```
1  timescale 1ns / 1ps
2
3  module ee201_numlock_sm(Clk, reset, U, Z, Unlock,
4      q_I, q_G1get, q_G1, q_G10get, q_G10, q_G101get,
5      q_G101, q_G1011get, q_G1011, q_Opening, q_Bad);
6
7
8      /*  INPUTS */
9      // Clock & Reset
10     input       Clk, reset;
11     input       U, Z;
12
13     /*  OUTPUTS */
14     // store current state
15     output      q_I, q_G1get, q_G1, q_G10get, q_G10, q_G101get,
16                 q_G101, q_G1011get, q_G1011, q_Opening, q_Bad;
17     //Unlock Output
18     output      Unlock;
19     reg [10:0]  state;
20     reg [1:0] UZ;
21     reg [2:0] Timerout_count;
22
23     assign {U, Z} = UZ;
24     assign {q_I, q_G1get, q_G1, q_G10get, q_G10, q_G101get,
25                 q_G101, q_G1011get, q_G1011, q_Opening, q_Bad} = state;
26
27
28     // lets make accessing the state information easier within the state
…  machine code
29     // each line aliases the approriate state bits and sets up a 1-hot
…  code
30     localparam
31
32     Q_I         = 11'b00000000001;
33     Q_G1GET     = 11'b00000000010;
34     Q_G1        = 11'b00000000100;
35     Q_G10GET    = 11'b00000001000;
36     Q_G10       = 11'b00000010000;
37     Q_G101GET   = 11'b00000100000;
38     Q_G101      = 11'b00001000000;
39     Q_G1011GET  = 11'b00010000000;
40     Q_G1011     = 11'b00100000000;
41     Q_OPENING   = 11'b01000000000;
42     Q_BAD       = 11'b10000000000;
43
```

```
44      //wire variables
45      wire = Timerout;
46
47
48      // NSL AND SM
49      always @ (posedge Clk, posedge reset)
50      begin
51          if(reset)
52              state <= Q_I;
53          else
54          begin
55              case(state)
56              Q_I:
57                  begin
58                      if (UZ == 2'b10)
59                          state <= Q_G1GET;
60                  end
61              Q_G1GET:
62                  begin
63                      if ( U == 0)
64                          state <= Q_G1;
65                  end
66              Q_G1:
67                  begin
68                      if (UZ == 2'b01)
69                          state <= Q_G10GET;
70                      else if (UZ == 2'b10)
71                          state <= Q_BAD;
72                  end
73              Q_G10GET:
74                  begin
75                      if ( Z == 0)
76                          state <= Q_G10;
77                  end
78              Q_G10:
79                  begin
80                      if ( UZ == 2'b10 )
81                          state <= Q_G101GET;
82                      else if (UZ == 2'b01)
83                          state <= Q_BAD;
84                  end
85              Q_G101GET:
86                  begin
87                      if (U == 0)
88                          state <= Q_G101;
```

```
89                  end
90              Q_G101:
91                  begin
92                      if ( UZ == 2'b10 )
93                          state <= Q_G1011GET;
94                      else if (UZ == 2'b01)
95                          state <= Q_BAD;
96                  end
97              Q_G1011GET:
98                  begin
99                      if (U == 0)
100                         state <= Q_G1011;
101                 end
102             Q_G1011:
103                 state <= Q_OPENING;
104             Q_OPENING:
105                 if (Timerout)
106                     state <= Q_I;
107             Q_BAD:
108                 begin
109                     if ( UZ = 2'b00)
110                         state <= Q_I;
111                 end
112             endcase
113         end
114     end
115
116     always @ (posedge Clk, posedge reset)
117     begin
118         if (Q_OPENING)
119             Timerout_count <= Timerout_count + 1;
120         else
121             Timerout_count <= 0;
122     end
123
124     // OFL
125     assign Unlock = state && Q_OPENING;
126     assign Timerout = Timerout_count && 3'b101;
127
128 endmodule
```