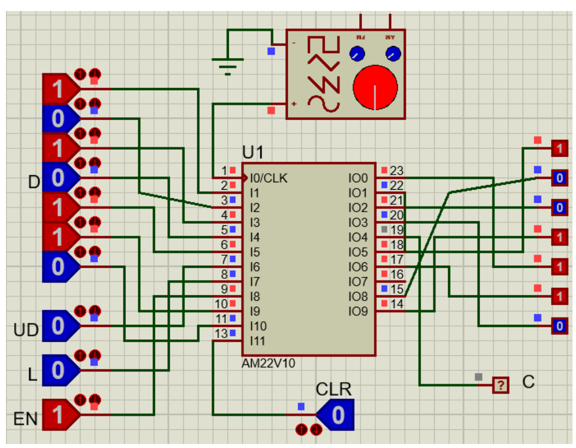

U1
AM22V10
D
UD
L
EN
CLR
C


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
D
UD
L
EN
CLR
C

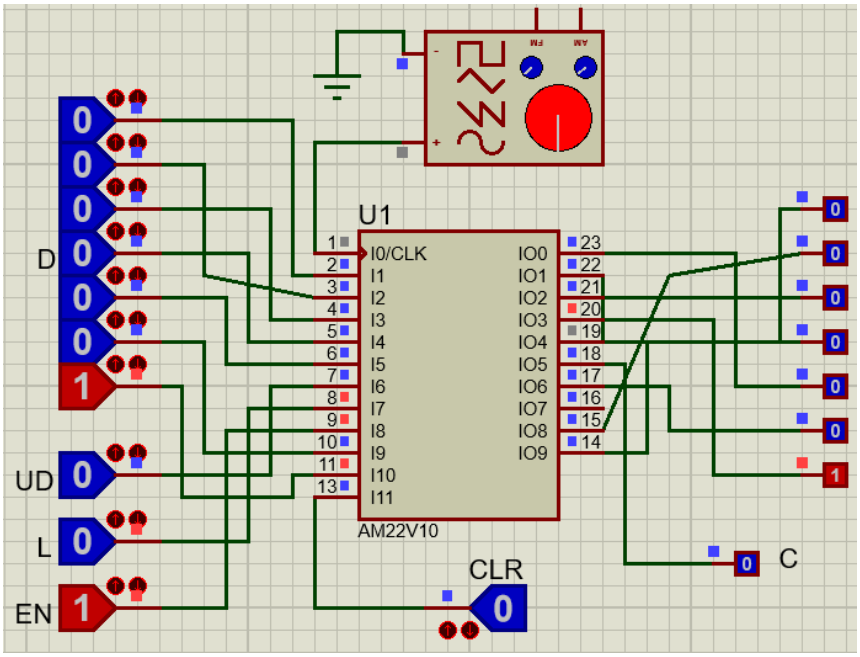

U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
D
UD
L
EN
CLR
C


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
D
UD
L
EN
CLR
C

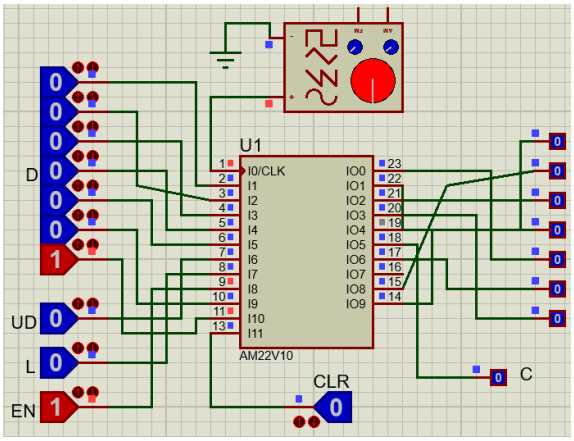

U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
D
UD
L
EN
CLR
C

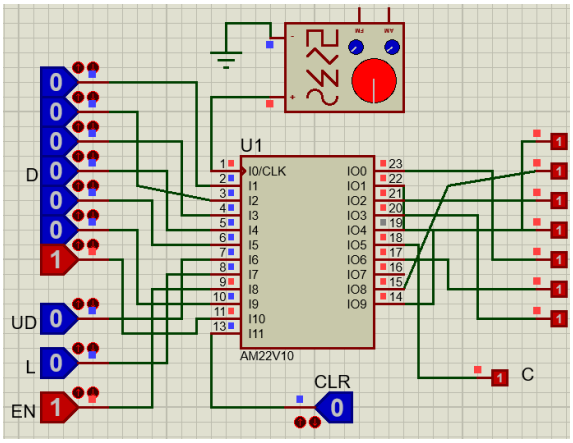

U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
D
UD
L
EN
CLR
C

Cargar 1 (En binario 0000001)

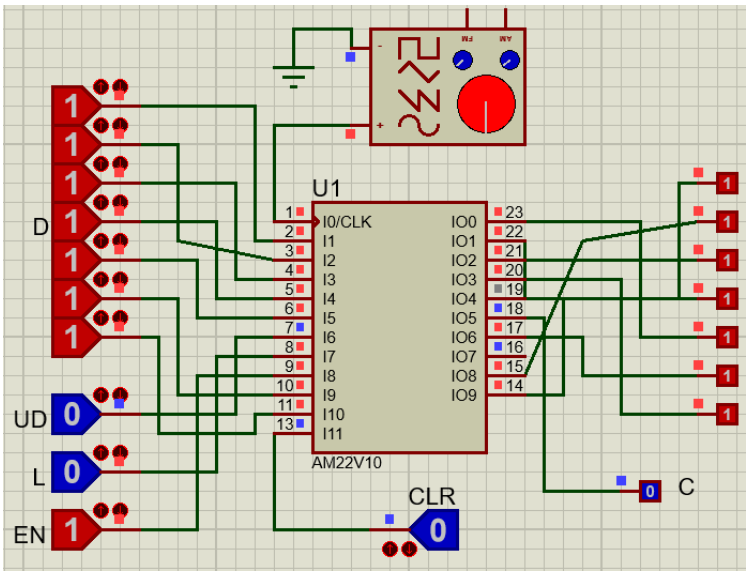


Mostrar la salida en C (Dos flancos hacia arriba)

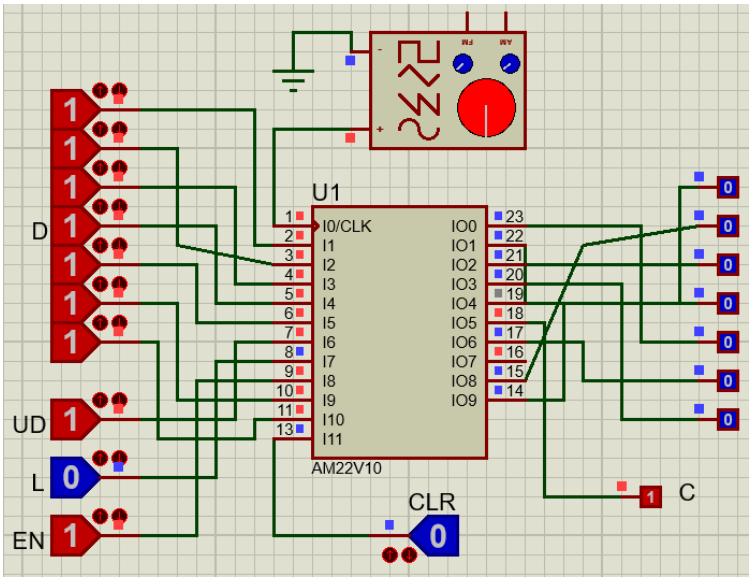




Cargar el número más grande (1111111)



Mostrar acarreo



CUESTIONARIO

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?

1 para cada circuito

2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?

Considerablemente muchos más, tan solo para el segundo circuito serían 16.

3. ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?

Para el segundo, se usaron 21 pines en total.

4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?

65 para el segundo.

5. ¿Por qué se tienen que usar variables para implementar la ecuación genérica del contador con señal de control enable?

Hace mucho más fácil el código, y mucho más fácil la abstracción.

6. ¿Qué nivel de diseño se implementó al usar los operadores + y – en el contador?

Alto.

7. ¿Cuáles son las señales que funcionan de manera síncrona y cuáles de manera asíncrona?

Enable, L, UD, D son síncronas, CLR, CLK son asíncronas.

8. ¿Qué puedes concluir de esta práctica?

Los contadores deben de ser la base de los generadores de números pseudoaleatorios en las computadoras. Cada vez que entiendo más Diseño Digital, entiendo qué hacen por dentro las computadoras cuando programamos.