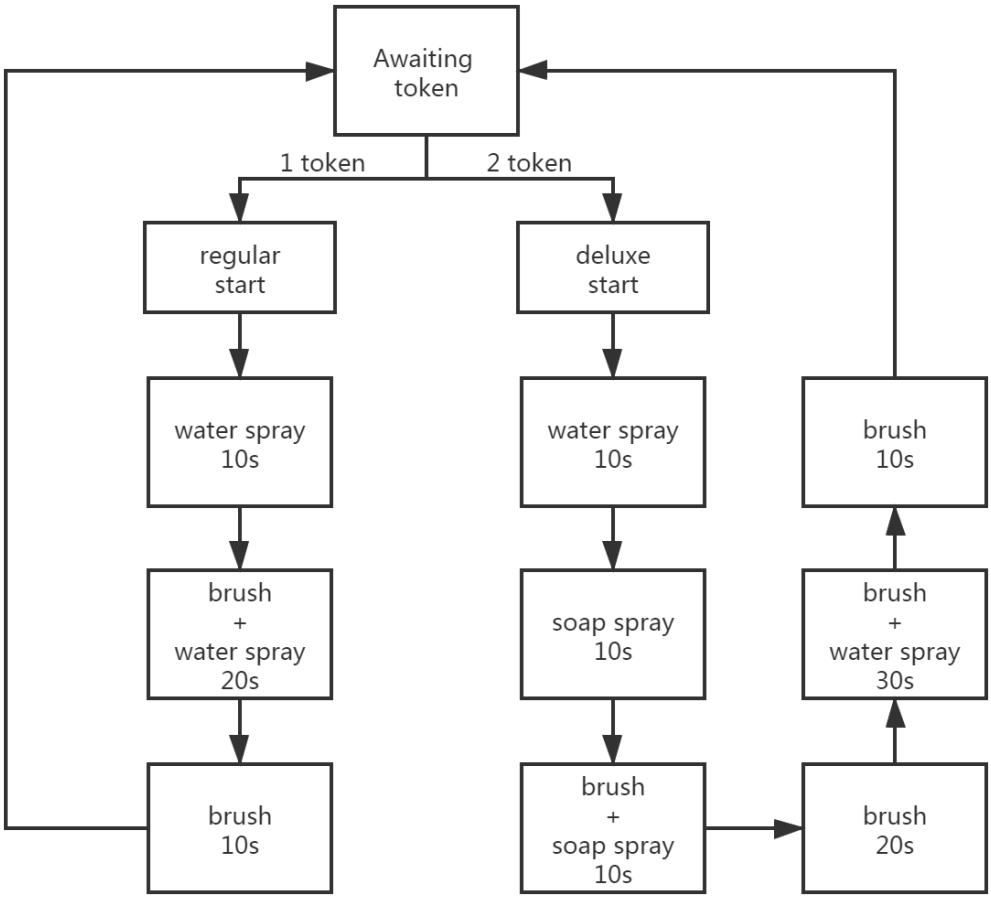
1.

Block diagram:



2.

Regular system

State diagram:

State:

S0: initial state

S1: 10s water spray

S2: 20s water spray + brush

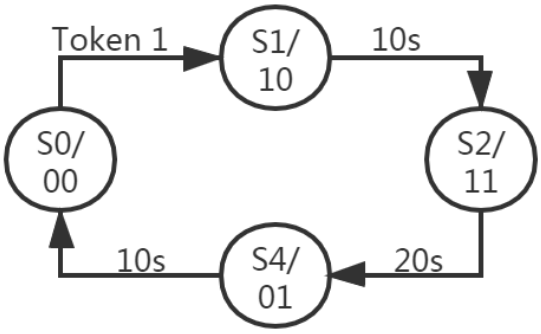
S4: 10s brush

Output:

Water spray: 10

Brush: 01

Water spray + brush: 11



Code:

5bit counter:

counter5bit.vhd | regular.vhd

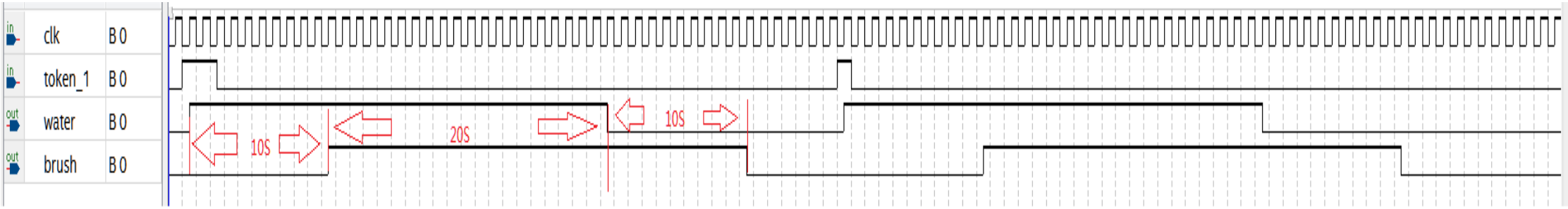
```
library ieee ;
use ieee . std_logic_1164 . all ;
use ieee . numeric_std . all ;
ENTITY counter5bit IS
   PORT ( clk , rst : in std_logic ;
          q: out std_logic_vector (4 downto 0));
END counter5bit;

ARCHITECTURE rtl OF counter5bit IS
signal cnt : unsigned (4 downto 0) := "00001";
BEGIN
   process (clk)
      begin
         if (clk'event and clk = '1') then
            if (rst='1') then
               cnt <= "00001";
            else
               cnt <= cnt + "1";
            end if ;
         end if ;
      end process ;
q <= std_logic_vector ( cnt );
END rtl ;
```

Regular wash:

```
library ieee; use ieee.std_logic_1164.all;

ENTITY regular IS
   PORT( clk, token_1 : IN STD_LOGIC;
         water, brush : OUT STD_LOGIC);
END regular;

ARCHITECTURE design OF regular IS
component counter5bit is
   port ( clk , rst : in std_logic ;
   q: out std_logic_vector (4 downto 0) );
end component;

signal start_timer, time_10sec, time_20sec: std_logic;
signal q_out : std_logic_vector (4 downto 0);

-- Declare a state type
type state_type is (Initial, W10s, WB20s, B10s);

-- Declare current and next state variables .
signal current_state , next_state : state_type := Initial;

BEGIN

t0 : counter5bit port map ( clk , start_timer , q_out );

time_10sec <= q_out (3) and q_out (1);
time_20sec <= q_out (4) and q_out (2);

process ( clk )
begin
   if (clk'event and clk='1') then
         current_state <= next_state ;
   end if ;
end process ;
-- Process to determine next state

process ( current_state, time_10sec, time_20sec)
begin
   case current_state is
      when Initial =>
         if (token_1 = '1') then
            next_state <= W10s ;
         else
            next_state <= current_state;
         end if ;
```

```
47  ├
48  │       when W10s =>
49  ⊟          if (time_10sec = '1') then
50  ├             next_state <= WB20s ;
51  ⊟          else
52  │             next_state <= current_state;
53  │          end if ;
54  ├
55  │       when WB20s =>
56  ⊟          if (time_20sec = '1') then
57  ├             next_state <= B10s ;
58  ⊟          else
59  │             next_state <= current_state;
60  │          end if ;
61  ├
62  │       when B10s =>
63  ⊟          if (time_10sec = '1') then
64  ├             next_state <= Initial ;
65  ⊟          else
66  │             next_state <= current_state;
67  │          end if ;
68  ├
69  ├end case ;
70  │end process ;
71  ├
72  │-- Conditional assignments for outputs:
73  │water <= '1' when ((current_state = W10s) or (current_state = WB20s)) else '0';
74  │brush <= '1' when ((current_state = B10s) or (current_state = WB20s)) else '0';
75  │
76  │start_timer <= '1' when (current_state /= next_state) else '0';
77  │
78  └END design ;
```

Test: 2 times of regular wash



3.

Deluxe + Regular

State diagram:

State:

S0: initial state

S11: 10s water spray

S12: 20s water spray + brush

S13: 10s brush

S21: 10s water spray

S22: 10s soap spray

S23: 10s soap spray + brush

S24: 20s brush

S25: 30s water spray + brush

S26: 10s brush

Output:

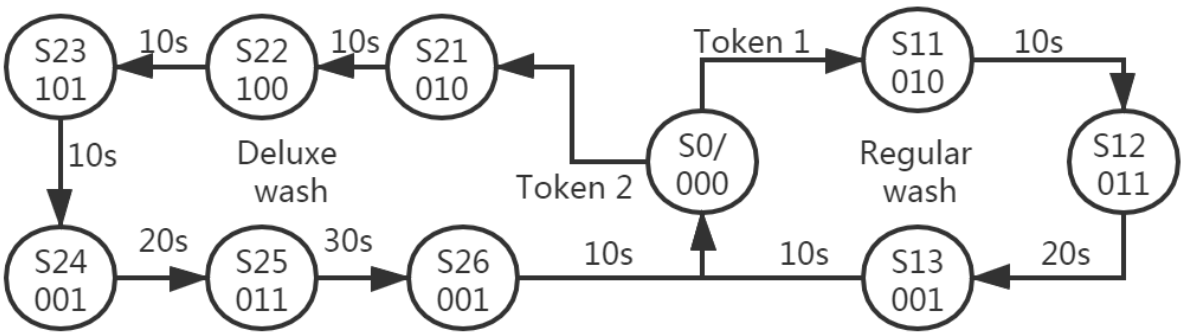
Water spray: 010

Brush: 001

Soap: 100

Water spray + brush: 011

Soap spray + brush: 101



Code:

deluxe_regular.vhd counter5bit.vhd

```
1  library ieee; use ieee.std_logic_1164.all;
2
3  ENTITY Deluxe_Regular IS
4     PORT( clk, token_1, token_2 : IN STD_LOGIC;
5           soap, water, brush : OUT STD_LOGIC);
6  END Deluxe_Regular;
7
8  ARCHITECTURE design OF Deluxe_Regular IS
9  component counter5bit is
10    port ( clk , rst : in std_logic ;
11    q: out std_logic_vector (4 downto 0) );
12 end component;
13
14 signal start_timer, time_10sec, time_20sec, time_30sec: std_logic;
15 signal q_out : std_logic_vector (4 downto 0);
16
17 -- Declare a state type
18 type state_type is (Initial, s11, s12, s13,
19                     s21, s22, s23, s24, s25, s26);
20
```

```
20
21  -- Declare current and next state variables .
22  signal current_state , next_state : state_type := Initial;
23
24  BEGIN
25
26  t0 : counter5bit port map ( clk , start_timer , q_out );
27
28  time_10sec <= q_out (3) and q_out (1);
29  time_20sec <= q_out (4) and q_out (2);
30  time_30sec <= q_out (4) and q_out (3) and q_out (2) and q_out (1);
31
32  process ( clk )
33  begin
34     if (clk'event and clk='1') then
35           current_state <= next_state ;
36     end if ;
37  end process ;
38  -- Process to determine next state
39
40  process ( current_state, time_10sec, time_20sec, time_30sec)
41  begin
42     case current_state is
43        when Initial =>
44           if (token_2 = '1') then
45              next_state <= s21 ;
46           elsif (token_1 = '1') then
47              next_state <= s11 ;
48           else
49              next_state <= current_state;
50           end if ;
51
52        when s11 =>
53           if (time_10sec = '1') then
54              next_state <= s12 ;
55           else
56              next_state <= current_state;
57           end if ;
```

```
58
59        when s12 =>
60           if (time_20sec = '1') then
61              next_state <= s13 ;
62           else
63              next_state <= current_state;
64           end if ;
65
66        when s13 =>
67           if (time_10sec = '1') then
68              next_state <= Initial ;
69           else
70              next_state <= current_state;
71           end if ;
72
73        when s21 =>
74           if (time_10sec = '1') then
75              next_state <= s22 ;
76           else
77              next_state <= current_state;
78           end if ;
79
80        when s22 =>
81           if (time_10sec = '1') then
82              next_state <= s23 ;
83           else
84              next_state <= current_state;
85           end if ;
86
87        when s23 =>
88           if (time_10sec = '1') then
89              next_state <= s24 ;
90           else
91              next_state <= current_state;
92           end if ;
```

```
93  
94          when s24 =>
95             if (time_20sec = '1') then
96                next_state <= s25 ;
97             else
98                next_state <= current_state;
99             end if ;
100 
101         when s25 =>
102            if (time_30sec = '1') then
103               next_state <= s26 ;
104            else
105               next_state <= current_state;
106            end if ;
107 
108         when s26 =>
109            if (time_10sec = '1') then
110               next_state <= Initial ;
111            else
112               next_state <= current_state;
113            end if ;
114 
115 end case ;
116 end process ;
117 
118 -- Conditional assignments for outputs:
119 soap  <= '1' when ((current_state = s22) or (current_state = s23)) else '0';
120 
121 water <= '1' when ((current_state = s11) or (current_state = s12)
122                    or (current_state = s21) or (current_state = s25)) else '0';
123 
124 brush <= '1' when ((current_state = s12) or (current_state = s13)
125                    or (current_state = s23) or (current_state = s24)
126                    or (current_state = s25) or (current_state = s26)) else '0';
127 
128 start_timer <= '1' when (current_state /= next_state) else '0';
129 
130 END design ;
```

Test: 1 time of regular wash, then 1 time of deluxe wash: