```
1         --  E:\PROJECTS\...\STATE_COUNTER_DIAGRAM.vhd
2         --  VHDL code created by Xilinx's StateCAD 9.2i
3         --  Thu Jan 17 17:50:44 2008
4
5         --  This VHDL code (for use with Xilinx XST) was generated using:
6         --  binary encoded state assignment with structured code format.
7         --  Minimization is enabled,  implied else is enabled,
8         --  and outputs are speed optimized.
9
10        LIBRARY ieee;
11        USE ieee.std_logic_1164.all;
12
13        ENTITY SHELL_STATE_COUNTER_DIAGRAM IS
14           PORT (CLK,RESET,w: IN std_logic;
15              Q0,Q1,Q2 : OUT std_logic);
16        END;
17
18        ARCHITECTURE BEHAVIOR OF SHELL_STATE_COUNTER_DIAGRAM IS
19           SIGNAL sreg : std_logic_vector (2 DOWNTO 0);
20           SIGNAL next_sreg : std_logic_vector (2 DOWNTO 0);
21           CONSTANT STATE0 : std_logic_vector (2 DOWNTO 0) :="000";
22           CONSTANT STATE1 : std_logic_vector (2 DOWNTO 0) :="001";
23           CONSTANT STATE2 : std_logic_vector (2 DOWNTO 0) :="010";
24           CONSTANT STATE3 : std_logic_vector (2 DOWNTO 0) :="011";
25           CONSTANT STATE4 : std_logic_vector (2 DOWNTO 0) :="100";
26           CONSTANT STATE5 : std_logic_vector (2 DOWNTO 0) :="101";
27           CONSTANT STATE6 : std_logic_vector (2 DOWNTO 0) :="110";
28           CONSTANT STATE7 : std_logic_vector (2 DOWNTO 0) :="111";
29
30           SIGNAL next_Q0,next_Q1,next_Q2 : std_logic;
31           SIGNAL Q : std_logic_vector (2 DOWNTO 0);
32        BEGIN
33           PROCESS (CLK, RESET, next_sreg, next_Q2, next_Q1, next_Q0)
34           BEGIN
35              IF ( RESET='1' ) THEN
36                 sreg <= STATE0;
37                 Q2 <= '0';
38                 Q1 <= '0';
39                 Q0 <= '0';
40              ELSIF CLK='1' AND CLK'event THEN
41                 sreg <= next_sreg;
42                 Q2 <= next_Q2;
43                 Q1 <= next_Q1;
44                 Q0 <= next_Q0;
45              END IF;
46           END PROCESS;
47
48           PROCESS (sreg,w,Q)
49           BEGIN
50              next_Q0 <= '0'; next_Q1 <= '0'; next_Q2 <= '0';
51              Q<=std_logic_vector'("000");
52
53              next_sreg<=STATE0;
54
55              CASE sreg IS
56                 WHEN STATE0 =>
57                    IF ( w='1' ) THEN
58                       next_sreg<=STATE1;
59
60                       Q <= (std_logic_vector'("001"));
61                    END IF;
```

```
62                      IF ( w='0' ) THEN
63                         next_sreg<=STATE0;
64
65                         Q <= (std_logic_vector'("000"));
66                      END IF;
67                   WHEN STATE1 =>
68                      IF ( w='1' ) THEN
69                         next_sreg<=STATE2;
70
71                         Q <= (std_logic_vector'("010"));
72                      END IF;
73                      IF ( w='0' ) THEN
74                         next_sreg<=STATE1;
75
76                         Q <= (std_logic_vector'("001"));
77                      END IF;
78                   WHEN STATE2 =>
79                      IF ( w='1' ) THEN
80                         next_sreg<=STATE3;
81
82                         Q <= (std_logic_vector'("011"));
83                      END IF;
84                      IF ( w='0' ) THEN
85                         next_sreg<=STATE2;
86
87                         Q <= (std_logic_vector'("010"));
88                      END IF;
89                   WHEN STATE3 =>
90                      IF ( w='1' ) THEN
91                         next_sreg<=STATE4;
92
93                         Q <= (std_logic_vector'("100"));
94                      END IF;
95                      IF ( w='0' ) THEN
96                         next_sreg<=STATE3;
97
98                         Q <= (std_logic_vector'("011"));
99                      END IF;
100                  WHEN STATE4 =>
101                     IF ( w='1' ) THEN
102                        next_sreg<=STATE5;
103
104                        Q <= (std_logic_vector'("101"));
105                     END IF;
106                     IF ( w='0' ) THEN
107                        next_sreg<=STATE4;
108
109                        Q <= (std_logic_vector'("100"));
110                     END IF;
111                  WHEN STATE5 =>
112                     IF ( w='1' ) THEN
113                        next_sreg<=STATE6;
114
115                        Q <= (std_logic_vector'("110"));
116                     END IF;
117                     IF ( w='0' ) THEN
118                        next_sreg<=STATE5;
119
120                        Q <= (std_logic_vector'("101"));
121                     END IF;
122                  WHEN STATE6 =>
```

```
123                   IF ( w='1' ) THEN
124                      next_sreg<=STATE7;
125
126                      Q <= (std_logic_vector'("111"));
127                   END IF;
128                   IF ( w='0' ) THEN
129                      next_sreg<=STATE6;
130
131                      Q <= (std_logic_vector'("110"));
132                   END IF;
133                WHEN STATE7 =>
134                   IF ( w='1' ) THEN
135                      next_sreg<=STATE0;
136
137                      Q <= (std_logic_vector'("000"));
138                   END IF;
139                   IF ( w='0' ) THEN
140                      next_sreg<=STATE7;
141
142                      Q <= (std_logic_vector'("111"));
143                   END IF;
144                WHEN OTHERS =>
145             END CASE;
146
147             next_Q2 <= Q(2);
148             next_Q1 <= Q(1);
149             next_Q0 <= Q(0);
150          END PROCESS;
151       END BEHAVIOR;
152
153       LIBRARY ieee;
154       USE ieee.std_logic_1164.all;
155
156       ENTITY STATE_COUNTER_DIAGRAM IS
157          PORT (Q : OUT std_logic_vector (2 DOWNTO 0);
158             CLK,RESET,w: IN std_logic);
159       END;
160
161       ARCHITECTURE BEHAVIOR OF STATE_COUNTER_DIAGRAM IS
162          COMPONENT SHELL_STATE_COUNTER_DIAGRAM
163             PORT (CLK,RESET,w: IN std_logic;
164                Q0,Q1,Q2 : OUT std_logic);
165          END COMPONENT;
166       BEGIN
167          SHELL1_STATE_COUNTER_DIAGRAM : SHELL_STATE_COUNTER_DIAGRAM PORT MAP (CLK=>
168             CLK,RESET=>RESET,w=>w,Q0=>Q(0),Q1=>Q(1),Q2=>Q(2));
169       END BEHAVIOR;
170
```