

CODIGO

```
library ieee;
use ieee.std_logic_1164.all;

entity contGen is port (
e : in std_logic_vector (2 downto 0);
display: out std_logic_vector(6 downto 0);
an: out std_logic_vector(2 downto 0);
clk, clr :in std_logic
);
end contGen;
--
architecture aCont of contGen is --POLI
constant l1 : std_logic_vector( 6 downto 0) := "0011000"; --P
constant l2 : std_logic_vector( 6 downto 0) := "0000001"; --O
constant l3 : std_logic_vector( 6 downto 0) := "0001110"; --L
constant l4 : std_logic_vector( 6 downto 0) := "0000110"; --I

--depende el tipo de componente
constant d0 : std_logic_vector(2 downto 0) := "110";
constant d1 : std_logic_vector(2 downto 0) := "101";
constant d2 : std_logic_vector(2 downto 0) := "011";

--posibles estados de automata
constant edo0 : std_logic_vector( 9 downto 0) := "0000000000";
constant edo1 : std_logic_vector( 9 downto 0) := d0 & l1;
constant edo2 : std_logic_vector( 9 downto 0) := d1 & l1;
constant edo3 : std_logic_vector( 9 downto 0) := d0 & l2;
constant edo4 : std_logic_vector( 9 downto 0) := d2 & l1;
constant edo5 : std_logic_vector( 9 downto 0) := d1 & l2;
constant edo6 : std_logic_vector( 9 downto 0) := d0 & l3;
constant edo7 : std_logic_vector( 9 downto 0) := d2 & l2;
constant edo8 : std_logic_vector( 9 downto 0) := d1 & l3;
constant edo9 : std_logic_vector( 9 downto 0) := d0 & l4;
constant edo10 : std_logic_vector( 9 downto 0) := d2 & l3;
constant edo11 : std_logic_vector( 9 downto 0) := d1 & l4;
constant edo12 : std_logic_vector( 9 downto 0) := d2 & l4;

signal estado : std_logic_vector(9 downto 0);

begin

process(clk, clr)
begin
    if (clr = '1') then
```

```

    estado <= edo0;
elsif (rising_edge(clk)) then
    case estado is

        when edo0 =>
            if (e = "000" or e = "001") then
                estado <= edo0;
            elsif (e = "010") then
                estado <= edo1;
            else
                estado <= edo0;
            end if;

        when edo1 =>
            if(e = "010") then
                estado <= edo1;
            elsif(e = "011") then
                estado <= edo2;
            else
                estado <= edo0;
            end if;

        when edo2 =>
            if(e = "100") then
                estado <= edo4;
            elsif(e = "011") then
                estado <= edo3;
            else
                estado <= edo0;
            end if;

        when edo3 =>
            if(e = "100") then
                estado <= edo4;
            elsif(e = "011") then
                estado <= edo2;
            else
                estado <= edo0;
            end if;

        when edo4 =>
            if(e = "100") then
                estado <= edo5;
            elsif(e = "101") then
                estado <= edo7;
            end if;
    end case;
end if;

```

```

else
    estado <= edo0;
end if;

when edo5 =>
    if(e = "100") then
        estado <= edo6;
    elsif(e = "101") then
        estado <= edo7;
    else
        estado <= edo0;
    end if;

when edo6 =>
    if(e = "101") then
        estado <= edo7;
    elsif(e = "100") then
        estado <= edo4;
    else
        estado <= edo0;
    end if;

when edo7 =>
    if(e = "101") then
        estado <= edo8;
    elsif(e = "110") then
        estado <= edo10;
    else
        estado <= edo0;
    end if;

when edo8 =>
    if(e = "101") then
        estado <= edo9;
    elsif(e = "110") then
        estado <= edo10;
    else
        estado <= edo0;
    end if;

when edo9 =>
    if(e = "110") then
        estado <= edo10;
    elsif(e = "101") then
        estado <= edo7;

```

```



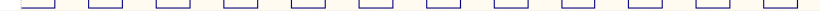
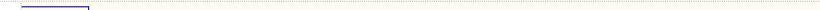
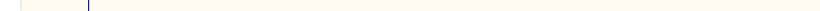
        else
            estado <= edo0;
        end if;

when edo10 =>
    if(e = "110") then
        estado <= edo11;
    elsif(e = "111") then
        estado <= edo12;
    else
        estado <= edo0;
    end if;

when edo11 =>
    if(e = "111") then
        estado <= edo12;
    elsif(e = "110") then
        estado <= edo10;
    else
        estado <= edo0;
    end if;

when edo12 =>
    if(e = "000") then
        estado <= edo0;
    elsif(e = "111") then
        estado <= edo12;
    else
        estado <= edo0;
    end if;
when others =>
    estado <= edo0;
end case;
end if;
end process;
display <= estado(6 downto 0);
an <= estado (9 downto 7);
end architecture;
```

### SIMULACION (arreglado)

Name	Value	Stimulator	
▷ clk	0	Clock	
▷ clr	0	<= 0	
⊕ ▷ e	0	<= "000"	
⊕ ▷ display	00		
⊕ ▷ an	0		

## CUESTIONARIO

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

 $R = 2$ 

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

R = 20-30

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

R = 3 de entrada y 9 de salida + 3 de salida del contador

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

$$R = 72 / 121 = 59\%$$

5. ¿Qué codificación se usa con la directiva TYPE?

R = Cuando el sintetizador realizará la asignación de código y obtendrá las ecuaciones del diseño.

6. ¿Cuál codificación es la que finalmente se pudo sintetizar?

R = El definido por el usuario

7. ¿Qué puedes concluir de esta práctica?

R = Ya que la practica ocupaba muchos términos fue necesario optimizar algunos pasos, en este caso el del estado 1 se le asigno el valor de 0 para que funcionara con normalidad.