```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity bitToBcd13bitsToD7Seg is
5     port (SW_in: in std_logic_vector(12 downto 0);
6           HEX0,HEX1,HEX2,HEX3: out std_logic_vector(6 downto 0));
7   end bitToBcd13bitsToD7Seg ;
8
9   architecture ckt of bitToBcd13bitsToD7Seg is
10    component bitToBcd13bits is
11      port (bit_in: in std_logic_vector(12 downto 0);
12            bcd_out: out std_logic_vector(15 downto 0));
13    end component;
14    component d7Seg is
15       port(S_in: in std_logic_vector(3 downto 0);
16            D_out: out std_logic_vector(6 downto 0));
17    end component;
18
19    signal bcd_ax_out: std_logic_vector(15 downto 0);
20    signal hex0_ax_out,hex1_ax_out,hex2_ax_out,hex3_ax_out : std_logic_vector(6 downto 0);
21
22    begin
23        BtB: BitToBcd13bits port map(
24             bit_in => SW_in,
25             bcd_out => bcd_ax_out);
26
27        D7S3: d7Seg port map(
28              S_in => bcd_ax_out(15 downto 12),
29              D_out => hex3_ax_out);
30
31        D7S2: d7Seg port map(
32              S_in => bcd_ax_out(11 downto 8),
33              D_out => hex2_ax_out);
34
35        D7S1: d7Seg port map(
36              S_in => bcd_ax_out(7 downto 4),
37              D_out => hex1_ax_out);
38
39        D7S0: d7Seg port map(
40              S_in => bcd_ax_out(3 downto 0),
41              D_out => hex0_ax_out);
42        HEX0 <= hex0_ax_out;
43        HEX1 <= hex1_ax_out;
44        HEX2 <= hex2_ax_out;
45        HEX3 <= hex3_ax_out;
46  end ckt;
```

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity d7Seg is
5     port(S_in: in std_logic_vector(3 downto 0);
6          D_out: out std_logic_vector(6 downto 0));
7   end d7Seg;
8
9   architecture display of d7Seg is
10    begin
11      D_out(0) <= not ((S_in(3) or S_in(1) or (S_in(2) and S_in(0)) or (S_in(2) nor S_in(0
    ))));
12
13      D_out(1) <= not (((not S_in(2)) or (S_in(1) nor S_in(0)) or (S_in(1) and S_in(0))));
14
15      D_out(2) <= not (((not(S_in(3)) and ((S_in(1) and S_in(0)) or S_in(2))) or (not(S_in(2
    ) or S_in(1)))));
16
17      D_out(3) <= not((S_in(3) or (S_in(1) and (S_in(2) nand S_in(0))) or (S_in(2) nor S_in(
    0)) or (S_in(2) and (not S_in(1)) and S_in(0))));
18
19      D_out(4) <= not (((S_in(1) and (not S_in(0))) or (S_in(2) nor S_in(0))));
20
21      D_out(5) <= not ((S_in(3) or (S_in(1) nor S_in(0)) or (S_in(2) and (S_in(1) nand S_in(
    0)))));
22
23      D_out(6) <= not ((S_in(3) or (S_in(1) and (not S_in(2))) or (S_in(2) and (S_in(1) nand
     S_in(0)))));
24
25  end display;
```

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity bitToBcd13bits is
5     port (bit_in: in std_logic_vector(12 downto 0);
6           bcd_out: out std_logic_vector(15 downto 0));
7   end bitToBcd13bits ;
8
9   architecture ckt of bitToBcd13bits is
10    component ciBitToBcd is
11      port (BtB_in: in std_logic_vector(3 downto 0);
12            BtB_out: out std_logic_vector(3 downto 0));
13    end component;
14
15    signal ciBtB_01_out, ciBtB_02_out, ciBtB_03_out, ciBtB_04_out, ciBtB_05_out,
    ciBtB_06_out, ciBtB_07_out, ciBtB_08_out, ciBtB_09_out: std_logic_vector(3 downto 0);
16    signal ciBtB_10_out, ciBtB_11_out, ciBtB_12_out, ciBtB_13_out, ciBtB_14_out,
    ciBtB_15_out, ciBtB_16_out, ciBtB_17_out, ciBtB_18_out: std_logic_vector(3 downto 0);
17    signal ciBtB_19_out, ciBtB_20_out, ciBtB_21_out: std_logic_vector(3 downto 0);
18
19    begin
20       ciBtB01: ciBitToBcd port map(
21                BtB_in(3) => '0',
22                BtB_in(2 downto 0) => bit_in(12 downto 10),
23                BtB_out => ciBtB_01_out);
24
25       ciBtB02: ciBitToBcd port map(
26                BtB_in(3 downto 1) => ciBtB_01_out(2 downto 0),
27                BtB_in(0) => bit_in(9),
28                BtB_out => ciBtB_02_out);
29
30       ciBtB03: ciBitToBcd port map(
31                BtB_in(3 downto 1) => ciBtB_02_out(2 downto 0),
32                BtB_in(0) => bit_in(8),
33                BtB_out => ciBtB_03_out);
34
35       ciBtB04: ciBitToBcd port map(
36                BtB_in(3) => '0',
37                BtB_in(2) => ciBtB_01_out(3),
38                BtB_in(1) => ciBtB_02_out(3),
39                BtB_in(0) => ciBtB_03_out(3),
40                BtB_out => ciBtB_04_out);
41
42       ciBtB05: ciBitToBcd port map(
43                BtB_in(3 downto 1)=> ciBtB_03_out(2 downto 0),
44                BtB_in(0) => bit_in(7),
45                BtB_out => ciBtB_05_out);
46
47       ciBtB06: ciBitToBcd port map(
48                BtB_in(3 downto 1)=> ciBtB_04_out(2 downto 0),
49                BtB_in(0) => ciBtB_05_out(3),
50                BtB_out => ciBtB_06_out);
51
52       ciBtB07: ciBitToBcd port map(
53                BtB_in(3 downto 1)=> ciBtB_05_out(2 downto 0),
54                BtB_in(0) => bit_in(6),
55                BtB_out => ciBtB_07_out);
56
57       ciBtB08: ciBitToBcd port map(
58                BtB_in(3 downto 1)=> ciBtB_06_out(2 downto 0),
59                BtB_in(0) => ciBtB_07_out(3),
60                BtB_out => ciBtB_08_out);
61
62       ciBtB09: ciBitToBcd port map(
63                BtB_in(3 downto 1)=> ciBtB_07_out(2 downto 0),
64                BtB_in(0) => bit_in(5),
```

```
65                BtB_out => ciBtB_09_out);
66
67        ciBtB10: ciBitToBcd port map(
68                BtB_in(3) => '0',
69                BtB_in(2) => ciBtB_04_out(3),
70                BtB_in(1) => ciBtB_06_out(3),
71                BtB_in(0) => ciBtB_08_out(3),
72                BtB_out => ciBtB_10_out);
73
74        ciBtB11: ciBitToBcd port map(
75                BtB_in(3 downto 1)=> ciBtB_08_out(2 downto 0),
76                BtB_in(0) => ciBtB_09_out(3),
77                BtB_out => ciBtB_11_out);
78
79        ciBtB12: ciBitToBcd port map(
80                BtB_in(3 downto 1)=> ciBtB_09_out(2 downto 0),
81                BtB_in(0) => bit_in(4),
82                BtB_out => ciBtB_12_out);
83
84        ciBtB13: ciBitToBcd port map(
85                BtB_in(3 downto 1)=> ciBtB_10_out(2 downto 0),
86                BtB_in(0) => ciBtB_11_out(3),
87                BtB_out => ciBtB_13_out);
88
89        ciBtB14: ciBitToBcd port map(
90                BtB_in(3 downto 1)=> ciBtB_11_out(2 downto 0),
91                BtB_in(0) => ciBtB_12_out(3),
92                BtB_out => ciBtB_14_out);
93
94        ciBtB15: ciBitToBcd port map(
95                BtB_in(3 downto 1)=> ciBtB_12_out(2 downto 0),
96                BtB_in(0) => bit_in(3),
97                BtB_out => ciBtB_15_out);
98
99        ciBtB16: ciBitToBcd port map(
100               BtB_in(3 downto 1)=> ciBtB_13_out(2 downto 0),
101               BtB_in(0) => ciBtB_14_out(3),
102               BtB_out => ciBtB_16_out);
103
104       ciBtB17: ciBitToBcd port map(
105               BtB_in(3 downto 1)=> ciBtB_14_out(2 downto 0),
106               BtB_in(0) => ciBtB_15_out(3),
107               BtB_out => ciBtB_17_out);
108
109       ciBtB18: ciBitToBcd port map(
110               BtB_in(3 downto 1)=> ciBtB_15_out(2 downto 0),
111               BtB_in(0) => bit_in(2),
112               BtB_out => ciBtB_18_out);
113
114       ciBtB19: ciBitToBcd port map(
115               BtB_in(3 downto 1)=> ciBtB_16_out(2 downto 0),
116               BtB_in(0) => ciBtB_17_out(3),
117               BtB_out => ciBtB_19_out);
118
119       ciBtB20: ciBitToBcd port map(
120               BtB_in(3 downto 1)=> ciBtB_17_out(2 downto 0),
121               BtB_in(0) => ciBtB_18_out(3),
122               BtB_out => ciBtB_20_out);
123
124       ciBtB21: ciBitToBcd port map(
125               BtB_in(3 downto 1)=> ciBtB_18_out(2 downto 0),
126               BtB_in(0) => bit_in(1),
127               BtB_out => ciBtB_21_out);
128
129       bcd_out(15) <= ciBtB_10_out(3);
130       bcd_out(14) <= ciBtB_13_out(3);
```

```
131         bcd_out(13) <= ciBtB_16_out(3);
132         bcd_out(12 downto 9) <= ciBtB_19_out;
133         bcd_out(8 downto 5) <= ciBtB_20_out;
134         bcd_out(4 downto 1) <= ciBtB_21_out;
135         bcd_out(0) <= bit_in(0);
136 
137     end ckt;
```

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity ciBitToBcd is
5     port (BtB_in: in bit_vector(3 downto 0);
6           BtB_out: out bit_vector(3 downto 0));
7   end ciBitToBcd;
8
9   architecture ckt of ciBitToBcd is
10  begin
11    BtB_out(3) <= (BtB_in(3) or (BtB_in(2) and (BtB_in(1) or BtB_in(0))));
12    BtB_out(2)<= ((BtB_in(2) and (not BtB_in(1)) and (not BtB_in(0))) or (BtB_in(3) and
    BtB_in(0)));
13    BtB_out(1)<= ((BtB_in(3) and (not BtB_in(0))) or (BtB_in(1) and ((not BtB_in(2)) or
    BtB_in(0))));
14    BtB_out(0)<= (((not BtB_in(3)) and (not BtB_in(2)) and BtB_in(0)) or ((not BtB_in(0))
    and (BtB_in(3) or (BtB_in(2) and BtB_in(1)))));
15  end ckt;
16
```