



Prelab 2 demo 1

thí nghiệm kĩ thuật số (rường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh)



Scan to open on Studocu

BÀI CHUẨN BỊ THÍ NGHIỆM 2

THỰC HIỆN CÁC IC CHỨC NĂNG CƠ BẢN TRÊN KIT DE-2

Họ và tên:	Lớp TN:
MSSV:	Ngày:

LƯU Ý:

Nội dung trong các bài prelab thường là các hướng dẫn đi kèm các câu hỏi, bài tập có liên quan đến bài thí nghiệm tương ứng. Các bài prelab được biên soạn nhằm mục đích cho sinh viên có kiến thức nền để tiến hành thí nghiệm nhanh chóng và chính xác vì thời gian thí nghiệm là có hạn. Vì thế yêu cầu sinh viên tự thực hiện prelab, mọi trường hợp gian dối đều sẽ bị xử lý nặng.

Sinh viên tham khảo đoạn hướng dẫn sau đồng thời trả lời các câu hỏi tương ứng.

HƯỚNG DẪN CƠ BẢN VỀ KIT DE 2

Sinh viên xem Lab 0 và Kit De2 Manual để hiểu cách sử dụng Kit DE 2, nối dây, các thiết bị ngoại vi, cách sử dụng phần mềm Quartus để mô phỏng, tổng hợp mạch. Sinh viên tham khảo các tài liệu để hiểu cách viết thiết kế phần cứng bằng ngôn ngữ SystemVerilog.

Với Lab 2 sinh viên sẽ sử dụng các ngoại vi LEDR, LEDG, và SW. Dưới đây là header nối dây căn bản của các ngoại vi này khi sinh viên tổng hợp project trên Intel Quartus. (Lưu ý import file DE2_pin_list/assignment.



Lưu ý, top-level của project phải đặt tên là tên của file header (nổi pin), ví dụ là “lab2tn1_wrapper”, với module “lab2tn1” là module chứa đoạn mã mô tả thiết kế của thí nghiệm 1.

Giả sử, với thí nghiệm viết thiết kế phần cứng bằng ngôn ngữ SystemVerilog cho hàm là thí nghiệm 1 của lab 2, ta có đoạn mã SystemVerilog như sau:

```
lab2tn1.v
1  module lab2tn1 (
2      input x,y,z,
3      output f
4  );
5
6  assign f = !x & y & z
7
8  endmodule
```

Hình 1 Ví dụ về thí nghiệm 1

Vậy đoạn mã của top level “lab2tn1_wrapper” như sau:

```
lab2tn1_wrapper.v
1  module lab2tn1_wrapper (
2      input [2:0] SW,
3      output [2:0] LEDR,
4      output [0:0] LEDG
5  );
6
7  lab2tn1 ilab2tn1 (SW[2],SW[1],SW[0],LEDG);
8
9  assign LEDR = SW;
10
11 endmodule
```

Như vậy ta đã hoàn thành phần viết đoạn mã của chương trình và project Quartus, sinh viên thực hiện mô phỏng, nạp Kit như đã được hướng dẫn tại Lab 0.

Với các bài thí nghiệm yêu cầu sử dụng IC 74LS151/74LS138, sinh viên viết đoạn mã mô tả IC này và tiến hành gọi module như trên.

CÂU HỎI CHUẨN BỊ:

- Header cho một module SystemVerilog viết như thế nào

☐ **Module** Một design (thiết kế) gồm có ba thành phần

- Ngõ vào (input)
- Ngõ ra (output)
- Mach logic (logic circuit)

- Các toán tử Logic trong SystemVerilog sử dụng các kí hiệu nào (AND OR XOR NOT)

$\& \Rightarrow \text{and}$	$ \Rightarrow \text{or}$
$\wedge \Rightarrow \text{xor}$	$\sim \Rightarrow \text{not}$


```
assign data_and = foo & bar;
assign data_or  = foo | bar;
assign data_xor = foo ^ bar;
assign data_not = ~foo;
```

- Chế độ Netlist > RTL Viewer có ý nghĩa gì

Cho phép chúng ta quan sát code của mình dưới dạng các cổng logic, để dễ dàng xem ngược lại mã nguồn của mạch.

- Phân biệt sự khác nhau giữa Functional Simulation và Timing Simulation

- Functional Simulation:



- Mục đích: Functional Simulation tập trung vào kiểm tra tính đúng đắn của chức năng logic của mạch. Nó kiểm tra xem mạch hoạt động đúng theo yêu cầu logic được đặt ra không.
- Tín hiệu đầu vào: Các tín hiệu đầu vào được cung cấp để mô phỏng hoạt động của mạch.
- Delay Model: Trong Functional Simulation, không có mô hình trễ (timing delay). Tất cả các cổng logic được giả định hoạt động ngay lập tức.
- Ứng dụng: Functional Simulation thường được sử dụng trong giai đoạn mô hình hóa ngôn ngữ mô tả phần cứng (HDL) và tổng hợp để đảm bảo logic là chính xác.

Timing Simulation:

- Mục đích: Timing Simulation tập trung vào kiểm tra xem mạch hoạt động đúng với các thời gian hoạt động (timings) được xác định không. Nó kiểm tra xem mạch có hoạt động chính xác trong thực tế với tất cả các yếu tố thời gian (delays) được tính toán.
- Tín hiệu đầu vào: Các tín hiệu đầu vào và các tín hiệu điều khiển thời gian (clocks, resets) được cung cấp. Ngoài ra, mô hình trễ cũng được áp dụng.
- Delay Model: Trong Timing Simulation, mô hình trễ đóng vai trò quan trọng. Nó bao gồm các thời gian trễ từ các cổng logic, nút mạch, và các linh kiện khác.
- Ứng dụng: Timing Simulation thường được sử dụng để đảm bảo rằng mạch hoạt động trong các điều kiện thực tế, kể cả các yếu tố thời gian và các điều kiện môi trường.

Tóm lại, Functional Simulation kiểm tra chức năng logic của mạch, trong khi Timing Simulation kiểm tra cả chức năng logic lẫn tính đúng đắn với các yếu tố thời gian được tính toán. Cả hai loại mô phỏng đều cần được sử dụng để đảm bảo tính đúng đắn và hoạt động chính xác của mạch điện tử.

5. Sinh viên tìm hiểu datasheet của IC 74LS138, viết module SystemVerilog mô tả hoạt động của IC này



```

module LS138 (
    input logic A, B, C,
    input logic G1, G2A, G2B,
    output logic [7:0] tmp
);
    always @ (A | B | C | G1 | G2A | G2B)
        case({G2B, G2A, G1, C, B, A})
            6'b001000: tmp = 8'b11111110;
            6'b001001: tmp = 8'b11111101;
            6'b001010: tmp = 8'b11111011;
            6'b001011: tmp = 8'b11110111;
            6'b001100: tmp = 8'b11101111;
            6'b001101: tmp = 8'b11011111;
            6'b001110: tmp = 8'b10111111;
            6'b001111: tmp = 8'b01111111;
        endcase
endmodule : LS138

```

6. Sinh viên tìm hiểu datasheet của IC 74LS151, viết module SystemVerilog mô tả hoạt động của IC này

```

module LS151 (
    input logic S1, S2, S0, GN,
    input logic D0, D1, D2, D3, D4, D5, D6, D7,
    output logic Y
);
    always @(GN | S2 | S1 | S0)
        case({GN, S2, S1, S0})
            4'b0000 : Y = D0;
            4'b0001 : Y = D1;
            4'b0010 : Y = D2;
            4'b0011 : Y = D3;
            4'b0100 : Y = D4;
            4'b0101 : Y = D5;
            4'b0110 : Y = D6;
            4'b0111 : Y = D7;
        endcase
endmodule : LS151

```

7. Nếu mục đích của top level wrapper, tại sao phải đặt tên theo pin assignment đã import

- Kết nối các thành phần chính:** Top-level wrapper kết nối các thành phần chính của mạch, như các module, IP cores, hoặc các phần tử vi mạch tổng hợp (synthesized components).
- Định rõ giao diện:** Nó định rõ các giao diện (ports) của mạch. Các ports này thường đại diện cho các tín hiệu vào ra của mạch, và đóng vai trò quan trọng trong việc kết nối và giao tiếp với các thành phần khác.
- Quản lý đồng bộ:** Trong các mạch lớn, top-level wrapper cũng có thể có trách nhiệm quản lý các tín hiệu đồng bộ như clock và reset, đảm bảo rằng chúng được



phân phối đúng cách đến các phần tử khác.

4. **Tạo ra nguyên mẫu của mạch:** Nó cung cấp một nguyên mẫu (template) để tổ chức và cấu trúc của mạch. Điều này giúp làm cho quá trình thiết kế và hiểu rõ mạch dễ dàng hơn.

Đặt tên của top-level wrapper theo pin assignment đã import có ý nghĩa quan trọng vì:

1. **Bảo đảm tính đúng đắn của kết nối:** Khi bạn đặt tên theo pin assignment, bạn đang đảm bảo rằng các tín hiệu được kết nối đúng với vị trí và mục đích của chúng. Điều này đặc biệt quan trọng trong các thiết kế nơi vị trí vật lý của các tín hiệu có vai trò quan trọng.
2. **Dễ dàng quản lý và xác định:** Khi một người khác hoặc bạn sau này xem lại mạch, việc sử dụng tên theo pin assignment giúp họ dễ dàng xác định nơi các tín hiệu được kết nối.
3. **Hạn chế lỗi trong quá trình thiết kế:** Đặt tên theo pin assignment giúp tránh lỗi kết nối không chính xác, đặc biệt trong các mạch lớn có nhiều tín hiệu và nhiều phần tử kết nối.

Tóm lại, top-level wrapper là một phần quan trọng trong thiết kế vi mạch số, và việc đặt tên theo pin assignment giúp đảm bảo tính đúng đắn và dễ dàng quản lý của kết nối trong mạch.

