

```
1  --
2  -- Work by : Chaari Mahdi and Daboussi Mariem , Engineering Students
3  --
4  library IEEE;
5  use IEEE.STD_LOGIC_1164.ALL;
6  use ieee.numeric_std.all;
7
8
9  entity traffic_lights is
10     Port ( clk : in  STD_LOGIC;
11           cnf : in  STD_LOGIC;--Bouton Configuration
12           rst : in  STD_LOGIC;--Bouton Reset
13
14           --Les entrées de configuration
15           switch_dip_main_timing : in std_logic_vector (7 downto 0);
16           switch_dip_turning_left : in std_logic_vector (5 downto 0);
17           switch_dip_orange_light : in std_logic_vector (3 downto 0);
18           switch_dip_security_time : in std_logic_vector (1 downto 0);
19           switch_dip_pedestrian_time : in std_logic_vector (5 downto 0);
20           switch_dip_security_code : in std_logic_vector (7 downto 0);
21           bouton_okay : in std_logic;
22
23           --Les entrées et sorties du passage piéton
24           Bouton_appel_pieton :in std_logic_vector (3 downto 0);--4 capteurs pour
chaque passage
25           lights_pieton : out std_logic_vector (3 downto 0);--4 feux verts pour
chaque passage
26
27           --Les feux
28           --feux voie "UP"
29           Up_light : out  STD_LOGIC_vector(2 downto 0);
30           Up_left_light : out  STD_LOGIC_vector(2 downto 0);
31           Up_right_light : out  STD_LOGIC;
32           --feux voie "DOWN"
33           Down_light : out  STD_LOGIC_vector(2 downto 0);
34           Down_left_light : out  STD_LOGIC_vector(2 downto 0);
35           Down_right_light : out  STD_LOGIC;
36           --feux voie "LEFT"
37           Left_light : out  STD_LOGIC_vector(2 downto 0);
38           Left_left_light : out  STD_LOGIC_vector(2 downto 0);
39           Left_right_light : out  STD_LOGIC;
40           --feux voie "RIGHT"
41           Right_light : out  STD_LOGIC_vector(2 downto 0);
42           Right_left_light : out  STD_LOGIC_vector(2 downto 0);
43           Right_right_light : out  STD_LOGIC);
44
45  end traffic_lights;
46
47  architecture Behavioral of traffic_lights is
48
49     type state is (initial_state, state_vertical_1, state_vertical_2, state_vertical_3,
state_vertical_4,
50 state_vertical_5, state_vertical_6, state_vertical_7,
state_vertical_8,
51 state_horizontal_1, state_horizontal_2, state_horizontal_3,
state_horizontal_4,
52 state_horizontal_5, state_horizontal_6, state_horizontal_7,
```

```
state_horizontal_8);
53     type functioning is (day, night , configuration_process);
54
55     --Les signals des états
56     signal actual_functioning : functioning;
57     signal actual_state , futur_state, previous_state : state;
58
59     --Les signals des périodes de fonctionnement de chaque état
60     signal main_timing: integer :=10;
61     signal turning_left_time : integer :=5;
62     signal orange_light_time : integer := 4;
63     signal security_time : integer :=2;
64     signal pedestrian_time : integer := 10;
65
66 begin
67
68     --Le process du changement du type de fonctionnement du système
69     process(clk,rst)
70     begin
71         if rst='1' then
72             actual_functioning <= night;
73             if cnf='1' then
74                 actual_functioning <= configuration_process;
75             end if;
76         else actual_functioning <= day; actual_state<=futur_state;
77         end if;
78     end process;
79
80
81     --Le processus de configuration du système
82     process(bouton_okay)
83     begin
84         if (rising_edge(bouton_okay) and actual_functioning= configuration_process)
85     then
86         if (switch_dip_security_code="10010110") then
87             main_timing<= to_integer(unsigned(switch_dip_main_timing));
88             turning_left_time<= to_integer(unsigned(switch_dip_turning_left));
89             orange_light_time<= to_integer(unsigned(switch_dip_orange_light));
90             security_time<= to_integer(unsigned(switch_dip_security_time));
91             pedestrian_time<= to_integer(unsigned(switch_dip_pedestrian_time));
92         end if;
93     end if;
94     end process;
95
96     --Le process qui contrôle le fonctionnement des feux de tourner à droite qui sont
97     tous orangés
98     process(clk)
99     variable counter : integer :=0;
100    begin
101        case actual_functioning is
102        when day => --les feux clignotent
103            if counter < 4 then
104                Up_right_light <='1';
105                Down_right_light <='1';
106                Left_right_light <='1';
107                Right_right_light <='1';
```

```
107         counter:= counter + 1;
108     else
109         Up_right_light <='0';
110         Down_right_light <='0';
111         Left_right_light <='0';
112         Right_right_light <='0';
113         counter:= 0;
114     end if;
115     when others => --les feux sont à l'état on sans clignotement
116         Up_right_light <='1';
117         Down_right_light <='1';
118         Left_right_light <='1';
119         Right_right_light <='1';
120     end case;
121 end process;
122
123
124 --Le process de control d'état des feux de direction directe et ceux de tourner à
gauche
125 process (clk)
126     variable counter: integer :=0;
127     variable counter_pieton: integer :=0;
128     variable appel_pieton: boolean :=false;
129     variable bouton_appel_pieton_memory : std_logic_vector (3 downto 0) :="0000";
130
131     begin
132         if actual_functioning = day then
133
134             if bouton_appel_pieton/= "0000" then -- s'il y'a un appel piéton
135                 appel_pieton:= true;
136                 bouton_appel_pieton_memory:=bouton_appel_pieton; --mémoire pour les
boutons poussoirs d'appel piéton
137             end if;
138
139             if appel_pieton= false then --s'il n'y a pas d'appel piéton
140                 case actual_state is
141
142                     --état initial de sécurité où tous les feux sont rouges
143                     when initial_state =>
144                         Up_light <= "100";
145                         Up_left_light <= "100";
146                         Down_light <= "100";
147                         Down_left_light <= "100";
148                         Left_light <= "100";
149                         Left_left_light <= "100";
150                         Right_light <= "100";
151                         Right_left_light <= "100";
152                         counter :=counter+1;
153                         if counter> security_time then
154                             counter :=0;
155                             --l'état suivant dépend de l'état précédent
156                             if previous_state = state_vertical_8 then futur_state<=
state_horizontal_1;
157                             else futur_state<= state_vertical_1;
158                             end if;
159                         end if;
160
```

```
161      --La voie "Up" a ses feux de direction directe et de tourner à gauche
      VERT
162      when state_vertical_1 =>
163          Up_light <= "001";
164          Up_left_light <= "001";
165          Down_light <= "100";
166          Down_left_light <= "100";
167          Left_light <= "100";
168          Left_left_light <= "100";
169          Right_light <= "100";
170          Right_left_light <= "100";
171          counter :=counter+1;
172          if counter> turning_left_time then counter :=0; futur_state<=
state_vertical_2;
173      end if;
174
175      --La voie "Up" a son feu de direction directe "VERT" et de tourner à
gauche "ORANGE"
176      when state_vertical_2 =>
177          Up_light <= "001";
178          Up_left_light <= "010";
179          Down_light <= "100";
180          Down_left_light <= "100";
181          Left_light <= "100";
182          Left_left_light <= "100";
183          Right_light <= "100";
184          Right_left_light <= "100";
185          counter :=counter+1;
186          if counter> orange_light_time then counter :=0; futur_state<=
state_vertical_3;
187      end if;
188
189      --état de sécurité, La voie "Up" a son feu de direction directe "VERT"
190      when state_vertical_3 =>
191          Up_light <= "001";
192          Up_left_light <= "100";
193          Down_light <= "100";
194          Down_left_light <= "100";
195          Left_light <= "100";
196          Left_left_light <= "100";
197          Right_light <= "100";
198          Right_left_light <= "100";
199          counter :=counter+1;
200          if counter> security_time then counter :=0; futur_state<=
state_vertical_4;
201      end if;
202
203      --La voie "Up" et "Down" ont leurs feux de direction directe "VERT"
204      when state_vertical_4 =>
205          Up_light <= "001";
206          Up_left_light <= "100";
207          Down_light <= "001";
208          Down_left_light <= "100";
209          Left_light <= "100";
210          Left_left_light <= "100";
211          Right_light <= "100";
212          Right_left_light <= "100";
```

```
213         counter :=counter+1;
214         if counter> main_timing then counter :=0; futur_state<=
state_vertical_5;
215         end if;
216
217         --La voie "Down" a son feu de direction directe "VERT" , la voie "UP"
a son feu de direction directe "ORANGE"
218         when state_vertical_5 =>
219             Up_light <= "010";
220             Up_left_light <= "100";
221             Down_light <= "001";
222             Down_left_light <= "100";
223             Left_light <= "100";
224             Left_left_light <= "100";
225             Right_light <= "100";
226             Right_left_light <= "100";
227             counter :=counter+1;
228             if counter> orange_light_time then counter :=0; futur_state<=
state_vertical_6;
229             end if;
230
231             --état de sécurité, La voie "Down" a son feu de direction directe "VERT"
232             when state_vertical_6 =>
233                 Up_light <= "100";
234                 Up_left_light <= "100";
235                 Down_light <= "001";
236                 Down_left_light <= "100";
237                 Left_light <= "100";
238                 Left_left_light <= "100";
239                 Right_light <= "100";
240                 Right_left_light <= "100";
241                 counter :=counter+1;
242                 if counter> security_time then counter :=0; futur_state<=
state_vertical_7;
243                 end if;
244
245                 --La voie "Down" a ses feux de direction directe et de tourner à
gauche VERT
246                 when state_vertical_7 =>
247                     Up_light <= "100";
248                     Up_left_light <= "100";
249                     Down_light <= "001";
250                     Down_left_light <= "001";
251                     Left_light <= "100";
252                     Left_left_light <= "100";
253                     Right_light <= "100";
254                     Right_left_light <= "100";
255                     counter :=counter+1;
256                     if counter> turning_left_time then counter :=0; futur_state<=
state_vertical_8;
257                     end if;
258
259                     --La voie "Down" a ses feux de direction directe et de tourner à
gauche ORANGE
260                     when state_vertical_8 =>
261                         Up_light <= "100";
262                         Up_left_light <= "100";
```

```
263         Down_light <= "010";
264         Down_left_light <= "010";
265         Left_light <= "100";
266         Left_left_light <= "100";
267         Right_light <= "100";
268         Right_left_light <= "100";
269         counter :=counter+1;
270         if counter> orange_light_time then counter :=0; futur_state<=
initial_state; previous_state<= state_vertical_8;
271         end if;
272
273         --La voie "Left" a ses feux de direction directe et de tourner à
gauche VERT
274         when state_horizontal_1 =>
275             Up_light <= "100";
276             Up_left_light <= "100";
277             Down_light <= "100";
278             Down_left_light <= "100";
279             Left_light <= "001";
280             Left_left_light <= "001";
281             Right_light <= "100";
282             Right_left_light <= "100";
283             counter :=counter+1;
284             if counter> turning_left_time then counter :=0; futur_state<=
state_horizontal_2;
285             end if;
286
287         --La voie "Left" a son feu de direction directe "VERT" et de tourner à
gauche "ORANGE"
288         when state_horizontal_2 =>
289             Up_light <= "100";
290             Up_left_light <= "100";
291             Down_light <= "100";
292             Down_left_light <= "100";
293             Left_light <= "001";
294             Left_left_light <= "010";
295             Right_light <= "100";
296             Right_left_light <= "100";
297             counter :=counter+1;
298             if counter> orange_light_time then counter :=0; futur_state<=
state_horizontal_3;
299             end if;
300
301         --état de sécurité, La voie "Left" a son feu de direction directe
"VERT"
302         when state_horizontal_3 =>
303             Up_light <= "100";
304             Up_left_light <= "100";
305             Down_light <= "100";
306             Down_left_light <= "100";
307             Left_light <= "001";
308             Left_left_light <= "100";
309             Right_light <= "100";
310             Right_left_light <= "100";
311             counter :=counter+1;
312             if counter> security_time then counter :=0; futur_state<=
state_horizontal_4;
```

```
313         end if;
314
315         --La voie "Left" et "Right" ont leurs feux de direction directe "VERT"
316         when state_horizontal_4 =>
317             Up_light <= "100";
318             Up_left_light <= "100";
319             Down_light <= "100";
320             Down_left_light <= "100";
321             Left_light <= "001";
322             Left_left_light <= "100";
323             Right_light <= "001";
324             Right_left_light <= "100";
325             counter :=counter+1;
326             if counter> main_timing then counter :=0; futur_state<=
state_horizontal_5;
327         end if;
328
329         --La voie "Right" a son feu de direction directe "VERT" , la voie
"Left" a son feu de direction directe "ORANGE"
330         when state_horizontal_5 =>
331             Up_light <= "100";
332             Up_left_light <= "100";
333             Down_light <= "100";
334             Down_left_light <= "100";
335             Left_light <= "010";
336             Left_left_light <= "100";
337             Right_light <= "001";
338             Right_left_light <= "100";
339             counter :=counter+1;
340             if counter> orange_light_time then counter :=0; futur_state<=
state_horizontal_6;
341         end if;
342
343         --état de sécurité, La voie "Right" a son feu de direction directe
"VERT"
344         when state_horizontal_6 =>
345             Up_light <= "100";
346             Up_left_light <= "100";
347             Down_light <= "100";
348             Down_left_light <= "100";
349             Left_light <= "100";
350             Left_left_light <= "100";
351             Right_light <= "001";
352             Right_left_light <= "100";
353             counter :=counter+1;
354             if counter> security_time then counter :=0; futur_state<=
state_horizontal_7;
355         end if;
356
357         --La voie "Right" a ses feux de direction directe et de tourner à
gauche VERT
358         when state_horizontal_7 =>
359             Up_light <= "100";
360             Up_left_light <= "100";
361             Down_light <= "100";
362             Down_left_light <= "100";
363             Left_light <= "100";
```

```
364         Left_left_light <= "100";
365         Right_light <= "001";
366         Right_left_light <= "001";
367         counter :=counter+1;
368         if counter> turning_left_time then counter :=0; futur_state<=
state_horizontal_8;
369         end if;
370
371         --La voie "Right" a ses feux de direction directe et de tourner à
gauche ORANGE
372         when state_horizontal_8 =>
373             Up_light <= "100";
374             Up_left_light <= "100";
375             Down_light <= "100";
376             Down_left_light <= "100";
377             Left_light <= "100";
378             Left_left_light <= "100";
379             Right_light <= "010";
380             Right_left_light <= "010";
381             counter :=counter+1;
382             if counter> orange_light_time then counter :=0; futur_state<=
initial_state; previous_state<=state_horizontal_8;
383             end if;
384
385             when others =>
386         end case;
387
388         else -- On a un appel piéton
389
390             case bouton_appel_pieton is
391                 --lorsque seul le capteur du voie horizontale est actionné de la
côté droite
392                 when "0001" =>
393                     Down_left_light <= "100";
394                     Left_light <= "100";
395                     Left_left_light <= "100";
396                     Right_light <= "100";
397                 --lorsque seul le capteur du voie horizontale est actionné de la
côté gauche
398                 when "0010" =>
399                     Up_left_light <= "100";
400                     Left_light <= "100";
401                     Right_light <= "100";
402                     Right_left_light <= "100";
403                 --lorsque les deux capteurs de la voie horizontale sont actionnés
404                 when "0011" =>
405                     Up_left_light <= "100";
406                     Down_left_light <= "100";
407                     Left_light <= "100";
408                     Left_left_light <= "100";
409                     Right_light <= "100";
410                     Right_left_light <= "100";
411                 --lorsque seul le capteur du voie verticale est actionné du haut
412                 when "0100" =>
413                     Up_light <= "100";
414                     Down_light <= "100";
415                     Down_left_light <= "100";
```



```
416         Right_left_light <= "100";
417     --lorsque seul le capteur du voie verticale est actionné du bas
418     when "1000" =>
419         Up_light <= "100";
420         Up_left_light <= "100";
421         Down_light <= "100";
422         Left_left_light <= "100";
423     --lorsque les deux capteurs de la voie verticale sont actionnés
424     when "1100" =>
425         Up_light <= "100";
426         Up_left_light <= "100";
427         Down_light <= "100";
428         Down_left_light <= "100";
429         Left_left_light <= "100";
430         Right_left_light <= "100";
431     when others =>
432         Up_light <= "100";
433         Up_left_light <= "100";
434         Down_light <= "100";
435         Down_left_light <= "100";
436         Left_light <= "100";
437         Left_left_light <= "100";
438         Right_light <= "100";
439         Right_left_light <= "100";
440     end case;
441     counter_pieton:= counter_pieton+1;
442     if counter_pieton>security_time and counter_pieton<=security_time+
pedestrian_time then
443         -- Les feux correspondants à l'appel piéton s'allume après un
durée de sécurité
444         lights_pieton<=bouton_appel_pieton_memory;
445     else
446         lights_pieton<="0000"; appel_pieton:= false;
447     end if;
448
449     end if;
450
451     else
452         lights_pieton<="0000";
453         Up_light <="010";
454         Up_left_light <="010";
455         Down_light <="010";
456         Down_left_light<="010";
457         Left_light <="010";
458         Left_left_light <="010";
459         Right_light <="010";
460         Right_left_light <="010";
461         futur_state<= initial_state;
462         counter :=0;
463     end if;
464 end process;
465 end Behavioral;
466
467
```