Angie Paola Marchena Mondell 604650904
Proyecto corto 2 Dispositivos de lógica Programable

Tabla de la verdad y Convertidor binario de 7

Valor Hexa	Binario				Decodificador						
	X3	X2	X1	X0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
b	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
d	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

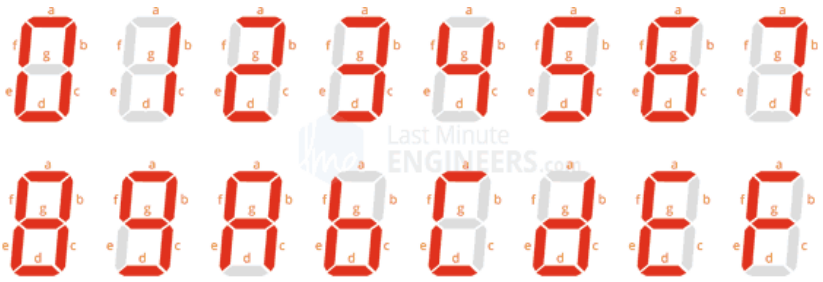

Figura 1: Visualización en 7 segmentos de los valores de entrada

Se realizó este decodificador mediante un estudio de de casos de entrada como se muestra en la imagen.

```
if auxVectIn = "0000" then auxVectOut := "1111110";    -- 0
elsif auxVectIn = "0001" then auxVectOut := "0110000"; -- 1
elsif auxVectIn = "0010" then auxVectOut := "1101101"; -- 2
elsif auxVectIn = "0011" then auxVectOut := "1111001"; -- 3
elsif auxVectIn = "0100" then auxVectOut := "0110011"; -- 4
elsif auxVectIn = "0101" then auxVectOut := "1011011"; -- 5
elsif auxVectIn = "0110" then auxVectOut := "1011111"; -- 6
elsif auxVectIn = "0111" then auxVectOut := "1110000"; -- 7
elsif auxVectIn = "1000" then auxVectOut := "1111111"; -- 8
elsif auxVectIn = "1001" then auxVectOut := "1110011"; -- 9
elsif auxVectIn = "1010" then auxVectOut := "1110111"; -- A
elsif auxVectIn = "1011" then auxVectOut := "0011111"; -- b
elsif auxVectIn = "1100" then auxVectOut := "1001110"; -- C
elsif auxVectIn = "1101" then auxVectOut := "0111101"; -- d
elsif auxVectIn = "1110" then auxVectOut := "1001111"; -- E
elsif auxVectIn = "1111" then auxVectOut := "1000111"; -- F
else auxVectOut := "0000000";
end if;
```

Figura 2: Código del decodificador.

Se analizó los casos mediante el siguiente testbench

```
generate_input_signals : process
   begin
       tbVectIn <= "0000"; wait for 100 ns;
       tbVectIn <= "0001"; wait for 100 ns;
       tbVectIn <= "0010"; wait for 100 ns;
       tbVectIn <= "0011"; wait for 100 ns;
       tbVectIn <= "0100"; wait for 100 ns;
       tbVectIn <= "0101"; wait for 100 ns;
       tbVectIn <= "0110"; wait for 100 ns;
       tbVectIn <= "0111"; wait for 100 ns;
       tbVectIn <= "1000"; wait for 100 ns;
       tbVectIn <= "1001"; wait for 100 ns;
       tbVectIn <= "1010"; wait for 100 ns;
       tbVectIn <= "1011"; wait for 100 ns;
       tbVectIn <= "1100"; wait for 100 ns;
       tbVectIn <= "1101"; wait for 100 ns;
       tbVectIn <= "1110"; wait for 100 ns;
       tbVectIn <= "1111"; wait for 100 ns;
       report "Fin de la simulacion!";
       -- Final de la simulacion con los resultados.
       wait;
   end process;
```

Figura 3: Test Bench con los casos de prueba.