

OBTENCIÓN DE ECUACIONES

Tabla inicial

Estado Inicial	Entrada	Nuevo Estado	Salida
A	0	A	0
A	1	B	0
B	0	A	0
B	1	C	0
C	0	D	0
C	1	C	0
D	0	A	0
D	1	E	1
E	0	A	0
E	1	B	0

Tabla sin redundancia

Estado Inicial	Entrada	Nuevo Estado	Salida
A	0	A	0
A	1	B	0
B	0	A	0
B	1	C	0
C	0	D	0
C	1	C	0
D	0	A	0
D	1	A	1

Código de Estados

Estado Inicial	Código
A	00
B	01
C	10
D	11

Tabla de Estados Codificados

Estado Inicial		Entrada	Nuevo Estado		Salida
Q_1	Q_0		Q_1+	Q_0+	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	1

Tabla de excitación del Flip flop D

Q	$Q+$	D
0	0	0
0	1	1
1	0	0
1	1	1

Mapas de Karnaugh

Q_1/Q_0E	00	01	11	10
0			1	
1	1	1		

Q_1/Q_0E	00	01	11	10
0		1		
1	1			

Ecuaciones

$$D_1 = (Q_1 * !Q_0) + (!Q_1 * Q_0 * E)$$

$$D_0 = (Q_1 * !Q_0 * !E) + (!Q_1 * !Q_0 * E) = !Q_0 * (Q_1 \oplus E)$$

$$S = Q_1Q_0E$$

CÓDIGO FUENTE

Máquina de Mealy con Convertidor de código

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity P4 is port(
5     D, CLR, CLK : in std_logic;
6     S : out std_logic_vector (6 downto 0)
7 );
8 end P4;
9
10 architecture AP4 of P4 is
11     signal Y, Q1, Q0, D1, D0 : std_logic;
12 begin
13     process (CLK, CLR)
14     begin
15         if (CLR = '1') then
16             Q0 <= '0';
17             Q1 <= '0';
18         elsif (rising_edge(CLK)) then
19             Q0 <= D0;
20             Q1 <= D1;
21         end if;
22     end process;
23
24     D1 <= (Q1 and not Q0) or (not Q1 and Q0 and D);
25     D0 <= not Q0 and (Q1 xor D);
26     Y <= Q1 and Q0 and D;
27
28     with Y select
29         S <= "1001111" when '0',
30             "1110111" when others;
31 end AP4;
```

SIMULACIONES EN GALAXY

SECUENCIA 1101 (VÁLIDA)

Name	Value	Stimulator	
clk	0	Clock	
clr	0	<= 0	
d	1	<= 1	
s	5F		

SECUENCIA 11101 (VÁLIDA)

Name	Value	Stimulator	
clk	0	Clock	
clr	0	<= 0	
d	1	<= 1	
s	5F		

SECUENCIA 1001 (INVÁLIDA)

Name	Value	Stimulator	
clk	0	Clock	
clr	0	<= 0	
d	1	<= 1	
s	4F		

SECUENCIA 1111 (INVÁLIDA)

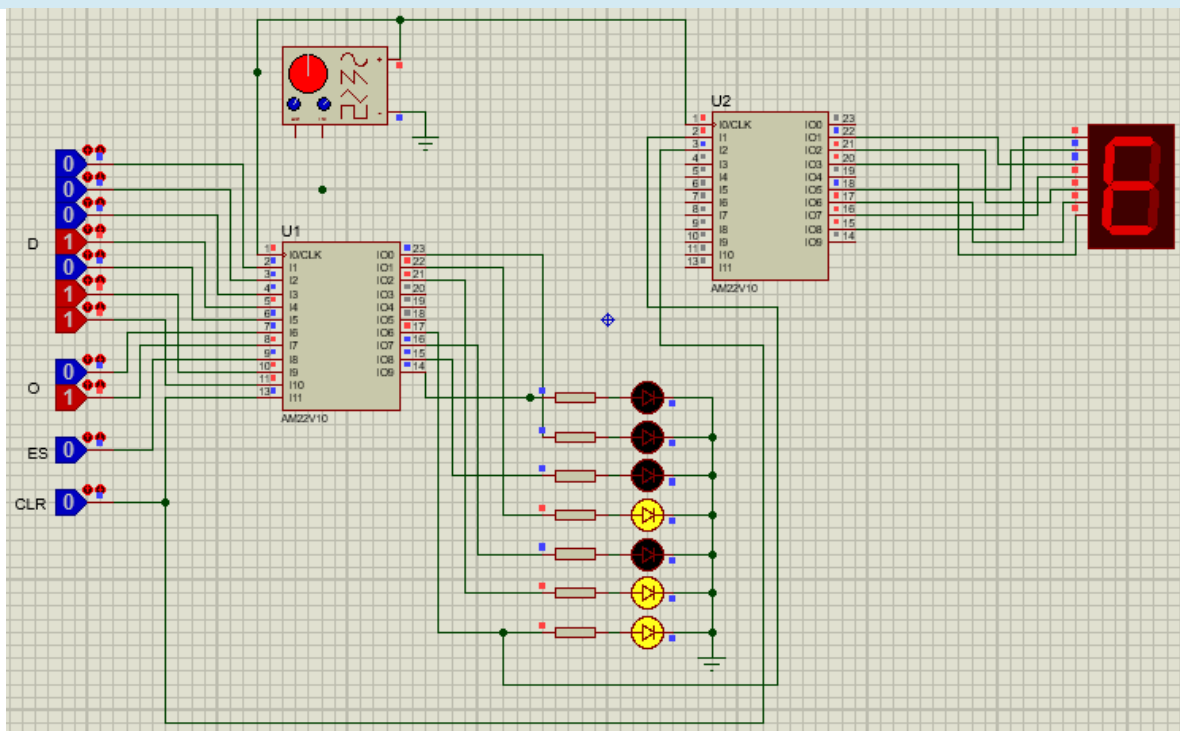
Name	Value	Stimulator	
clk	0	Clock	
clr	0	<= 0	
d	1	<= 1	
s	4F		

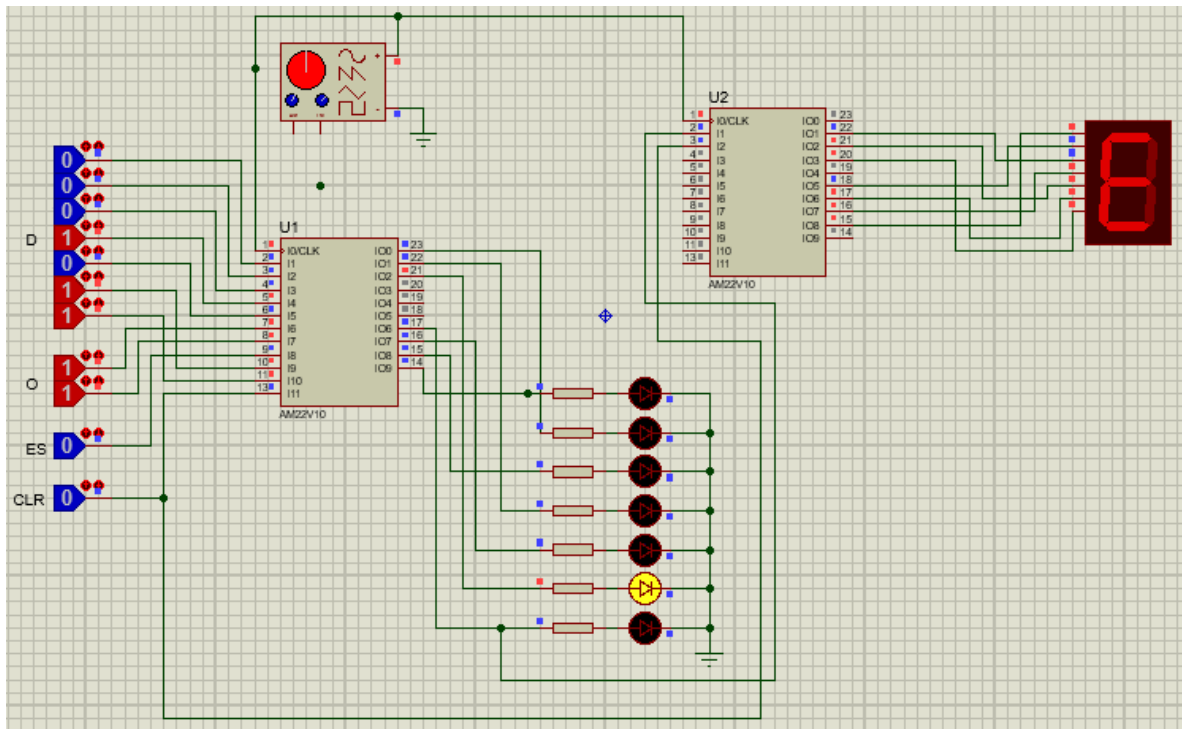
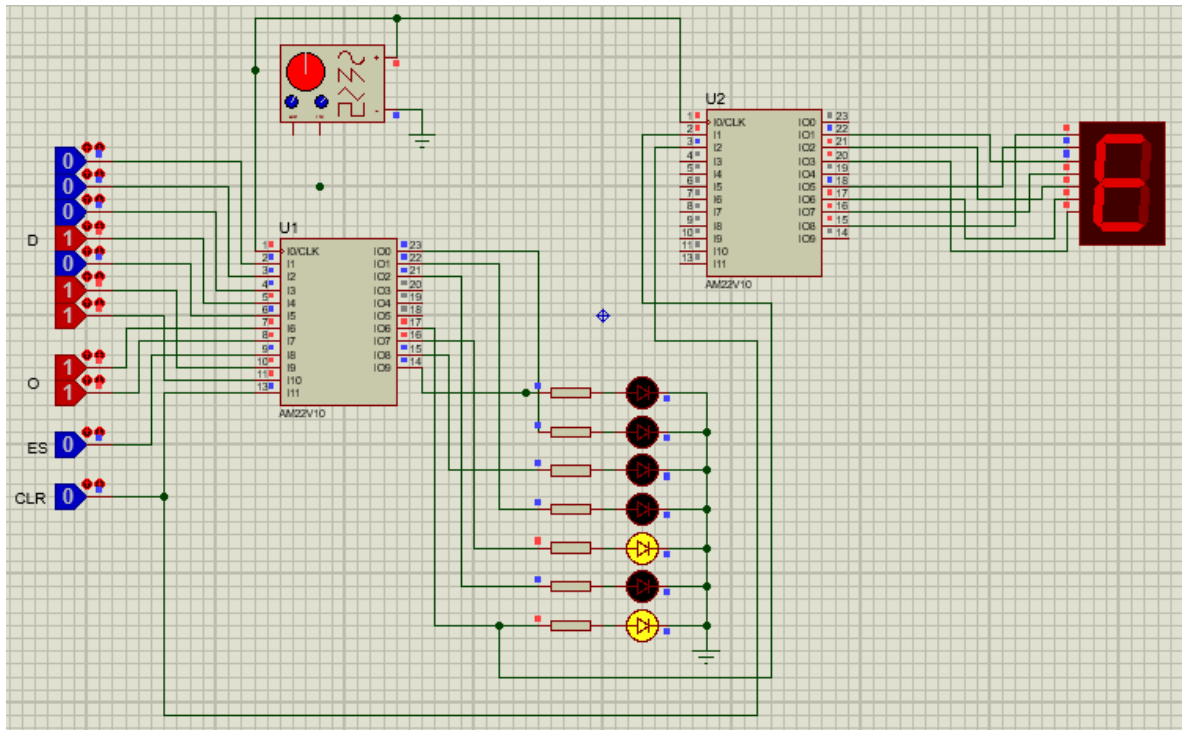
SIMULACIÓN EN PROTEUS

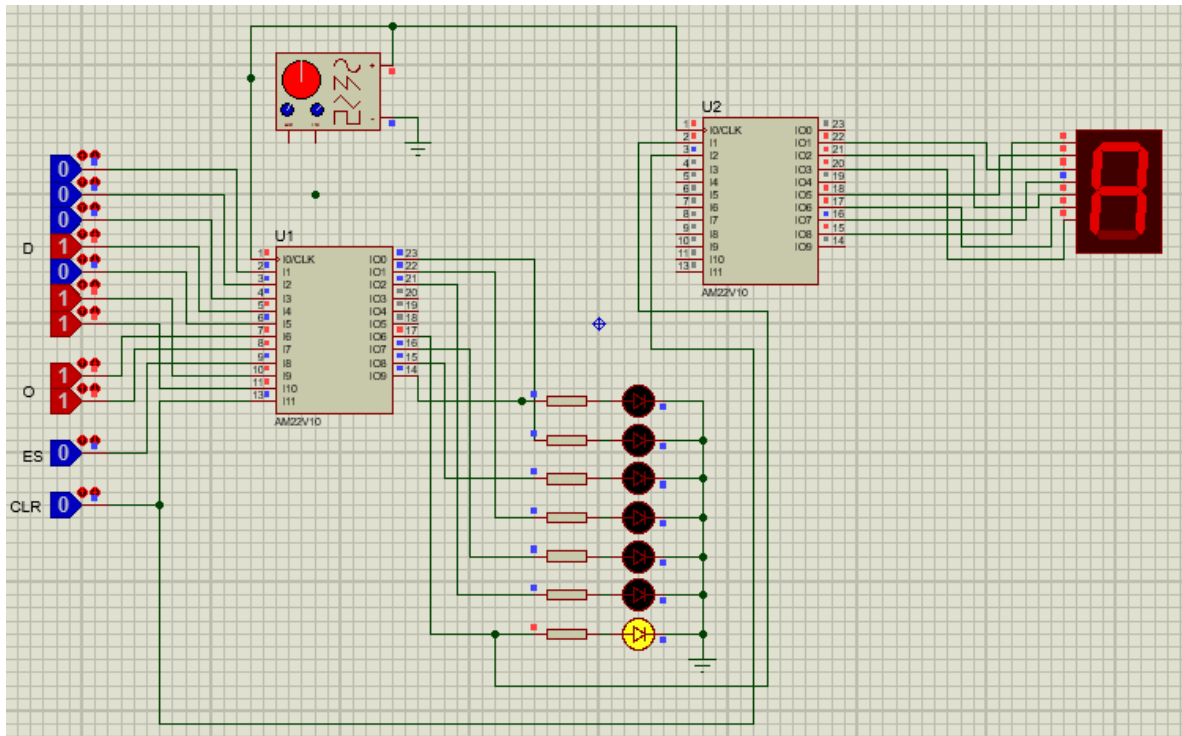
C22V10

clk = 1	24 * not used
d = 2	23 = (q0)
clr = 3	22 = s (4)
not used * 4	21 = s (2)
not used * 5	20 = s (0)
not used * 6	19 * not used
not used * 7	18 = s (5)
not used * 8	17 = s (1)
not used * 9	16 = s (3)
not used * 10	15 = s (6)
not used * 11	14 = (q1)
not used * 12	13 * not used

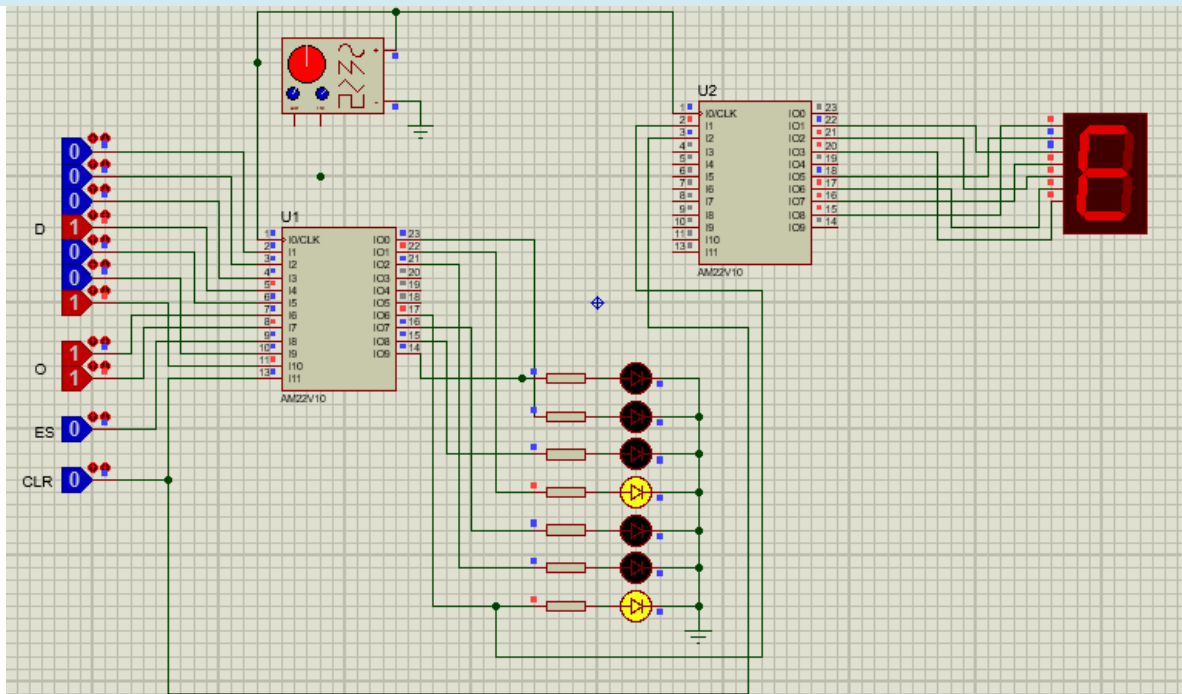
SECUENCIA 1101 (VÁLIDA)

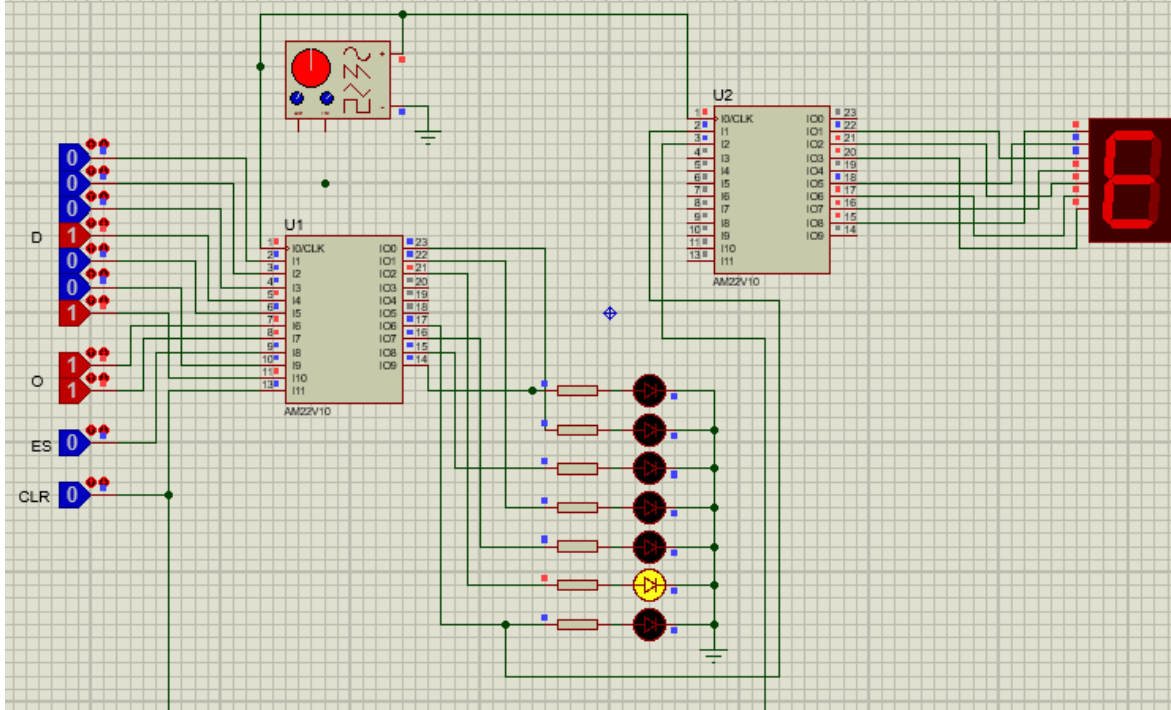


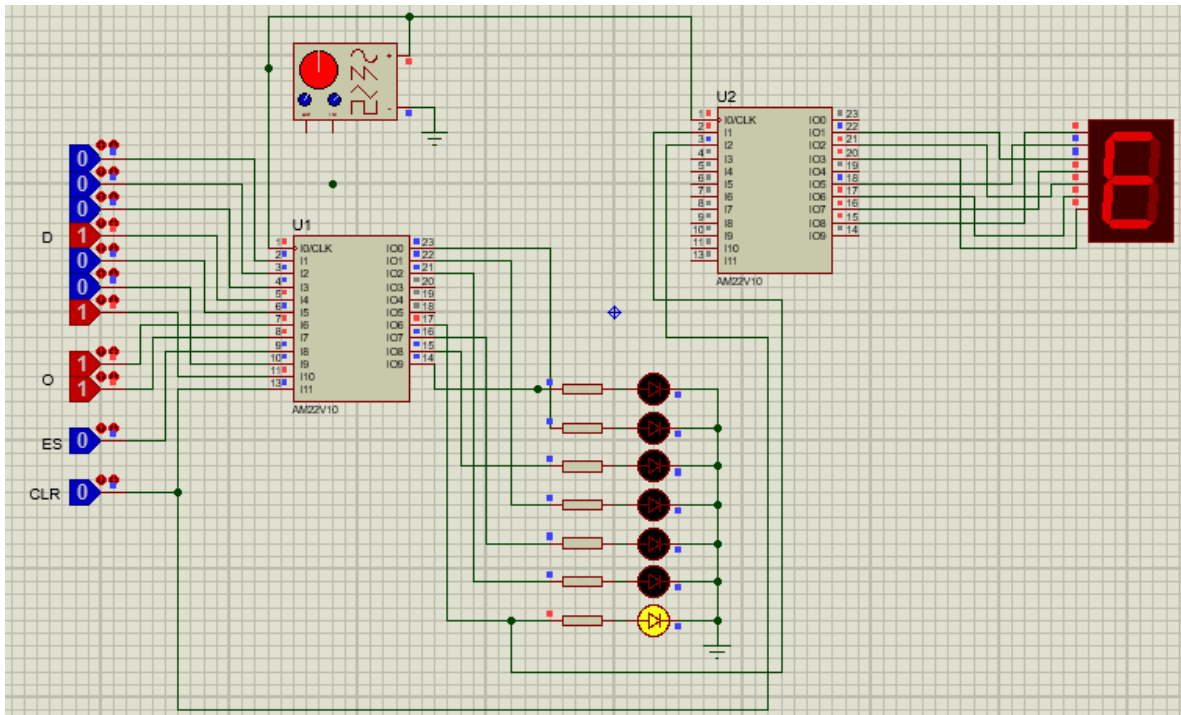




SECUENCIA 1001 (INVÁLIDA)







CUESTIONARIO

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

2

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

El doble que la práctica anterior.

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

12

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

11 de 121, es decir, 9%; de la segunda PLD.

5. ¿Cuántos FF se ocupan en el PLD para implementar la máquina Mealy del detector de secuencia?

2

6. ¿Cuántas terminales de salida se usan en PLD2?

7 explícitas y 2 implícitas.

7. ¿Qué puedes concluir de esta práctica?

Debió haber sido un gran avance, esto permite la construcción de AFD, entonces con eso se podía comprobar la célula de McCulloch-Pitts, con esto debió de comprobarse que era un avance sobre la teoría de Turing y Alonzo Church.