

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 -- Definición de señales de entrada y salida
5 entity dec_4b_2d is
6     Port ( DIN : in std_logic_vector(3 downto 0);
7           CLK: in std_logic;
8           ca : out std_logic;
9           cb : out std_logic;
10          cc : out std_logic;
11          cd : out std_logic;
12          ce : out std_logic;
13          cf : out std_logic;
14          cg : out std_logic;
15          EN0 : out std_logic;
16          EN1 : out std_logic);
17 end dec_4b_2d;
18
19 architecture Behavioral of dec_4b_2d is
20 -- Definición de señales internas
21     type type_sreg is (STATE0,STATE1);
22     signal sreg, next_sreg: type_sreg;
23     signal next_ca,next_cb,next_cc,next_cd,next_ce,next_cf,next_cg: std_logic;
24     signal next_EN0,next_EN1: std_logic;
25     signal a0,b0,c0,d0,e0,f0,g0: std_logic;
26     signal a1,b1,c1,d1,e1,f1,g1: std_logic;
27 begin
28 -- Proceso donde calcula el valor de las 7 señales de activación de cada uno
29 -- de los LEDs de los 7 segmentos para cada dígito (2 dígitos).
30 -- Se calcula para todas las combinaciones de la señal de entrada
31 -- (4 bits -> 16 combinaciones)
32     Process (DIN)
33     begin
34         case DIN is
35             when "0000" =>
36                 a0<='0';b0<='0';c0<='0';d0<='0';e0<='0';f0<='0';g0<='1';
37                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
38             when "0001" =>
39                 a0<='1';b0<='0';c0<='0';d0<='1';e0<='1';f0<='1';g0<='1';
40                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
41             when "0010" =>
42                 a0<='0';b0<='0';c0<='1';d0<='0';e0<='0';f0<='1';g0<='0';
43                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
44             when "0011" =>
45                 a0<='0';b0<='0';c0<='0';d0<='0';e0<='1';f0<='1';g0<='0';
46                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
47             when "0100" =>
48                 a0<='1';b0<='0';c0<='0';d0<='1';e0<='1';f0<='0';g0<='0';
49                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
50             when "0101" =>
51                 a0<='0';b0<='1';c0<='0';d0<='0';e0<='1';f0<='0';g0<='0';
52                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
53             when "0110" =>
54                 a0<='0';b0<='1';c0<='0';d0<='0';e0<='0';f0<='0';g0<='0';
55                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
56             when "0111" =>
57                 a0<='0';b0<='0';c0<='0';d0<='1';e0<='1';f0<='1';g0<='1';
58                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
59             when "1000" =>
60                 a0<='0';b0<='0';c0<='0';d0<='0';e0<='0';f0<='0';g0<='0';
61                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
62             when "1001" =>
63                 a0<='0';b0<='0';c0<='0';d0<='0';e0<='1';f0<='0';g0<='0';
64                 a1<='0';b1<='0';c1<='0';d1<='0';e1<='0';f1<='0';g1<='1';
65             when "1010" =>
66                 a0<='0';b0<='0';c0<='0';d0<='0';e0<='0';f0<='0';g0<='1';
67                 a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
68             when "1011" =>
```

```

69         a0<='1';b0<='0';c0<='0';d0<='1';e0<='1';f0<='1';g0<='1';
70         a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
71     when "1100" =>
72         a0<='0';b0<='0';c0<='1';d0<='0';e0<='0';f0<='1';g0<='0';
73         a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
74     when "1101" =>
75         a0<='0';b0<='0';c0<='0';d0<='0';e0<='1';f0<='1';g0<='0';
76         a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
77     when "1110" =>
78         a0<='1';b0<='0';c0<='0';d0<='1';e0<='1';f0<='0';g0<='0';
79         a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
80     when "1111" =>
81         a0<='0';b0<='1';c0<='0';d0<='0';e0<='1';f0<='0';g0<='0';
82         a1<='1';b1<='0';c1<='0';d1<='1';e1<='1';f1<='1';g1<='1';
83     when others => null;
84     end case;
85 end process;
86
87 -- Proceso donde se actualizan los valores de las salidas y el estado al
88 -- recibir un flanco de subida del reloj utilizado para alternar la iluminación
89 -- de los 2 dígitos del display
90     process (CLK,next_sreg, next_ca, next_cb, next_cc, next_cd, next_ce, next_cf, next
_cg, next_EN0, next_EN1)
91     begin
92         if CLK='1' and CLK'event then
93             sreg <= next_sreg;
94             ca <= next_ca;
95             cb <= next_cb;
96             cc <= next_cc;
97             cd <= next_cd;
98             ce <= next_ce;
99             cf <= next_cf;
100            cg <= next_cg;
101            EN0 <= next_EN0;
102            EN1 <= next_EN1;
103        end if;
104    end process;
105
106 -- Proceso para ir asignando a las salidas los valores correspondientes para
107 -- representar el número decimal adecuado en el dígito que se desinhibe su iluminación.
108 -- A cada flanco de reloj se van desinhibiendo alternativamente los dos dígitos del display
109     process (sreg)
110     begin
111         next_ca<= a0; next_cb<= b0; next_cc<= c0;
112         next_cd<= d0; next_ce<= e0; next_cf<= f0; next_cg<= g0;
113         next_EN0<= '0'; next_EN1<= '1';
114         next_sreg<= STATE0;
115
116         case sreg is
117             when STATE0 =>
118                 next_sreg<= STATE1;
119                 next_ca<= a1;
120                 next_cb<= b1;
121                 next_cc<= c1;
122                 next_cd<= d1;
123                 next_ce<= e1;
124                 next_cf<= f1;
125                 next_cg<= g1;
126                 next_EN0<= '1';
127                 next_EN1<= '0';
128             when STATE1 =>
129                 next_sreg<= STATE0;
130                 next_ca<= a0;
131                 next_cb<= b0;
132                 next_cc<= c0;
133                 next_cd<= d0;
134                 next_ce<= e0;

```

```
135         next_cf<= f0;
136         next_cg<= g0;
137         next_EN0<= '0';
138         next_EN1<= '1';
139         when others => null;
140     end case;
141 end process;
142
143 end Behavioral;
144
```