**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**BÀI THỰC HÀNH 3**

**THIẾT KẾ MÁY TRẠNG THÁI HỮU HẠN**

|  |  |
| --- | --- |
| **HỌ VÀ TÊN:** | **VÕ ĐÔNG THÀNH** |
| **MSSV:** | **21520457** |
| **LỚP:** | **CE213.O11.1** |

**GIẢNG VIÊN HƯỚNG DẪN:**

**HỒ NGỌC DIỄM**

**I. Mục tiêu**

- Trong bài thực hành này, sinh viên sẽ dùng sử dụng ngôn ngữ Verilog HDL thiết kế mạch tuần tự theo mô hình máy trạng thái hữu hạn (FSM).

**II. Chuẩn bị thực hành**

- Sinh viên phải chuẩn bị code Verilog cho tất cả các câu trong phần nội dung thực hành và nộp cho GVHD vào đầu buổi học.

- Sinh viên nào không có bài chuẩn bị được xem là vắng buổi học hôm đó.

- Điểm bài chuẩn bị được tính vào điểm bài báo cáo của Lab.

**III. Nội dung thực hành**

**Câu 1.**

Sử dụng Verilog HDL thiết kế một mạch tuần tự theo mô hình máy trạng thái có chức năng phát hiện hai chuỗi cụ thể của ngõ vào, cụ thể là bốn số 1 liên tiếp hoặc bốn số 0 liên tiếp. Mạch có một ngõ vào x và một ngõ ra z. Bất cứ khi nào x = 1 hoặc x = 0 trong bốn xung đồng hồ liên tiếp, giá trị của z = 1; mặt khác, z = 0. Cho phép chuỗi ngõ vào được chồng lấp nhau (overlapped), tức là nếu x = 1 trong năm xung clock liên tiếp thì ngõ ra z sẽ bằng 1 sau xung thứ tư và thứ năm. Sử dụng công tắc SW0 trên bo Altera DE2 làm ngõ vào x, LEDG0 làm ngõ ra z và nút ấn KEY0 làm xung clock được áp dụng thủ công. Mô phỏng hoạt động của mạch và kiểm tra chức năng của mạch trên board DE2.State:

A diagram of a diagram

Description automatically generated

Code:

|  |
| --- |
| module Cau1(  input w, clk, reset,  output reg z  );  reg [3:0]state, next\_state;  parameter s0 = 4'b0000,  s1 = 4'b0001,  s2 = 4'b0011,  s3 = 4'b0010,  s4 = 4'b0110,  s5 = 4'b0111,  s6 = 4'b0101,  s7 = 4'b0100,  s8 = 4'b1100;  always @(state or w)  case (state)    s0:  begin  z = 0;  if (w == 1'b0)  next\_state = s1;  else  next\_state = s5;  end    s1:  begin  z = 0;  if (w == 1'b0)  next\_state = s2;  else  next\_state = s5;  end    s2:  begin  z = 0;  if (w == 1'b0)  next\_state = s3;  else  next\_state = s5;  end    s3:  begin  z = 0;  if (w == 1'b0)  next\_state = s4;  else  next\_state = s5;  end    s4:  begin  z = 1;  if (w == 1'b0)  next\_state = s4;  else  next\_state = s5;  end    s5:  begin  z = 0;  if (w == 1'b1)  next\_state = s6;  else  next\_state = s1;  end    s6:  begin  z = 0;  if (w == 1'b1)  next\_state = s7;  else  next\_state = s1;  end    s7:  begin  z = 0;  if (w == 1'b1)  next\_state = s8;  else  next\_state = s1;  end    s8:  begin  z = 1;  if (w == 1'b1)  next\_state = s8;  else  next\_state = s1;  end    endcase    always @(posedge clk or posedge reset)  begin  if (reset == 1'b1)  state <= s0;  else  state <= next\_state;  end    endmodule |

RTL Viewer:

A yellow rectangular object with text

Description automatically generated

Câu 2:

Hiện thực mạch báo đèn giao thông như mình hoạ trong Hình 3.1. Các đèn giao thông được đặt ở ngã tư giao nhau của một trục đường hướng bắc-nam và một trục đường hướng đông-tây. Tập các đèn giao thông được thể hiện trong Bảng 3.1 và giản đồ chuyển trạng thái cho các đèn trên hai trục Bắc-Nam và Đông-Tây được biểu diễn trong Hình 3.2.

A screenshot of a paper

Description automatically generated

Sử dụng đèn LED trên board DE2 để hiển thị các đèn Đỏ, Xanh, Vàng; CLOCK\_50 để kiểm soát thời gian của mạch. Viết chương trình và kiểm tra chức năng của mạch trên board DE2.

Code:

Clock\_50:

|  |
| --- |
| module clock\_1Hz (  input wire clk\_50,  output reg clk\_1Hz  );  reg [24:0] count;  parameter DIVIDER = 25000000;  always @ (posedge clk\_50)  begin  if (count == DIVIDER - 1)  begin  count <= 0;  clk\_1Hz <= ~clk\_1Hz;  end  else begin  count <= count + 1;  end  end  endmodule |