library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity prac10 is port

(

señal: in std\_logic\_vector(1 downto 0);

clr,clk: in std\_logic;

uni: inout std\_logic\_vector(3 downto 0);

dec: inout std\_logic\_vector(2 downto 0)

);

end entity;

architecture p10 of prac10 is

type estados is (q0,q1,q2,q3,q4,q5,q6,q7);

signal act, sig: estados;

signal salida: std\_logic\_vector(1 downto 0);

begin

process(clk,clr)

begin

if(clr='1')then

act<=q0;

elsif(rising\_edge(clk))then

act<=sig;

end if;

end process;

process(act,señal)

begin

case act is

when q0=>

if(señal ="00")then

salida <="00";

sig<=q0;

elsif(señal ="10")then

salida <="00";

sig<=q1;

elsif(señal ="11")then

salida <="00";

sig<=q7;

else

salida <="00";

sig<=q4;

end if;

when q1=>

case señal is

when"10"=>

salida <="00";

sig<=q1;

when"11"=>

salida <="00";

sig<=q2;

when"01"=>

salida <="00";

sig<=q4;

when others=>

salida <="00";

sig<=q0;

end case;

when q2=>

case señal is

when"11"=>

salida <="00";

sig<=q2;

when"01"=>

salida <="00";

sig<=q3;

when"10"=>

salida <="00";

sig<=q1;

when others=>

salida <="00";

sig<=q0;

end case;

when q3=>

case señal is

when"10"=>

salida <="01";

sig<=q1;

when"01"=>

salida <="00";

sig<=q3;

when"11"=>

salida <="00";

sig<=q2;

when others=>

salida <="01";

sig<=q0;

end case;

when q4=>

case señal is

when"10"=>

salida <="00";

sig<=q1;

when"11"=>

salida <="00";

sig<=q5;

when"01"=>

salida <="00";

sig<=q4;

when others=>

salida <="00";

sig<=q0;

end case;

when q5=>

case señal is

when"10"=>

salida <="00";

sig<=q6;

when"11"=>

salida <="00";

sig<=q5;

when"01"=>

salida <="00";

sig<=q4;

when others=>

salida <="00";

sig<=q0;

end case;

when q6=>

case señal is

when"10"=>

salida <="00";

sig<=q6;

when"01"=>

salida <="10";

sig<=q4;

when"11"=>

salida <="00";

sig<=q5;

when others=>

salida <="10";

sig<=q0;

end case;

when q7=>

case señal is

when"10"=>

salida <="00";

sig<=q1;

when"11"=>

salida <="00";

sig<=q7;

when"01"=>

salida <="00";

sig<=q4;

when others=>

salida <="00";

sig<=q0;

end case;

when others=>

salida <="00";

sig<=q0;

end case;

end process;

process(clk,clr)

begin

if(clr='1')then

uni<="0000";

dec<="000";

elsif(rising\_edge(clk))then

if(salida ="00")then

uni<=uni;

dec<=dec;

elsif(salida ="01")then

if(uni="1001")then

uni<="0000";

dec<=dec+1;

else

uni<=uni+1;

end if;

elsif(salida ="10")then

if(uni="0000")then

uni<="1001";

dec<=dec-1;

else

uni<=uni-1;

end if;

end if;

end if;

end process;

end architecture;

library ieee;

use ieee.std\_logic\_1164.all;

entity prac102 is port

(

uni: in std\_logic\_vector(3 downto 0);

dec: in std\_logic\_vector(2 downto 0);

clk,clr: in std\_logic;

display: out std\_logic\_vector(6 downto 0);

cat: inout std\_logic\_vector(2 downto 0)

);

end entity;

architecture aprac102 of prac102 is

signal edo: std\_logic;

begin

process(clk,clr)

begin

if(clr='1')then

cat<="011";

elsif(rising\_edge(clk))then

cat<=TO\_STDLOGICVECTOR(TO\_BITVECTOR(cat)ror 1);

end if;

end process;

--MULTIPLEXOR--

process(cat)

begin

if(cat="110")then

edo<='0';

elsif(cat="101")then

edo<='1';

else

edo<='-';

end if;

end process;

process(edo,clr)

begin

if(clr='1')then

display<="0000000";

elsif(edo='0')then

case uni is

when"0000"=>display<="1111110";

when"0001"=>display<="0110000";

when"0010"=>display<="1101101";

when"0011"=>display<="1111001";

when"0100"=>display<="0110011";

when"0101"=>display<="1011011";

when"0110"=>display<="1011111";

when"0111"=>display<="1110000";

when"1000"=>display<="1111111";

when"1001"=>display<="1111011";

when others=>display<="0000000";

end case;

elsif(edo='1')then

case dec is

when"000"=>display<="1111110";

when"001"=>display<="0110000";

when"010"=>display<="1101101";

when"011"=>display<="1111001";

when"100"=>display<="0110011";

when"101"=>display<="1011011";

when"110"=>display<="1011111";

when"111"=>display<="1110000";

when others=>display<="0000000";

end case;

end if;

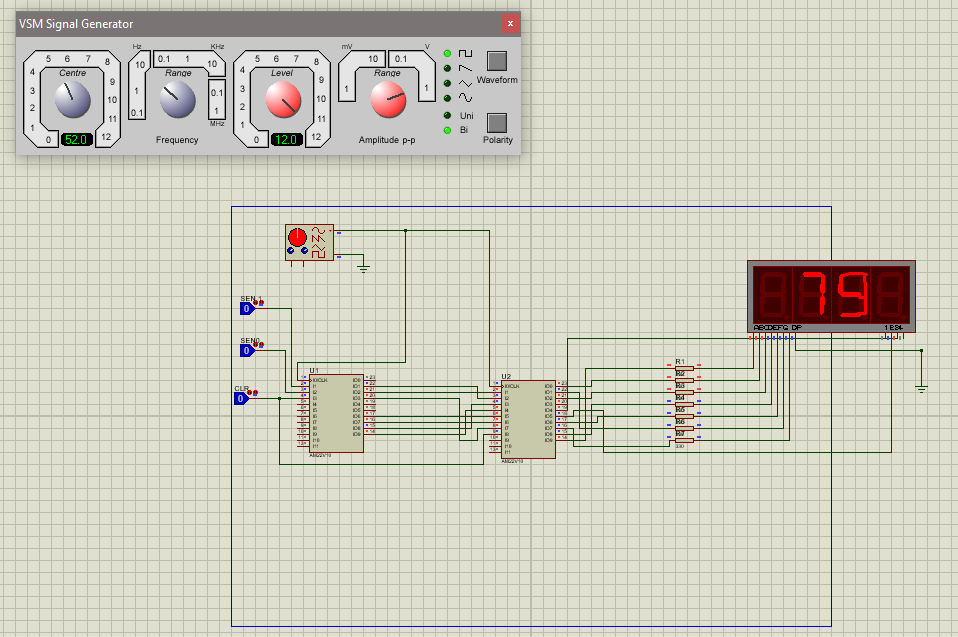
end process;

end architecture;

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated



CUESTIONARIO

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

esta práctica?

**Se utilizaron 2 dispositivos.**

2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras

necesitado para el desarrollo de esta práctica?

**Más de 100**

3. ¿Cuántos pines de entrada/salida del PLD1 22V10 y PLD2 22V10 se

usan en el diseño?

**En el PLD1 se utilizan 3 entradas dedicadas, 1 entrada de reloj y las 10 macroceldas de input/output.**

**En el PLD2 se utilizan 8 entradas dedicadas, 1 de reloj y las 10 macroceldas de input/output.**

4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD1 22V10 y PLD2 22V10?

**En el PLD1 se utilizan 88 términos producto y se usa un 72% del total.**

**En el PLD2 se utilizan 49 términos producto y se usa un 40% del total.**

5. ¿Qué puedes concluir de esta práctica?

**Fue una práctica interesante ya que en el proceso que diseñar el autómata en VHDL utilizamos una nueva herramienta que nos permite describir cualquier autómata ya sea de MEALY o de MOORE, por lo que es aun otra opción que podemos utilizar si queremos describir un circuito secuencial, se escribe un poco más de código pero en mi opinión se mantiene más claro el propósito de cada componente y estado.**