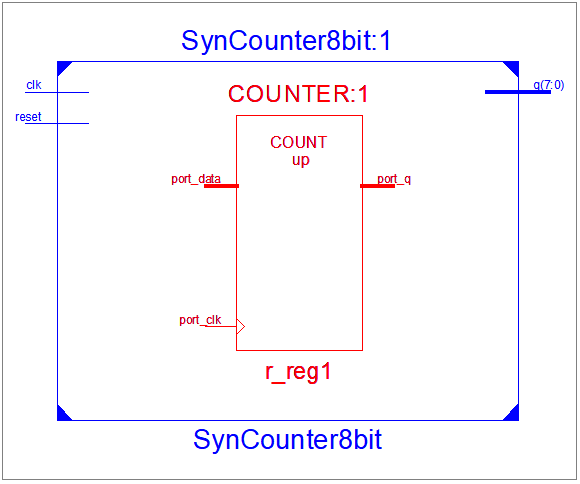
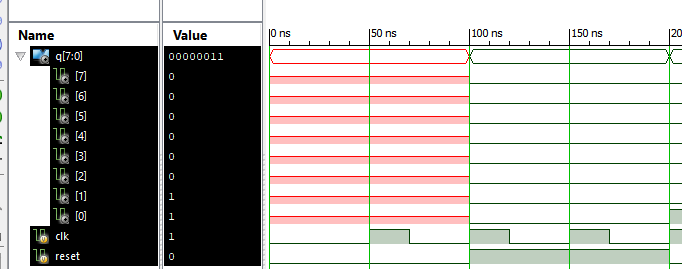
|  |  |
| --- | --- |
| BÁO CÁO THỰC HÀNH | TKHT - VMTH |
| Họ và tên | Nguyễn Hồng Đăng |
| MSSV | 20119127 |
| Lớp | 20119CLCA5 |
| Tuần | 3 |

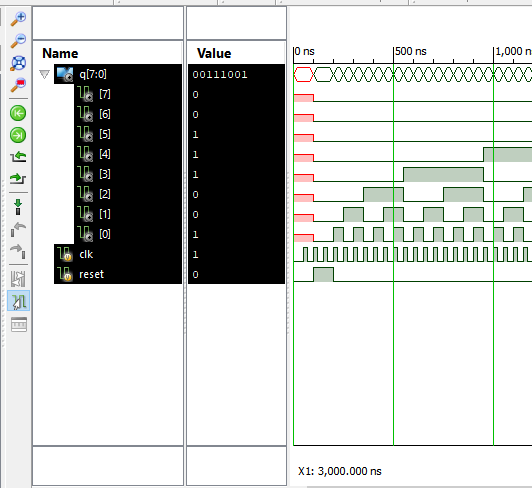
1. Thiết kế và mô phỏng mạch đếm đồng bộ 8 bit, tần số ngõ vào có độ động 20ns, chu kỳ 50ns.
   1. RTL Schematic



* 1. Dạng sóng mô phỏng và phân tích

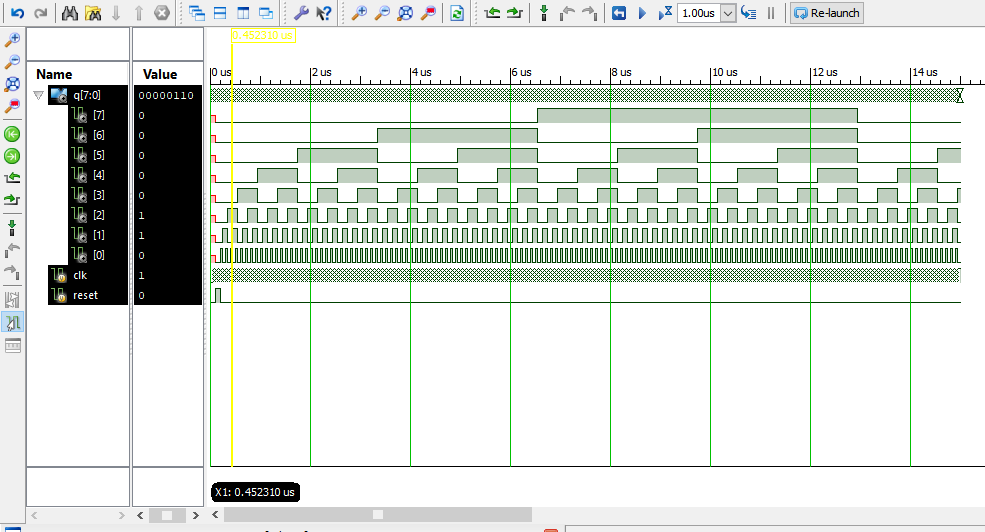


Dạng sóng xung clock đầu vào đúng với yêu cầu chu kì 50ns, độ đọng 20ns



Khi có tín hiệu reset mức cao, tất cả ngõ ra đều bằng 0.

Khi tín hiệu reset mức thấp, mạch bắt đầu hoạt dộng đếm lên



Khi đếm đến 1111 1111 mạch bắt đầu quay lại đếm từ 0000 0001 lần nữa.

1. Thiết kế mạch chia xung, sử dụng mô hình mạch đếm tuần tự đồng bộ, ngõ vào clkIn tần số 50Mhz, ngõ ra clkOut có tần số thấp hơn.
   1. Code Verilog

module machchiaxung50M#(parameter N= 26)(

input wire clk, reset,

output wire clkOut

);

reg [N-1 : 0]r\_reg;

wire [N-1 : 0]r\_next;

// body, register

always @(posedge clk, posedge reset)

if (reset)

r\_reg <= 0;

else

r\_reg <= r\_next; // <= is non-blocking statement

// next state logic

assign r\_next = r\_reg + 1;

// output logic

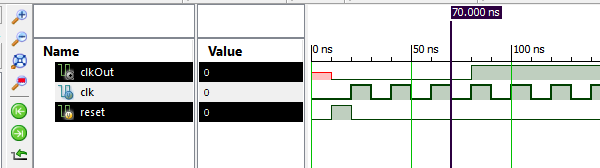
assign clkOut = r\_reg[2];

endmodule

* 1. Tính tần số xung ngõ ra

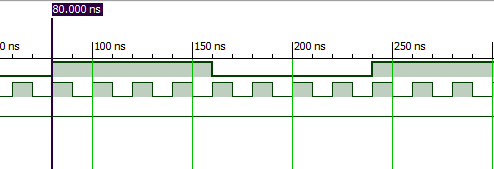
Ta có thể dễ dàng tính được tần số ngõ ra bằng công thức:

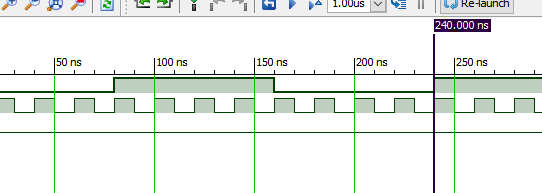
F(input)/2^(n) = 50\*(10^6)/2^2 = 12.5\*(10^6) Hz



Ta có thể thấy chu kì xung clock lúc này là 70-50=20ns => f=1/(20\*10^-9)=50\*10^(-6) Hz

⬄ 50MHz

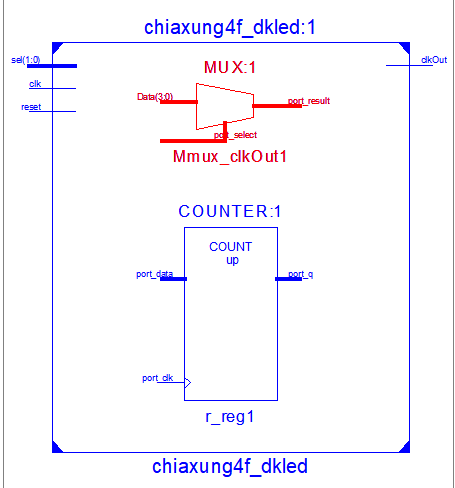




Ta có thể thấy rõ chu kì clkOut lúc này là 240-8=160 ns => f =12.5 MHz đúng với công thức trên.

1. Thiết kế mạch chia xung với 1 tần số ngõ vào clkIn = 50Mhz, tần số ngõ ra clkOut thấp hơn được dùng để điểu khiển LED, 2 chân lựa chọn S1, S0 cho phép chọn 4 tần số khác nhau

3.1 RTL schematic



3.2. Verilog Code

module chiaxung4f\_dkled #(parameter N=26)(

input wire clk,reset,

input wire [1:0]sel,

output reg clkOut

);

reg [N-1 : 0]r\_reg;

wire [N-1 : 0]r\_next;

always@(posedge clk ,posedge reset)

if(reset)

r\_reg <= 0;

else

r\_reg <= r\_next;

assign r\_next = r\_reg + 1;

always@(sel)

case(sel)

0 : clkOut = r\_reg[22];

1 : clkOut = r\_reg[23];

2 : clkOut = r\_reg[24];

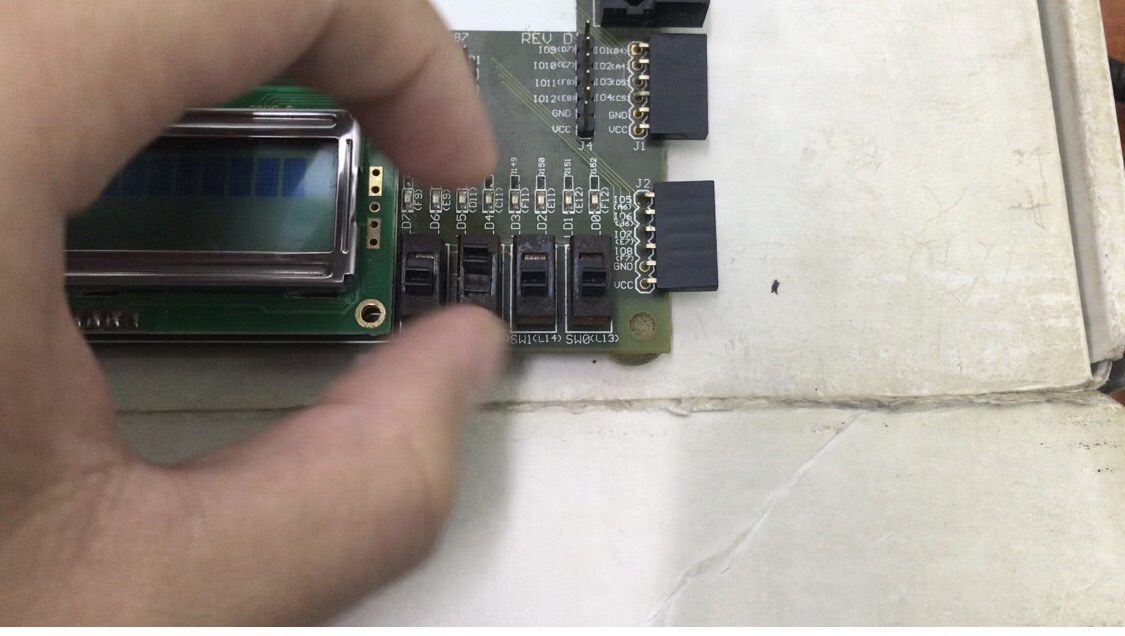
3 : clkOut = r\_reg[25];

endcase

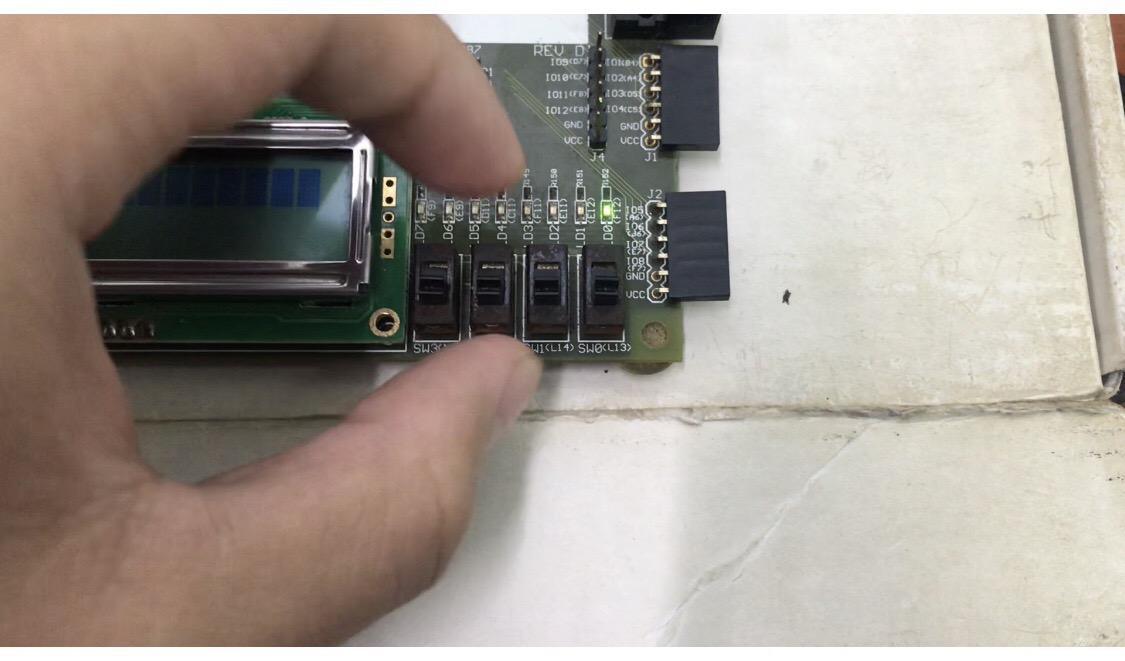
endmodule

3.3. Kết quả thực nghiệm

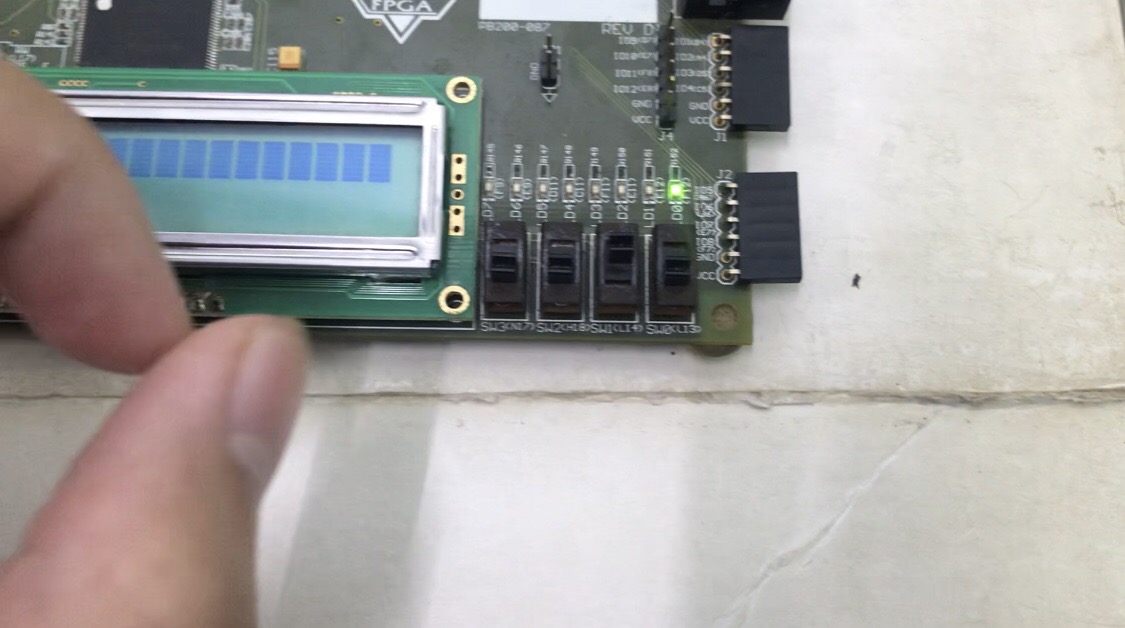
(sinh viên mô tả kết quả thực nghiệm, thông qua hình ảnh thực tế)



Reset high, led off.

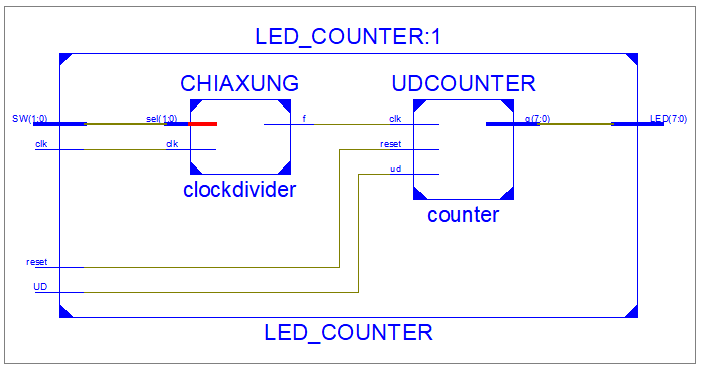


Reset low, sel = 2’b00. Frequency = 11.9 Hz



Reset low, sel = 2’b10, The frequency lower, F = 2.98 Hz

1. Thiết kế mạch tự động đếm lên 8bit kết nối với 8 led đơn, lựa chọn tần số đếm bằng 2 switch S1, S0, lựa chọn đếm lên hoặc đếm xuống bằng switch S2. Mỗi mô đun được thiết kế với 1 tập mã nguồn (.v) khác nhau.
   1. Sơ đồ RTL



* 1. Verilog code
     1. Module (Chiaxung):

module CHIAXUNG

(input wire clk,

input wire [1:0]sel,

output reg f

);

reg [25:0]cnt;

initial

begin

cnt = 1;

f=0;

end

always@(posedge clk)

begin

case(sel)

0 : begin

if(cnt == 2) //25MHz

begin f = ~f; cnt =1;end

else cnt = cnt +1;

end

1: begin

if(cnt == 12\_500\_000) //2Hz

begin f = ~f; cnt =1;end

else cnt = cnt +1;

end

2: begin

if(cnt == 25\_000\_000) //1Hz

begin f = ~f; cnt =1;end

else cnt = cnt +1;

end

3: begin

if(cnt == 35\_355\_339) //0.1Hz

begin f = ~f; cnt =1;end

else cnt = cnt +1;

end

endcase

end

endmodule

* + 1. Module UDCOUNTER

module UDCOUNTER // 500,000,000 for 0.1Hz

(input wire clk,reset,ud,

output wire [7:0]q

);

integer cnt=0;

always @(posedge clk, posedge reset)

if (reset)

cnt = 0 ;

else if(cnt==8'b1111\_1111)

cnt=0;

else if(ud==0)

cnt = cnt + 1;

else cnt = cnt -1;

assign q = cnt;

endmodule

* + 1. Module LEDCOUNTER

module LED\_COUNTER(

input wire clk, reset,

input wire [1:0] SW,

input wire UD,

output wire [7:0] LED

);

// wire declaration

wire n1;

// module instance

CHIAXUNG clockdivider (.clk(clk),.sel(SW),.f(n1));

UDCOUNTER counter (.clk(n1),.reset(reset),.ud(UD),.q(LED));

endmodule

* 1. Kết quả thực nghiệm

1. Thiết kế mạch điều khiển 8 led sáng dần rồi tắt dần từ trái sang phải hoặc từ phải sang trái, được lựa chọn bởi 1 switch, tần số dịch được chia từ 50Mhz.
   1. Verilog Code
      1. Module điều chế tần số 50Mhz về 1Hz

module chiaxung50Mto1Hz(

input clk50m,

output reg clk1hz

);

reg [24:0]cnt;

initial

begin

cnt <= 1; clk1hz <= 0;

end

always@(posedge clk50m)

if(cnt == 25000000)

begin clk1hz <= ~clk1hz; cnt <= 1; end

else cnt <= cnt + 1;

endmodule

* + 1. Modulde điều khiển led

module dkledsangtatdan(

input wire clk,

input wire reset,

input wire mode,

output wire [7:0]led );

reg [7:0]out;

integer a; //tat dan

integer b;//tat dan

always@(posedge clk)

if(reset) begin out = 8'b0000\_0000;a=1;b=1;end

else

if(mode == 0) //TSP

if (out==8'b1111\_1111)

begin out=out>>1; a =1;end

else if(out== 8'b0000\_0000)

begin out[7]=1; a=0;end

else if(a==1)

begin out=out>>1;end

else begin out=out>>1;out[7]=1;end

else //mode=1 //PST

if (out==8'b1111\_1111)

begin out=out<<1 ; b=1; end

else if(out== 8'b0000\_0000)

begin out[0] = 1;b=0; end

else if(b==1)

begin out = out<<1; end

else begin out=out<<1 ;out[0]=1;end

assign led = out;

endmodule

* + 1. Modulde General (tổng quát)

module generalmodule(

input wire clk50M,

input wire reset,

input wire mode,

output wire [7:0]led

);

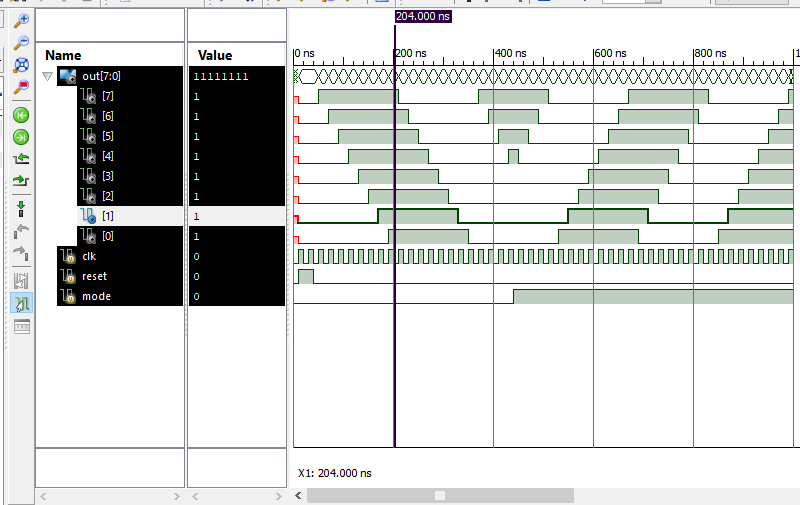
wire n1;

chiaxung50Mto1Hz dieuche1hz (.clk50m(clk50M),.clk1hz(n1));

dkledsangtatdan dieukhienled (.clk(n1),.reset(reset),.mode(mode),.out(led));

endmodule

* 1. Kết quả thực nghiệm



1. Thiết kê mạch điều khiển 1 led chạy từ trái sang phải rồi tự động chạy từ phải sang trái
   1. Verilog code
      1. Module dieuchexung1Hz

module dieuchexung1Hz(

input clk,

output reg clk1hz

);

reg [24:0]cnt;

initial

begin

cnt <= 1; clk1hz <= 0;

end

always@(posedge clk)

if(cnt == 25000000)

begin clk1hz <= ~clk1hz; cnt <= 1; end

else cnt <= cnt + 1;

endmodule

* + 1. Module dịch led từ trái sang phải và phải sang trái tự dộng

module dichledTSP\_PST(

input wire reset,

input wire clk,

output reg [7:0]led

);

initial

begin

led = 8'b1000\_0000;

end

reg s =0;

always@(posedge clk,posedge reset)

if(reset==1)

begin led=8'b0000\_0000; end

else

if(led == 8'b0000\_0000)

begin led = 8'b1000\_0000; end

else if(led == 8'b0000\_0001)

begin led = led <<1; s=1; end //s=1 dich sang trai

else if(led == 8'b1000\_0000)

begin led = led >>1; s=0; end

else if(s==0) //s=0 dich sang phai

begin led = led >>1; end

else begin led = led <<1;end

endmodule

* + 1. Module tổng

module ledtudongTSP\_PST(

input wire clk,

input wire reset,

output wire [7:0]Led

);

wire n1;

dieuchexung1Hz xung (.clk(clk),.clk1hz(n1));

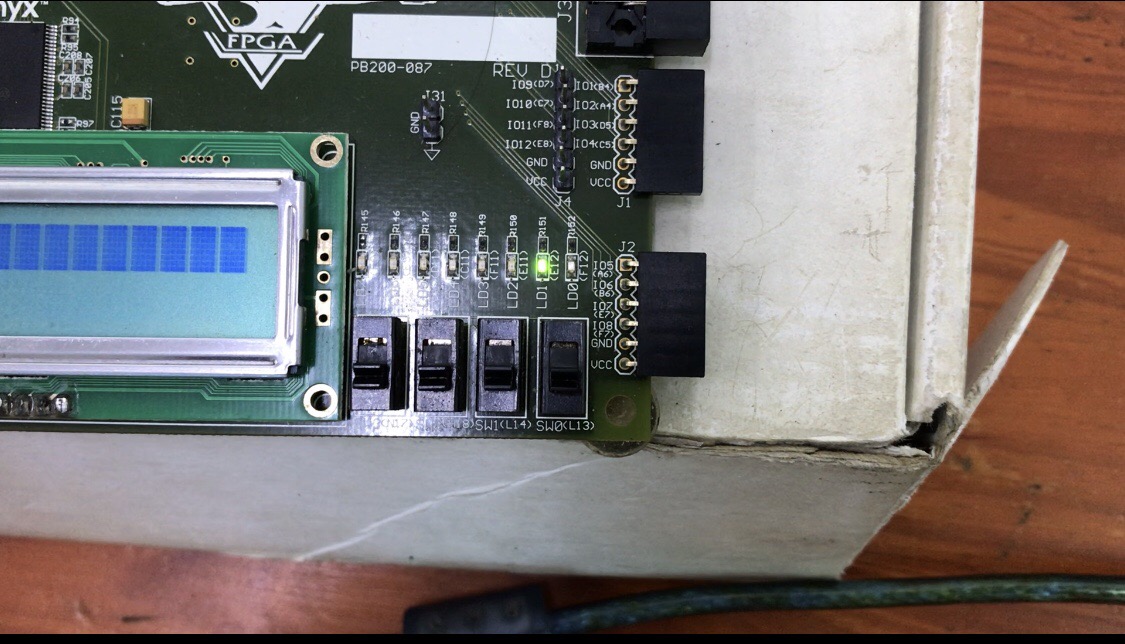
dichledTSP\_PST dichled (.reset(reset),.clk(n1),.led(Led));

endmodule

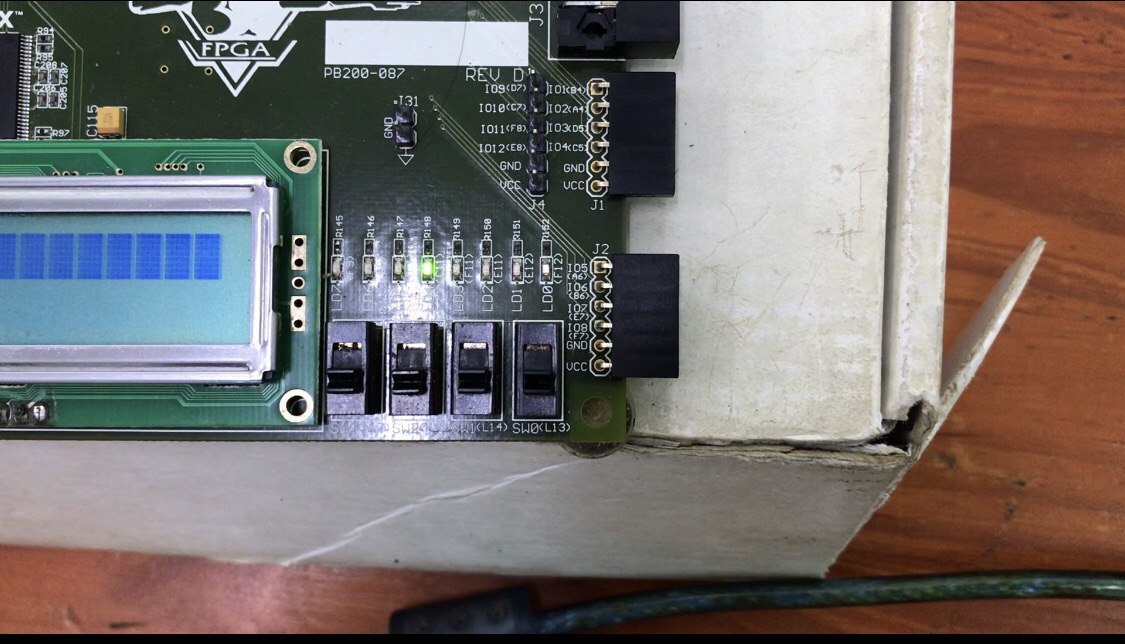
* 1. Kết quả thực hiện



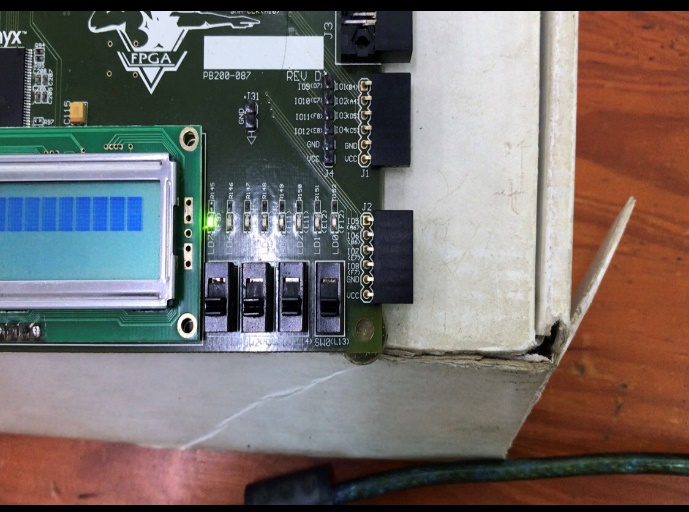
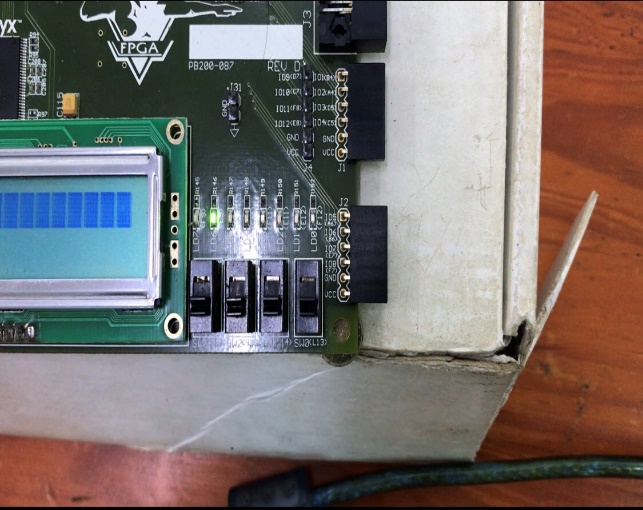
Led đầu tiên sáng, sau đó dịch đến led cuối cùng bên phải .



Sau khi led cuối cùng sáng thì led tự dộng dịch sang trái.



Ảnh trên là quá trình led đang dịch sang trái.

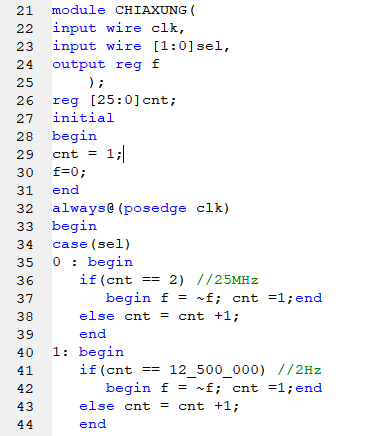
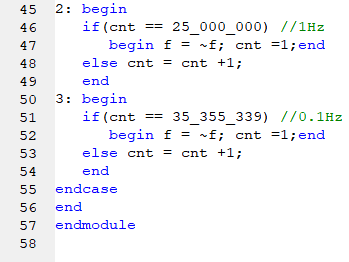
Sau khi tới led đầu tiên bên trái led bắt đầu tự dộng dịch ngược lại sang phải.

1. Thiết kế mạch gồm 8 led đơn, 4 switch S1, S2, S3, S4

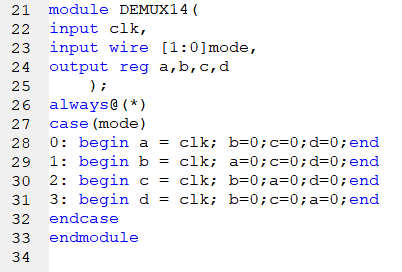
* S1, S2, lựa chọn 1 trong 4 tần số
* S3, S4, lựa chọn mode cho 8 led.
* Mode =1: 8 led chớp tắt,
* Mode =2: 8 led sáng dần, tắt dần,
* Mode = 3: 1 led sáng chạy từ trái sang phải, rồi từ phải sang trái.
* Mode = 4: 8 led sáng dồn

7.1 Verilog code

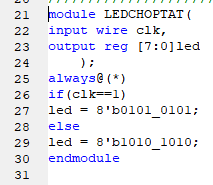
Module chia xung:

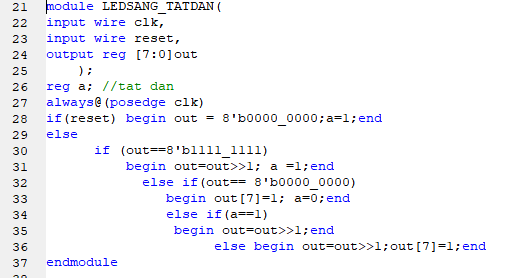
Module DEMUX14



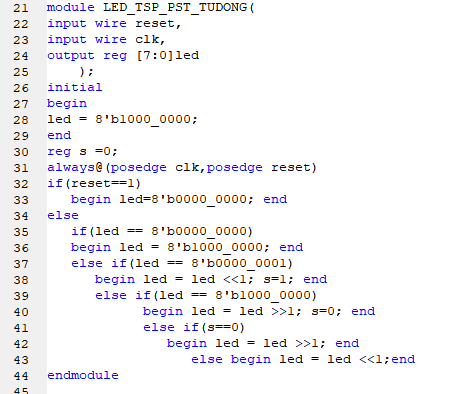
Module Led chớp tắt:



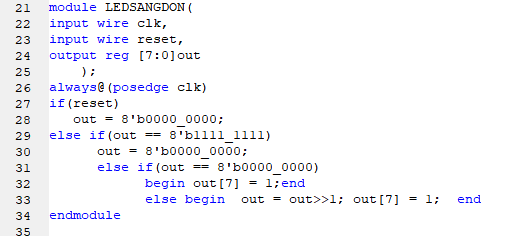
Module Led sáng dần rồi tắt dần từ trái qua phải



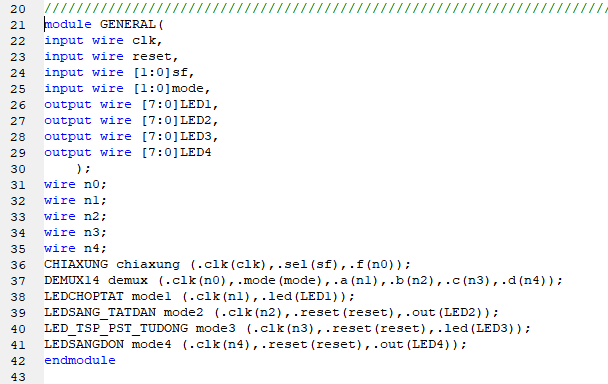
Module Led tự động dịch sang trái và dịch sang phải



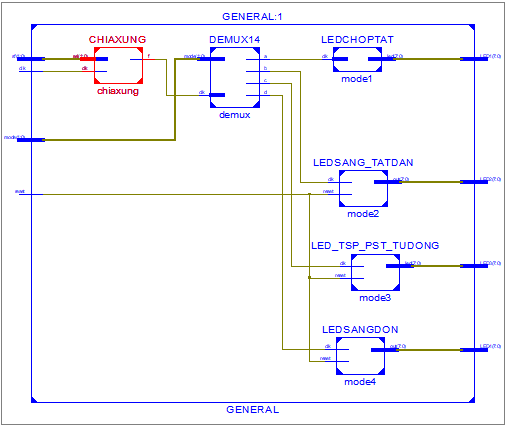
Module Led sáng dồn



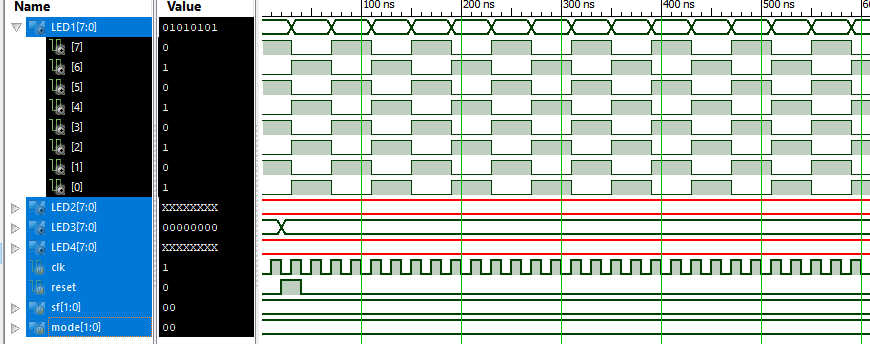
Module Tổng hợp



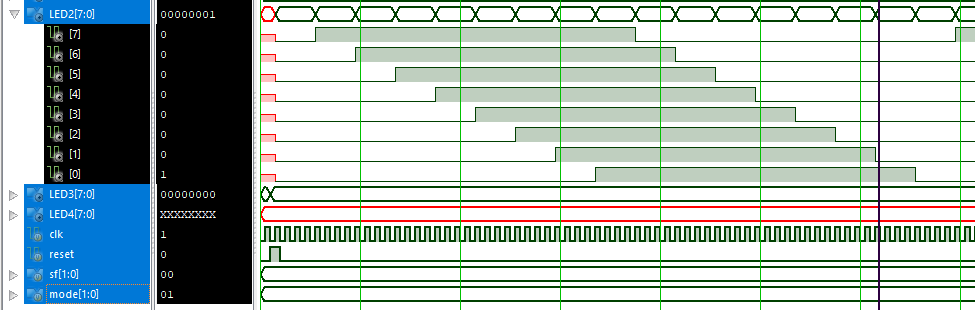
7.2 Kết quả thực hiện



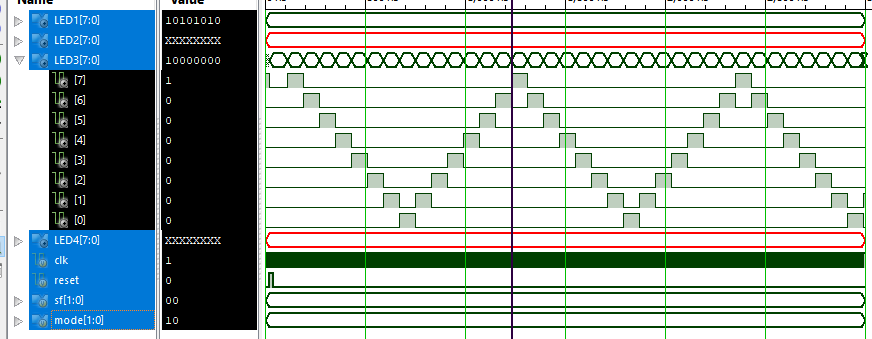
Mode 1:



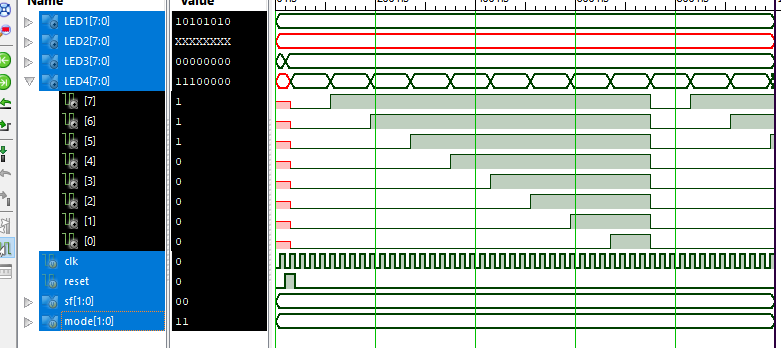
Mode 2:



Mode 3:



Mode 4:



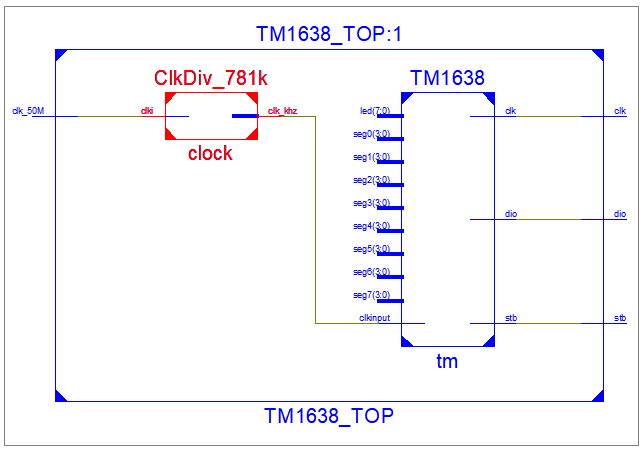
1. Thiết kế module mở rộng led TM1638.
   1. Sinh viên đọc tại liệu kỹ thuật board mở rộng TM1638

* Giải thích quá trình điều khiển led (đơn, 7 đoạn) trên Môđun TM1638

(giải thích dạng sóng tín hiệu giao tiếp, phương pháp giao tiếp và điều khiển, địa chỉ các vùng nhớ trên TM1638)

* Viết chương trình điều khiển module TM1638- test led đơn và led 7 đoạn

RTL schematic and verilog Code



* 1. Viết các chương trình điều khiển led trên module TM1638
     1. Điều khiển 8 led đơn sáng dần rồi tắt hết

Verilog code:

* + 1. Điều khiển đếm lên, đếm xuống từ 0000 để 9999 trên led 7 đoạn, tần số đếm gần 1Hx

Verilog Code

* + 1. Viết chương trình hiển thị giờ phút giây trên led 7 đoạn

Verilog Code