library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

entity key\_light is

port(clk: in std\_logic;

rst: in std\_logic;

key: in std\_logic;

sw1\_led, sw2\_led, sw3\_led, sw4\_led: out std\_logic);

end entity;

architecture behav of key\_light is

signal key\_rst: std\_logic;

signal key\_rst\_an: std\_logic;

signal key\_rst\_r: std\_logic;

signal low\_sw: std\_logic;

signal low\_sw\_an: std\_logic;

signal low\_sw\_r: std\_logic;

signal cnt: integer range 10 downto 0;

begin

process (clk, rst, key)

type state\_type is(S0, S1, S2, S3, S4);

variable present\_state: state\_type;

variable key\_available: std\_logic;

variable turn\_on : integer range 1 downto 0;

begin

if rst='1' then

present\_state:=S0;

key\_available := '0';

sw1\_led <= '1';

sw2\_led <= '1';

sw3\_led <= '1';

sw4\_led <= '1';

turn\_on := 0;

cnt <= 0;

elsif rising\_edge(clk) then

key\_rst <= key;

key\_rst\_an <= (key\_rst and key\_rst\_r) or (not key\_rst\_r and not key\_rst);

if key\_rst\_an = '1' then

cnt <= cnt + 1;

else

cnt <= 0;

end if;

key\_rst\_r <= key\_rst;

if cnt = 3 then

key\_available := key;

cnt <= 0;

else

null;

end if;

case present\_state is

when S0 =>

sw1\_led <= '1';

sw2\_led <= '1';

sw3\_led <= '1';

sw4\_led <= '1';

if (key\_available = '1' and turn\_on = 0) then

turn\_on := 1;

present\_state := S1;

elsif key\_available = '1' and turn\_on = 1 then

present\_state := S0;

elsif key\_available = '0' and turn\_on = 1 then

turn\_on := 0;

else

null;

end if;

when S1 =>

sw1\_led <= '0';

sw2\_led <= '1';

sw3\_led <= '1';

sw4\_led <= '1';

if key\_available = '1' and turn\_on = 0 then

turn\_on := 1;

present\_state := S2;

elsif key\_available = '1' and turn\_on = 1 then

present\_state := S1;

elsif key\_available = '0' and turn\_on = 1 then

present\_state := S1;

turn\_on := 0;

else

null;

end if;

when S2 =>

sw1\_led <= '1';

sw2\_led <= '0';

sw3\_led <= '1';

sw4\_led <= '1';

if key\_available = '1' and turn\_on = 0 then

turn\_on := 1;

present\_state := S3;

elsif key\_available = '1' and turn\_on = 1 then

present\_state := S2;

elsif key\_available = '0' and turn\_on = 1 then

present\_state := S2;

turn\_on := 0;

else

null;

end if;

when S3 =>

sw1\_led <= '1';

sw2\_led <= '1';

sw3\_led <= '0';

sw4\_led <= '1';

if (key\_available = '1' and turn\_on = 0) then

turn\_on := 1;

present\_state := S4;

elsif key\_available = '1' and turn\_on = 1 then

present\_state := S3;

elsif key\_available = '0' and turn\_on = 1 then

present\_state := S3;

turn\_on := 0;

end if;

when S4 =>

sw1\_led <= '1';

sw2\_led <= '1';

sw3\_led <= '1';

sw4\_led <= '0';

if key\_available = '1' and turn\_on = 0 then

turn\_on := 1;

present\_state := S1;

elsif key\_available = '1' and turn\_on = 1 then

present\_state := S4;

elsif key\_available = '0' and turn\_on = 1 then

present\_state := S4;

turn\_on := 0;

end if;

end case;

end if;

end process;

end behav;