

Bài thi module 1:

FPGA - SOC – LẬP TRÌNH FPGA (VERILOG + VITIS)

Đề tài:

Thiết kế và lập trình **bộ điều khiển 8 LED WS2812**, có khả năng điều khiển **kiểu chớp (type)** và **màu sắc (color)** thông qua **giao diện console UART** hoặc **4 nút nhấn** trên board.

1. Mục tiêu

- Ứng dụng kiến thức về lập trình phần cứng Verilog, FSM, và giao tiếp ngoại vi.
- Thực hành phối hợp giữa phần PL (FPGA) và phần PS (ARM) trong Zynq.
- Thiết kế giao diện điều khiển linh hoạt qua **UART console** và **phím nhấn**.

2. Yêu cầu kỹ thuật

2.1. Phần cứng

- Board: **EBZA4205** hoặc **Zybo Z7/Zynq-7000** hoặc **ZUBOARD**
- Module LED: **8 LED WS2812B** nối chuỗi
- 4 nút nhấn:
 - **KEY1**: đổi kiểu chớp `type`
 - **KEY2**: đổi màu `color`
 - **KEY3**: tăng tốc độ chớp (+)
 - **KEY4**: giảm tốc độ chớp (–)

2.2. Chức năng điều khiển LED

Tín hiệu	Mô tả	Ghi chú
<code>type[1:0]</code>	4 kiểu chớp LED	00: Toàn bộ sáng 01: Chạy tuần tự 10: Hai chiều 11: Nhấp nháy toàn bộ
<code>color[2:0]</code>	8 màu cơ bản	000: Red, 001: Green, 010: Blue, 011: Yellow, 100: Cyan, 101: Magenta, 110: White, 111: Off
<code>speed</code>	tốc độ delay giữa các hiệu ứng	tăng giảm bằng KEY3 / KEY4

2.3. Điều khiển qua UART Console

- Khi bật hệ thống, console hiển thị menu như sau:

```
-----  
WS2812 LED CONTROL MENU  
-----  
Type   : 01 (Running)  
Color  : 010 (Blue)  
Speed  : 300 ms  
  
Commands:  
t - Change Type  
c - Change Color  
+ - Increase Speed  
- - Decrease Speed  
-----  
Enter command:
```

- Mỗi khi người dùng nhập lệnh, hệ thống cập nhật ngay và hiển thị lại thông tin mới.

2.4. Điều khiển bằng 4 nút nhấn

- Mỗi lần nhấn nút, hệ thống thay đổi giá trị tương ứng như lệnh trên console.
 - Dữ liệu từ nút nhấn và console đều cập nhật chung vào cùng một **FSM điều khiển LED**.
-

3. Cấu trúc hệ thống

3.1. Phần PL (Verilog)

- Module chính: `ws2812_controller.v`
- Input:
 - `clk, reset`
 - `type[1:0], color[2:0]`
 - `speed`
- Output:
 - `led_data` (data line đến WS2812)
- Nhiệm vụ:
 - Sinh tín hiệu điều khiển 8 LED WS2812.
 - Cập nhật kiểu chớp và màu dựa theo tham số nhập từ phần PS.
 - Có thể tách thành 2 module:
 - `ws2812_driver.v`: sinh xung theo chuẩn WS2812.
 - `pattern_controller.v`: tạo hiệu ứng LED dựa theo type/color/speed.

3.2. Phần PS (Vitis – C code)

- Viết chương trình:
 - Kết nối UART để nhập lệnh từ console.
 - Đọc giá trị nút nhấn (GPIO).
 - Hiển thị menu và cập nhật lệnh.
 - Gửi giá trị `type, color, speed` xuống PL qua GPIO hoặc EMIO.
-

4. Hướng dẫn thực hiện

Bước 1: Thiết kế module `ws2812_driver` sinh tín hiệu 24-bit RGB

- Gợi ý thời gian chuẩn WS2812B (Nhớ kiểm tra kỹ loại module LED mã hiệu loại 2812 hay 2812B):
 - Bit “0”: $T_{high} = 0.4 \mu s$, $T_{low} = 0.85 \mu s$.
 - Bit “1”: $T_{high} = 0.8 \mu s$, $T_{low} = 0.45 \mu s$.
 - Tổng chu kỳ 1 bit = $1.25 \mu s$.
- RESET: giữ mức thấp $\geq 50 \mu s$ (một số datasheet khuyến nghị $\sim 300 \mu s$ cho an toàn với một vài bản clone)

Bước 2: Thiết kế FSM trong `pattern_controller`

- 4 trạng thái tương ứng `type[1:0]`
- Mỗi trạng thái cập nhật giá trị màu LED tương ứng.

Bước 3: Kết nối PL \leftrightarrow PS GPIO / EMIO.

Bước 4: Viết chương trình C trong Vitis:

- Đọc nút, nhận lệnh console, cập nhật thông số.
 - Ghi giá trị xuống thanh ghi điều khiển LED.
 - In lại thông tin trên terminal sau mỗi thay đổi.
-

5. Mức đánh giá

Mức	Mô tả	Điểm
A (9-10)	Hoàn chỉnh cả 2 phần Console & Key Input, hiển thị menu đẹp, LED hiển thị đúng 4 kiểu và 8 màu	100
B (8)	Hoàn thiện phần Console + 4 kiểu LED, có thể thiếu đổi màu	80
C (6-7)	Chạy được hiệu ứng LED, có thể đổi type nhưng không có console	70
D (5)	LED sáng được 1 kiểu cố định	50
F (<5)	Không chạy được LED hoặc lỗi hệ thống	40

6. Gợi ý cấu trúc file

```
├── src/
│   ├── ws2812_driver.v
│   ├── pattern_controller.v
│   └── top_led_controller.v
├── vitis_app/
│   ├── main.c
│   ├── led_menu.c
│   └── gpio_control.c
├── constraint/
│   ├── led_ws2812.xdc
│   └── button_io.xdc
```