LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.NUMERIC\_STD.ALL;

ENTITY TRAFIC\_LIGHT IS

PORT ( CLK : IN STD\_LOGIC;

RST : IN STD\_LOGIC;

START : IN STD\_LOGIC;

SEG\_0 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_1 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_2 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_3 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_4 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_5 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_6 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

SEG\_7 : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0)

);

END TRAFIC\_LIGHT;

ARCHITECTURE BEHAVE OF TRAFIC\_LIGHT IS

TYPE STATE IS (INIT, S0, S1, S2, S3);

SIGNAL CURRENT\_STATE : STATE := INIT;

SIGNAL CYCLE\_CNT : STD\_LOGIC\_VECTOR(31 DOWNTO 0) := X"00000000";

CONSTANT ONE\_SEC\_CST : STD\_LOGIC\_VECTOR := X"05F5E100";

CONSTANT FOUR\_SEC\_CST : STD\_LOGIC\_VECTOR := X"17D78400";

BEGIN

SEQ\_PROC: PROCESS(CLK, RST, START)

BEGIN

IF(RST = '1') THEN

CURRENT\_STATE <= INIT;

ELSIF(RISING\_EDGE(CLK)) THEN

CASE CURRENT\_STATE IS

WHEN S0=>

--Feu1=Vert

SEG\_0 <= "00000001";

SEG\_1 <= "00000011";

SEG\_2 <= "00000011";

SEG\_3 <= "00000011";

--Feu2=Rouge

SEG\_4 <= "00000011";

SEG\_5 <= "00000011";

SEG\_6 <= "00000000";

SEG\_7 <= "00000011";

-- On décrémente le compteur

CYCLE\_CNT <= STD\_LOGIC\_VECTOR(UNSIGNED(CYCLE\_CNT) - 1);

-- Si le compteur est arrivé a zero, il faut changer d'état

IF(CYCLE\_CNT = X"00000000") THEN

-- On set le temps a passer dans le prochain etat et on

-- change l'etat courrant

CURRENT\_STATE <= S1;

CYCLE\_CNT <= ONE\_SEC\_CST;

END IF;

WHEN S1=>

--Feu1=Orange

SEG\_0 <= "00000011";

SEG\_1 <= "00000010";

SEG\_2 <= "00000011";

SEG\_3 <= "00000011";

--Feu2=Rouge

SEG\_4 <= "00000011";

SEG\_5 <= "00000011";

SEG\_6 <= "00000000";

SEG\_7 <= "00000011";

-- On décrémente le compteur

CYCLE\_CNT <= STD\_LOGIC\_VECTOR(UNSIGNED(CYCLE\_CNT) - 1);

-- Si le compteur est arrivé a zero, il faut changer d'état

IF(CYCLE\_CNT = X"00000000") THEN

-- On set le temps a passer dans le prochain etat et on

-- change l'etat courrant

CURRENT\_STATE <= S2;

CYCLE\_CNT <= FOUR\_SEC\_CST;

END IF;

WHEN S2=>

--Feu1=Rouge

SEG\_0 <= "00000011";

SEG\_1 <= "00000011";

SEG\_2 <= "00000000";

SEG\_3 <= "00000011";

--Feu2=Vert

SEG\_4 <= "00000001";

SEG\_5 <= "00000011";

SEG\_6 <= "00000011";

SEG\_7 <= "00000011";

-- On décrémente le compteur

CYCLE\_CNT <= STD\_LOGIC\_VECTOR(UNSIGNED(CYCLE\_CNT) - 1);

-- Si le compteur est arrivé a zero, il faut changer d'état

IF(CYCLE\_CNT = X"00000000") THEN

-- On set le temps a passer dans le prochain etat et on

-- change l'etat courrant

CURRENT\_STATE <= S3;

CYCLE\_CNT <= ONE\_SEC\_CST;

END IF;

WHEN S3=>

--Feu1=Rouge

SEG\_0 <= "00000011";

SEG\_1 <= "00000011";

SEG\_2 <= "00000000";

SEG\_3 <= "00000011";

--Feu2=Orange

SEG\_4 <= "00000011";

SEG\_5 <= "00000010";

SEG\_6 <= "00000011";

SEG\_7 <= "00000011";

-- On décrémente le compteur

CYCLE\_CNT <= STD\_LOGIC\_VECTOR(UNSIGNED(CYCLE\_CNT) - 1);

-- Si le compteur est arrivé a zero, il faut changer d'état

IF(CYCLE\_CNT = X"00000000") THEN

-- On set le temps a passer dans le prochain etat et on

-- change l'etat courrant

CURRENT\_STATE <= S0;

CYCLE\_CNT <= FOUR\_SEC\_CST;

END IF;

WHEN INIT =>

--Feu1=----

SEG\_0 <= "00000011";

SEG\_1 <= "00000011";

SEG\_2 <= "00000011";

SEG\_3 <= "00000011";

--Feu2=----

SEG\_4 <= "00000011";

SEG\_5 <= "00000011";

SEG\_6 <= "00000011";

SEG\_7 <= "00000011";

IF(START = '1') THEN

CURRENT\_STATE <= S0;

CYCLE\_CNT <= FOUR\_SEC\_CST;

END IF;

END CASE;

END IF;

END PROCESS SEQ\_PROC;

END BEHAVE;