```vhdl
--
-- Work by : Chaari Mahdi and Daboussi Mariem , Engineering Students
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;


entity traffic_lights is
    Port ( clk : in  STD_LOGIC;
           cnf : in STD_logic;--Bouton Configuration
           rst : in  STD_LOGIC;--Bouton Reset

           --Les entrées de configuration
           switch_dip_main_timing : in std_logic_vector (7 downto 0);
           switch_dip_turning_left : in std_logic_vector (5 downto 0);
           switch_dip_orange_light : in std_logic_vector (3 downto 0);
           switch_dip_security_time : in std_logic_vector (1 downto 0);
           switch_dip_pedestrian_time : in std_logic_vector (5 downto 0);
           switch_dip_security_code : in std_logic_vector (7 downto 0);
           bouton_okay : in std_logic;

           --Les entrées et sorties du passage piéton
           Bouton_appel_pieton :in std_logic_vector (3 downto 0);--4 capteurs pour chaque passage
           lights_pieton : out std_logic_vector (3 downto 0);--4 feux verts pour chaque passage

           --Les feux
           --feux voie "UP"
           Up_light : out  STD_LOGIC_vector(2 downto 0);
           Up_left_light : out  STD_LOGIC_vector(2 downto 0);
           Up_right_light : out  STD_LOGIC;
           --feux voie "DOWN"
           Down_light : out  STD_LOGIC_vector(2 downto 0);
           Down_left_light : out  STD_LOGIC_vector(2 downto 0);
           Down_right_light : out  STD_LOGIC;
           --feux voie "LEFT"
           Left_light : out  STD_LOGIC_vector(2 downto 0);
           Left_left_light : out  STD_LOGIC_vector(2 downto 0);
           Left_right_light : out  STD_LOGIC;
           --feux voie "RIGHT"
           Right_light : out  STD_LOGIC_vector(2 downto 0);
           Right_left_light : out  STD_LOGIC_vector(2 downto 0);
           Right_right_light : out  STD_LOGIC);

end traffic_lights;

architecture Behavioral of traffic_lights is

   type state is (initial_state, state_vertical_1, state_vertical_2, state_vertical_3, state_vertical_4,
                  state_vertical_5, state_vertical_6, state_vertical_7, state_vertical_8,
                  state_horizontal_1, state_horizontal_2, state_horizontal_3, state_horizontal_4,
                  state_horizontal_5, state_horizontal_6, state_horizontal_7,
state_horizontal_8);
   type functioning is (day, night , configuration_process);

   --Les signals des états
   signal actual_functioning : functioning;
   signal actual_state , futur_state, previous_state : state;

   --Les signals des périodes de fonctionnement de chaque état
   signal main_timing: integer :=10;
   signal turning_left_time : integer :=5;
   signal orange_light_time : integer := 4;
   signal security_time : integer :=2;
   signal pedestrian_time : integer := 10;

begin

   --Le process du changement du type de fonctionnement du système
   process(clk,rst)
      begin
         if rst='1' then
            actual_functioning <= night;
            if cnf='1' then
               actual_functioning <= configuration_process;
            end if;
         else actual_functioning <= day; actual_state<=futur_state;
         end if;
      end process;


   --Le processus de configuration du système
   process(bouton_okay)
      begin
         if (rising_edge(bouton_okay) and actual_functioning= configuration_process) then
            if (switch_dip_security_code="10010110") then
               main_timing<= to_integer(unsigned(switch_dip_main_timing));
               turning_left_time<= to_integer(unsigned(switch_dip_turning_left));
               orange_light_time<= to_integer(unsigned(switch_dip_orange_light));
               security_time<= to_integer(unsigned(switch_dip_security_time));
               pedestrian_time<= to_integer(unsigned(switch_dip_pedestrian_time));
               end if;
            end if;
      end process;


   --Le process qui contrôle le fonctionnement des feux de tourner à droite qui sont tous orangés
   process(clk)
      variable counter : integer :=0;
      begin
         case actual_functioning is
         when day => --les feux clignotent
            if counter < 4 then
               Up_right_light <='1';
               Down_right_light <='1';
               Left_right_light <='1';
               Right_right_light <='1';
               counter:= counter + 1;
            else
               Up_right_light <='0';
               Down_right_light <='0';
               Left_right_light <='0';
               Right_right_light <='0';
               counter:= 0;
            end if;
         when others => --les feux sont à l'état on sans clignotement
               Up_right_light <='1';
               Down_right_light <='1';
               Left_right_light <='1';
               Right_right_light <='1';
         end case;
      end process;


   --Le process de control d'état des feux de direction directe et ceux de tourner à gauche
   process (clk)
      variable counter: integer :=0;
      variable counter_pieton: integer :=0;
      variable appel_pieton: boolean :=false;
      variable bouton_appel_pieton_memory : std_logic_vector (3 downto 0) :="0000";

      begin
         if actual_functioning = day then

            if bouton_appel_pieton/= "0000" then -- s'il y'a un appel piéton
               appel_pieton:= true;
               bouton_appel_pieton_memory:=bouton_appel_pieton; --mémoire pour les boutons poussoirs d'appel pitéton
               end if;

            if appel_pieton= false then --s'il n'y a pas d'appel piéton
            case actual_state is

               --état initial de sécurité où tous les feux sont rouges
               when initial_state =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> security_time then
                     counter :=0;
                     --l'état suivant dépend de l'état précedent
                     if previous_state = state_vertical_8 then futur_state<= state_horizontal_1;
                     else futur_state<= state_vertical_1;
                     end if;
                  end if;

               --La voie "Up" a ses feux de direction directe et de tourner à gauche VERT
               when state_vertical_1 =>
                  Up_light <= "001";
                  Up_left_light <= "001";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> turning_left_time then counter :=0; futur_state<= state_vertical_2;
                  end if;

               --La voie "Up" a son feu de direction directe "VERT" et de tourner à gauche "ORANGE"
               when state_vertical_2 =>
                  Up_light <= "001";
                  Up_left_light <= "010";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= state_vertical_3;
                  end if;

               --état de sécurité, La voie "Up" a son feu de direction directe "VERT"
               when state_vertical_3 =>
                  Up_light <= "001";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> security_time then counter :=0; futur_state<= state_vertical_4;
                  end if;

               --La voie "Up" et "Down" ont leurs feux de direction directe "VERT"
               when state_vertical_4 =>
                  Up_light <= "001";
                  Up_left_light <= "100";
                  Down_light <= "001";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> main_timing then counter :=0; futur_state<= state_vertical_5;
                  end if;

               --La voie "Down" a son feu de direction directe "VERT" , la voie "UP" a son feu de direction directe "ORANGE"
               when state_vertical_5 =>
                  Up_light <= "010";
                  Up_left_light <= "100";
                  Down_light <= "001";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= state_vertical_6;
                  end if;

               --état de sécurité, La voie "Down" a son feu de direction directe "VERT"
               when state_vertical_6 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "001";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> security_time then counter :=0; futur_state<= state_vertical_7;
                  end if;

               --La voie "Down" a ses feux de direction directe et de tourner à gauche VERT
               when state_vertical_7 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "001";
                  Down_left_light <= "001";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> turning_left_time then counter :=0; futur_state<= state_vertical_8;
                  end if;

               --La voie "Down" a ses feux de direction directe et de tourner à gauche ORANGE
               when state_vertical_8 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "010";
                  Down_left_light <= "010";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= initial_state; previous_state<= state_vertical_8;
                  end if;

               --La voie "Left" a ses feux de direction directe et de tourner à gauche VERT
               when state_horizontal_1 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "001";
                  Left_left_light <= "001";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> turning_left_time then counter :=0; futur_state<= state_horizontal_2;
                  end if;

               --La voie "Left" a son feu de direction directe "VERT" et de tourner à gauche "ORANGE"
               when state_horizontal_2 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "001";
                  Left_left_light <= "010";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= state_horizontal_3;
                  end if;

               --état de sécurité, La voie "Left" a son feu de direction directe "VERT"
               when state_horizontal_3 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "001";
                  Left_left_light <= "100";
                  Right_light <= "100";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> security_time then counter :=0; futur_state<= state_horizontal_4;
                  end if;

               --La voie "Left"  et "Right" ont leurs feux de direction directe "VERT"
               when state_horizontal_4 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "001";
                  Left_left_light <= "100";
                  Right_light <= "001";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> main_timing then counter :=0; futur_state<= state_horizontal_5;
                  end if;

               --La voie "Right" a son feu de direction directe "VERT" , la voie "Left" a son feu de direction directe "ORANGE"
               when state_horizontal_5 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "010";
                  Left_left_light <= "100";
                  Right_light <= "001";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= state_horizontal_6;
                  end if;

               --état de sécurité, La voie "Right" a son feu de direction directe "VERT"
               when state_horizontal_6 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "001";
                  Right_left_light <= "100";
                  counter :=counter+1;
                  if counter> security_time then counter :=0; futur_state<= state_horizontal_7;
                  end if;

               --La voie "Right" a ses feux de direction directe et de tourner à gauche VERT
               when state_horizontal_7 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "001";
                  Right_left_light <= "001";
                  counter :=counter+1;
                  if counter> turning_left_time then counter :=0; futur_state<= state_horizontal_8;
                  end if;

               --La voie "Right" a ses feux de direction directe et de tourner à gauche ORANGE
               when state_horizontal_8 =>
                  Up_light <= "100";
                  Up_left_light <= "100";
                  Down_light <= "100";
                  Down_left_light <= "100";
                  Left_light <= "100";
                  Left_left_light <= "100";
                  Right_light <= "010";
                  Right_left_light <= "010";
                  counter :=counter+1;
                  if counter> orange_light_time then counter :=0; futur_state<= initial_state; previous_state<=state_horizontal_8;
                  end if;

               when others =>
            end case;

            else -- On a un appel piéton

               case bouton_appel_pieton is
                  --lorsque seul le capteur du voie horizontale est actionné de la côté droite
                  when "0001" =>
                        Down_left_light <= "100";
                        Left_light <= "100";
                        Left_left_light <= "100";
                        Right_light <= "100";
                  --lorsque seul le capteur du voie horizontale est actionné de la côté gauche
                  when "0010" =>
                        Up_left_light <= "100";
                        Left_light <= "100";
                        Right_light <= "100";
                        Right_left_light <= "100";
                  --lorsque les deux capteurs de la voie horizontale sont actionnés
                  when "0011" =>
                        Up_left_light <= "100";
                        Down_left_light <= "100";
                        Left_light <= "100";
                        Left_left_light <= "100";
                        Right_light <= "100";
                        Right_left_light <= "100";
                  --lorsque seul le capteur du voie verticale est actionné du haut
                  when "0100" =>
                        Up_light <= "100";
                        Down_light <= "100";
                        Down_left_light <= "100";
```

```vhdl
                        Right_left_light <= "100";
                  --lorsque seul le capteur du voie verticale est actionné du bas
                  when "1000" =>
                        Up_light <= "100";
                        Up_left_light <= "100";
                        Down_light <= "100";
                        Left_left_light <= "100";
                  --lorsque les deux capteurs de la voie verticale sont actionnés
                  when "1100" =>
                        Up_light <= "100";
                        Up_left_light <= "100";
                        Down_light <= "100";
                        Down_left_light <= "100";
                        Left_left_light <= "100";
                        Right_left_light <= "100";
                  when others =>
                        Up_light <= "100";
                        Up_left_light <= "100";
                        Down_light <= "100";
                        Down_left_light <= "100";
                        Left_light <= "100";
                        Left_left_light <= "100";
                        Right_light <= "100";
                        Right_left_light <= "100";
                  end case;
                  counter_pieton:= counter_pieton+1;
                  if counter_pieton>security_time and counter_pieton<=security_time+pedestrian_time then
                     -- Les feux correspondants à l'appel piéton s'allume après un durée de sécurité
                     lights_pieton<=bouton_appel_pieton_memory;
                  else
                     lights_pieton<="0000"; appel_pieton:= false;
                  end if;

            end if;

         else
            lights_pieton<="0000";
            Up_light <="010";
            Up_left_light <="010";
            Down_light <="010";
            Down_left_light<="010";
            Left_light <="010";
            Left_left_light <="010";
            Right_light <="010";
            Right_left_light <="010";
            futur_state<= initial_state;
            counter :=0;
         end if;
      end process;
end Behavioral;
```