**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**


# BÁO CÁO BÀI TẬP LỚN
# THIẾT KẾ LUẬN LÝ (TH) (CO1026)

## ***ĐỀ TÀI:***
## *ĐÈN GIAO THÔNG CHO NGƯỜI ĐI BỘ*

**NHÓM 12 --- L01 --- HK 222**
**Giảng viên hướng dẫn: Huỳnh Hoàng Kha**

*Thành phố Hồ Chí Minh – 2023*

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

# DANH SÁCH THÀNH VIÊN
# NHÓM 12 – L01
# ID: 2213226

| Thành viên thực hiện | Mã số sinh viên |
|---|---|
| CAO THUẬN THIÊN | 2213226 |
| TRẦN TRUNG ĐỨC | 2210817 |
| NGÔ MINH THÁI | 2213106 |

# MỤC LỤC

**DANH MỤC HÌNH** ..... **3**

**BẢNG THUẬT NGỮ** ..... **3**

**GIỚI THIỆU ĐỀ TÀI** ..... **4**

1. Lời mở đầu ..... 4

2. Đặc tả đề tài ..... 5

**CHƯƠNG 1: SCHEMATIC (Elaborate Design)** ..... **8**

1.1. Sơ đồ khối ..... 8

1.2. Nội dung code ..... 9

1.3. Schematic (Elaborate Design) ..... 12

**CHƯƠNG 2: HIERARCHICAL TREE** ..... **14**

**CHƯƠNG 3: KẾT QUẢ SIMULATION** ..... **15**

3.1. Testcase code ..... 15

3.2. Kết quả simulation ..... 16

3.2.a..Kết quả testcase 01 ..... 16

3.2.b. Kết quả testcase 02 ..... 17

3.2.c. Kết quả testcase 03 ..... 18

**CHƯƠNG 4: PIN ASSIGNMENTS** ..... **19**

**DANH MỤC TÀI LIỆU THAM KHẢO** ..... **20**

# DANH MỤC HÌNH


| Tên hình ảnh | Trang |
|---|---|
| *Hình 1a. Tên đề tài* | 4 |
| *Hình 1b. Hình mô phỏng của đề tài* | 6 |
| *Hình 1c. Bảng trạng thái hoạt động của hệ thống đèn* | 7 |
| *Hình 1.1. Sơ đồ khối của đề tài* | 8 |
| *Hình 1.2. Schematic của sub-module div_clock* | 12 |
| *Hình 1.3. Schematic của sub-module TransTo7segment* | 12 |
| *Hình 1.4. Schematic của TrafficLightForWalking* | 13 |
| *Hình 2. Sơ đồ thể hiện quan hệ module – sub-module của đề tài* | 14 |
| *Hình 3.1. Kết quả simulation của testcase 01* | 16 |
| *Hình 3.2. Kết quả simulation của testcase 02* | 17 |
| *Hình 3.3. Kết quả simulation của testcase 03* | 18 |
| *Hỉnh 4.1. Bảng chi tiết pin assignment* | 19 |
| *Hình 4.2. Cài đặt pin assignment trên Vivado* | 19 |


# BẢNG THUẬT NGỮ

| Thuật ngữ | Ý nghĩa | Trang |
|---|---|---|
| *Default state* | Trạng thái mặc định | 5, 6, 16, 17 |
| *Ped state* | Trạng thái dành cho người đi bộ qua đường (đèn đi bộ màu xanh, đèn phương tiện màu đỏ) | 5. 6, 16, 18 |
| *Wait state* | Trạng thái đèn giao thông cho người đi bộ màu đỏ, chờ qua đường | 5, 6, 18 |
| *Normal mode* | Chế độ cho phép hệ thống hoạt động bình thường | 6 |
| *Setup mode* | Chế độ chỉnh sửa thời gian đèn sáng | 6 |
| *Normal zone* | Vùng người sử dụng được phép tương tác | 6 |
| *Setup zone* | Vùng tương tác dành riêng cho cơ quan chức năng chỉnh sửa hoạt động của đèn | 6 |
| *TrafficLight* | Đèn giao thông dành cho các phương tiện đang lưu thông trên tuyến đường | 5, 8 |
| *WalkingLight* | Đèn giao thông dành cho người đi bộ cần qua đường | 5, 8 |

# GIỚI THIỆU ĐỀ TÀI


*Hình 1a. Tên đề tài*

## *1. <u>Lời mở đầu:</u>*

Giao thông đang trở nên ngày càng phức tạp hơn, đặc biệt với số lượng người đi bộ trong các khu vực đô thị đang ngày càng tăng. Trong bối cảnh này, đèn giao thông cho người đi bộ đã trở thành một phần quan trọng của hệ thống giao thông đô thị, giúp bảo đảm an toàn cho người đi bộ khi đi qua đường.

Trong báo cáo này, chúng em sẽ đề xuất một thiết kế đèn giao thông cho người đi bộ, bao gồm đặc tả đề tải và sơ đồ khối của đề tài. Bên cạnh đó, chúng em sẽ đưa các hình ảnh liên quan tới quá trình hoàn thiện thiết kế, bao gồm schematic design, hierarchical tree (sơ đồ thể hiện quan hệ module - sub module), kết quả simulation trên app và pin assignments trên board Arty Z7.

## *2. Đặc tả đề tài:*

*** Đèn giao thông cho người đi bộ bao gồm 9 thiết bị ngoại vi sau:**

1. Nút nhấn: đặt ở bên lề đường, nơi có vạch kẻ đường, dùng để người đi bộ nhấn khi có nhu cầu sang đường.

2. Nút reset: dùng để đưa hệ thống đèn về default state.

3. Công tắc mode: dùng khi có nhu cầu chỉnh sửa thời gian đèn sáng.

4. Công tắc select: dùng để lựa chọn chỉnh sửa thời gian wait state hoặc ped state.

5. Nút (+): dùng để tăng thời gian, mỗi chu kì clk tăng thời gian lên 1 giây.

6. Nút (-): dùng để giảm thời gian, mỗi chu kì clk giảm thời gian đi 1 giây.

7. WalkingLight: có biểu tượng người đi bộ, dùng để ra hiệu cho người đi bộ khi sang đường.

8. TrafficLight: là đèn giao thông ra hiệu cho các phương tiện đang lưu thông trên đường.

9. Đèn hiển thị thời gian (LED 7 đoạn): bố trí cùng với đèn giao thông, cho biết thời gian còn lại của mỗi trạng thái.

*** Cách hoạt động:**

Đèn giao thông cho người đi bộ được đặt tại các tuyến đường xung quanh phố đi bộ, nơi có nhu cầu đi bộ thường xuyên, dùng để dừng các phương tiện đang lưu thông trên đường để dành đường cho người đi bộ sang đường.

- Ở default state, đèn dành cho người đi bộ hiển thị màu đỏ và đèn giao thông hiển thị màu xanh.

- Khi người đi bộ có nhu cầu sang đường, người đi bộ sẽ đến nơi có vạch kẻ đường và có bố trí đèn. Người đi bộ sẽ nhấn vào nút nhấn và đợi đến khi đèn tín hiệu của mình chuyển sang màu xanh để sang đường.

- Khi nút nhấn được nhấn, hệ thống chuyển sang wait state đèn giao thông bắt đầu chuyển sang vàng trong vòng 5 giây, LED 7 đoạn sẽ hiển thị thời gian đếm ngược từ 4 đến 0. Sau đó đèn chuyển sang màu đỏ, đồng thời đèn dành cho người đi bộ chuyển sang

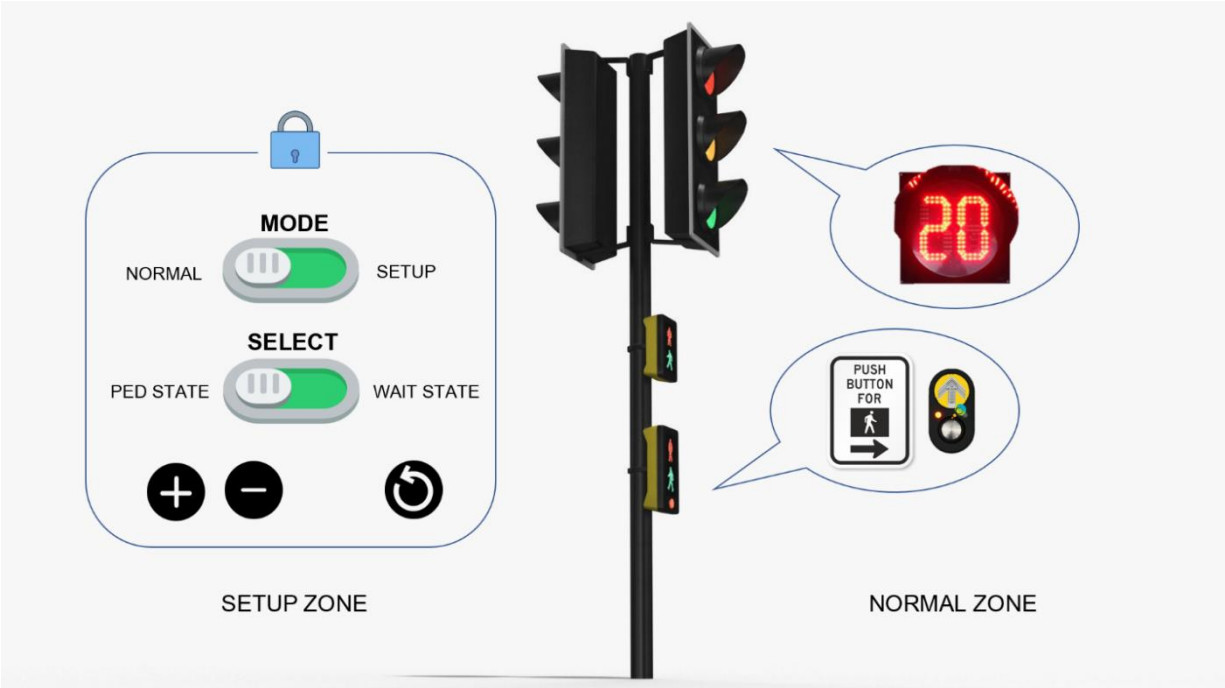
ped state trong vòng 9 giây, LED 7 đoạn hiển thị thời gian đếm ngược từ 8 đến 0 (đây là thời gian mặc định khi chưa được chỉnh sửa theo nhu cầu).

- Cuối cùng, sau khi hoàn thành chu trình hoạt động của 1 lần nhấn nút, đèn trở về default state chờ tín hiệu nút tiếp theo.

- Khi chưa bật công tắc mode, đèn giao thông vẫn hoạt động như mô tả ở trên (normal mode). Việc bật công tắc mode sẽ kích hoạt đèn chuyển sang setup mode. Tại thời điểm đó, hệ thống chỉ cho phép chỉnh sửa thời gian ped state hoặc wait state tùy thuộc vào trạng thái của công tắc select. Thời gian sẽ được điều chỉnh bằng 2 nút nhấn (+) và (-) bên trong setup zone.

- Việc nhấn nút reset sẽ đưa đèn từ bất kì trạng thái nào trước đó về default state.

Dưới đây là hình mô phỏng của đề tài, cũng như bảng trạng thái hoạt động của hệ thống đèn:




*Hình 1b. Hình mô phỏng của đề tài*

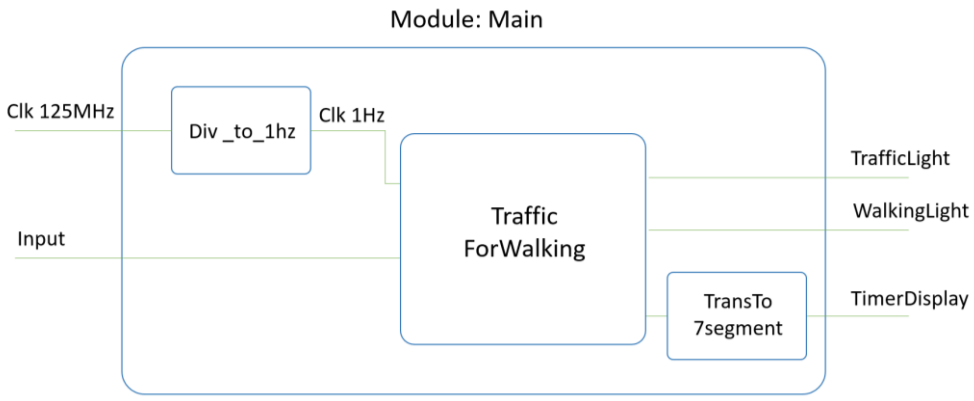
| Input | | | | | | | Function |
|---|---|---|---|---|---|---|---|
| clk | but | reset | mode | sel | (+) | (-) | |
| ↓ | X | X | X | X | X | X | Không nhận tín hiệu. |
| ↑ | X | X | **H** | X | X | X | Kích hoạt setup mode. |
| ↑ | X | X | **H** | **L** | X | X | Chọn chỉnh sửa ped state. |
| ↑ | X | X | **H** | **H** | X | X | Chọn chỉnh sửa wait state. |
| ↑ | X | X | **H** | **L** | **H** | **L** | Tăng thời gian ped state 1s. |
| ↑ | X | X | **H** | **L** | **L** | **H** | Giảm thời gian ped state 1s. |
| ↑ | X | X | **H** | **H** | **H** | **L** | Tăng thời gian wait state 1s. |
| ↑ | X | X | **H** | **H** | **L** | **H** | Giảm thời gian wait state 1s. |
| ↑ | X | X | **L** | X | X | X | Kích hoạt normal mode. |
| ↑ | **H** | **L** | **L** | X | X | X | Nhận nút nhấn và bắt đầu hoạt động tuần tự các trạng thái. |
| ↑ | **L** | **H** | **L** | X | X | X | Chuyển về default state. |

*Hình 1c. Bảng trạng thái hoạt động của hệ thống đèn*

# CHƯƠNG 1: SCHEMATIC (Elaborate Design)

## *1.1 Sơ đồ khối:*

Dựa vào đặc tả đề tài, chúng em đã thiết kế được sơ đồ khối chức năng như sau:




*Hình 1.1. Sơ đồ khối của đề tài*

- Module Div_to_1hz: module này nhận tín hiệu clk có tần số 125MHz từ Board Arty Z7 và chia tín hiệu này sang dạng clk có tầng số 1Hz, tần số này sẽ điều khiển toàn bộ hệ thống hoạt động tuần tự.
- Module TrafficForWalking: được xem như bộ xử lý trung tâm, nó có nhiệm vụ nhận tín hiệu của các input, clk1Hz, xử lý chúng và xuất ra tín hiệu logic cho đèn TrafficLight và WalkingLight, riêng tín hiệu của các led 7 đoạn dùng để hiển thị thời gian thì sẽ được xuất ra dưới dạng dãy số BCD.
- Module TransTo7segment: module này nhận tín hiệu là dãy số BCD được chuyển đến từ module TrafficForWalking, phân tích nó và xử lý thành các tín hiệu logic để điều khiển led 7 đoạn.
- Module Main: được ví như là một chiếc main của máy vi tính, nó mang chức năng nhận input, xuất ra output và kết nối các module con lại với nhau.

Sở dĩ chức năng chuyển đổi BCD sang tín hiệu logic cho led 7 đoạn được tách ra làm một module riêng là có mục đích. Trong trường hợp chúng ta cần sử dụng trực tiếp led 7 đoạn thì sẽ gọi nó ra trong module main. Còn trường hợp sử dụng led 7 đoạn có trên Extension Board được cung cấp thì sẽ gửi trực tiếp tín hiệu BCD, vì Extension Board đã được tích hợp IC giải mã CD4511BM có chức năng tương tự module này.

## 1.2 ***Nội dung code:***

Dựa vào thiết kế trên, chúng em xây dựng các module như sau:

- Module Div_to_1hz:

```
module div_to_1Hz(input clkin, output reg clkout);
    reg [28:0] counter;
    initial begin
        counter = 0;
        clkout = 0;
    end
    always @(posedge clkin) begin
        if (counter == 0) begin
            counter <= 64;
            clkout <= ~clkout;
        end
        else begin
            counter <= counter - 1;
        end
    end
endmodule
```

- Module TrafficLightForWalking:

```
module trafficLightForWalking(
    input clk,             // clock signal
    input reset,           // reset signal
    input buttonPressed,   // button press signal
    input mode,
    input sel,
    input btplus,
    input btminus,
    output reg [2:0] pedestrianLight,    // pedestrian traffic light signal
    output reg [2:0] trafficLight,       // vehicular traffic light signal
    output reg [3:0] timeCounter         // countdown time signal
    );
// internal signals
    reg [1:0] state;        // state variable
    reg [3:0] count;        // countdown time counter
    reg [3:0] countWait;
    reg [3:0] countPed;
// constants
    parameter IDLE_STATE = 2'b00;        // idle state
    parameter WAIT_STATE = 2'b01;        // countdown state
    parameter PED_GREEN_STATE = 2'b10;   // pedestrian green state
    initial begin
        state = IDLE_STATE;
        count = 0;
        countWait = 4;
        countPed = 9;
    end

    always @(posedge clk) begin
        if (reset) begin
            state <= IDLE_STATE;
            count = 0;
            end
        else begin
        if (mode) begin
        if (sel) begin timeCounter = countWait;
if (btplus&&countWait<9) begin countWait <= countWait +1; timeCounter = countWait; end
if (btminus&&countWait>1) begin countWait <= countWait - 1;
timeCounter = countWait; end
```

```
        end
        else begin timeCounter = countPed;
  if (btplus&&countPed<9) begin countPed <= countPed +1;timeCounter = countPed; end
  if (btminus&&countPed>1) begin countPed <= countPed -1;timeCounter = countPed; end
        end

        end
        else begin
            case(state)
            IDLE_STATE: begin
                pedestrianLight <= 3'b001;      // pedestrians: red
                trafficLight <= 3'b010;         // vehicles: green
                timeCounter <= 4'b0000;
                if (buttonPressed) begin
                    state <= WAIT_STATE;
                    count = countWait;
                    end
            end

            WAIT_STATE: begin
                pedestrianLight <= 3'b011;      // pedestrians: yellow
                trafficLight <= 3'b011;         //  vehicles: yellow
                timeCounter <= count;           // Time countdown
                if (count == 0) begin
                    state <= PED_GREEN_STATE;
                    count = countPed;
                    end
            end

            PED_GREEN_STATE: begin
                pedestrianLight <= 3'b010;      // pedestrians: green
                trafficLight <= 3'b001;         // vehicles: red
                timeCounter <= count;
                if (count == 0) begin
                    state <= IDLE_STATE;
                    end
            end

            default: state <= IDLE_STATE;
            endcase
            end
// Time countdown
        if (state == WAIT_STATE || state == PED_GREEN_STATE) begin
            count <= count - 1;
            end
        end
    end
endmodule
```

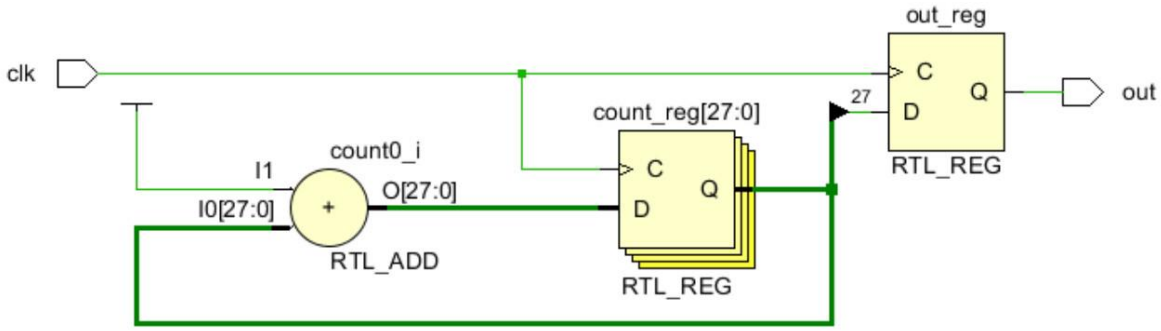
- Module TransTo7segment:

```
module TransTo7segment(
  input [3:0] bcd_input,
  output reg [6:0] segment_output
);

  always @(*) begin
    case (bcd_input)
      4'b0000: segment_output = 7'b1000000; // 0
      4'b0001: segment_output = 7'b1111001; // 1
      4'b0010: segment_output = 7'b0100100; // 2
      4'b0011: segment_output = 7'b0110000; // 3
      4'b0100: segment_output = 7'b0011001; // 4
      4'b0101: segment_output = 7'b0010010; // 5
      4'b0110: segment_output = 7'b0000010; // 6
      4'b0111: segment_output = 7'b1111000; // 7
      4'b1000: segment_output = 7'b0000000; // 8
      4'b1001: segment_output = 7'b0011000; // 9
      default: segment_output = 7'b1111111; // blank
    endcase
  end
endmodule
```

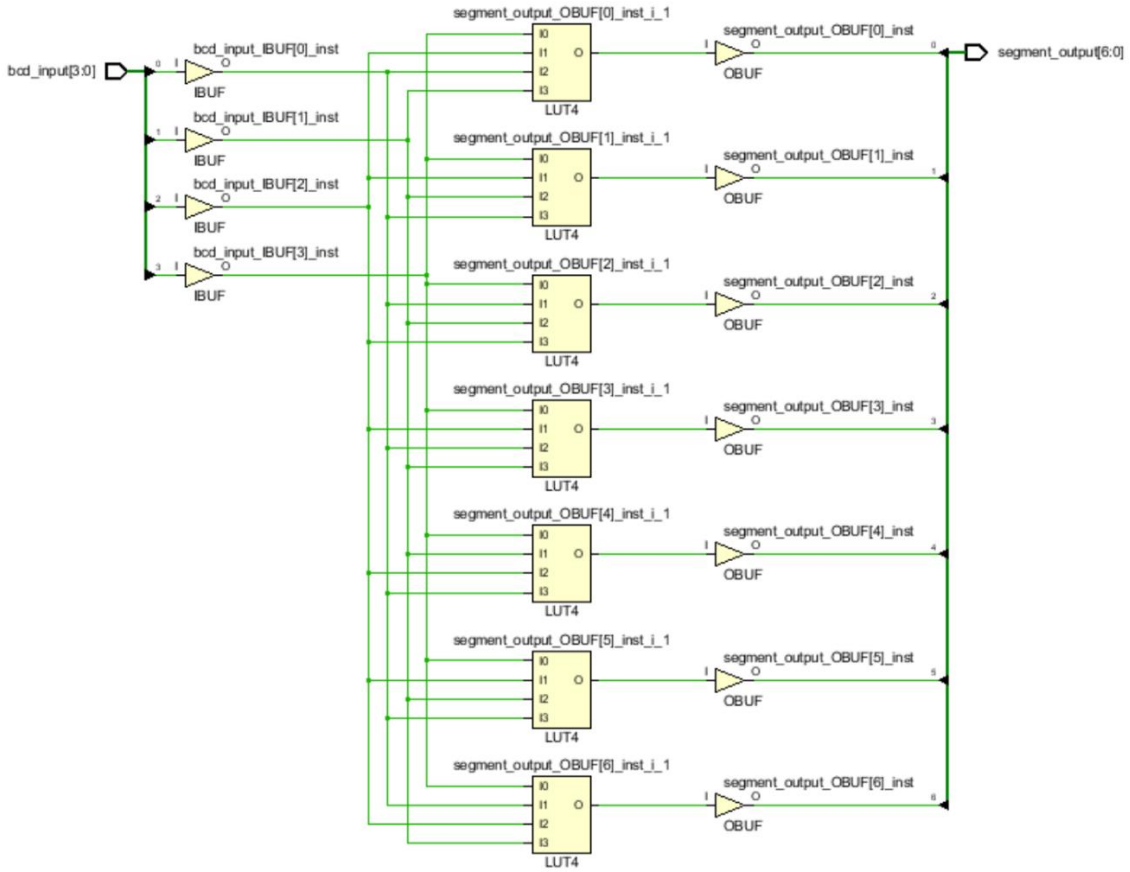
- Module Main:

```
module main(
    input clk,                    // clock signal
    input reset,                  // reset signal
    input buttonPressed,          // button press signal
    input mode,
    input sel,
    input btplus,
    input btminus,
    output  [2:0] pedestrianLight,       // pedestrian traffic light signal
    output  [2:0] trafficLight,          // vehicular traffic light signal
    output  [6:0] display                // countdown time signal
    );

    div_to_1Hz  Div(clk,clk1Hz);
    trafficLightForWalking Traf(clk1Hz, reset, buttonPressed, mode, sel,
btplus, btminus, pedestrianLight, trafficLight, BCD);
    TransTo7segment Trans(BCD, display);

endmodule
```

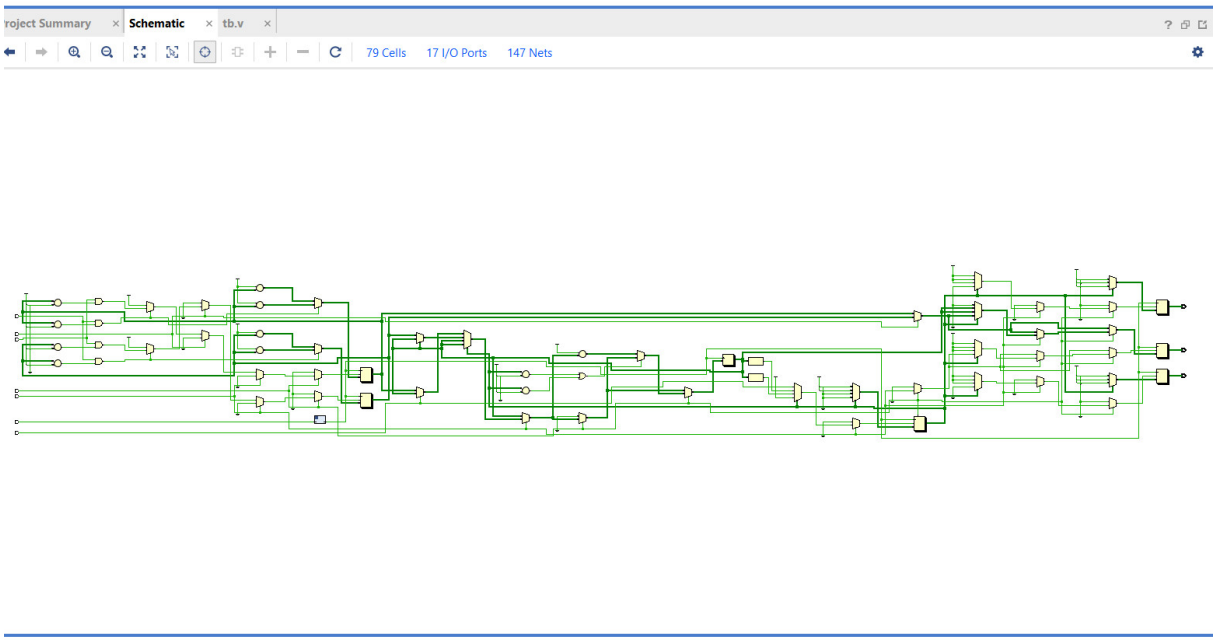
## *1.3 Schematic (Elaborate Design):*




*Hình 1.2. Schematic của sub-module div_clock*




*Hình 1.3. Schematic của sub-module TransTo7segment*




*Hình 1.4. Schematic của TrafficLightForWalking*

Như vậy, schematic của module main và các sub-module đã trùng khớp với sơ đồ khối thiết kế ban đầu.

# CHƯƠNG 2: HIERARCHICAL TREE

Trong Vivado, cây phân cấp (hierarchical tree) là một biểu diễn đồ họa của phân cấp thiết kế của một dự án. Nó hiển thị các module chính, các sub-module của thiết kế và các kết nối của chúng với nhau. Cây phân cấp sẽ hữu ích trong việc xem xét một dự án lớn, giúp ta hiểu mối quan hệ giữa các module khác nhau.

Dưới đây là sơ đồ thể hiện quan hệ module – sub-module (hierarchical tree) của đề tài chúng em.


*Hình 2. Sơ đồ thể hiện quan hệ module – sub-module của đề tài*

Các sub-module là div_to_1Hz, trafficLightForWalking, TransTo7segment. Như vậy mô hình sơ đồ cây này đã trùng khớp với sơ đồ khối ban đầu được chúng em thiết kế.

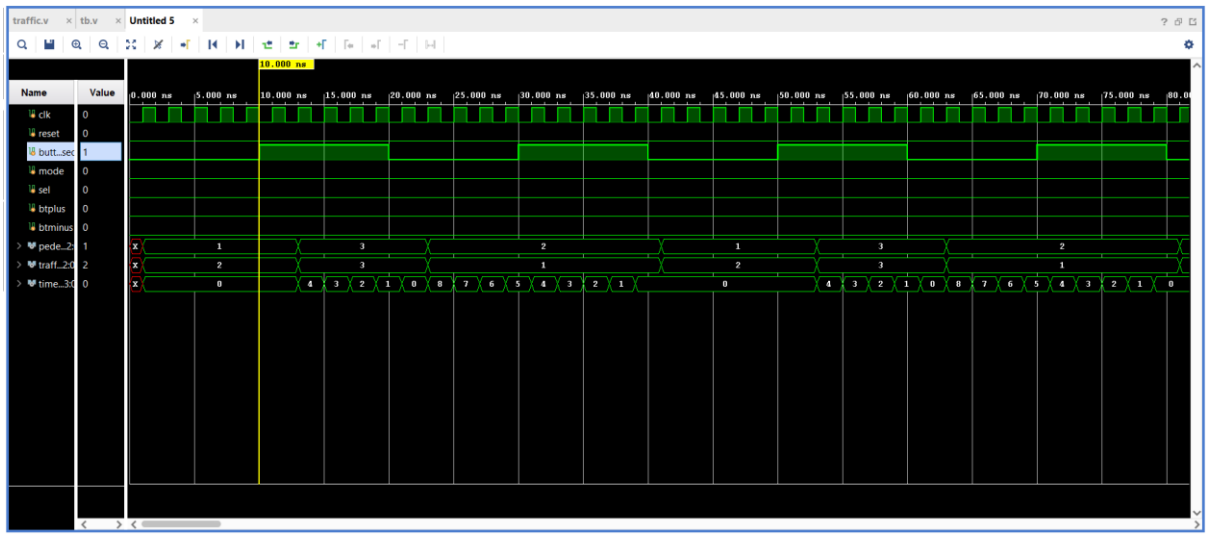
# CHƯƠNG 3: KẾT QUẢ SIMULATION

## 3.1 Testcase code:

```
module trafficLightForWalking_tb;
reg clk;
reg reset;
reg buttonPressed;
reg mode;
reg sel;
reg btplus;
reg btminus;
wire [2:0] pedestrianLight;
wire [2:0] trafficLight;
wire [3:0] timeCounter;
trafficLightForWalking dut(.clk(clk), .reset(reset),
    .buttonPressed(buttonPressed), .mode(mode), .sel(sel), .btplus(btplus),
    .btminus(btminus), .pedestrianLight(pedestrianLight),
    .trafficLight(trafficLight), .timeCounter(timeCounter));
always #1 clk = ~clk;
// Testcase 01: Pedestrian button is pressed
initial begin
clk = 0; reset = 0; buttonPressed = 0; mode = 0;
sel = 0; btplus = 0; btminus = 0;
end
always #10 buttonPressed = ~buttonPressed;
// Testcase 02: Reset button is pressed while the system's working
initial begin
#2 buttonPressed = 1;
#5 buttonPressed = 0;
#10 reset = 1;
#2 reset = 0;
end
// Testcase 03: Switch mode & sel check
initial begin
#5;
mode = 1;
sel = 1;
btplus = 1;
#3 btplus = 0; #3 btplus = 1;
#3 btplus = 0; #3 btplus = 1;
#3 btplus = 0; #3 btplus = 1;
#3 btplus = 0; #3 btplus = 1;
#3 btplus = 0;
#10 sel = 0;
btminus = 1;
#3 btminus = 0; #3 btminus = 1;
#3 btminus = 0; #3 btminus = 1;
#3 btminus = 0; #3 btminus = 1;
#3 btminus = 0; #3 btminus = 1;
#3 btminus = 0; #3 btminus = 1;
#3 btminus = 0;
#10 mode = 0;
#5 buttonPressed = 1; #5 buttonPressed = 0;
#100 $finish;
end
endmodule
```

## 3.2 ***Kết quả simulation:***

### a. ***Kết quả testcase 01:***




*Hình 3.1. Kết quả simulation của testcase 01*

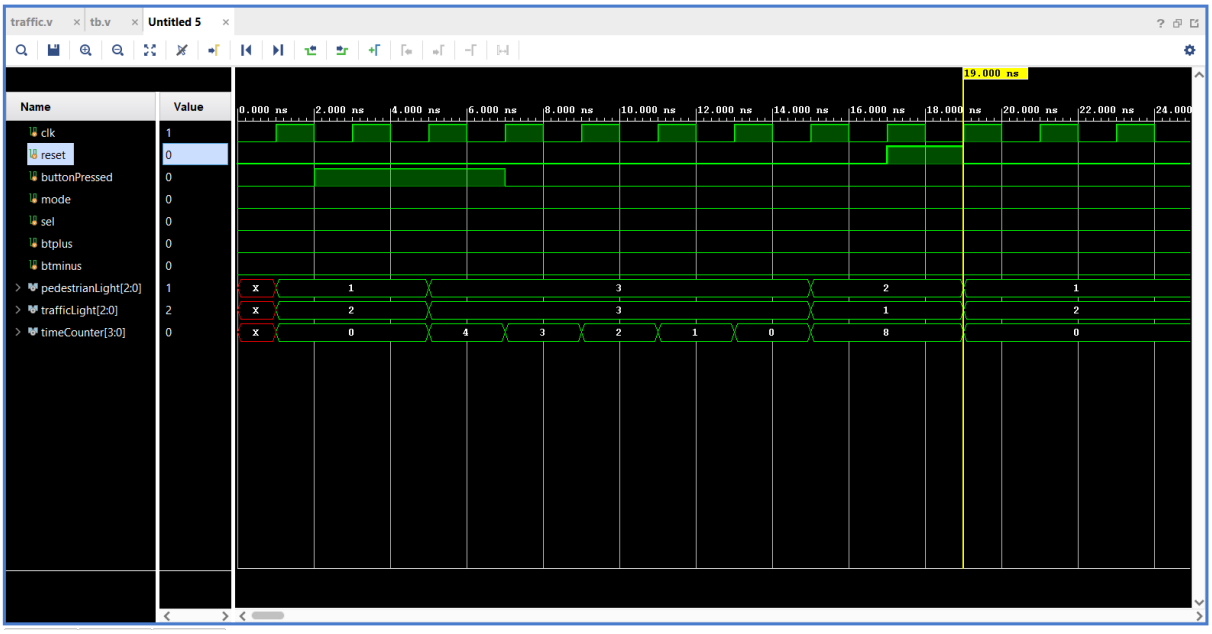
Khi nút nhấn được nhấn lần đầu, LED 7 đoạn timeCounter hiển thị thời gian đếm ngược từ 4 đến 0 (chính là trạng thái đèn giao thông chuyển sang vàng trong 5 giây). Sau đó LED 7 đoạn timeCounter hiển thị thời gian đếm ngược từ 8 đến 0 (chính là trạng thái đèn chuyển sang màu đỏ, và đèn dành cho người đi bộ chuyển sang ped state trong 9 giây).

Lần thứ hai nhấn nút khi đèn giao thông đi bộ đang trong trạng thái hoạt động, hệ thống sẽ bỏ qua lần nhấn nút này và tiếp tục hoạt động bình thường tới khi quay trở lại default state.

Lần thứ ba nhấn nút tương tự lần đầu, và lần thứ tư nhấn nút tương tự lần thứ hai.

**Kết luận:** Kết quả simulation hoàn toàn khớp với đặc tả đề tài.

## b. ***Kết quả testcase 02:***



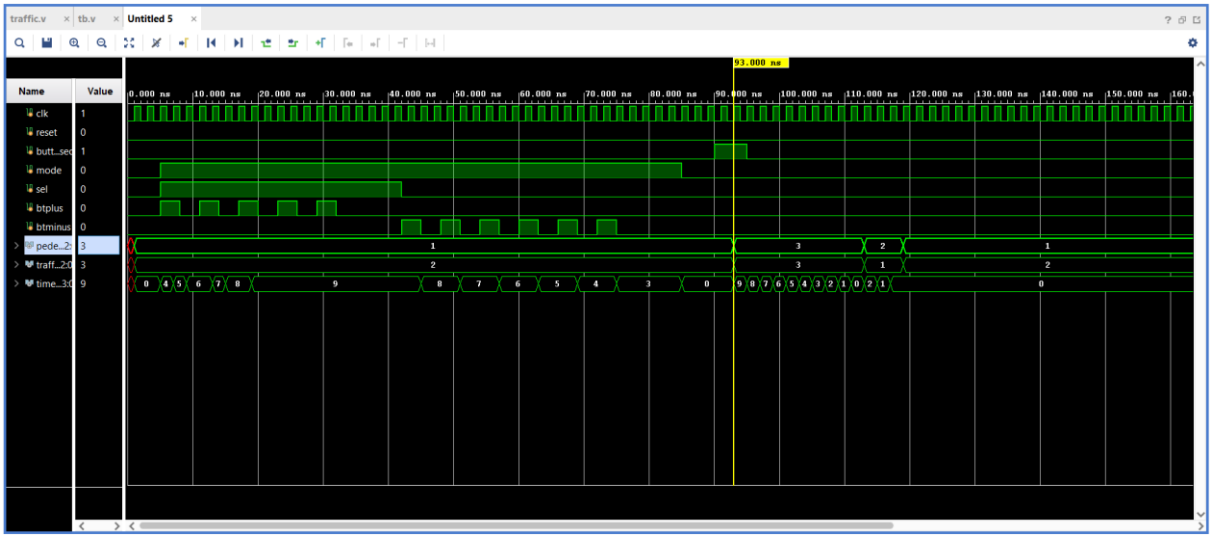

*Hình 3.2. Kết quả simulation của testcase 02*

Khi nút nhấn được nhấn lần đầu, LED 7 đoạn timeCounter hiển thị thời gian đếm ngược từ 4 đến 0 (chính là trạng thái đèn giao thông chuyển sang vàng trong 5 giây).

Khi nhấn nút reset ngay sau đó, hệ thống đèn sẽ lập tức quay về default state.

**Kết luận:** Kết quả simulation hoàn toàn khớp với đặc tả đề tài.

### c. ***Kết quả testcase 03:***




*Hình 3.3. Kết quả simulation của testcase 03*

Khi bật công tắc mode và select cùng 1 lúc, hệ thống đèn sẽ cho phép chỉnh sửa thời gian wait state. Nhấn nút (+) 5 lần giúp tăng thêm 5 giây, tức tổng thời gian wait state là 10 giây (đếm ngược từ 9 đến 0).

Sau khi tắt công tắc select, hệ thống đèn cho phép chỉnh sửa thời gian ped state. Nhấn nút (-) 6 lần làm giảm 6 giây, tức thời gian ped state còn 3 giây (đếm ngược từ 2 đến 0).

Khi nút nhấn được nhấn, LED 7 đoạn timeCounter hiển thị thời gian đếm ngược từ 9 đến 0, sau đó đếm ngược từ 2 đến 0, hoàn toàn khớp với những phân tích ở trên.

**Kết luận:** Kết quả simulation hoàn toàn khớp với đặc tả đề tài.

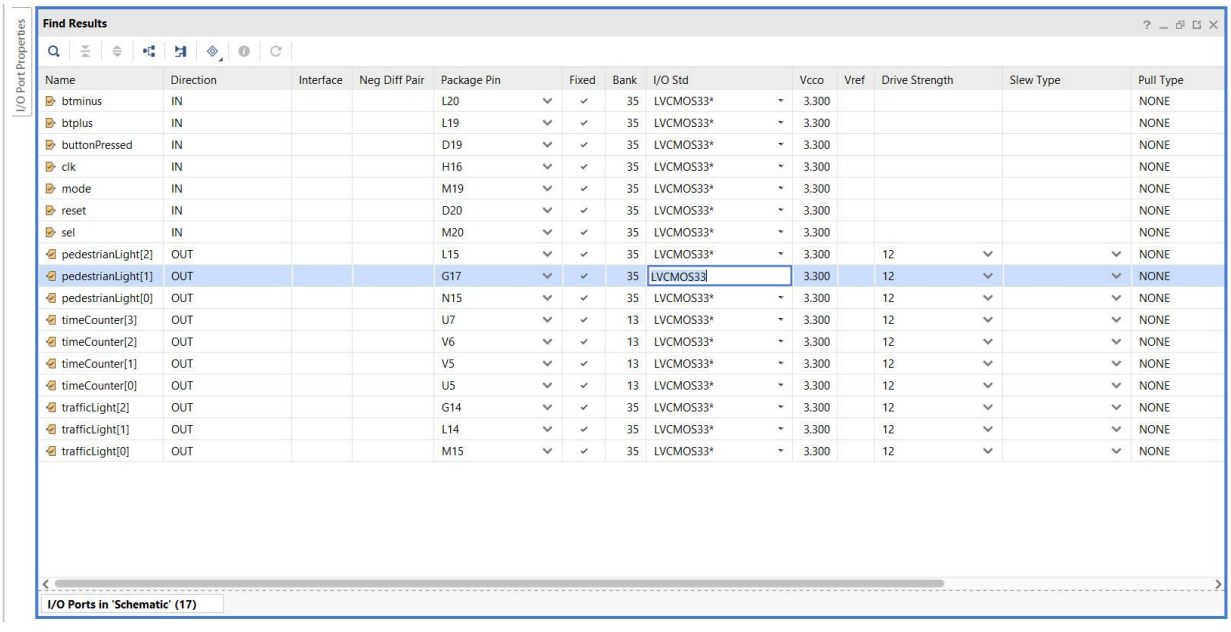
# CHƯƠNG 4: PIN ASSIGNMENTS

Các chân I/O đều phải được cài đặt và gán pin assignment đúng trước khi kết nối với board để tránh việc extension board hoạt động sai hoặc bị hỏng.

Dưới đây là bảng trình bày chi tiết pin assignment thông qua chức năng và vai trò của các chân.

| Name | Direction | Type | Port | Package Bin | I/O Std |
|---|---|---|---|---|---|
| clk | INPUT | Clock Signal | clk | H16 | LVCMOS33 |
| buttonPressed | | Button | btn[0] | D19 | |
| reset | | | btn[1] | D20 | |
| btminus | | | btn[2] | L20 | |
| btplus | | | btn[3] | L19 | |
| sel | | Switch | sw[0] | M20 | |
| mode | | | sw[1] | M19 | |
| pedestrianLight[0] | OUTPUT | RGB LEDs | led4_b | L15 | |
| pedestrianLight[1] | | | led4_g | G17 | |
| pedestrianLight[2] | | | led4_r | N15 | |
| trafficLight[0] | | | led5_b | G14 | |
| trafficLight[1] | | | led5_g | L14 | |
| trafficLight[2] | | | led5_r | M15 | |
| timeCounter[0] | | 7-segment LEDs | ex_D0 | U7 | |
| timeCounter[1] | | | ex_C0 | V6 | |
| timeCounter[2] | | | ex_B0 | V5 | |
| timeCounter[3] | | | ex_A0 | U5 | |

*Hình 4.1. Bảng chi tiết pin assignment*




| Name | Direction | Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std | Vcco | Vref | Drive Strength | Slew Type | Pull Type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| btminus | IN | | | L20 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| btplus | IN | | | L19 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| buttonPressed | IN | | | D19 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| clk | IN | | | H16 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| mode | IN | | | M19 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| reset | IN | | | D20 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| sel | IN | | | M20 | ✓ | 35 | LVCMOS33* | 3.300 | | | | NONE |
| pedestrianLight[2] | OUT | | | L15 | ✓ | 35 | LVCMOS33* | 3.300 | | 12 | | NONE |
| pedestrianLight[1] | OUT | | | G17 | ✓ | 35 | LVCMOS33 | 3.300 | | 12 | | NONE |
| pedestrianLight[0] | OUT | | | N15 | ✓ | 35 | LVCMOS33* | 3.300 | | 12 | | NONE |
| timeCounter[3] | OUT | | | U7 | ✓ | 13 | LVCMOS33* | 3.300 | | 12 | | NONE |
| timeCounter[2] | OUT | | | V6 | ✓ | 13 | LVCMOS33* | 3.300 | | 12 | | NONE |
| timeCounter[1] | OUT | | | V5 | ✓ | 13 | LVCMOS33* | 3.300 | | 12 | | NONE |
| timeCounter[0] | OUT | | | U5 | ✓ | 13 | LVCMOS33* | 3.300 | | 12 | | NONE |
| trafficLight[2] | OUT | | | G14 | ✓ | 35 | LVCMOS33* | 3.300 | | 12 | | NONE |
| trafficLight[1] | OUT | | | L14 | ✓ | 35 | LVCMOS33* | 3.300 | | 12 | | NONE |
| trafficLight[0] | OUT | | | M15 | ✓ | 35 | LVCMOS33* | 3.300 | | 12 | | NONE |

*Hình 4.2. Cài đặt pin assignment trên Vivado*

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] *3D traffic light - TurboSquid 1682361*,

https://www.turbosquid.com/3d-models/3d-traffic-light-1682361

[2] *Arty Z7 Reference Manual*,

https://digilent.com/reference/programmable-logic/arty-z7/reference-manual

[3] *Vivado Design Suite User Guide: I/O and Clock Planning (UG899)*,

https://docs.xilinx.com/v/u/2018.2-English/ug899-vivado-io-clock-planning

[4] *UltraFast Design Methodology Guide for the Vivado Design Suite (UG949)*,

https://docs.xilinx.com/v/u/2018.2-English/ug949-vivado-design-methodology

**HẾT**