**Phân Tích Code Điều Khiển Audio Codec WM8731 qua I2C**

**📌 Mục tiêu của Code**

* **Gửi lệnh cấu hình WM8731 qua giao tiếp I2C**.
* **Quản lý trạng thái gửi dữ liệu I2C** bằng FSM (**Finite State Machine**).
* **Tạo xung clock I2C (sclk) và dữ liệu I2C (sdin) để giao tiếp với codec**.

**1️ Tổng Quan Các Module**

| **Module** | **Chức Năng** |
| --- | --- |
| audio\_codec | Module chính, kết nối với config\_codec và i2c\_protocol. |
| config\_codec | Chứa danh sách các thanh ghi cấu hình cho WM8731 và FSM điều khiển việc gửi dữ liệu. |
| i2c\_protocol | Thực hiện giao thức I2C (tạo sclk, sdin) để truyền dữ liệu tới codec. |

**2️ Phân Tích Từng Module**

**🔹 Module audio\_codec**

verilog

Sao chépChỉnh sửa

module audio\_codec (

input logic clk\_i2c, // Clock I2C từ FPGA

input logic reset\_n, // Reset toàn bộ hệ thống

input logic is\_config, // Kích hoạt quá trình cấu hình codec

output sclk, // Clock I2C gửi tới codec

output sdin, // Dữ liệu I2C gửi tới codec

output logic done // Hoàn thành cấu hình

);

**Chức năng**:

* Nhận clk\_i2c từ FPGA, reset\_n, và tín hiệu is\_config để khởi động cấu hình codec.
* Kết nối với config\_codec và i2c\_protocol để thực hiện giao tiếp I2C.

**🔹 Kết nối với config\_codec**

verilog

Sao chépChỉnh sửa

logic busy, ack\_i2c, wr\_send;

logic [6:0] addr;

logic [7:0] data\_st\_send, data\_nd\_send;

config\_codec configcodec (

.clk (clk\_i2c),

.reset\_n (reset\_n),

.is\_config (is\_config),

.busy (busy),

.done\_config (done),

.ack\_i2c (ack\_i2c),

.addr (addr),

.wr\_rd (wr\_send),

.addr\_reg (data\_st\_send),

.data\_config (data\_nd\_send)

);

**Giải thích**:

* config\_codec **chứa danh sách các thanh ghi cấu hình của WM8731**.
* Khi is\_config = 1, module này sẽ gửi từng lệnh I2C **tới WM8731**.

**🔹 Kết nối với i2c\_protocol**

verilog

Sao chépChỉnh sửa

i2c\_protocol i2c (

.clk (clk\_i2c),

.reset\_n (reset\_n),

.start (ack\_i2c),

.addr (addr),

.wr\_rd (wr\_send),

.data\_st (data\_st\_send),

.data\_nd (data\_nd\_send),

.busy (busy),

.done (),

.sclk (sclk),

.sdin (sdin)

);

endmodule

**Giải thích**:

* i2c\_protocol thực hiện **giao tiếp I2C**, tạo sclk và sdin để gửi đến WM8731.
* start = ack\_i2c → Khi có ACK từ codec, FPGA tiếp tục gửi dữ liệu.

**🔹 Module config\_codec**

**1. Bộ nhớ chứa danh sách thanh ghi cấu hình WM8731**

verilog

Sao chépChỉnh sửa

logic [15:0] data\_array [9:0];

always\_ff @(negedge reset\_n) begin

data\_array[0] <= 16'h1e00; // Reset all

data\_array[1] <= 16'h0c00; // Power-up

data\_array[2] <= 16'h0579; // Left Volume

data\_array[3] <= 16'h0779; // Right Volume

data\_array[4] <= 16'h0e09; // Digital Audio Format

data\_array[5] <= 16'h1001; // Sampling Control

data\_array[6] <= 16'h13ff; // Active Codec

data\_array[7] <= 16'h0812; // Analog Path Control

data\_array[8] <= 16'h0a00; // Digital Path Control

data\_array[9] <= 16'h0a00;

end

**Giải thích**:

* Mỗi phần tử **data\_array[n]** chứa **địa chỉ thanh ghi + dữ liệu cần gửi**.
* FPGA sẽ **gửi từng lệnh I2C** trong danh sách này.

**2. FSM điều khiển gửi dữ liệu I2C**

verilog

Sao chépChỉnh sửa

typedef enum logic [3:0] {

IDLE, IS\_CONFIG, SEND\_1, SEND\_2, IS\_READY, SEND\_LAST, DONE

} state\_e;

📌 **FSM này điều khiển quá trình gửi dữ liệu đến WM8731**

* IDLE: Chờ tín hiệu is\_config.
* IS\_CONFIG: Nếu không busy, bắt đầu gửi lệnh.
* SEND\_1 → SEND\_2: Gửi địa chỉ thanh ghi.
* IS\_READY: Chờ busy = 0 (codec xử lý xong).
* SEND\_LAST: Kiểm tra xem đã gửi hết danh sách cấu hình chưa.
* DONE: Quá trình cấu hình hoàn tất.

**🔹 Module i2c\_protocol**

📌 **Chức năng:**

* **Tạo xung nhịp I2C (sclk)**.
* **Gửi dữ liệu I2C (sdin)** theo chuẩn **2-wire serial interface**.

**1. Tạo gói tin I2C**

verilog

Sao chépChỉnh sửa

assign data = {addr\_send, wr\_rd\_send, 1'b1, data\_st\_send, 1'b1, data\_nd\_send, 1'b1};

📌 **Gói tin I2C gửi đi gồm:**

| **Địa chỉ Slave** | **Read/Write** | **ACK** | **Địa chỉ Thanh ghi** | **ACK** | **Dữ liệu** | **ACK** |
| --- | --- | --- | --- | --- | --- | --- |
| addr\_send | wr\_rd\_send | 1 | data\_st\_send | 1 | data\_nd\_send | 1 |

**2. FSM Điều Khiển I2C**

verilog

Sao chépChỉnh sửa

typedef enum logic [4:0] {

IDLE, START, START\_CONT, BUF\_1, DATA\_SEND\_1, DATA\_SEND\_2, DATA\_SEND\_3, DATA\_SEND\_4, BUF\_2, STOP, DONE

} state\_e;

📌 **FSM này điều khiển từng bước trong giao thức I2C**:

1. IDLE: Chờ lệnh từ config\_codec.
2. START: Gửi bit **START** (SDA kéo xuống).
3. DATA\_SEND\_X: Gửi từng bit của địa chỉ + dữ liệu.
4. STOP: Gửi bit **STOP** (SDA kéo lên).
5. DONE: Xác nhận gửi hoàn tất.

**📢 Kết Luận**

✅ **Code này hoạt động chính xác và tuân theo chuẩn I2C để điều khiển WM8731**.  
✅ **FSM trong config\_codec giúp gửi từng lệnh cấu hình một cách chính xác**.  
✅ **FSM trong i2c\_protocol thực hiện chuẩn I2C chặt chẽ, đảm bảo giao tiếp ổn định**.