library ieee;

use ieee.std\_logic\_1164.all;

entity Automata is port(

clr, clk : in std\_logic;

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

estado : out std\_logic\_vector(9 downto 0)

);

end Automata;

--AUTOMATA DE MARQUESINA

architecture aAutomata of Automata is

--DISPLAYS

CONSTANT ND : std\_logic\_vector(2 downto 0) := "000";

CONSTANT D0 : std\_logic\_vector(2 downto 0) := "001";

CONSTANT D1 : std\_logic\_vector(2 downto 0) := "010";

CONSTANT D2 : std\_logic\_vector(2 downto 0):= "100";

--LETRAS DEL MENSAJE

CONSTANT NL : std\_logic\_vector(6 downto 0) := "1111111";

CONSTANT L1 : std\_logic\_vector(6 downto 0) := "0110111";

CONSTANT L2 : std\_logic\_vector(6 downto 0) := "1111110";

CONSTANT L3 : std\_logic\_vector(6 downto 0) := "0001110";

CONSTANT L4 : std\_logic\_vector(6 downto 0) := "1110111";

--ESTADOS DEL AUTÓMATA

CONSTANT Q0 : std\_logic\_vector(9 downto 0) := ND&NL;

CONSTANT Q1 : std\_logic\_vector(9 downto 0) := D0&L1;

CONSTANT Q2 : std\_logic\_vector(9 downto 0) := D1&L1;

CONSTANT Q3 : std\_logic\_vector(9 downto 0) := D0&L2;

CONSTANT Q4 : std\_logic\_vector(9 downto 0) := D2&L1;

CONSTANT Q5 : std\_logic\_vector(9 downto 0) := D1&L2;

CONSTANT Q6 : std\_logic\_vector(9 downto 0) := D0&L3;

CONSTANT Q7 : std\_logic\_vector(9 downto 0) := D2&L2;

CONSTANT Q8 : std\_logic\_vector(9 downto 0) := D1&L3;

CONSTANT Q9 : std\_logic\_vector(9 downto 0) := D0&L4;

CONSTANT Q10 : std\_logic\_vector(9 downto 0) := D2&L3;

CONSTANT Q11 : std\_logic\_vector(9 downto 0) := D1&L4;

CONSTANT Q12 : std\_logic\_vector(9 downto 0) := D2&L4;

signal aux : std\_logic\_vector(9 downto 0) := "0001111111";

begin

process(clk, clr)

begin

if (clr = '1') then

aux <= Q0;

elsif(rising\_edge(clk)) then

case aux is

when "0001111111" =>

if(e = "000" or e = "001") then

aux <= Q0;

elsif( e = "010") then

aux <= Q1;

else

aux <= "----------";

end if;

when "0010110111" =>

if(e = "010") then

aux <= Q1;

elsif( e = "011") then

aux <= Q2;

else

aux <= "----------";

end if;

when "0100110111" =>

if(e = "011") then

aux <= Q3;

elsif( e = "100") then

aux <= Q4;

else

aux <= "----------";

end if;

when "0011111110" =>

if(e = "011") then

aux <= Q2;

elsif( e = "100") then

aux <= Q4;

else

aux <= "----------";

end if;

when "1000110111" =>

if(e = "100") then

aux <= Q5;

elsif( e = "101") then

aux <= Q7;

else

aux <= "----------";

end if;

when "0101111110" =>

if(e = "100") then

aux <= Q6;

elsif( e = "101") then

aux <= Q7;

else

aux <= "----------";

end if;

when "0010001110" =>

if(e = "100") then

aux <= Q4;

elsif( e = "101") then

aux <= Q7;

else

aux <= "----------";

end if;

when "1001111110" =>

if(e = "101") then

aux <= Q8;

elsif( e = "110") then

aux <= Q10;

else

aux <= "----------";

end if;

when "0100001110" =>

if(e = "101") then

aux <= Q9;

elsif( e = "110") then

aux <= Q10;

else

aux <= "----------";

end if;

when "0011110111" =>

if(e = "101") then

aux <= Q7;

elsif( e = "110") then

aux <= Q10;

else

aux <= "----------";

end if;

when "1000001110" =>

if(e = "110") then

aux <= Q11;

elsif( e = "111") then

aux <= Q12;

else

aux <= "----------";

end if;

when "0101110111" =>

if(e = "110") then

aux <= Q10;

elsif( e = "111") then

aux <= Q12;

else

aux <= "----------";

end if;

when "1001110111" =>

if(e = "111") then

aux <= Q12;

elsif( e = "000") then

aux <= Q0;

else

aux <= "----------";

end if;

when others => aux <= "----------";

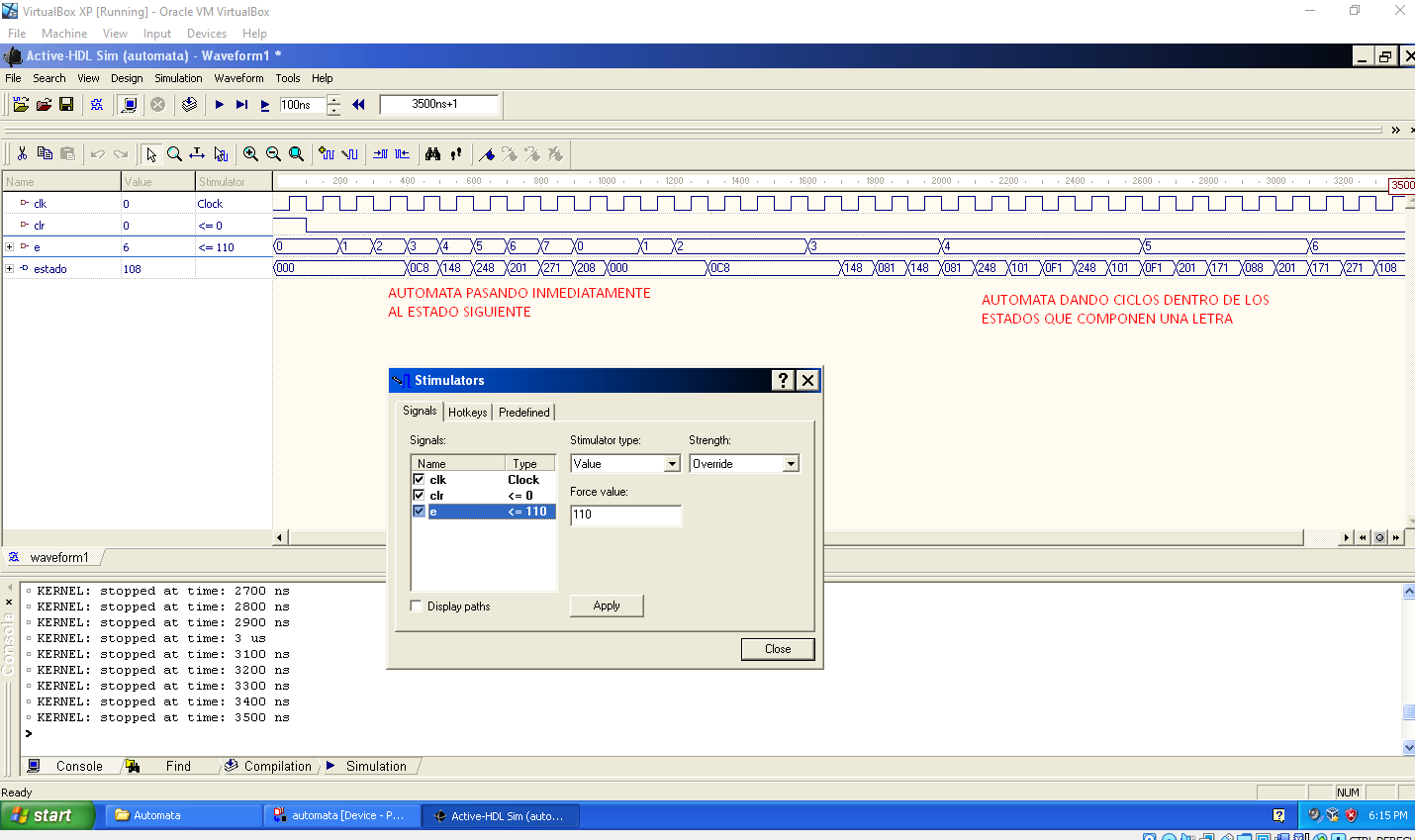
end case;

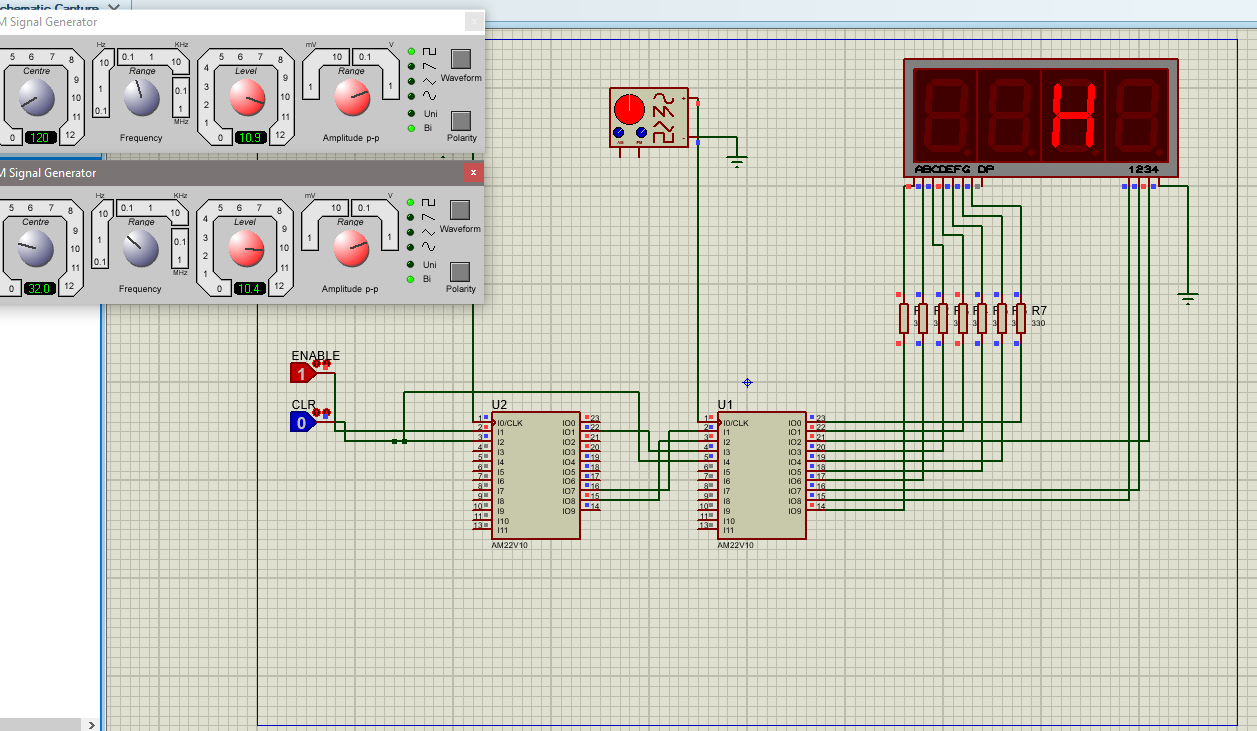
end if;

estado <= aux; --los primeros 3 bits pertenecen al selector de display, el resto a la salida de display.

end process;

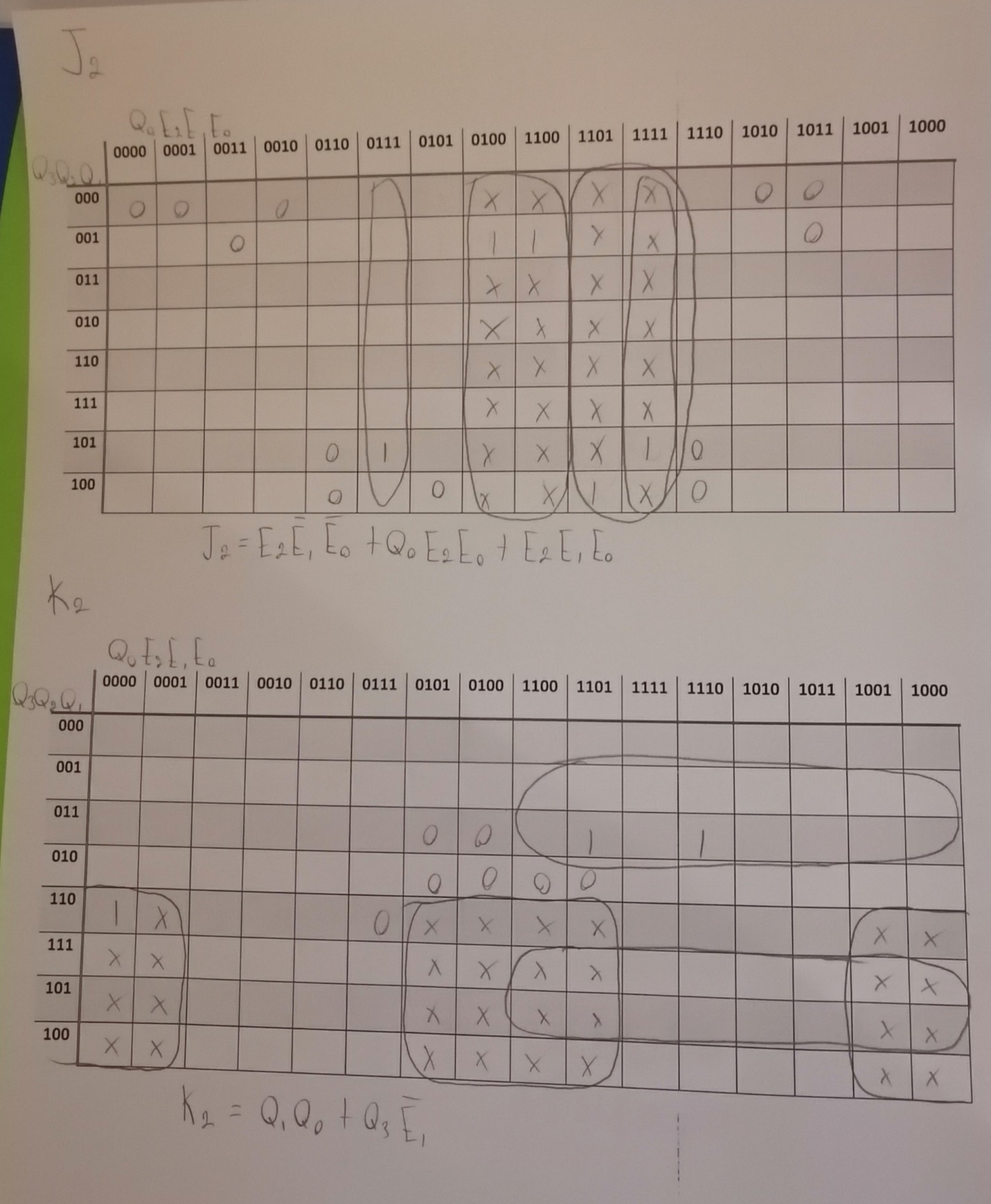
end architecture;





**A picture containing calendar

Description automatically generatedA picture containing calendar

Description automatically generatedDiagram, schematic

Description automatically generatedA picture containing diagram

Description automatically generated**

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

**esta práctica?**

2, uno para el contador y otro para el autómata de marquesina

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

**necesitado para el desarrollo de esta práctica?**

Aproximadamente 80 tomando en cuenta los AND y OR de las ecuaciones producidas por el RPT

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

Para el autómata se utilizan los 10 pines de entrada/salida de la GAL, para el contador también se utilizan los 10 pines de entrada/salida

**4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de**

**salida y que porcentaje se usa en total del PLD 22V10?**

Para el autómata se utilizan 52 términos producto, siendo esto el 42% de la GAL, para el contador se utilizan 65, siendo esto el 53%.

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

---------------------------------------------

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

----------------------------------------------

**7. ¿Qué puedes concluir de esta práctica?**

Fue una práctica muy compleja en el análisis a mano, ya que requería de 4 flip flops JK para representar Q, y aparte necesitaba 3 bits de entrada del contador, por lo que nos quedaron mapas de Karnaugh de 7 variables, los cuales son difíciles de resolver a mano, referente al diseño, nos dimos cuenta que utilizando las sentencias de VHDL podemos diseñar circuitos un tanto complejos sin necesidad de utilizar las ecuaciones de display, o de flip-flops, simplemente utilizando un poco de ingenio con las herramientas que VHDL nos ofrece, así, logramos crear una marquesina que muestra el mensaje “HOLA” recorriendo de derecha a izquierda utilizando un simple proceso, un “case” y siguiendo las instrucciones del autómata