```
1
2
3   begin
4
5       -- Calcul de l'état suivant
6       -- Comme on est en std_logic,"elsif ='0'" et non "else", car le signal peux
        avoir d'autre valeur
7       process (state, athome, findfood, lostfood, closetofood, success, aboverestth,
        abovesearchth, scantimeup)
8       begin
9           case state is
10              when IDLE => nextstate <= RESTING;
11              when RESTING =>
12                  if aboverestth = '1' then nextstate <= RANDOMWALK;
13                  else--elsif aboverestth = '0' then
14                  nextstate <= RESTING;
15                  end if;
16
17              when RANDOMWALK =>
18                  if abovesearchth = '1' then nextstate <= HOMING;
19                  else--  abovesearchth = '0' then
20                      if findfood = '1' then nextstate <= MOVETOFOOD;
21                      else--elsif findfood = '0' then
22                          nextstate <= RANDOMWALK;
23                      end if;
24                  end if;
25
26              when SCANAREA =>
27                  if abovesearchth = '1' then nextstate <= HOMING;
28                  else--elsif  abovesearchth = '0' then
29                      if findfood = '1' then nextstate <= MOVETOFOOD;
30                      else--elsif findfood = '0' then
31                          if scantimeup = '1' then nextstate <= RANDOMWALK;
32                          else--elsif scantimeup = '0' then
33                          nextstate <= SCANAREA;
34                          end if;
35                      end if;
36                  end if;
37              when HOMING => if(athome = '1') then nextstate <= RESTING; else
                nextstate <= HOMING; end if;
38              when MOVETOFOOD =>
39                  if abovesearchth = '1' then nextstate <= HOMING;
40                  else--elsif  abovesearchth = '0' then
41                      if lostfood = '1' then nextstate <= SCANAREA;
42                      else--elsif lostfood = '0' then
43                          if closetofood = '1' then nextstate <= GRABFOOD;
44                          else--elsif closetofood = '0' then
45                              nextstate <= MOVETOFOOD;
46                          end if;
47                      end if;
48                  end if;
49              when GRABFOOD =>
50                  if success = '1' then nextstate <= MOVETOHOME;
51                  else--elsif success = '0' then
52                   nextstate <= GRABFOOD;
53                  end if;
54              when MOVETOHOME =>
55                  if athome = '1' then nextstate <= DEPOSIT;
56                  else--elsif athome = '0' then
57                   nextstate <= MOVETOHOME;
58                  end if;
59              when DEPOSIT =>
60                  if success = '1' then nextstate <= RESTING;
61                  else--elsif success = '0' then
62                   nextstate <= DEPOSIT;
63                  end if;
64          end case;
65      end process;
66
```