

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỆN TỬ**

**BÁO CÁO THÍ NGHIỆM 5**  
**LAB 4: TỔNG HỢP VÀ THỰC HIỆN MẠCH SỐ TRÊN**  
**BREADBOARD – GIAO TIẾP VỚI FPGA**  
**HỆ TUẦN TỰ CƠ BẢN TRÊN FPGA**  
**MÔN: KỸ THUẬT SỐ (TN) (EE1010)**

**GVHD:** Nguyễn Trung Hiếu

**Sinh viên thực hiện**

**Nhóm 6 – Lớp L21**

- |                     |               |
|---------------------|---------------|
| 1) Lâm Thành Phát   | MSSV: 2111974 |
| 2) Nguyễn Đăng Khoa | MSSV: 2111529 |
| 3) Trần Thanh Tâm   | MSSV: 2114720 |

**TPHCM 12, 2022**

## A. HƯỚNG DẪN THÍ NGHIỆM

### I. MỤC TIÊU

- Áp dụng lý thuyết vào thiết kế trong thực tế.
- Kết hợp kiến thức ngôn ngữ mô tả phần cứng và mạch kỹ thuật số.

### II. CHUẨN BỊ:

- Hoàn thành đầy đủ các bài thí nghiệm 1, 2, 3, 4.
- Sinh viên phải hoàn thành và nộp PRELAB 5 trước khi vào lớp.

### III. HƯỚNG DẪN THÍ NGHIỆM

#### THÍ NGHIỆM 1

**Mục tiêu:** Thiết kế hệ thống đèn giao thông điều khiển thủ công

**Yêu cầu:** Thiết kế một hệ thống đèn giao thông cho một ngã tư có các thông số sau:

Đầu vào (input):

- Các nút nhấn bao gồm:
  - o 1 nút **RESET**. (Sử dụng nút Key0 trên Kit De 2)
  - o 1 Switch thay đổi hướng xe được di chuyển (Sử dụng nút SW1 trên Kit De2)

Ngõ ra (output):

- Sinh viên sử dụng ngõ ra Expansion Header trên Kit De 2 (dùng GPIO\_0) để kết nối tới các linh kiện sau:
- Hai bộ đèn (Hai bộ còn lại đối diện ở mỗi hướng là tùy chọn) bao gồm:
  - o 1 Led 7 đoạn đếm giây
  - o 1 Led Đỏ
  - o 1 Led Xanh lá
  - o 1 Led Vàng

Mô tả hoạt động:

- Sau khi RESET, hệ thống đèn giao thông tự động hoạt động theo chế độ
- Ở chế độ điều khiển:
  - o Hướng xe được cho phép di chuyển sẽ sáng đèn xanh lá
  - o Hướng còn lại sáng đèn đỏ
  - o Tất cả đèn đếm hiển thị 9 giây và không thay đổi giá trị
- Khi hướng xe thay đổi (ở chế độ điều khiển)
  - o Hướng xe được di chuyển trước đó chuyển từ đèn xanh lá sang đèn vàng, đèn đếm hiển thị 3 giây và đếm ngược xuống 0 rồi chuyển sang đèn đỏ
  - o Hướng còn lại giữ nguyên đèn đỏ cho đến khi hướng trước kia chuyển sang đèn đỏ, thời gian đếm ngược từ 3 giây đếm xuống 0.
  - o Lúc này, hướng được chuyển chuyển sang đèn xanh. Tất cả đèn đếm hiển thị 9 giây và không thay đổi giá trị

### **Kiểm tra:**

Sinh viên sử dụng kit De 2 với ngôn ngữ mô tả phần cứng SystemVerilog cho mạch mô tả phần cứng hoạt động như trên, kết nối từ GPIO 0 đến breadboard để thể hiện hệ thống đèn. Hệ thống nút nhấn, switch sử dụng trên kit như yêu cầu đề bài.

- o Viết lại đoạn mã SystemVerilog wrapper (top\_level) của thiết kế

```
module tn1(  
    input logic rst,x,clk,  
    output logic [2:0] led1,  
    output logic [2:0] led2,  
    output logic [3:0] dem1, dem2,  
    output logic out1,out2  
);  
parameter s0 = 3'b000,s1 = 3'b001, s2 = 3'b010, s3 = 3'b011, s4 = 3'b100, s5 = 3'b101, s6 = 3'b110;  
  
logic [2:0] pre,next;
```

```
integer j=0;

always_ff@(posedge clk) begin
j <= j + 1;
if(j==1000000000) begin
    out2 <= ~out2;
    j <= 0;
end
end

always @(posedge out2) begin
    if (rst) begin
        pre <=s0;
        end
    else pre <=next;
end

integer i=0;
always_ff@(posedge clk) begin
i <= i + 1;
if(i==250000000) begin
    out1 <= ~out1;
    i <= 0;
end
end

always @(pre or x) begin
case(pre)
    s0: if(x=='0') next = s1; else next = s4;
    s1: if(x=='0') next = s1; else next = s2;
    s2: if(x=='0') next = s0; else next = s3;
```

```
s3: if(x=='0') next = s0; else next = s4;  
s4: if(x=='0') next = s5; else next = s4;  
s5: if(x=='0') next = s6; else next = s0;  
s6: if(x=='0') next = s1; else next = s0;  
default: next = s0;
```

```
endcase
```

```
end
```

```
always @(posedge out1) begin
```

```
    if (pre ==s0) begin dem1 <= 4'b0000; dem2 <= 4'b0000; led1 = 3'b000; led2 = 3'b000;  
end
```

```
    if (pre ==s1) begin dem1 <= 4'b1001; dem2 <= 4'b1001; led1 = 3'b001; led2 = 3'b100;  
end
```

```
    if (pre ==s4) begin dem1 <= 4'b1001; dem2 <= 4'b1001; led1 = 3'b100; led2 = 3'b001;  
end
```

```
    if (pre ==s2) begin led2 = 3'b100; dem2 <= 4'b1001; led1 = 3'b010;
```

```
    if(dem1 == 4'b1001) dem1 <=4'b0011;
```

```
    else if(dem1 ==4'b0000) dem1 <= 4'b0000;
```

```
    else dem1 <= dem1 - 4'b0001;
```

```
    end
```

```
    if (pre ==s3) begin led2 = 3'b100; led1 = 3'b100; dem1 <= 4'b1001;
```

```
    if(dem2 == 4'b1001) dem2 <=4'b0011;
```

```
    else if(dem2 ==4'b0000) dem2 <= 4'b0000;
```

```
    else dem2 <= dem2 - 4'b0001;
```

```
    end
```

```
    if (pre ==s5) begin led1 = 3'b100; dem1 <= 4'b1001; led2 = 3'b010;
```

```
    if(dem2 == 4'b1001) dem2 <=4'b0011;
```

```
    else if(dem2 ==4'b0000) dem2 <= 4'b0000;
```

```
else dem2 <= dem2 - 4'b0001;  
end  
  
if (pre ==s6) begin led2 = 3'b100; led1 = 3'b100; dem2 <= 4'b1001;  
if(dem1 == 4'b1001) dem1 <=4'b0011;  
else if(dem1 ==4'b0000) dem1 <= 4'b0000;  
else dem1 <= dem1 - 4'b0001;  
end  
end  
endmodule
```

```
module seg7(  
    input logic [3:0] i,  
    output logic [6:0] o  
);  
  
always@(*) begin  
case(i)  
    4'b0000: o = 7'b1000000;  
    4'b0001: o = 7'b1111001;  
    4'b0010: o = 7'b0100100;  
    4'b0011: o = 7'b0110000;  
    4'b0100: o = 7'b0011001;  
    4'b0101: o = 7'b0010010;  
    4'b0110: o = 7'b0000010;  
    4'b0111: o = 7'b1111000;  
    4'b1000: o = 7'b0000000;  
endcase  
end
```



```
4'b1001: o = 7'b0010000;  
default: o = 7'b00000000;  
endcase  
  
end  
  
endmodule  
  
module lab1(  
input logic [1:0] SW,  
input logic CLOCK_50,  
output logic [4:0] LEDR, LEDG,  
output logic [6:0] HEX0, HEX1  
);  
//led 2, 1, 0 theo thứ tự đỏ vàng xanh//  
logic [3:0] dem1, dem2;  
tn1 tn(  
    .rst(SW[0]), .x(SW[1]), .clk(CLOCK_50), .led1(LEDR[2:0]), .led2(LEDG[2:0]),  
    .dem1(dem1), .dem2(dem2), .out1(LEDR[3]),  
    .out2(LEDR[4])  
);  
seg7 seg1(.i(dem1), .o(HEX0));  
  
seg7 seg2(.i(dem2), .o(HEX1));  
  
endmodule
```

## THÍ NGHIỆM 2

**Mục tiêu:** Thiết kế hệ thống đèn giao thông tự động, hai chế độ

**Yêu cầu:**

Thiết kế một hệ thống đèn giao thông cho một ngã tư có các thông số sau:

Đầu vào (input):

- Các nút nhấn bao gồm:
  - 1 nút **RESET**. (Sử dụng nút Key0 trên Kit De 2)
  - 1 Switch chuyển giữa chế độ tự động và điều khiển (Sử dụng SW0 trên Kit De2)
  - 1 Switch thay đổi hướng xe được di chuyển (Sử dụng nút SW1 trên Kit De2)

Ngõ ra (output):

- Thay thế kết nối giữa GPIO\_0 Kit De 2 đến Led 7 đoạn thành Kết nối từ GPIO\_0 Kit De 2 đến IC 74LS47 đến Led 7 đoạn.

Mô tả hoạt động:

- Sau khi RESET, hệ thống đèn giao thông tự động hoạt động theo chế độ cài ở SW0.
- Chế độ điều khiển như thí nghiệm 1
- Ở chế độ tự động:
  - Mỗi hướng xe thay nhau di chuyển
  - Hướng được di chuyển có đèn xanh trong thời hạn 5 giây đếm xuống 0, chuyển sang đèn vàng và từ 2 giây đếm xuống 0. Lúc này hướng chuyển sang đèn đỏ, thời gian đếm ngược 9 đếm xuống 0.
  - Hướng còn lại hiển thị đèn đỏ và thời gian đếm ngược 9 đếm xuống 0 (Trùng thời điểm hướng còn lại vừa đèn đỏ được 1 giây) và chuyển sang đèn xanh trong thời hạn 5 giây đếm xuống 0, chuyển sang đèn vàng và từ 2 giây đếm xuống 0. Và cứ thế tiếp tục.

**Lưu ý:**



Thời gian trong yêu cầu có tính chất tham khảo, sinh viên sử dụng và áp dụng cách đếm sao cho hệ thống hoạt động hợp lý.

### **Kiểm tra:**

Sinh viên sử dụng kit De 2 với ngôn ngữ mô tả phần cứng SystemVerilog cho mạch mô tả phần cứng hoạt động như trên, kết nối từ GPIO 0 đến breadboard để thể hiện hệ thống đèn. Hệ thống nút nhấn, switch sử dụng trên kit như yêu cầu đề bài

- Viết lại đoạn mã SystemVerilog wrapper (top\_level) của thiết kế

```
module tn2(
    input logic rst,x,clk,y,
    output logic [2:0] led1,led3,
    output logic [2:0] led2,led4,
    output logic [3:0] dem1, dem2, dem3, dem4,
    output logic out1,out2
);
parameter s0 = 3'b000,s1 = 3'b001, s2 = 3'b010, s3 = 3'b011, s4 = 3'b100, s5 = 3'b101, s6 = 3'b110;
/* cach 2
parameter z1 = 5'h0,z2 =5'h1,z3 =5'h2,z4 =5'h3,z5 = 5'h4,z6=5'h5,z7=5'h6,z8=5'h7,
z9 = 5'h8,z10 =5'h9,z11 =5'hA,z12 =5'hB,z13 = 5'hC,z14=4'hD,z15=5'hE, z16=5'hF,z17=5'h11,z18=5'h12,z19=5'h13,z20=5'h14;
logic [4:0] pre1,next1; */          //red blue yellow == bit2 bit1 bit0//
logic [2:0] pre,next;
logic [2:0] pre1,next1;
integer i = 0, k5=1, k6=1, k7=1; k8=1, j= 0;

always_ff@(posedge clk) begin
i <= i + 1;
if(i==250000000) begin
```



```
        out1 <= ~out1;

        i <= 0;

    end

end

always_ff@(posedge clk) begin
j <= j + 1;
if(j==1000000) begin
    out2 <= ~out2;

    j <= 0;

    end
end

always @(posedge out2) begin
if (y == '0) begin
    if (rst) begin
        pre <= s0;

        end

    else pre <= next;

end
end

always @(posedge out1) begin
if (y == '1) begin
    if (rst) begin
        pre1 <= z1;

        end

    else pre1 <= next1;

end
end
```

```
always @(pre or x) begin
if (y == '0) begin
case(pre)
s0: if(x=='0) next = s1; else next = s4;
s1: if(x=='0) next = s1; else next = s2;
s2: if(x=='0) next = s0; else next = s3;
s3: if(x=='0) next = s0; else next = s4;
s4: if(x=='0) next = s5; else next = s4;
s5: if(x=='0) next = s6; else next = s0;
s6: if(x=='0) next = s1; else next = s0;
default: next = s0;
endcase
end
end

always @(posedge out1) begin
if (y == '1) begin
case (pre1)
s1: begin
led3 = 3'b001; led4 = 3'b100;
if(dem3 > 4'h6||dem3 ==4'h0) begin
if(k5 == 0) begin dem3 = 4'b0000; k8 = 1; next1 = s2; end
else begin dem3 = 4'h6; k5 <= 0; end
end
else if(4'h0<dem3<=4'h6) dem3 = dem3 - 4'b0001;

if(dem4 > 4'h9||dem4 <=4'h3) begin
if(k5 == 0) begin dem4 = 4'b0011; end
else begin dem4 = 4'h9; end
```

```
        end
    else if(4'h3<dem4<=4'h9) dem4 = dem4 - 4'b0001;
    end

s2: begin
    led3 = 3'b010; led4 = 3'b100;
    if(dem3 > 4'h2 || dem3 ==4'h0) begin
        if(k6 == 0) begin dem3 <= 4'b0000; k5 = 1; next1 = s3; end
        else begin dem3 <= 4'h2; k6 <= 0; end
    end
    else if(4'h0<dem3<=4'h2) dem3 <= dem3 - 4'b0001;

    if(dem4 > 4'h2 || dem4 ==4'h0) begin
        if(k6 == 0) begin dem4 <= 4'b0000; end
        else begin dem4 <= 4'h2; end
    end
    else if(4'h0<dem4<=4'h2) dem4 <= dem4 - 4'b0001;
    end

s3: begin
    led4 = 3'b001; led3 = 3'b100;
    if(dem3 > 4'h9||dem3 <=4'h3) begin
        if(k7 == 0) begin dem3 = 4'b0011; k6 = 1; next1 = s4; end
        else begin dem3 = 4'h9; k7 <= 0; end
    end
    else if(4'h3<dem3<=4'h9) dem3 = dem3 - 4'b0001;

    if(dem4 > 4'h6||dem4 ==4'h0) begin
```

```
        if(k7 == 0) begin dem4 = 4'b0000; end
        else begin dem4 = 4'h6; end
        end
else if(4'h0 < dem4 <= 4'h6) dem4 = dem4 - 4'b0001;
end

s4: begin
led4 = 3'b010; led3 = 3'b100;
if(dem4 > 4'h2 || dem4 == 4'h0) begin
        if(k8 == 0) begin dem4 <= 4'b0000; k7 = 1; next1 = s1; end
        else begin dem4 <= 4'h2; k8 <= 0; end
        end
else if(4'h0 < dem4 <= 4'h2) dem4 <= dem4 - 4'b0001;

if(dem3 > 4'h2 || dem3 == 4'h0) begin
        if(k8 == 0) begin dem3 <= 4'b0000; end
        else begin dem3 <= 4'h2; end
        end
else if(4'h0 < dem3 <= 4'h2) dem3 <= dem3 - 4'b0001;
end

default: next = s1;
endcase
end
```

### THÍ NGHIỆM 3

**Mục tiêu:** Thiết kế hệ thống đèn giao thông tự động, hai chế độ như trên sử dụng IC 74LS47 kèm theo bảng quảng cáo LCD đặt ở góc ngã tư

**Yêu cầu:**

- Sinh viên kết nối đến module LCD 16 x 2 dùng GPIO\_1 tới breadboard, hiển thị đoạn quảng cáo “Thi Nghiem 5 KTS” ở hàng 1, “BM Dien Tu DHBK” ở hàng 2.



*Hình 5.6: LCD 2x16*

**Kiểm tra:**

Sinh viên sử dụng kit De 2 với ngôn ngữ mô tả phần cứng SystemVerilog cho mạch mô tả phần cứng hoạt động như trên, kết nối từ GPIO 0 đến breadboard để thể hiện hệ thống đèn. Hệ thống nút nhấn, switch sử dụng trên kit như yêu cầu đề bài.

```
module lcd(  
input logic CLOCK50,  
output logic [7:0] LCD_DATA,  
output logic LCD_RS,  
output logic LCD_EN,  
output logic LCD_ON,LCD_BLON  
);  
lcdmd lcd1 (  
.clk(CLOCK50),  
.lcd_e(LCD_EN),
```



```
.lcd_rs(LCD_RS),  
.data(LCD_DATA[7:0]),  
.lcd_on(LCD_ON),  
.lcd_blon(LCD_BLON)  
);  
endmodule
```

```
module lcdmd (  
    input logic clk,  
    output logic lcd_e, lcd_rs, lcd_on, lcd_blon,  
    output logic [7:0] data  
);  
assign lcd_on = '1;  
assign lcd_blon = '1;  
logic out;  
integer a = 0;  
always_ff@(posedge clk) begin  
    a <= a + 1;  
    if(a==25000) begin  
        out <= ~out;  
        a <= 0;  
    end  
end  
  
integer j = 1;  
logic [7:0] datas [1:38];  
always @(posedge out) begin
```

```
datas[1] = 8'h38;
datas[2] = 8'h0C;
datas[3] = 8'h06;
datas[4] = 8'h10;
datas[5] = 8'h80;
datas[6] = 8'h54; // T
datas[7] = 8'h68; // h
datas[8] = 8'h69; // i
datas[9] = 8'h20; // space
datas[10] = 8'h4E; // N
datas[11] = 8'h67; // g
datas[12] = 8'h68; // h
datas[13] = 8'h69; // i
datas[14] = 8'h65; // e
datas[15] = 8'h6D; // m
datas[16] = 8'h20; // space
datas[17] = 8'h35; // 5
datas[18] = 8'h20; // space
datas[19] = 8'h4B; // K
datas[20] = 8'h54; // T
datas[21] = 8'h53; // S
datas[22] = 8'h3C; // kích hoạt dòng 2
datas[23] = 8'hC1; // nhảy đến dòng 2 vì tri 1
datas[24] = 8'h42; // B
datas[25] = 8'h4D; // M
datas[26] = 8'h20; // space
datas[27] = 8'h44; // D
datas[28] = 8'h69; // i
```



```
datas[29] = 8'h65; // e
datas[30] = 8'h6E; // n
datas[31] = 8'h20; // space
datas[32] = 8'h54; // T
datas[33] = 8'h75; // u
datas[34] = 8'h20; // space
datas[35] = 8'h44; // D
datas[36] = 8'h48; // H
datas[37] = 8'h42; // B
datas[38] = 8'h4B; // K
end
integer i = 0;
always @(posedge out) begin
if (i<= 100000) begin i=i+1;
lcd_e = '1; data = datas[j]; end
else if(i>100000 && i<200000) begin
i = i+1;
lcd_e = '0; end
else if(i == 200000) begin
j=j+1;
i=0; end
else i=0;

if (j<=5 || j == 22 || j == 23) lcd_rs = 0;
else if (j>5 && j<21) lcd_rs = 1;

else if(j == 22 ) lcd_rs =0;
```



```
else if (j>24 && j<38) lcd_rs =1;  
  
else if(j >38) begin lcd_rs =1; j = 5; end  
else lcd_rs = 1;  
end  
endmodule
```



*Lab 5: Bài tổng hợp thực hiện mạch số trên breadboard – giao tiếp với FPGA.*

---