

```

library ieee;

use ieee.std_logic_1164.all;

entity Practica9 is port(
    clk, clr : in std_logic;
    filas : in std_logic_vector(3 downto 0); --3 a 0 son filas.
    anillo : inout std_logic_vector(2 downto 0);
    display : out std_logic_vector(6 downto 0)
);
end Practica9;

```

architecture aPractica9 of Practica9 is

```

signal L : std_logic;
signal regAux : std_logic_vector(6 downto 0);
--signal anillo : std_logic_vector(2 downto 0);
signal matrix : std_logic_vector(6 downto 0);

```

```

CONSTANT N1 : std_logic_vector(6 downto 0) := "1111001";
CONSTANT N2 : std_logic_vector(6 downto 0) := "0010010";
CONSTANT N3 : std_logic_vector(6 downto 0) := "0000110";
CONSTANT N4 : std_logic_vector(6 downto 0) := "1001100";
CONSTANT N5 : std_logic_vector(6 downto 0) := "0100100";
CONSTANT N6 : std_logic_vector(6 downto 0) := "0100000";
CONSTANT N7 : std_logic_vector(6 downto 0) := "0001111";
CONSTANT N8 : std_logic_vector(6 downto 0) := "0000000";
CONSTANT N9 : std_logic_vector(6 downto 0) := "0000100";
CONSTANT N0 : std_logic_vector(6 downto 0) := "0000001";

```

```
CONSTANT AS : std_logic_vector(6 downto 0) := "1001000";  
CONSTANT NG : std_logic_vector(6 downto 0) := "1000001";  
begin
```

```
--CODIFICADOR
```

```
matrix <= filas&anillo;
```

```
--REGISTRO
```

```
process(clk, clr)
```

```
begin
```

```
    if(clr = '1') then
```

```
        regAux <= "1111111";
```

```
    elsif(rising_edge(clk)) then
```

```
        if(L = '0') then --RETENCION
```

```
            regAux <= regAux;
```

```
        elsif(L = '1') then
```

```
            case matrix is
```

```
                when "1110011" =>
```

```
                    regAux <= N1;
```

```
                when "1110101" =>
```

```
                    regAux <= N2;
```

```
                when "1110110" =>
```

```
                    regAux <= N3;
```

```
                when "1101011" =>
```

```
                    regAux <= N4;
```

```
                when "1101101" =>
```

```
                    regAux <= N5;
```

```
                when "1101110" =>
```

```
                    regAux <= N6;
```

```

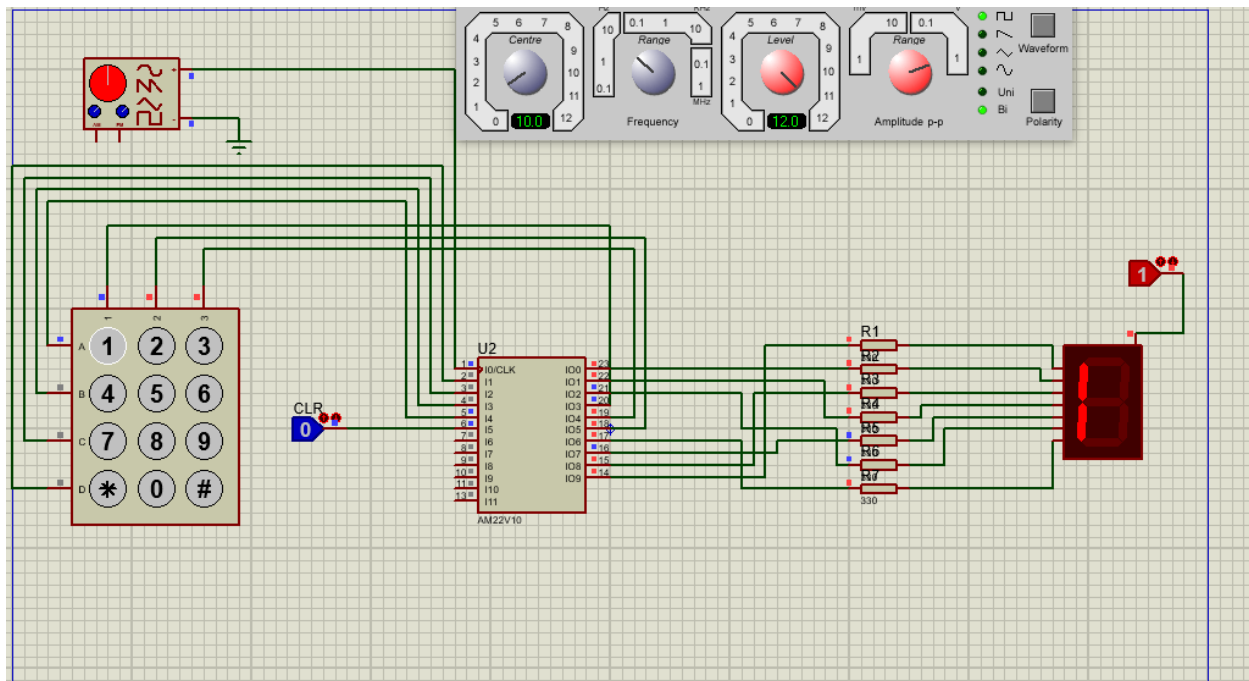
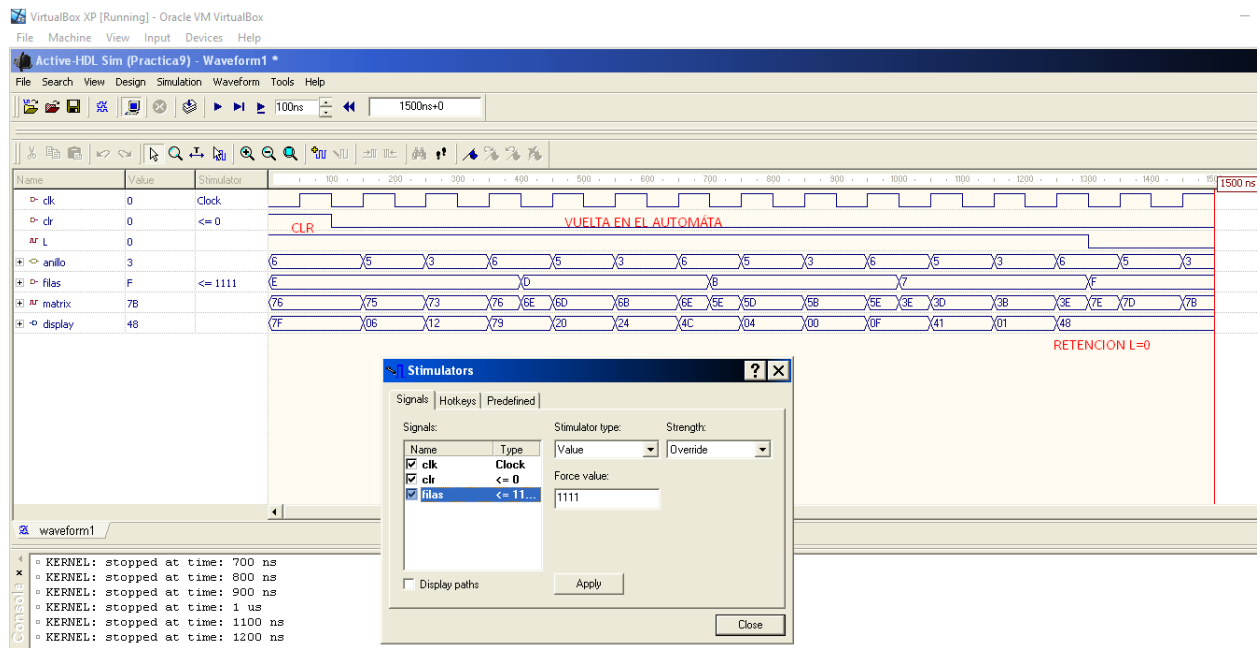
        when "1011011" =>
            regAux <= N7;
        when "1011101" =>
            regAux <= N8;
        when "1011110" =>
            regAux <= N9;
        when "0111011" =>
            regAux <= AS;
        when "0111101" =>
            regAux <= N0;
        when "0111110" =>
            regAux <= NG;
        when others => regAux <= "-----";
    end case;
end if;
end if;
end process;

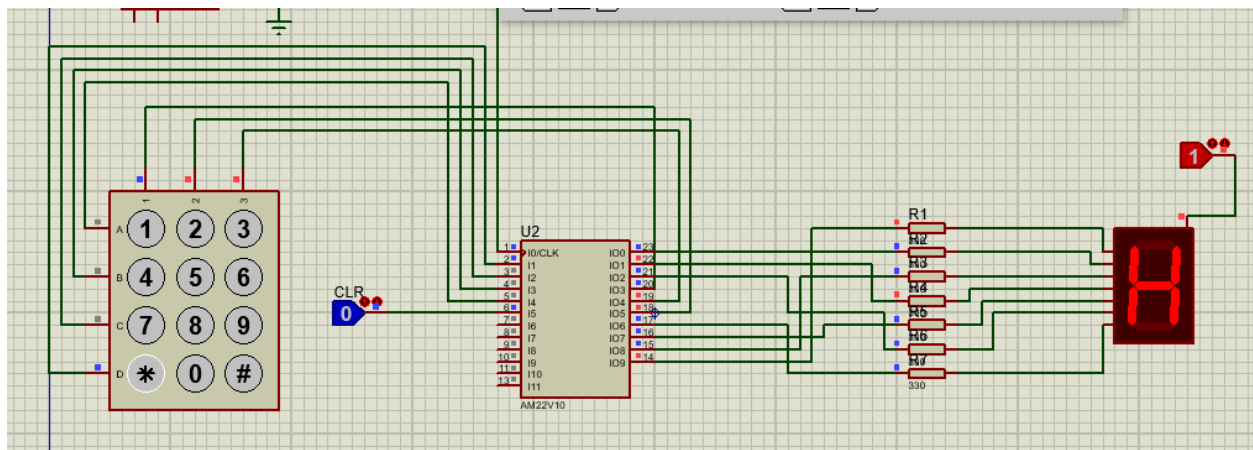
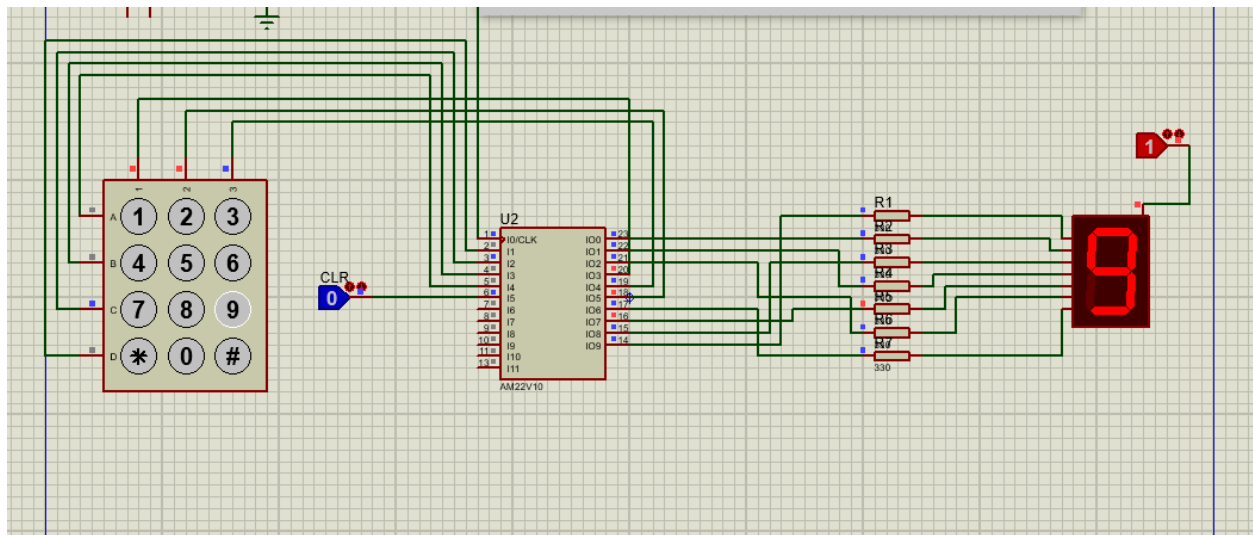
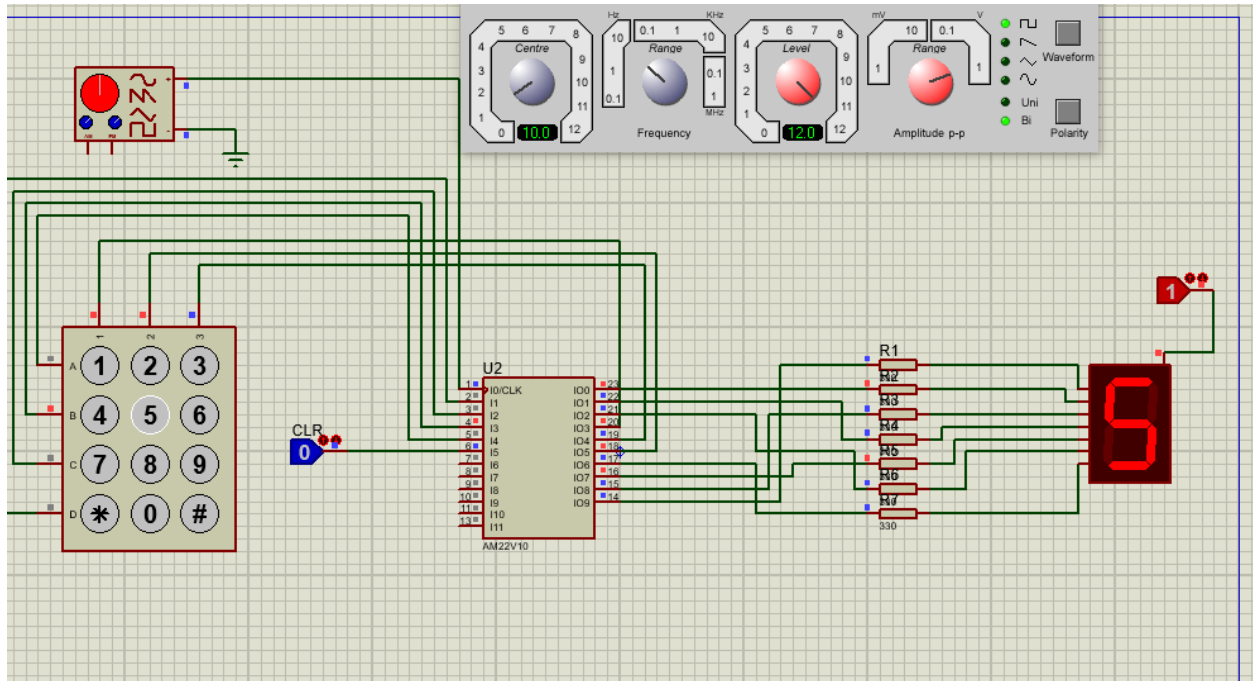
--CONTADOR DE ANILLO
process(clk, clr)
begin
    if(clr = '1') then
        anillo <= "110";
    elsif( rising_edge(clk)) then
        anillo <= to_stdlogicvector(to_bitvector(anillo) rol 1);
    end if;
end process;

L <= not(filas(3) and filas(2) and filas(1) and filas(0));

```

end architecture;





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

Solo 1

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

Alrededor de 40

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

10 de salida/entrada con macroceldas, 1 de reloj y 5 entradas

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

40 y se utiliza el 33% del PLD

5. ¿Qué frecuencia se debe utilizar para detectar la tecla presionada?

De preferencia una frecuencia alta para que se actualice rápidamente al presionar, 60Hz es ideal

6. ¿Cuáles son las señales que funcionan de manera síncrona y cuales de manera asíncrona?

Clk es la señal de reloj, Clr es el control asíncrono y todas las demás funcionan de manera síncrona (el registro y el contador)

7. ¿Qué puedes concluir de esta práctica?

A estas alturas ya tenemos la autonomía para completar un diseño con un registro, una salida, una codificación, un contador, una compuerta NAND y una entrada de teclado por nuestra cuenta, fue muy interesante ver como todos estos componentes trabajan juntos para lograr un objetivo mayor, la integración de todos estos componentes individuales es la clave para construir diseños mas avanzados.