```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    14:32:24 06/03/2010
6   -- Design Name:
7   -- Module Name:    START_UP - Behavioral
8   -- Project Name:
9   -- Target Devices:
10  -- Tool versions:
11  -- Description:
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  ---- Uncomment the following library declaration if instantiating
26  ---- any Xilinx primitives in this code.
27  --library UNISIM;
28  --use UNISIM.VComponents.all;
29
30  entity START_UP is
31      Port ( CLOCK  : in  STD_LOGIC;
32             ST_UP  : in  STD_LOGIC;
33             EN     : out STD_LOGIC;
34             RS     : out STD_LOGIC;
35             DONE   : out STD_LOGIC;
36             D_BUS  : out STD_LOGIC_VECTOR (3 downto 0));
37  end START_UP;
38
39  architecture Behavioral of START_UP is
40
41  signal counter       :  integer range 0 to 10000 := 0;
42  signal flag          :  std_logic := '0';
43  begin
44
45  process (CLOCK)
46  begin
47
48  if rising_edge(CLOCK) and flag = '0' and ST_UP = '1' then
49  counter <= counter + 1;
50  --
51
52     case counter is
53        when 1 =>
54        DONE <= '0';
55        RS    <= '0';
56        EN    <= '0';
57        D_BUS <= "0011";
58
59        when 2 =>
60        EN    <= '1';
61        D_BUS <= "0011";
```

```
 62
 63              when 3 =>
 64              EN    <= '0';
 65              D_BUS <= "0011";
 66
 67              when 4103 =>
 68              EN    <= '1';
 69              D_BUS <= "0011";
 70
 71              when 4104 =>
 72              EN    <= '0';
 73              D_BUS <= "0011";
 74
 75              when 4204 =>
 76              EN    <= '1';
 77              D_BUS <= "0011";
 78
 79              when 4205 =>
 80              EN    <= '0';
 81              D_BUS <= "0010";
 82
 83              when 4245 =>
 84              EN    <= '1';
 85              D_BUS <= "0010";
 86
 87              when 4246 =>
 88              EN    <= '0';
 89              D_BUS <= "0010";
 90
 91              when 4286 =>
 92              EN    <= '1';
 93              D_BUS <= "0010";
 94
 95              when 4287 =>
 96              EN    <= '0';
 97              D_BUS <= "1000";
 98
 99              when 4288 =>
100              EN    <= '1';
101              D_BUS <= "1000";
102
103              when 4289 =>
104              EN    <= '0';
105              D_BUS <= "0000";
106
107              when 4329 =>
108              EN    <= '1';
109              D_BUS <= "0000";
110
111              when 4330 =>
112              EN    <= '0';
113              D_BUS <= "0110";
114
115              when 4331 =>
116              EN    <= '1';
117              D_BUS <= "0110";
118
119              when 4332 =>
120              EN    <= '0';
121              D_BUS <= "0000";
122
```

```
123              when 4372 =>
124              EN    <= '1';
125              D_BUS <= "0000";
126
127              when 4373 =>
128              EN    <= '0';
129              D_BUS <= "1100";
130
131              when 4374 =>
132              EN    <= '1';
133              D_BUS <= "1100";
134
135              when 4375 =>
136              EN    <= '0';
137              D_BUS <= "0000";
138
139              when 4415 =>
140              EN    <= '1';
141              D_BUS <= "0000";
142
143              when 4416 =>
144              EN    <= '0';
145              D_BUS <= "0001";
146
147              when 4417 =>
148              EN    <= '1';
149              D_BUS <= "0001";
150
151              when 4418 =>
152              EN    <= '0';
153              D_BUS <= "0000";
154
155              when 6058 =>         --state 25
156              DONE <= '1';
157              flag <= '1';
158              --RS  <= '1';
159
160              when others =>
161           end case;
162
163        end if;
164
165        end process;
166
167        end Behavioral;
168
169
```