

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.all;
5
6  entity lcd_moduleC is
7      Port ( clk : in  STD_LOGIC;
8            reset : in  STD_LOGIC;
9            -- Señales para escribir
10           DECENASMIL,UNIDADES MIL,CENTENASUN,DECENASUN,UNIDADESUN,DECENASDEC,
11           UNIDADESDEC: in std_logic_vector ( 3 downto 0);
12           db : out  STD_LOGIC_VECTOR (7 downto 0);
13           rs : out  STD_LOGIC;
14           rw : out  STD_LOGIC;
15           ena : out  STD_LOGIC);
16
17  architecture Behavioral of lcd_moduleC is
18  type state is (sel,se2,se3,se4,se5,si1,si2,si3,s1ini2,s1ini3,s1ini4,s1ini5,s1lcur,
19  s1write,s2write,sfin,s3write,s4write,s5write,s6write,s7write,s8write);
20  signal mq : state;
21
22  type nom is ( inil,ini2,ini3,ini4,loccur,writx,writx1,writx2,writx3,writx4,writx5,
23  writx6,writx7);
24  signal var: nom;
25
26  signal tc : std_logic_vector ( 20 downto 0);
27
28  signal dbi: std_logic_vector ( 7 downto 0);
29
30  -- num es la columna que queremos en este caso numero de 0 a 9 (3)
31  signal num : std_logic_vector ( 3 downto 0);
32
33  begin
34
35  rw <='0';
36
37  process(clk,reset)
38  begin
39      if reset ='1' then
40          mq <= si1;
41          rs <= '0';
42          ena <= '0';
43          tc <= ( others => '0');
44          var <= inil;
45          db <= "00000000";
46          dbi <= "00100000";
47          num <= "0011";
48      elsif clk'event and clk ='1' then
49
50          case mq is
51          --- rutina de env?o de datos
52              when sel =>
53                  ena <= '1';
54                  mq <= se2;
```

```
55         if tc < 101 then
56             tc <= tc + 1;
57             mq <= se2;
58         else
59             tc <= ( others =>'0' );
60             mq <= se3;
61         end if;
62
63     when se3 =>
64         ena <= '0';
65         mq <= se4;
66
67     when se4 =>
68         if tc < 2501 then
69             tc <= tc + 1;
70             mq <= se4;
71         else
72             tc <= ( others =>'0' );
73             mq <= se5;
74         end if;
75
76     when se5 =>
77         case var is
78             when ini1 => mq <= slini2;
79             when ini2 => mq <= slini3;
80             when ini3 => mq <= slini4;
81             when ini4 => mq <= slini5;
82             when loccur => mq <= slwrite;
83             when writx  => mq <= s2write ;
84             when writx1 => mq <= s3write;
85             when writx2 => mq <= s4write;
86             when writx3 => mq <= s5write;
87             when writx4 => mq <= s6write;
88             when writx5 => mq <= s7write;
89             when writx6 => mq <= s8write;
90             when writx7 => mq <= sfin;
91
92             when others => null;
93         end case;
94
95     --inicializaci?n
96
97
98     when si1 =>  -- se configura LCD a 8 bits, 2 l?neas y caracter de 5 x 7
99
100     ena <= '0';
101     mq <= si2;
102
103     when si2 =>
104         if tc < 1000001 then
105             tc <= tc + 1;
106             mq <= si2;
107         else
108             tc <= (others =>'0');
109             mq <= si3;
110         end if;
111
```

```
112         when si3 =>
113             db <= x"38";
114             rs <= '0';
115             mq <= sel;
116
117     when slini2 =>    -- entry mode -- incremente apuntador y mueva cursor a
    la derecha
118         var <= ini2;
119         db <= x"06";
120         rs <= '0';
121         mq <= sel;
122
123     when slini3 =>    -- Display on, cursor off y est?tico.
124         var <= ini3;
125         db <= x"0C";
126         rs <= '0';
127         mq <= sel;
128
129     when slini4 =>    -- se borra display
130         var <= ini4;
131         db <= x"01";
132         rs <= '0';
133         mq <= sel;
134
135     when slini5 =>    -- se espera ejecuci?n del borrado
136         if tc < 1000001 then
137             tc <= tc + 1;
138             mq <= slini5;
139         else
140             tc <= (others => '0');
141             mq <= sllcur;
142         end if;
143
144     when sllcur =>    -- ubica cursor en la posici?n 00
145         var <= loccur;
146         db <= x"80";
147         rs <= '0';
148         mq <= sel;
149         --Aqu? ponemos que se escribe con cada selectror
150         --x"numnum" para escribir un simbolo em particular
151         --var&var para un simbolo desede una se?al
152         -- el var, se configura al siguiente en la lista
153     when slwrite =>
154         var <= writx;
155         rs <= '1';
156         db <= "0011"&DECENASMIL;
157         mq <= sel;
158     when s2write =>
159         var <= writx1;
160         rs <= '1';
161         db <= "0011"&UNIDADESMIL;
162         mq <= sel;
163     when s3write =>
164         var <= writx2;
165         rs <= '1';
166         db <= "0011"&CENTENASUN;
167         mq <= sel;
```

```
168
169         when s4write =>
170             var <= writx3;
171             rs <= '1';
172             db <= "0011"&DECENASUN;
173             mq <= sel;
174
175         when s5write =>
176             var <= writx4;
177             rs <= '1';
178             db <= "0011"&UNIDADESUN;
179             mq <= sel;
180
181         when s6write =>
182             var <= writx5;
183             rs <= '1';
184             db <= x"2E";--punto
185             mq <= sel;
186
187         when s7write =>
188             var <= writx6;
189             rs <= '1';
190             db <= "0011"&DECENASDEC;
191             mq <= sel;
192
193         when s8write =>
194             var <= writx7;
195             rs <= '1';
196             db <= "0011"&UNIDADESDEC;
197             mq <= sel;
198
199         when sfin => mq <= s11cur;
200
201
202
203         when others => null;
204     end case;
205 end if;
206 end process;
207
208
209
210 end Behavioral;
211
```