

LAB02

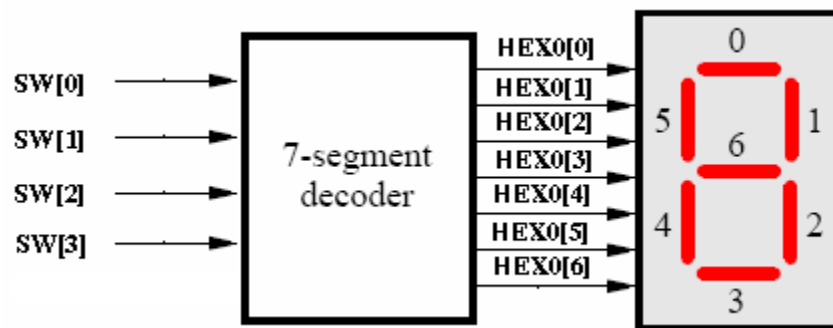
Mục đích

- Thực hành mạch giải mã cho LED 7 đoạn.
- Thực hành mạch chuyển số nhị phân sang thập phân.
- Thực hành mạch cộng/trừ 3 bit trên FPGA.

Nội dung thực hành

Part 1.

1. Cho mạch sau



Hình 2.1

- Dưới đây là bảng sự thật của mạch giải mã cho led 7 đoạn trên dùng hiển thị các kí tự từ 0 đến 9 (các giá trị khác không quan tâm)

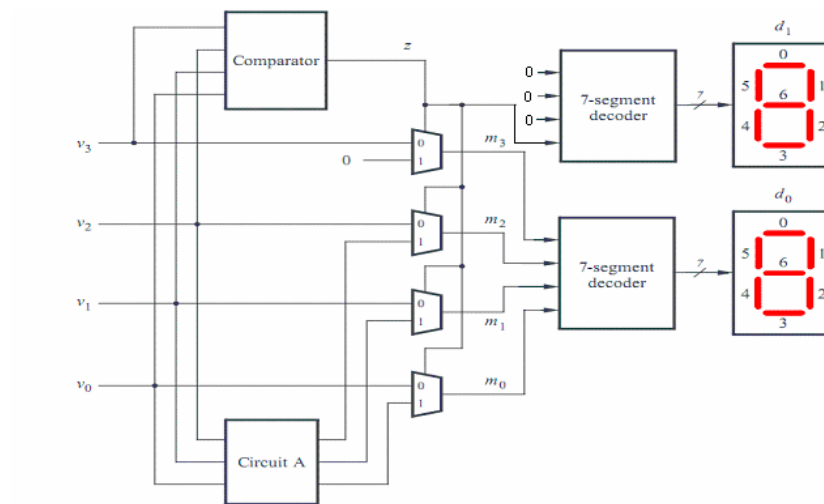
Input		Output						
SW[3:0]	Digit	HEX[0]	HEX[1]	HEX[2]	HEX[3]	HEX[4]	HEX[5]	HEX[6]
0000	0	?	?	?	?	?	?	?
0001	1	?	?	?	?	?	?	?
0010	2	?	?	?	?	?	?	?
0011	3	?	?	?	?	?	?	?
0100	4	?	?	?	?	?	?	?
0101	5	?	?	?	?	?	?	?
0110	6	?	?	?	?	?	?	?
0111	7	?	?	?	?	?	?	?
1000	8	?	?	?	?	?	?	?
1001	9	?	?	?	?	?	?	?
Others	X	X	X	X	X	X	X	X

- Sinh viên hoàn thành bảng sự thật trên.
Chú ý:
 - + Các đoạn led tích cực mức thấp – mức 0 đoạn led sáng, mức 1 đoạn led tắt
 - + Các giá trị X mang ý nghĩ không quan tâm (don't care)
- Dựa vào bảng sự thật, thiết mạch giải mã cho LED7-đoạn trên.
- Tạo project Quartus mới, đặt tên: **user_dir/lab2/lab2_MSSV_part1**
- Vẽ mạch đã thiết kế
- Biên dịch để phân tích, tổng hợp và tạo ra file ***.sof**
- Nạp file thực thi trên FPGA. Kiểm tra hoạt động của mạch.
- Note: **Sinh viên cần chuẩn bị ở nhà những công việc sau (Không có bài chuẩn bị không được vào lớp làm thí nghiệm → tính vắng buổi đó)**
* Đọc và thực hiện các bước 1,2,3,4,5

Part 2.

Thiết kế mạch hiển thị số BCD hai chữ số bằng cách sử dụng 2 led 7 đoạn để biểu diễn các số thập phân từ 10 đến 15 với input là 4 bit nhị phân.

Hình 2.3 là một gợi ý



Hình 2.2

Trong đó:

- ➡ Khối Comparator dùng để kiểm tra khi nào input (V3V2V1V0) lớn hơn 9

Từng bước thực hiện

- Trả lời các câu hỏi sau
 - Chức năng của mạch Circuit A
 - Nêu cách thức hoạt động của toàn mạch
- Thiết kế chức năng Comparator và Circuit A
- Tạo project Quartus mới, đặt tên: **user_dir/lab2/lab2_MSSV_part2**

4. Thiết kế toàn bộ mạch trên, sử dụng SW[3:0] cho input và HEX0[6:0], HEX1[6:0] cho output
5. Biên dịch để phân tích, tổng hợp và tạo ra file *.sof
6. Nạp file thực thi trên FPGA. Kiểm tra hoạt động của mạch.
7. Note: **Sinh viên cần chuẩn bị ở nhà những công việc sau (Không có bài chuẩn bị không được vào lớp làm thí nghiệm → tính vắng buổi đó)**
* Đọc và thực hiện các bước 1,2,3,4,5

Part 3.

Sử dụng kết quả ở part1 lab1 thiết kế một mạch cộng 4 bit nhị phân kiểu *carry ripple*.

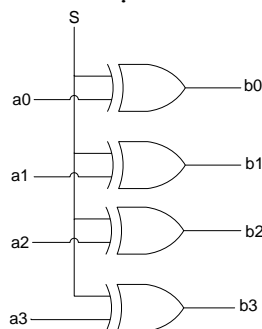
Gợi ý: xem bài giảng lý thuyết về combinational circuit

Từng bước thực hiện

1. Tạo project Quartus mới, đặt tên: **user_dir/lab2/lab2_MSSV_part3**
2. Thiết kế toàn bộ mạch trên, dùng switch làm input (C_i , a , b) cho mạch cộng, dùng LEDG[3:0] để hiển thị kết quả phép cộng $S3S2S1S0$ và LEDG[4] để hiển thị số dư C4
3. Biên dịch để phân tích, tổng hợp và tạo ra file ***.sof**
4. Nạp file thực thi trên FPGA. Kiểm tra hoạt động của mạch.
5. Note: **Sinh viên cần chuẩn bị ở nhà những công việc sau (Không có bài chuẩn bị không được vào lớp làm thí nghiệm → tính vắng buổi đó)**
* Đọc và thực hiện các bước 1,2,3,4,5

Part 4.

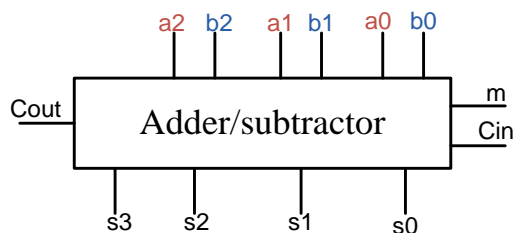
1. Cho mạch như hình vẽ



Hình 2.3

Anh/chị viết bảng chân trị của mạch ở hình 2.5. Quan sát bảng chân trị và cho biết ứng dụng của mạch.

2. Thiết kế mạch cộng hoặc trừ 4 bit nhị phân được mô tả như hình 2.6



Hình 2.4

Từng bước thực hiện

1. Tạo project Quartus mới, đặt tên: **user_dir/lab2/lab2_MSSV_part4**
 2. Thiết kế mạch cộng hoặc trừ trên
 - ➡ Dùng switch làm input ($b3b2b1b0$, $a3a2a1a0$ và m) cho mạch
 - ➡ Dùng LEDG[3:0] để hiển thị kết quả phép cộng/trừ S3S2S1S0 và LEDG[4] để hiển thị số dư Cout
 3. Biên dịch để phân tích, tổng hợp và tạo ra file ***.sof**
 4. Nạp file thực thi trên FPGA. Kiểm tra hoạt động của mạch.
 5. Note: **Sinh viên cần chuẩn bị ở nhà những công việc sau (Không có bài chuẩn bị không được vào lớp làm thí nghiệm → tính vắng buổi đó)**
- * Đọc và thực hiện các bước 1,2,3,4,5

Hết