```
1   -- Quartus II VHDL Template
2   -- Four-State Moore State Machine
3
4   -- A Moore machine's outputs are dependent only on the current state.
5   -- The output is written only when the state changes.  (State
6   -- transitions are synchronous.)
7
8   library ieee;
9   use ieee.std_logic_1164.all;
10  use ieee.std_logic_unsigned.all;
11  entity summation_control is
12     port(
13
14        clk : in std_logic;
15        go_i : in   std_logic;
16        idx : out std_logic;
17        idt : out std_logic;
18        selt : out std_logic;
19        idn : out std_logic;
20        seln : out std_logic;
21        idy  : out std_logic;
22        sely : out std_logic;
23        ids : out std_logic;
24        sels : out std_logic;
25        ltflg  :in  std_logic;
26
27        idout: out std_logic
28     );
29
30  end summation_control;
31
32  architecture summation of summation_control is
33
34
35     -- Build an enumerated type for the state machine
36     type state_type is (s0, s1, s2, s3,s4,s5,s6,s7,s8,
37     s9,s10,s11,s12,s13,s14,s15,s16,s17,s18,s19,s20,
38     s21,s22,s23,s24,s25,s26,s27,s28);
39
40     -- Register to hold the current state
41     signal state,nextstate   : state_type;
42
43  begin
44
45     -- Logic to advance to the next state
46     process (clk)
47     begin
48     if clk'event and clk = '1' then
49     state <=nextstate;
50     end if;
51     end process;
52
53     process(state,go_i,ltflg)
54     begin
55           case state is
56              when s0=>
57                 if go_i = '1' then
58                    nextstate <= s1;
59                 else
60                    nextstate <= s0;
61                 end if;
62
```

```
63            when s1=>
64            nextstate<=s2;
65            when s2=>
66            nextstate <= s3;
67            when s3=>
68            if ltflg = '1' then
69                     nextstate <= s4;
70
71                  else
72                     nextstate <= s28;
73                  end if;
74
75
76
77
78            when s4=>
79
80                     nextstate <= s5;
81            when s5=>
82                     nextstate <= s6;
83
84
85                  when s6=> nextstate<=s7;
86                  when s7=> nextstate<=s8;
87                  when s8=>nextstate<=s9;
88                  when s9=>nextstate<=s10;
89                  when s10=>nextstate<=s11;
90                  when s11=>nextstate<=s12;
91                  when s12=>nextstate<=s13;
92                  when s13=>nextstate<=s14;
93                  when s14=>nextstate<=s15;
94                  when s15=>nextstate<=s16;
95                  when s16=>nextstate<=s17;
96                  when s17=>nextstate<=s18;
97                  when s18=>nextstate<=s19;
98                  when s19=>nextstate<=s20;
99                  when s20=>nextstate<=s21;
100                 when s21=>nextstate<=s22;
101                 when s22=>nextstate<=s23;
102                 when s23=>nextstate<=s24;
103                 when s24=>nextstate<=s25;
104                 when s25=>nextstate<=s26;
105                 when s26=>
106           nextstate<=s27;
107                 when s27=>
108           nextstate<=s2;
109
110           when s28=>
111           nextstate<=s0;
112           when others=>
113           nextstate<=s0;
114           end case;
115     end process;
116
117     -- Output depends solely on the current state
118        idx <='1' when state <=s1 or state<=s4 else '0';
119        idt <='1' when state <=s1 or state<=s8 else '0';
120        selt <='1' when state<=s1 else '0';
121        idn <='1' when state <=s1 or state <=s27 else '0';
122        seln <='1' when state <=s1 else '0';
123        idy  <='1' when state <=s1 or state<=s13 else '0';
124        sely <='1' when state<=s1  else '0';
```

```
125         ids <='1' when state <=s1 or state<=s20 else '0';
126         sels <='1' when state<=s1 else '0';
127         idout<='1' when state <=s28 else '0';
128   end summation;
```