ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

--ଓ📖ଓ--

BÁO CÁO

BÀI TẬP LỚN

THÍ NGHIỆM – KỸ THUẬT SỐ

GVHD : Nguyễn Phan Hải Phú

SV thực hiện: Cao Văn Hiếu_2010251

Lớp : L12

Tp.HCM, Tháng 12 năm 2021

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ YÊU CẦU

1.1. Yêu cầu đặt ra

Thiết kế bộ đèn giao thông bằng các IC và cổng đã học trong học phần Thí nghiệm kỹ thuật số. Thời gian hiển thị như sau:

Mã số ID	Họ	Tên	Xanh	Vàng	Đỏ
2010251	Cao Văn	Hiếu	1	5	1

Sử dụng: Proteus (bắt buộc) + Quartus (tuỳ chọn)

1.2. Phân tích

Mạch đèn giao thông dùng IC số mô tả lại cách hoạt động với bảng hiển thị thời gian, 3 hiệu lệnh (xanh, đỏ, vàng). Theo yêu cầu đề bài, chúng ta chỉ cần bảng hiển thị 1 chữ số.

Do sử dụng các linh kiện chính trong mạch là IC số nên mạch chỉ mô tả được một chế độ duy nhất (không thể thay đổi thời gian hiển thị sau khi thiết đặt)

1.3. Phương pháp lựa chọn

Sinh viên sẽ thực hiện mô phỏng bằng cả Proteus và Quartus

1.4. Kết cấu bài làm

- Linh kiện sử dụng

- Thiết kế mạch

- Mô phỏng kết quả

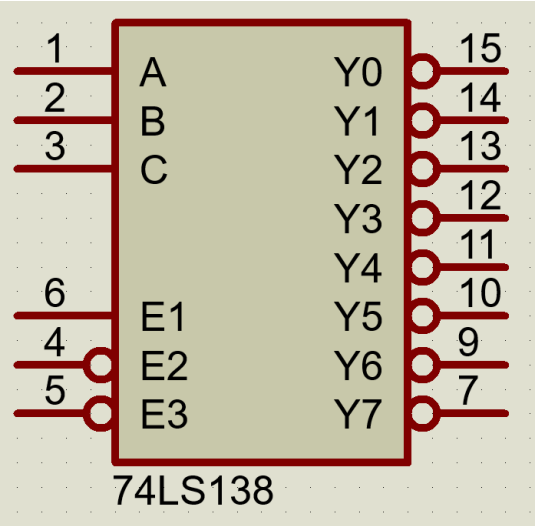
- Nhận xét.

CHƯƠNG 2. THIẾT KẾ SỬ DỤNG PROTEUS

2.1. Linh kiện sử dụng

2.1.1. IC 74LS138

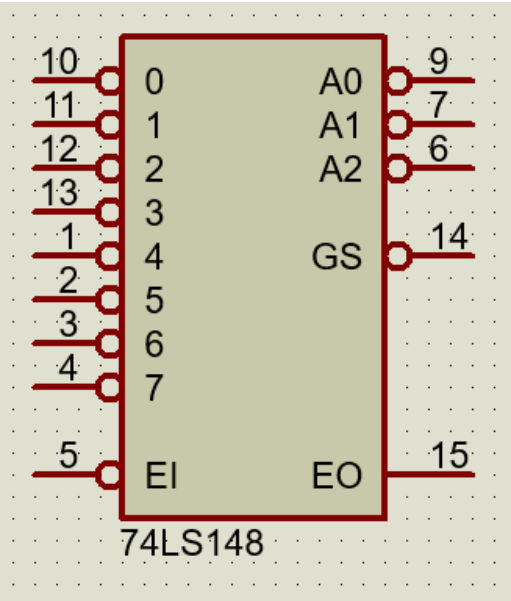
Con chip này được thiết kế để giải mã hoặc khử ghép và đi kèm với thiết lập 3 đầu vào đến 8 đầu ra.



Trong bài này, nó sẽ được sử dụng trong khối trạng thái với mục đích giải mã trạng thái.

2.1.2. IC 74LS148

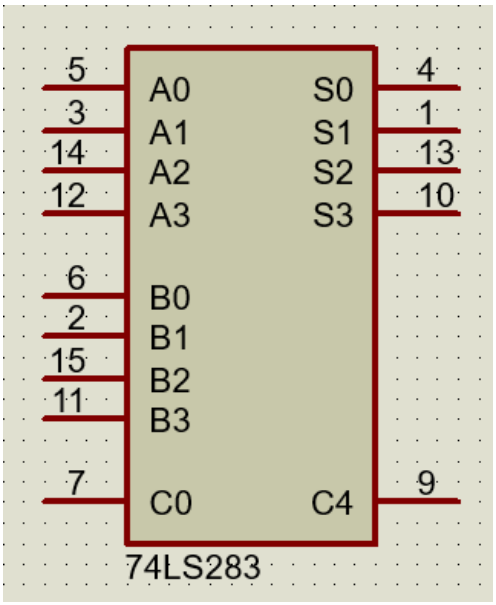
74LS148 là IC có điện áp làm việc đa dạng, nhiều điều kiện làm việc và giao tiếp trực tiếp với CMOS, NMOS và TTL. Nó có tính năng giải mã ưu tiên các đầu vào để đảm bảo chỉ dòng dữ liệu bậc cao nhất mới được mã hóa.



Trong bài này nó sẽ được sử dụng trong khối trạng thái để lựa chọn thời gian hiển thị.

2.1.3. IC 74LS283

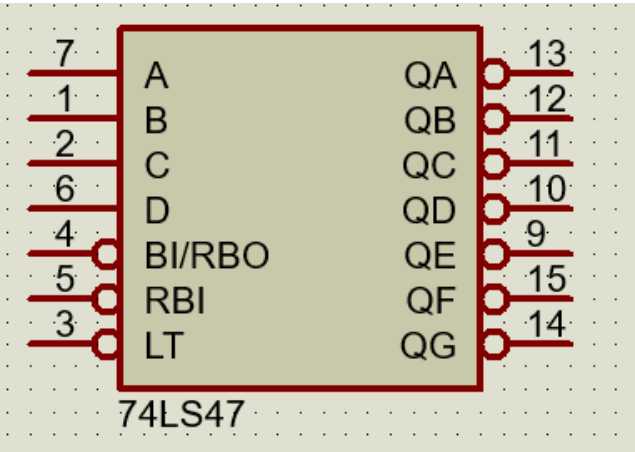
74LS283 là IC làm toán được sử dụng với mục đích cộng trừ 2 số nhị phân 4 bit.



Trong bài này, nó sẽ được sử dụng trong khối đếm với mục đích chuyển đếm lên thành đếm xuống.

2.1.4. IC 74LS47

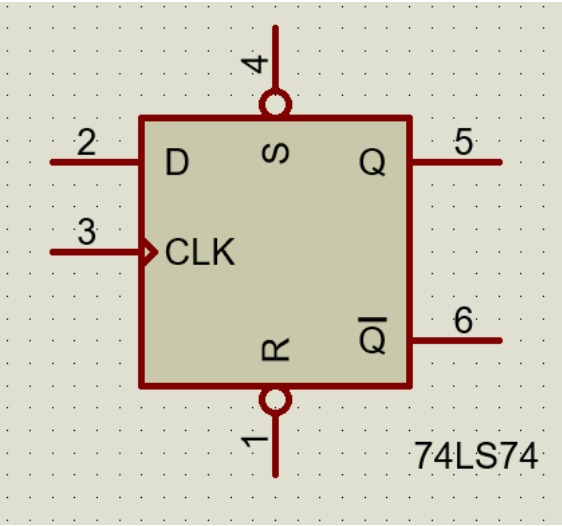
74LS47 là IC điều khiển / giải mã BCD sang 7 đoạn. Nó chấp nhận một số thập phân được mã hóa nhị phân làm đầu vào và chuyển đổi nó thành một mẫu để điều khiển 7 đoạn để hiển thị các chữ số từ 0 đến 9.



Trong bài này, nó sẽ được sử dụng trong khối hiển thị với mục đích hiện thời gian.

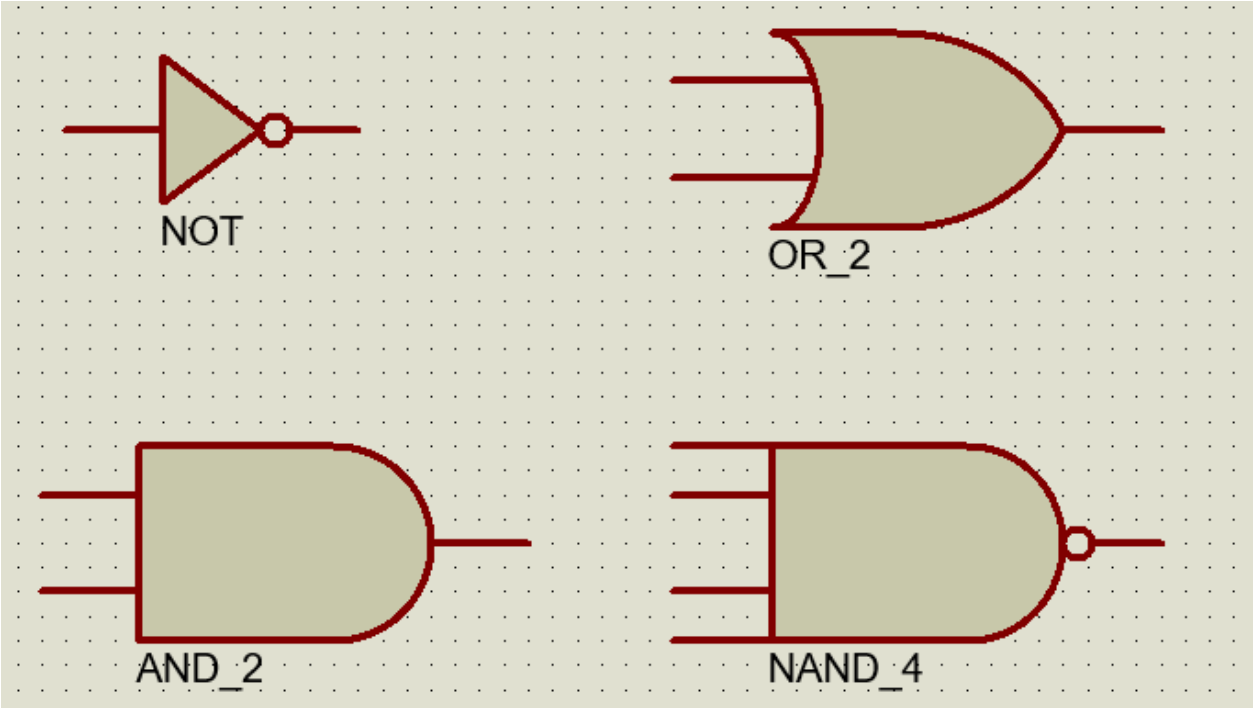
2.1.5. IC 74LS74

IC flip-flop 74LS74 là IC sử dụng mạch Schottky TTL để tạo ra flip-flop loại D tốc độ cao. Mỗi flip-flop có các đầu vào rõ ràng và thiết lập riêng lẻ, cũng như các đầu ra Q và Q` bổ sung.



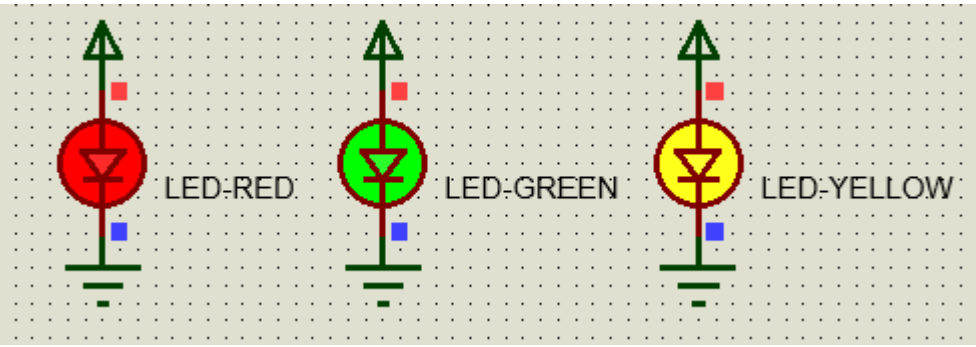
Trong bài này, nó sẽ được sử dụng trong khối trạng thái với mục đích chuyển trạng thái và khối đếm.

2.1.6. Cổng Logic

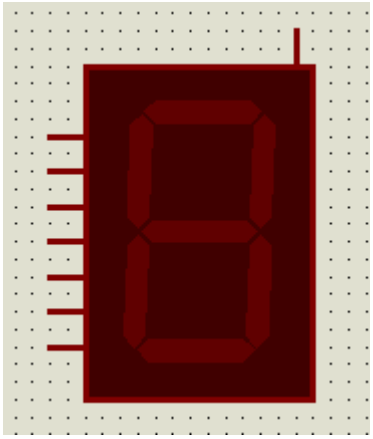


2.1.7. Các linh kiện khác

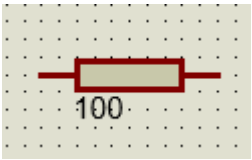
a. LED màu



b. LED 7 đoạn

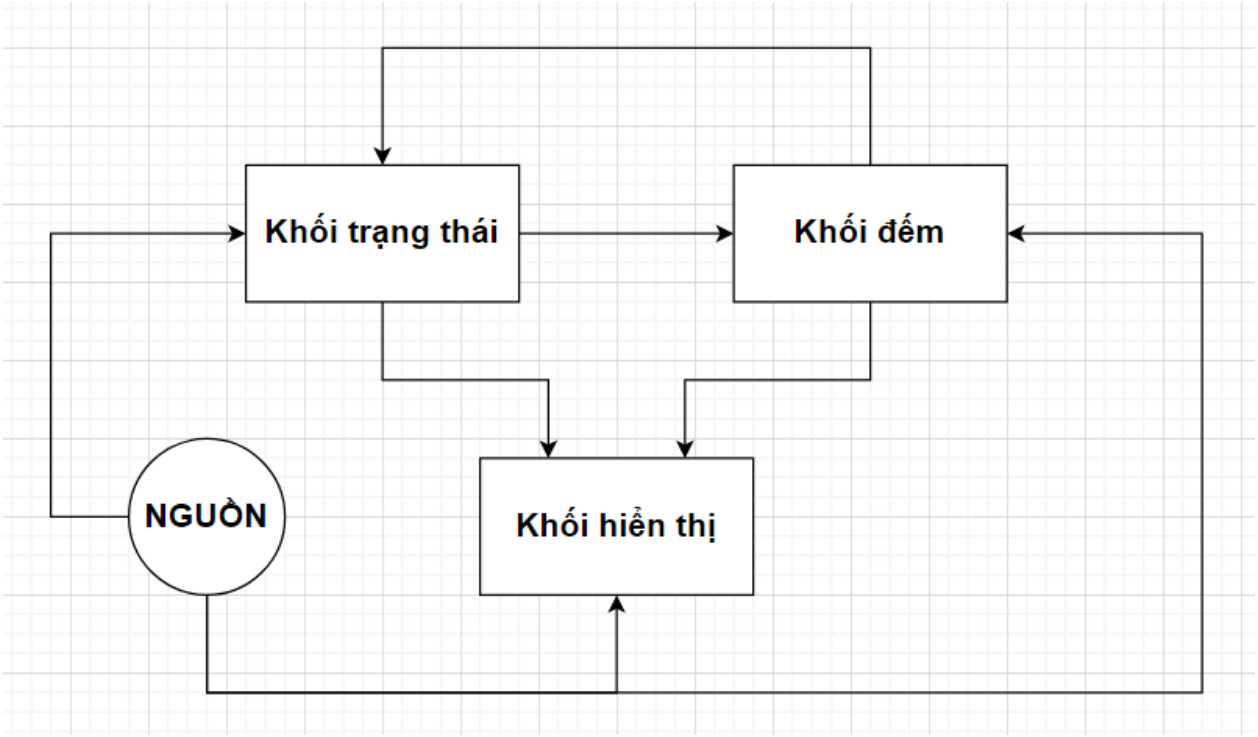


c. Điện trở 100 Ohm



2.2. Thiết kế mạch

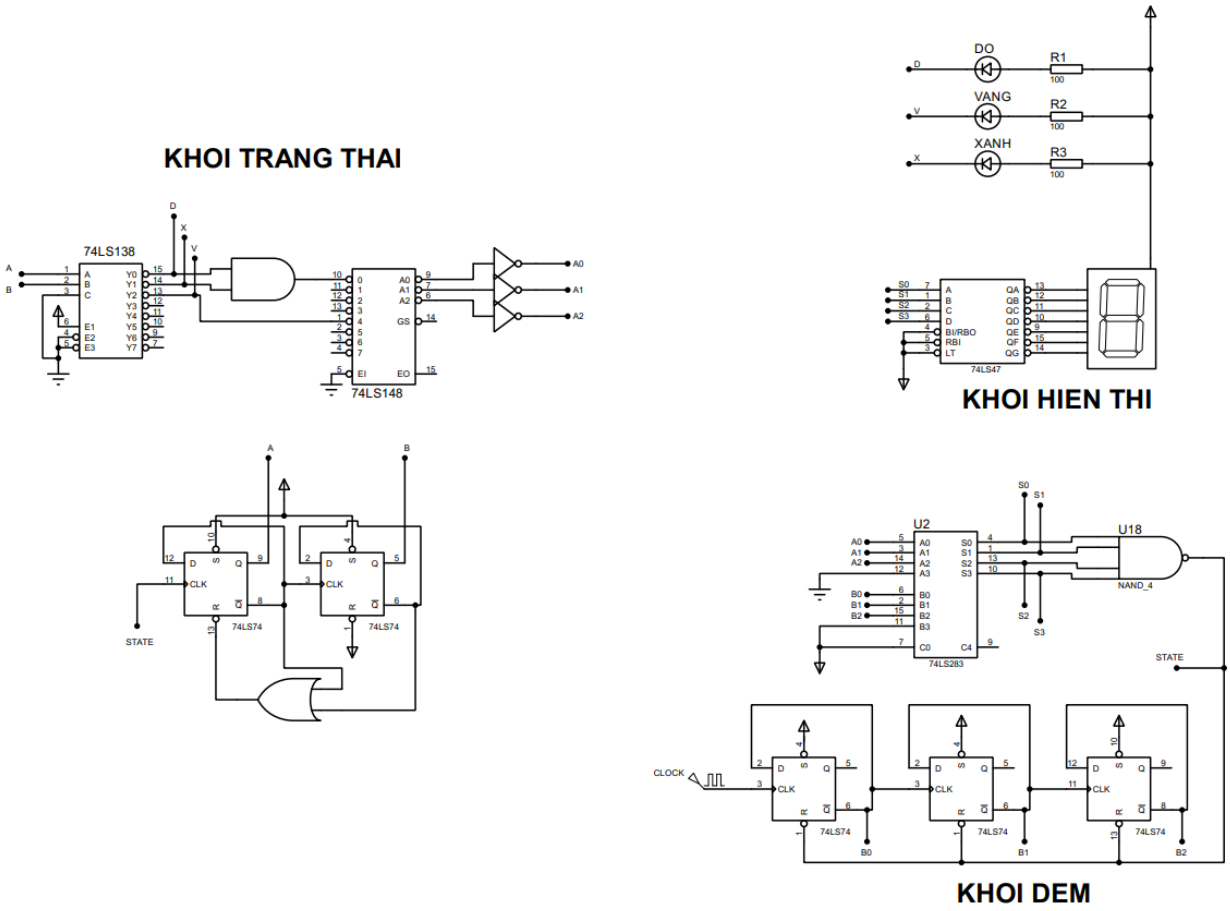
2.2.1. Sơ đồ khối



Không tính nguồn, mạch đèn giao thông của sinh viên làm gồm 3 khối chính:

- Khối trạng thái
- Khối đếm
- Khối hiển thị

2.2.2. Sơ đồ mạch



2.2.3. Chức năng các khối

a. Khối trạng thái

Khối trạng thái gồm 2 thành phần:

- Phần chuyển trạng thái: Gồm 2 Flip-flop D sao cho mỗi lần Khối đếm về 0 sẽ kích hoạt xung clock, làm chuyển trạng thái. Mỗi trạng thái tương ứng 1 mã nhị phân: 00, 01, 10, còn 11 sẽ làm cổng OR lập tức reset flip-flop.
- Phần giải mã: Mã trạng thái được chuyển đến IC giải mã 74LS138. Chân ra của IC 74LS138 sẽ lần lượt được tích cực tuỳ trạng thái từ Y0, Y1, Y2. Các chân này sẽ là ngõ vào của IC mã hoá 74LS148 (2 chân Y0, Y1 có cùng lựa chọn, ta đấu qua cổng AND_2). Các ngõ ra Y0, Y1, Y2 tích cực lần lượt kích hoạt led: Đỏ, Xanh, Vàng của khối hiển thị.

Sau khi mã hoá thời gian lựa chọn, mã sẽ qua cổng NOT (vì tích cực thấp) tới bộ đếm qua chân A0, A1, A2 sẽ được nói tới đây.

b. Khối đếm

Mã lựa chọn thời gian sẽ được chuyển đến chân A0, A1, A2 của IC 74LS283.

Bộ đếm gồm 3 con FlipFlop – D sẽ do xung clock kích, mã nhị phân của 3 con này cho ra sẽ thay đổi từng giây, quy cách đấu dây sẽ cho ta dạng đếm tăng dần. Cho Qbù vào B0, B1, B2 và tích cực chân B3, C0 ta sẽ được bộ trừ với tác dụng đếm lùi. Về 0, qua cổng NAND_4, ta gửi tín hiệu Reset đến các con Flip Flop.

Ví dụ: chân A0, A1, A2 nhận mã thời gian có giá trị là 5. Bộ đếm sẽ tăng dần theo giây, chân ra của 74LS283 sẽ giảm dần 4 – 3 – 2 – 1 – 0, và sau đó reset lại bộ đếm.

Cũng từ cổng NAND, ta gửi tín hiệu kích xung tới chân CLK của flip flop D tại bộ trạng thái, khiến trạng thái nhảy sang trạng thái mới.

Chân ra S0, S1, S2, S3 sẽ được chuyển tới khối hiển thị được nói tới đây.

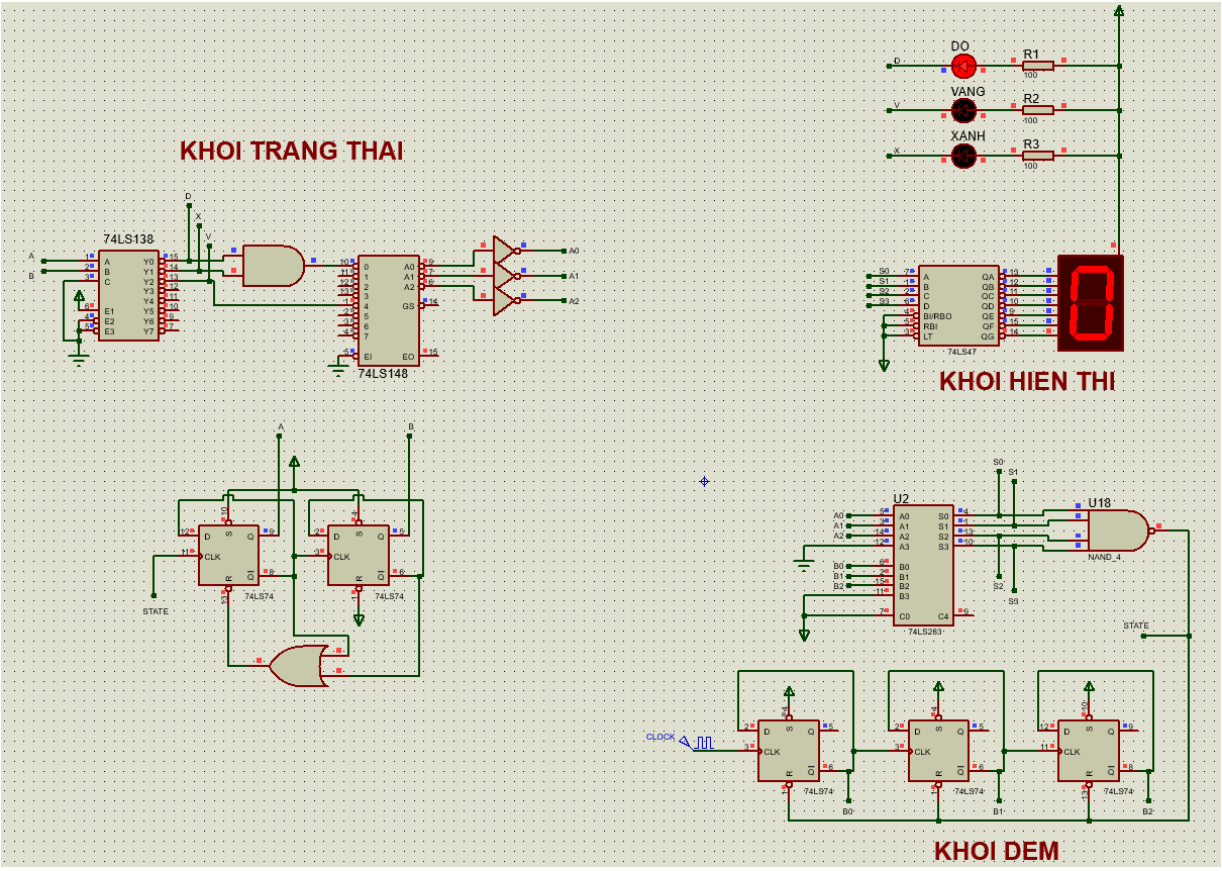
c. Khối hiển thị

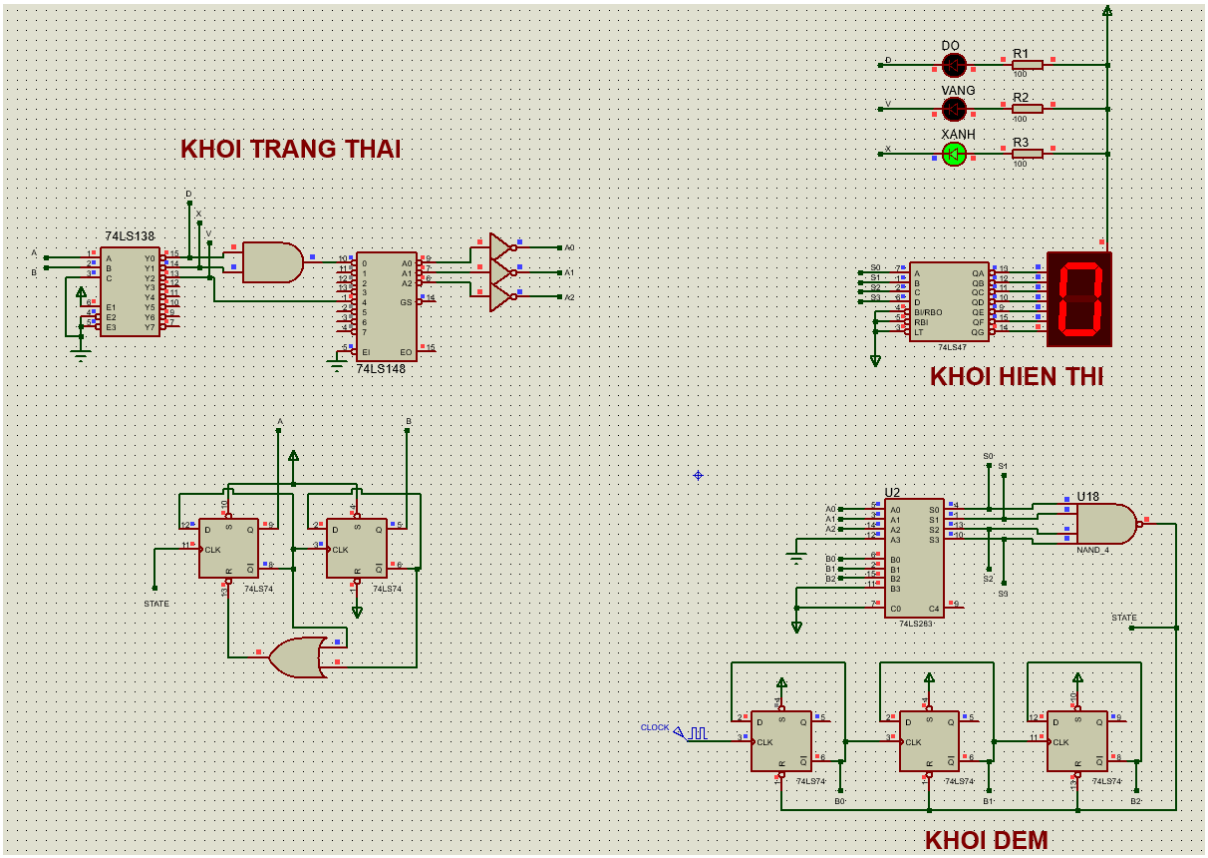
Khối hiển thị gồm 2 phần:

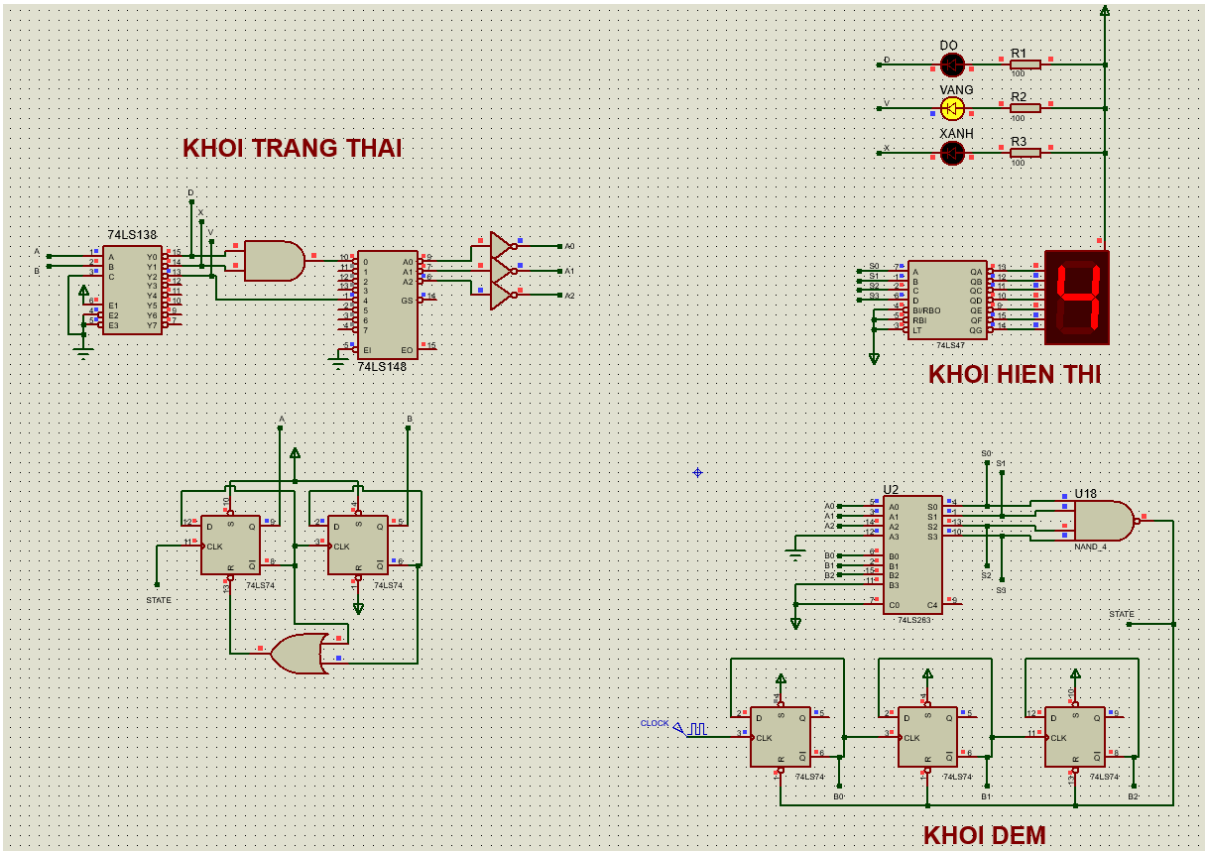
- Phần LED màu : Chân ra Y0, Y1, Y2 của IC 74LS138 sẽ tích cực lần lượt các led: Đỏ, Xanh , Vàng.
- Phần LED 7 đoạn: Hiển thị thời gian qua con 74LS47 dịch mã BCD sang 7 đoạn từ các chân S0, S1, S2, S3 từ IC 74LS283

2.3. Mô phỏng – Kết quả:

Ngõ ra 74LS138 ra sẽ lựa chọn được thời gian: Đấu vào chân 0, 0 , 4 của 74LS148 ta được thời gian hiển thị là 1, 1, 5.




KHOI TRANG THAI
74LS138
74LS148
DO
VANG
XANH
R1
R2
R3
74LS47
KHOI HIEN THI
STATE
U2
74LS263
U18
NAND_4
CLOCK
74LS74
KHOI DEM


KHOI TRANG THAI
74LS138
74LS148
DO
VANG
XANH
R1
R2
R3
74LS47
KHOI HIEN THI
STATE
U2
74LS263
U18
NAND_4
CLOCK
74LS74
KHOI DEM

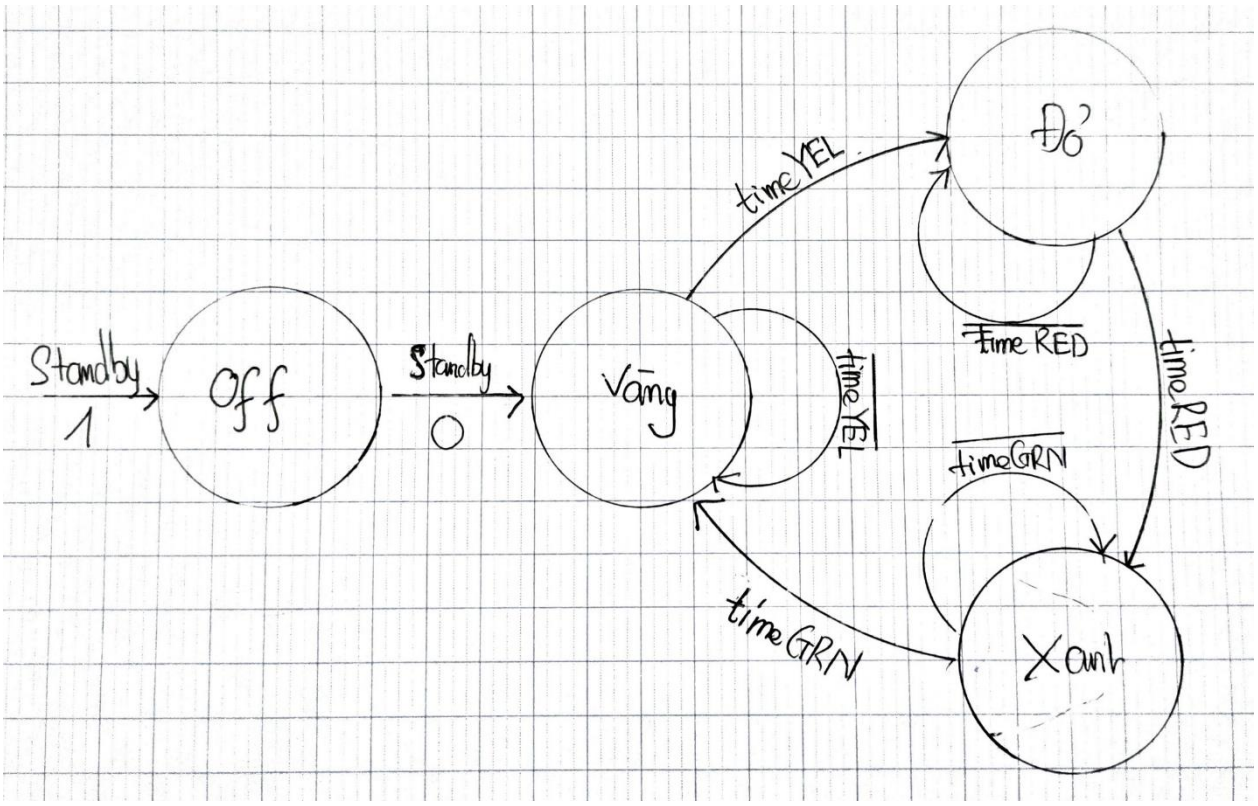
CHƯƠNG 3: SỬ DỤNG PHẦN MỀM QUARTUS 2.

3.1. Tổng quan

Thông qua ngôn ngữ VHDL, Quartus 2 hỗ trợ chúng ta máy trạng thái. Chúng ta sẽ tận dụng điều này để giải quyết yêu cầu.

3.2. Giản đồ máy trạng thái.

3.2.1. Giản đồ



3.2.2. Đặc điểm tóm lược

- Tín hiệu Standby = 1 mạch sẽ giữ ở chế độ Off (tắt)

- Tín hiệu Standby = 0 mạch sẽ bắt đầu tại đèn vàng

Với các trạng thái, thời gian tồn tại được cho như yêu cầu:

- Vàng: 5s

- Đỏ: 1s
- Xanh: 1s

Sau khi hết thời gian của 1 trạng thái, sẽ tự động chuyển sang trạng thái tiếp theo và reset bộ đếm thời gian.

3.3. CODE VHDL

Dưới đây là bản chụp code, phần chữ được ghi tại phần PHỤ LỤC:

```
----------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
----------------------------------------------
entity traffic_light is
   port (stby, clk : in std_logic;
         red, yellow, green : out std_logic
         );
end entity traffic_light;
--Ngõ vào Stanby ép đèn vào trạng thái off
----------------------------------------------
architecture state_machine of traffic_light is
   TYPE state is (off, vang , do , xanh);
   --Dùng xung 1Hz sẽ có đơn vị là second
   CONSTANT timeMAX : integer := 5;
   CONSTANT timeRED : integer := 1;
   CONSTANT timeGRN : integer := 1;
   CONSTANT timeYEL : integer := 5;
   SIGNAL pr_state, nx_state : state;
   --Trạng thái hiện tại pr_state
   --Trạng thái kế nx_state
   --Giá trị biến đếm hiện tại pr_state
   --Giá trị biến đếm kế nx_counter
   SIGNAL pr_counter, nx_counter : integer
                       range 0 to timeMAX := 1;
   SIGNAL reset_counter : std_logic;
   --Tín hiệu reset bộ counter
```

```
begin
-------------------Array FF--------------------
   process(clk, stby)
   begin
      if (stby = '1') then
         pr_state <= off;
      elsif (clk'event and clk = '1') then
   --Hoạt động với xung dương;
         pr_state <= nx_state;
         pr_counter <= nx_counter;
      end if;
   end process;

   nx_counter <= 1 when reset_counter = '1'
                        else pr_counter + 1;
--Nếu tín hiệu reset tích cực => reset bộ đếm
--Còn nếu không, thì counter lên 1
-----------------Mạch tổ hợp-----------------
   process(pr_counter, pr_state)
   begin
      case pr_state is
         when off => --OFF đến khi Stby tắt
            red <='0';green <='0';yellow <='0';
            nx_state <= vang;
            reset_counter <= '1';

         when vang => --vàng(5s) -> đỏ
            red <='0';green <='0';yellow <='1';
--Nếu bộ đếm = với timeYEL nghĩa là đã qua 5s
--=> chuyển sang đỏ
--Còn nếu không, thì counter + 1 để đếm tiếp
--Tương tự
            if pr_counter = timeYEL then
               nx_state <= do;
               reset_counter <= '1';
            else
               nx_state <= pr_state;
               reset_counter <= '0';
            end if;
```

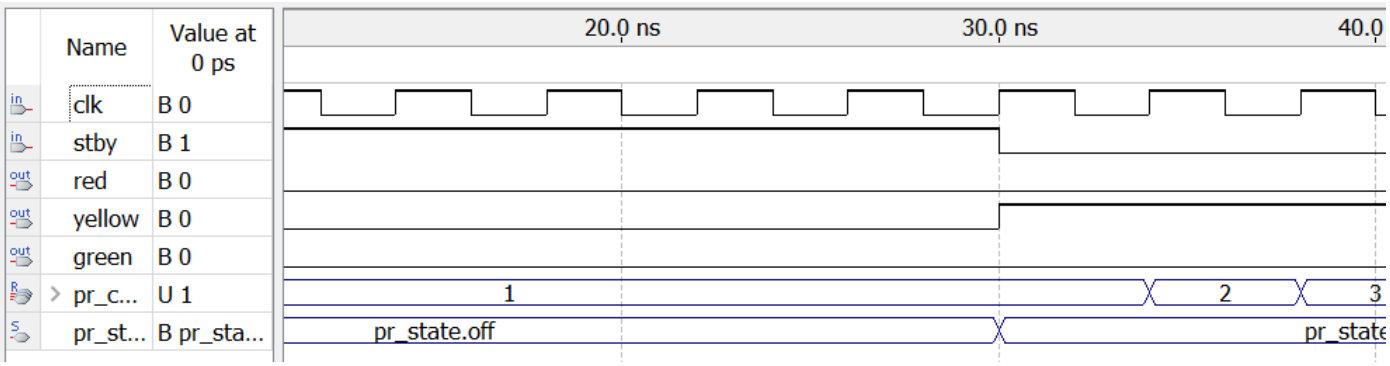
```
            when do => --đỏ(1s) -> xanh
               red <='1';green <='0';yellow <='0';
               if pr_counter = timeRED then
                  nx_state <= xanh;
                  reset_counter <= '1';
               else
                  nx_state <= pr_state;
                  reset_counter <= '0';
               end if;

            when xanh => --xanh(1s) -> vàng
               red <='0';green <='1';yellow <='0';
               if pr_counter = timeGRN then
                  nx_state <= vang;
                  reset_counter <= '1';
               else
                  nx_state <= pr_state;
                  reset_counter <= '0';
               end if;
         end case;
      end process;
end architecture state_machine;
```

Như ta đã thấy, số lượng Flip-Flop sử dụng là 5: 3 cái cho bộ đếm và 2 cái để lưu trữ trạng thái.

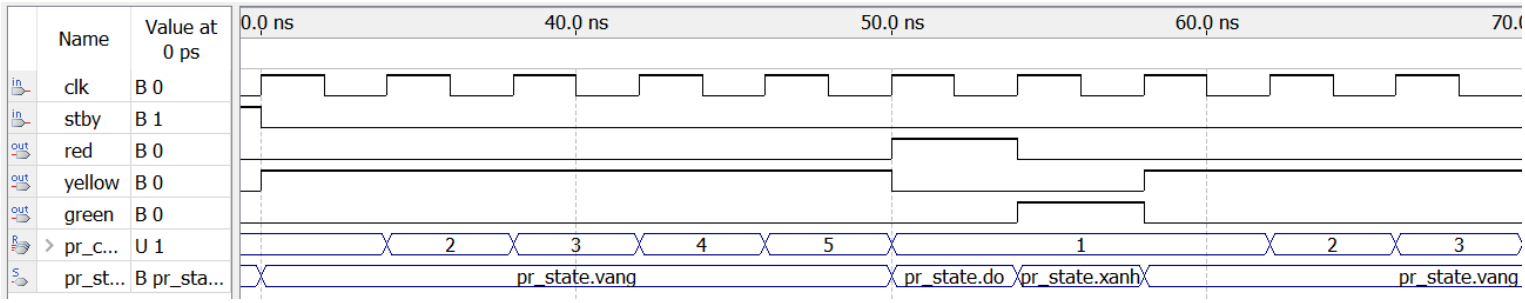
3.4. Mô phỏng – Kết quả

Kết quả mô phỏng sẽ được thể hiện dưới đây. Chúng ta giảm thời gian thực tế đi 250 000 000 lần để dễ thấy kết quả.

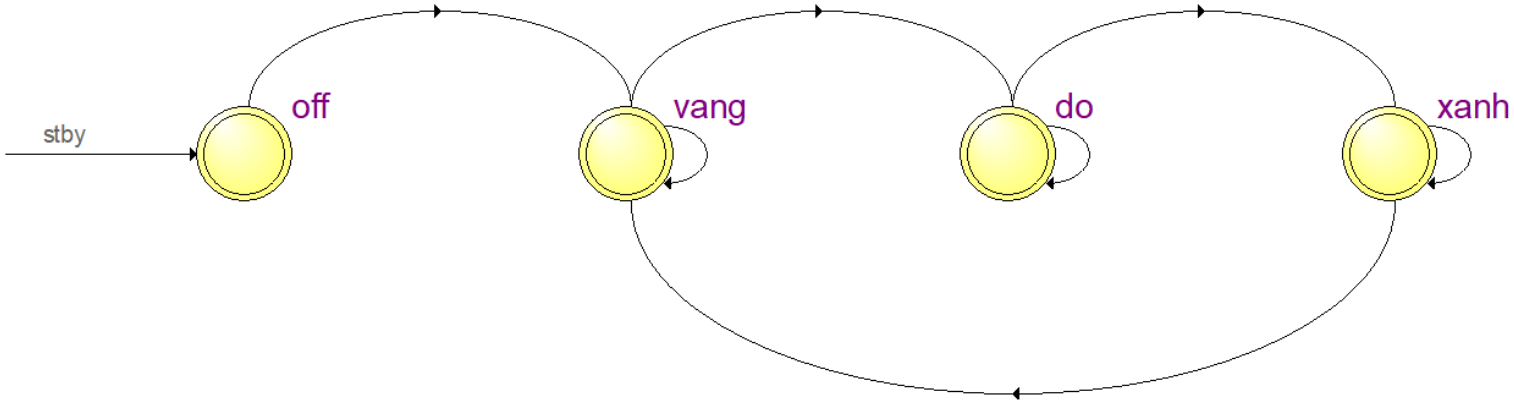
- Tín hiệu Standby tích cực:



- Sau đó tín hiệu Standby = 0:



- State Machine Viewer:



- State table:

State Table

	Source State	Destination State	Condition
1	do	xanh	(pr_counter[0]).(!pr_counter[1]).(!pr_counter[2])
2	do	do	(!pr_counter[0]) + (pr_counter[0]).(!pr_counter[1]).(pr_counter[2]) + (pr_counter[0]).(pr_counter[1])
3	off	vang	
4	vang	vang	(!pr_counter[0]) + (pr_counter[0]).(!pr_counter[1]).(!pr_counter[2]) + (pr_counter[0]).(pr_counter[1])
5	vang	do	(pr_counter[0]).(!pr_counter[1]).(pr_counter[2])
6	xanh	xanh	(!pr_counter[0]) + (pr_counter[0]).(!pr_counter[1]).(pr_counter[2]) + (pr_counter[0]).(pr_counter[1])
7	xanh	vang	(pr_counter[0]).(!pr_counter[1]).(!pr_counter[2])

Transitions | Encoding

PHỤ LỤC. CODE VHDL

```
------------------------------------------------

library ieee;

use ieee.std_logic_1164.all;

------------------------------------------------

entity traffic_light is

    port (stby, clk : in std_logic;

            red, yellow, green : out std_logic

            );

end entity traffic_light;

--Ngõ vào Stanby ép đèn vào trạng thái off

------------------------------------------------

architecture state_machine of traffic_light is

    TYPE state is (off, vang , do , xanh);

    --Dùng xung 1Hz sẽ có đơn vị là second
```

```
    CONSTANT timeMAX : integer := 5;
    CONSTANT timeRED : integer := 1;
    CONSTANT timeGRN : integer := 1;
    CONSTANT timeYEL : integer := 5;
    SIGNAL pr_state, nx_state : state;
    --Trạng thái hiện tại pr_state
    --Trạng thái kế nx_state
    --Giá trị biến đếm hiện tại pr_state
    --Giá trị biến đếm kế nx_counter
    SIGNAL pr_counter, nx_counter : integer
                                range 0 to timeMAX := 1;
    SIGNAL reset_counter : std_logic;
    --Tín hiệu reset bộ counter
begin
-------------------Array FF--------------------
    process(clk, stby)
    begin
        if (stby = '1') then
            pr_state <= off;
        elsif (clk'event and clk = '1') then
```

```
    --Hoạt động với xung dương;
            pr_state <= nx_state;
            pr_counter <= nx_counter;
        end if;
    end process;

    nx_counter <= 1 when reset_counter = '1'
                                    else pr_counter + 1;
--Nếu tín hiệu reset tích cực => reset bộ đếm
--Còn nếu không, thì counter lên 1
------------------Mạch tổ hợp------------------
    process(pr_counter, pr_state)
    begin
        case pr_state is
            when off => --OFF đến khi Stby tắt
                red <='0';green <='0';yellow <='0';
                nx_state <= vang;
                reset_counter <= '1';

            when vang => --vàng(5s) -> đỏ
```

```
                red <='0';green <='0';yellow <='1';
--Nếu bộ đếm = với timeYEL nghĩa là đã qua 5s
--=> chuyển sang đỏ
--Còn nếu không, thì counter + 1 để đếm tiếp
--Tương tự
                if pr_counter = timeYEL then
                    nx_state <= do;
                    reset_counter <= '1';
                else
                    nx_state <= pr_state;
                    reset_counter <= '0';
                end if;

            when do => --đỏ(1s) -> xanh
                red <='1';green <='0';yellow <='0';
                if pr_counter = timeRED then
                    nx_state <= xanh;
                    reset_counter <= '1';
                else
                    nx_state <= pr_state;
```

```
                    reset_counter <= '0';
                end if;

            when xanh => --xanh(1s) -> vàng
                red <='0';green <='1';yellow <='0';
                if pr_counter = timeGRN then
                    nx_state <= vang;
                    reset_counter <= '1';
                else
                    nx_state <= pr_state;
                    reset_counter <= '0';
                end if;
        end case;
    end process;
end architecture state_machine;
```