# Hệ Thống Số

Ôn tập giữa kì 20181 - Lưu

**Bài 1:** Hệ thống đèn giao thông tại một ngã tư

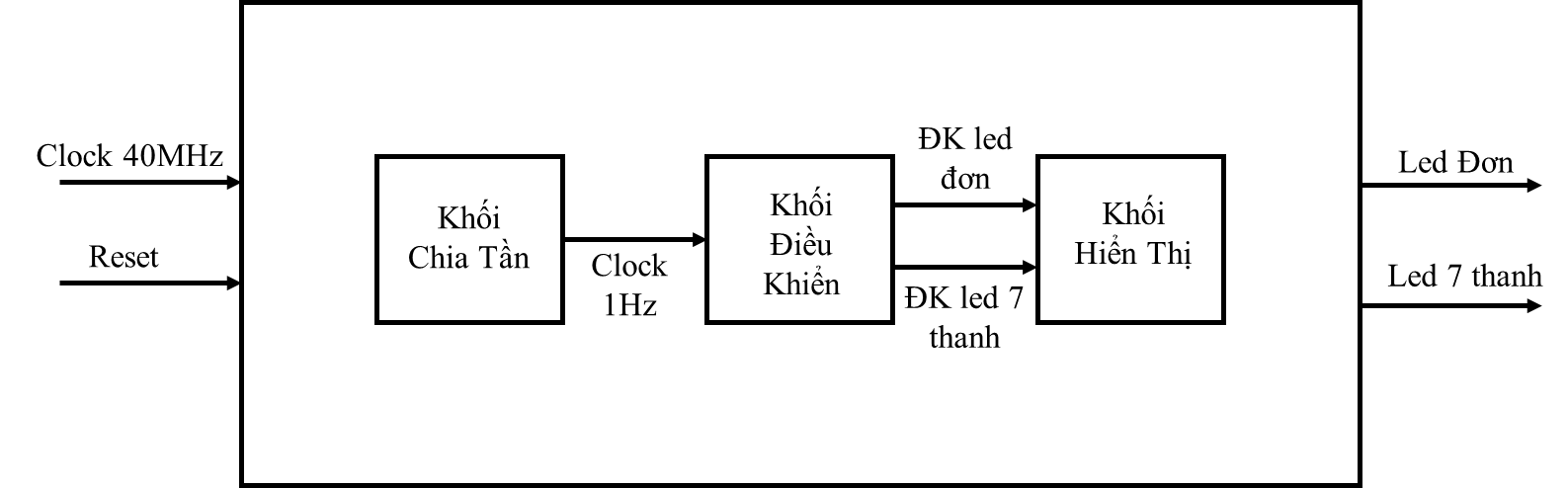
* Tần số chính 40 MHz
* Đèn đỏ dừng 50 giây
* Đèn xanh dừng 40 giây
* Đèn vàng dừng 10 giây
* Hệ thống gồm 3 led đơn điều khiển màu và 2 led 7 thanh hiển thị thời gian đếm lùi

Yêu cầu

1. Vẽ sơ đồ liên kết giữa các khối bên trong hệ thống
2. Viết code verilog cho các khối kể cả top level entity
3. Viết code testbench mô phỏng cho hệ thống

**Giải:**

1. **Sơ đồ liên kết giữa các khối bên trong hệ thống**

****

1. **Code verilog các khối**

Khối Top level entity

|  |
| --- |
| /\* Module Top level entity  \* Den Giao thong  \*/  module DenGiaoThong  (  clock, //Clock 40 MHz  reset, //Reset  ledRed, //Led don  ledGreen,  ledYellow,  led7Seg1, //Led 7 thanh hang chuc  led7Seg0 //Led 7 thanh hang don vi  );  // Khai bao tin hieu  input clock, reset;  output ledRed,ledGreen,ledYellow;  output [6:0] led7Seg1,led7Seg0;  wire [6:0] led7Seg1,led7Seg0;  wire clock1hz;  wire [3:0] controlLed7Seg1, controlLed7Seg0;  wire [2:0] ledSingle;  // Gan tin hieu vao cac khoi  DivisionFrequency T1  (  .clockIn (clock),  .clockOut (clock1hz)  );  Controller T2  (  .clock (clock1hz),  .reset (reset),  .ledSingle (ledSingle),  .controlLed7Seg1 (controlLed7Seg1),  .controlLed7Seg0 (controlLed7Seg0)  );  Display T3  (  .ledSingle (ledSingle),  .controlLed7Seg1 (controlLed7Seg1),  .controlLed7Seg0 (controlLed7Seg0),  .ledRed (ledRed),  .ledGreen (ledGreen),  .ledYellow (ledYellow),  .led7Seg1 (led7Seg1),  .led7Seg0 (led7Seg0)  );  endmodule |

Khối Chia Tần

|  |
| --- |
| module DivisionFrequency  (  clockIn, //Xung clock vao  clockOut //Xung clock ra  );  input clockIn;  output clockOut;  reg clockOut;  parameter scale = 40000000;  integer threshold = scale/2;  integer i = 0;  initial  clockOut = 0;  always @ (negedge clockIn )  begin  if(i == threshold-1)  begin  clockOut <= !clockOut;  i <= 0;  end  else  i <= i + 1;  end  endmodule |

Khối điều khiển

|  |
| --- |
| module Controller  (  clock, // Clock 40MHz  reset, // Reset tich cuc duong  ledSingle, // Tin hieu led don  controlLed7Seg1, // Tin hieu led 7 thanh hang chuc  controlLed7Seg0 // Tin hieu led 7 thanh hang don vi  );  // Khai bao tin hieu  input clock, reset;  output reg [2:0] ledSingle;  output reg [3:0] controlLed7Seg1,controlLed7Seg0;  parameter GreenTime = 40;  parameter YellowTime = 10;  parameter RedTime = 50;    parameter GreenState = 2'b00;  parameter YellowState = 2'b01;  parameter RedState = 2'b10;  reg [1:0] currentState, nextState;  initial  begin  currentState = GreenState;  nextState = YellowState;  controlLed7Seg1 <= GreenTime / 10;  controlLed7Seg0 <= GreenTime % 10;  end  always @ (negedge clock or posedge reset)  begin  if(reset)  begin  currentState = GreenState;  controlLed7Seg1 <= GreenTime / 10;  controlLed7Seg0 <= GreenTime % 10;  end  else  if( controlLed7Seg1 == 0 && controlLed7Seg0 == 0)  begin  currentState <= nextState;  end  else  begin  if (controlLed7Seg0 == 0)  begin  controlLed7Seg1 = controlLed7Seg1 - 1;  controlLed7Seg0 = 9;  end  else  controlLed7Seg0 = controlLed7Seg0 - 1;  end  end  always @ (currentState)  begin  case(currentState)  GreenState:  begin  nextState = YellowState;  controlLed7Seg1 = GreenTime / 10;  controlLed7Seg0 = GreenTime % 10;  end  YellowState:  begin  nextState = RedState;  controlLed7Seg1 = YellowTime / 10;  controlLed7Seg0 = YellowTime % 10;  end  RedState:  begin  nextState = GreenState;  controlLed7Seg1 = RedTime / 10;  controlLed7Seg0 = RedTime % 10;  end  endcase  end  always @ (currentState)  begin  case(currentState)  GreenState: ledSingle = 3'b100;  YellowState: ledSingle = 3'b010;  RedState: ledSingle = 3'b001;  endcase  end  endmodule |

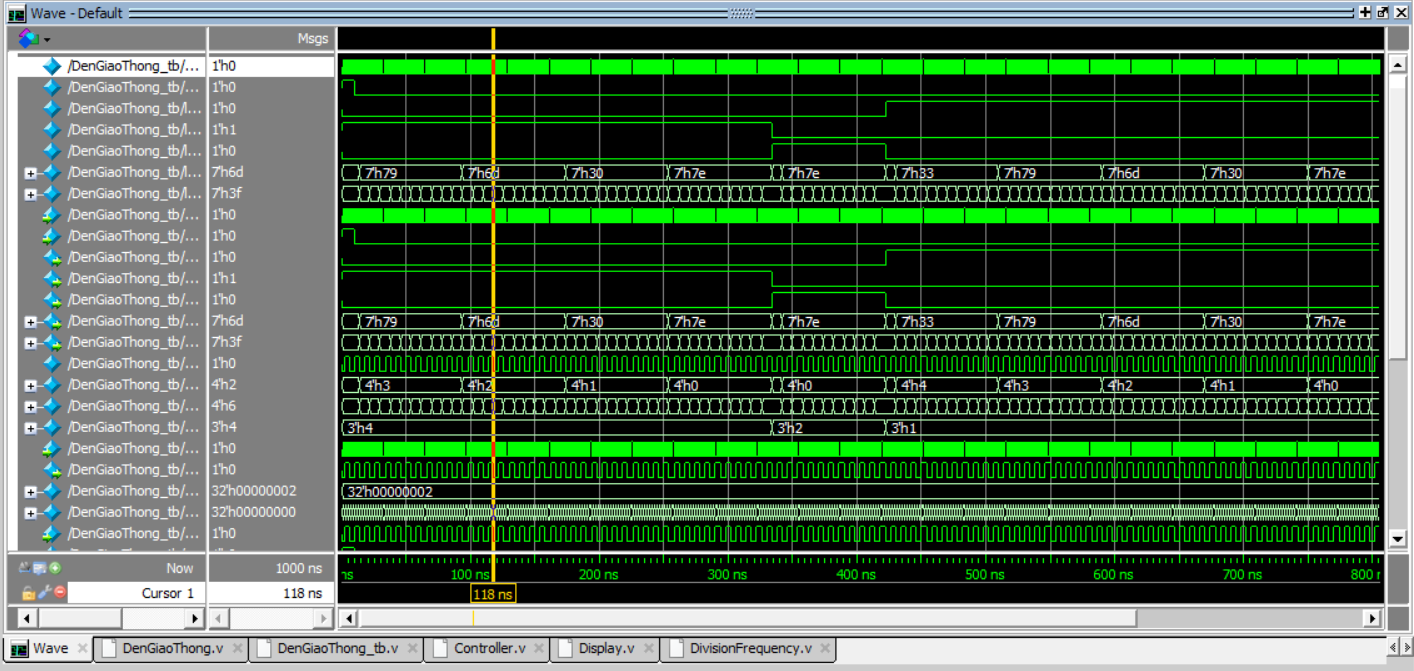
Khối hiển thị

|  |
| --- |
| module Display  (  ledSingle, // Input led don  controlLed7Seg1, // Input led 7 thanh hang chuc  controlLed7Seg0, // Input led 7 thanh hang don vi  ledRed, // Output led do  ledGreen, // Output led xanh  ledYellow, // Output led vang  led7Seg1, // Output led 7 thanh hang chuc  led7Seg0 // Output led 7 thanh hang don vi  );  input [2:0] ledSingle;  input [3:0] controlLed7Seg1,controlLed7Seg0;  output ledRed,ledGreen,ledYellow;  output reg [6:0] led7Seg1,led7Seg0;  parameter num0 = 7'b1111110, //7E  num1 = 7'b0110000, //30  num2 = 7'b1101101, //6D  num3 = 7'b1111001, //79  num4 = 7'b0110011, //33  num5 = 7'b1011011, //5B  num6 = 7'b0111111, //5F  num7 = 7'b1110000, //70  num8 = 7'b1111111, //7F  num9 = 7'b1111011; //7B  assign ledGreen = ledSingle[2];  assign ledYellow = ledSingle[1];  assign ledRed = ledSingle[0];  always @ (controlLed7Seg1)  begin  case(controlLed7Seg1)  4'b0000 : led7Seg1 = num0;  4'b0001 : led7Seg1 = num1;  4'b0010 : led7Seg1 = num2;  4'b0011 : led7Seg1 = num3;  4'b0100 : led7Seg1 = num4;  4'b0101 : led7Seg1 = num5;  4'b0110 : led7Seg1 = num6;  4'b0111 : led7Seg1 = num7;  4'b1000 : led7Seg1 = num8;  4'b1001 : led7Seg1 = num9;  default : led7Seg1 = num0;  endcase  end  always @ (controlLed7Seg0)  begin  case(controlLed7Seg0)  4'b0000 : led7Seg0 = num0;  4'b0001 : led7Seg0 = num1;  4'b0010 : led7Seg0 = num2;  4'b0011 : led7Seg0 = num3;  4'b0100 : led7Seg0 = num4;  4'b0101 : led7Seg0 = num5;  4'b0110 : led7Seg0 = num6;  4'b0111 : led7Seg0 = num7;  4'b1000 : led7Seg0 = num8;  4'b1001 : led7Seg0 = num9;  default : led7Seg0 = num0;  endcase  end  endmodule |

Khối testbench

|  |
| --- |
| module DenGiaoThong\_tb;  reg clock, //Clock 40 MHz  reset; //Reset  wire ledRed, //Led don  ledGreen,  ledYellow;  wire [6:0] led7Seg1, //Led 7 thanh hang chuc  led7Seg0;  DenGiaoThong DUT  (  .clock (clock), //Clock 40 MHz  .reset (reset), //Reset  .ledRed (ledRed), //Led don  .ledGreen (ledGreen),  .ledYellow (ledYellow),  .led7Seg1 (led7Seg1), //Led 7 thanh hang chuc  .led7Seg0 (led7Seg0) //Led 7 thanh hang don vi  );  initial  begin  clock = 0;  reset = 1;  #10  reset = 0;  end  always  #1 clock = !clock;  endmodule |

Kết quả chạy testbench trên ModelSim10.4a



Đóng gói code tại