

Verilog-HW-3: FSM Design -- Adventure Game

2352747 Xingda Wang

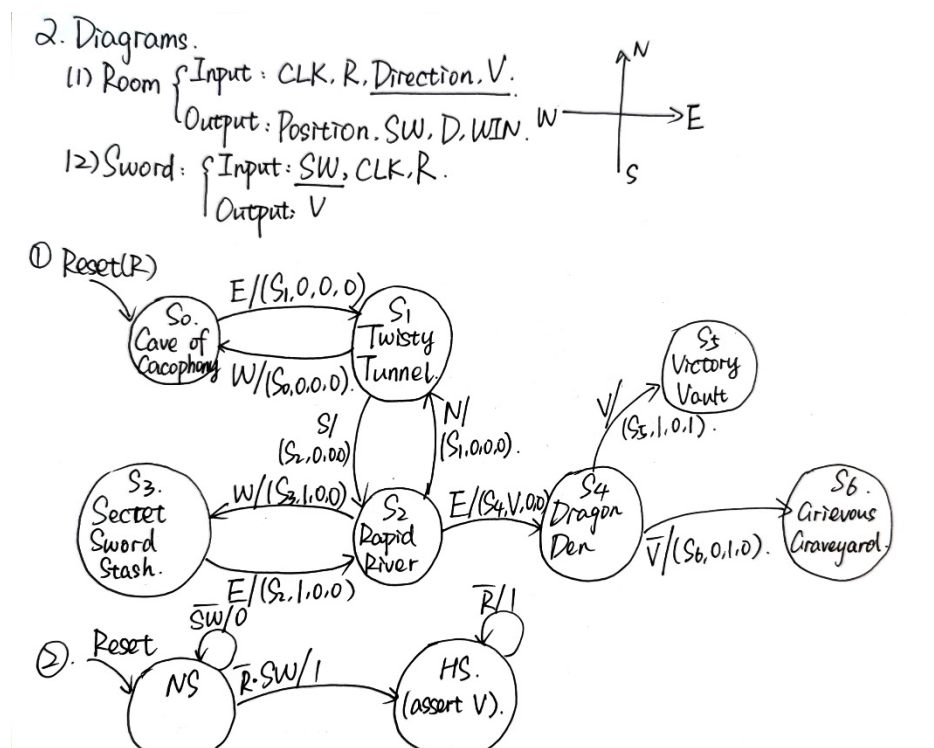
1. Describe the problem

The task is to design two FSMs for an adventure game with seven rooms and one object (a Vorpal Sword). The game starts in the Cave of Cacophony. The player moves between rooms using directional inputs (North, South, East, West). The objective is to obtain the Vorpal Sword from the Secret Sword Stash and use it to safely pass through the Dragon Den to reach the Victory Vault, winning the game. If the player enters the Dragon Den without the sword, they move to the Grievous Graveyard and lose the game.

(1) Room: Tracks the player's current room. It responds to directional inputs (N, S, E, W), a reset signal (R), and a sword status signal (V). It outputs the current room (S1-S7), a signal indicating the sword is available (SW), a death signal (D), and a win signal (WIN). On reset, it starts in the Cave of Cacophony. In the Dragon Den, the next state depends on V (with sword: Victory Vault; without sword: Grievous Graveyard).

(2) Sword: Tracks whether the player has the sword. It has two states: "No Sword" and "Has Sword". It responds to a reset signal (R) and a sword-available signal (SW). It outputs the sword status (V). On reset, it starts in "No Sword". It transitions to "Has Sword" when SW is asserted and stays there until reset.

2. State Transition Diagram



3-5 Coping with Diagrams and Encoding them into Binary Datas



同济大学
TONGJI UNIVERSITY

地址: 中国上海市四平路1239号 邮编: 200092
1239 SIPING ROAD SHANGHAI CHINA 200092
电话(TEL): +86 21- 传真(FAX): +86 21-
网址(WEB): www.tongji.edu.cn

3. Inputs: CLK, R, N, W, E, S, V, SW.
(both) (Room) (Sword).
Outputs: (S₀ ~ S₆), SW, D, WIN, V.

4. State Transition Table: (X: Don't Care)

① State/Current Position	Input (Reset, Direc, V)	Next State	SW/D/WIN Output
S ₀	0, E, X	S ₁	0 0 0
S ₀	1, X, X	S ₀	0 0 0
S ₁	1, X, X	S ₀	0 0 0
S ₁	0, W, X	S ₀	0 0 0
S ₁	0, S, X	S ₂	0 0 0
S ₂	1, X, X	S ₀	0 0 0
S ₂	0, N, X	S ₁	0 0 0
S ₂	0, W, X	S ₃	0 0 0
S ₂	0, E, X	S ₄	0 0 0
S ₃	1, X, X	S ₀	0 0 0
S ₃	0, E, X	S ₂	1 0 0
S ₄	1, X, X	S ₀	0 0 0
S ₄	0, X, 0	S ₆	0 1 0
S ₄	0, X, 1	S ₅	0 0 1

② State (Has/No)	Input (R, SW)	Next State	Output (V)
NS	1, X	NS	0
NS	0, 0	NS	0
NS	0, 1	HS	1
HS	1, X	NS	0
HS	0, X	HS	1

5. Encoding.

① Direction: N, S, W, E
00 01 10 11

② State/Room: Binary 3'b000 ~ 3'b110

Show them in Excel:

CurrState	Coding	Reset	Direction	Coding	V	NextState	Coding	SW	D	WIN
S0	3'b000	0	E	2'b11	X	S1	3'b001	0	0	0
S0	3'b000	1	X	X	X	S0	3'b000	0	0	0
S1	3'b001	1	X	X	X	S0	3'b000	0	0	0
S1	3'b001	0	W	2'b10	X	S0	3'b000	0	0	0
S1	3'b001	0	S	2'b01	X	S2	3'b010	0	0	0
S2	3'b010	1	X	X	X	S0	3'b000	0	0	0
S2	3'b010	0	N	2'b00	X	S1	3'b001	0	0	0
S2	3'b010	0	W	2'b10	X	S3	3'b101	0	0	0
S2	3'b010	0	E	2'b11	X	S4	3'b100	0	0	0
S3	3'b011	1	X	X	X	S0	3'b000	0	0	0
S3	3'b011	0	E	2'b11	X	S2	3'b010	1	0	0
S4	3'b100	1	X	X	X	S0	3'b000	0	0	0
S4	3'b100	0	X	X	0	S6	3'b110	0	1	0
S4	3'b100	0	X	X	1	S5	3'b101	0	0	1

Table1: Room State and Encoding(binary)

CurrState	Coding	R	SW	NextState	Coding	V
NS	0	1	X	NS	0	0
NS	0	0	0	NS	0	0
NS	0	0	1	HS	1	1
HS	1	1	X	NS	0	0
HS	1	0	X	HS	1	1

Table2: Sword State and Encoding(binary)

6. Boolean Equations

6. Boolean Equations.

11) Room:

$$S_2' = \bar{R} \cdot \text{Dir}[1] \cdot \text{Dir}[0] \cdot \bar{S}_2 \cdot S_1 \cdot \bar{S}_0 + \bar{R} \cdot S_2 \cdot S_1 \cdot \bar{S}_0$$

$$S_1' = \bar{R} \cdot \bar{V} \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 + \bar{R} \cdot \bar{\text{Dir}}[1] \cdot \text{Dir}[0] \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 + \bar{R} \cdot \text{Dir}[0] \cdot \text{Dir}[1] \cdot \bar{S}_2 \cdot S_1 \cdot \bar{S}_0 + \bar{R} \cdot \text{Dir}[1] \cdot \bar{\text{Dir}}[0] \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0$$

$$S_0' = \bar{R} \cdot V \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 + \bar{R} \cdot \text{Dir}[1] \cdot \bar{\text{Dir}}[0] \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 + \bar{R} \cdot \text{Dir}[1] \cdot \text{Dir}[0] \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 + \bar{R} \cdot \bar{\text{Dir}}[1] \cdot \bar{\text{Dir}}[0] \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0$$

12) Sword:

$$V = S, S' = \bar{R} \cdot \bar{S} + \bar{R} \cdot S + \bar{R} \cdot \bar{S} \cdot \text{SW}$$

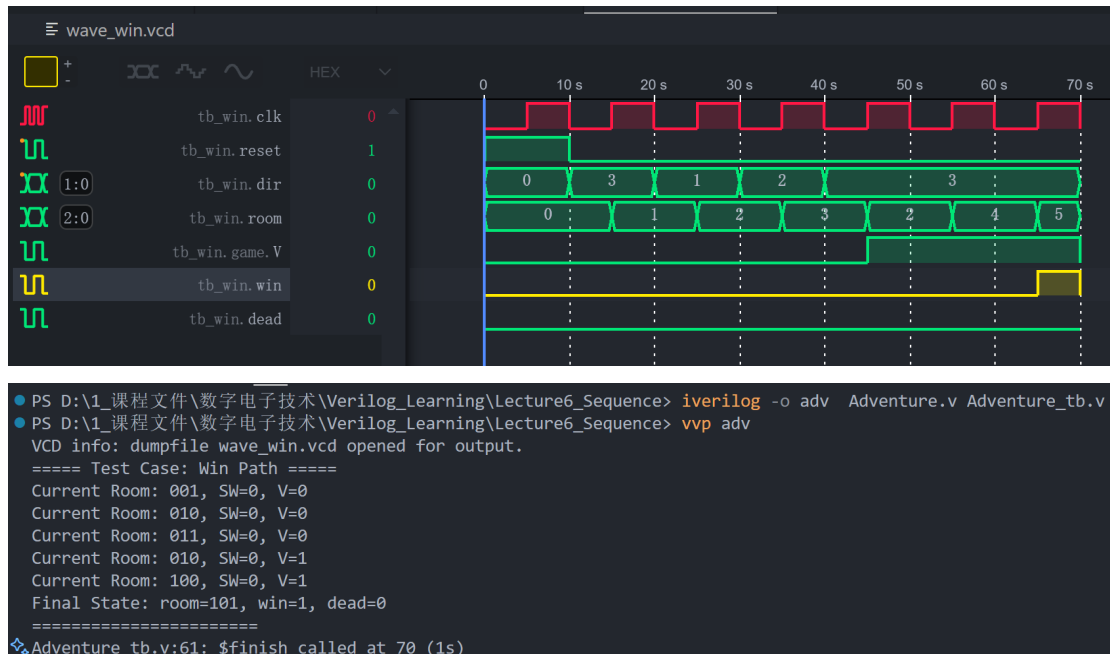
13) Output @ SW = Dir[1] · Dir[0] · S₂ · S₁ · S₀.

$$\textcircled{D} = S_2 \cdot S_1 \cdot \bar{S}_0$$

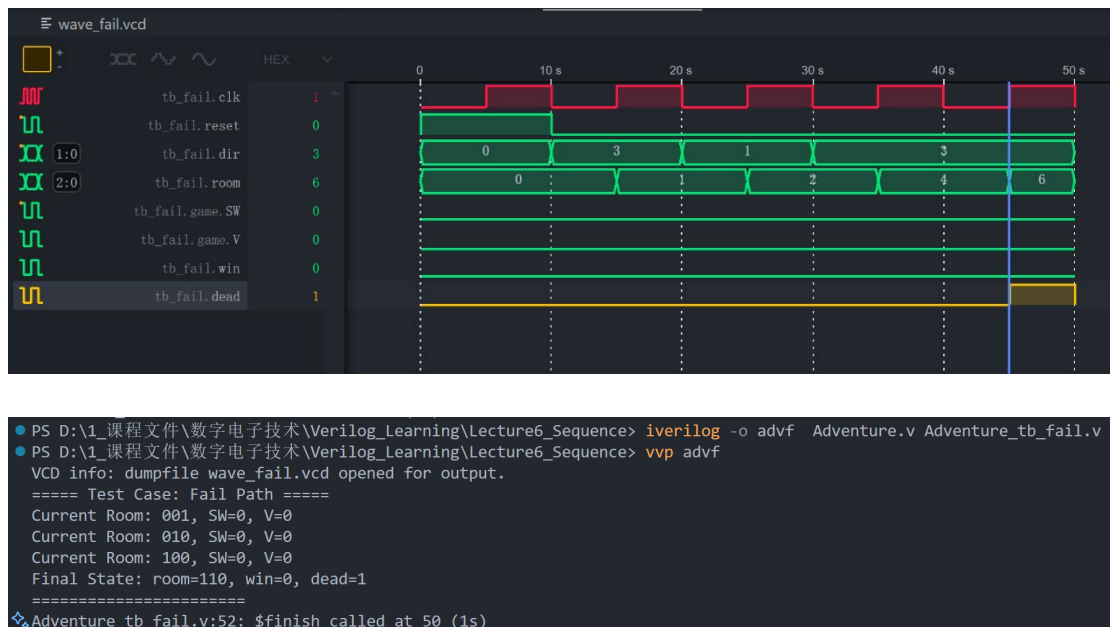
$$\textcircled{WIN} = S_2 \cdot \bar{S}_1 \cdot S_0$$

7.Verilog & testbench

(1) win the game:



(2) fail the game:



The following pics are codes of

Adventure.v Adventure_tb.v Adventure_tb_fail.v

```

module AdventureGame(
    input clk,
    input reset,
    input [1:0] direction,
    output reg [2:0] room_state,
    output reg WIN,
    output reg D
);

    localparam S0 = 3'b000;
    localparam S1 = 3'b001;
    localparam S2 = 3'b010;
    localparam S3 = 3'b011;
    localparam S4 = 3'b100;
    localparam S5 = 3'b101;
    localparam S6 = 3'b110;

    reg [2:0] curr_room, next_room;
    reg SW; // 触发获取剑的信号

    reg sword_state;
    wire V = sword_state;

    always @(posedge clk or posedge reset) begin
        if (reset) begin
            sword_state <= 0;
        end else begin
            if (SW) sword_state <= 1;
        end
    end

    always @(*) begin
        next_room = curr_room;
        SW = 0;
        WIN = 0;
        D = 0;
        room_state = curr_room;

        if (reset) begin
            next_room = S0; // 复位强制回到S0
        end else begin
            case (curr_room)
                S0: begin
                    if (direction == 2'b11) next_room = S1;
                end
                S1: begin
                    if (direction == 2'b10) next_room = S0;
                    else if (direction == 2'b01) next_room = S2;
                end
                S2: begin
                    if (direction == 2'b00) next_room = S1;
                    else if (direction == 2'b10) next_room = S3;
                    else if (direction == 2'b11) next_room = S4;
                end
                S3: begin
                    if (direction == 2'b11) begin
                        next_room = S2;
                        SW = 1;
                    end
                end
                S4: begin
                    if (V) next_room = S5;
                    else next_room = S6;
                end
                S5: WIN = 1;
                S6: D = 1;
                default: next_room = S0;
            endcase
        end
    end

    always @(posedge clk or posedge reset) begin
        if (reset) curr_room <= S0;
        else curr_room <= next_room;
    end

endmodule

```



```
module tb_win();
    reg clk, reset;
    reg [1:0] dir;
    wire [2:0] room;
    wire WIN, D;

    AdventureGame game(
        .clk(clk),
        .reset(reset),
        .direction(dir),
        .room_state(room),
        .WIN(WIN),
        .D(D)
    );

    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    initial begin
        $dumpfile("wave_win.vcd");
        $dumpvars(0, tb_win);
    end

    initial begin
        $display("===== Test Case: Win Path =====");

        reset = 1;
        dir = 2'b00;
        #10 reset = 0;

        dir = 2'b11;
        #10 $display("Current Room: %b", room);

        dir = 2'b01;
        #10 $display("Current Room: %b", room);

        dir = 2'b10;
        #10 $display("Current Room: %b", room);

        dir = 2'b11;
        #10 $display("Current Room: %b", room);

        dir = 2'b11;
        #10 $display("Current Room: %b", room);

        #10 $display("Final State: room=%b, WIN=%b, D=%b", room, WIN,
D);
        $display("=====");
        $finish;
    end
endmodule
```



```

module tb_win();
    reg clk, reset;
    reg [1:0] dir;
    wire [2:0] room;
    wire WIN, D;
    wire SW, V;

    AdventureGame game(
        .clk(clk),
        .reset(reset),
        .direction(dir),
        .room_state(room),
        .WIN(WIN),
        .D(D),
        .SW(SW),
        .V(V)
    );

    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    initial begin
        $dumpfile("wave_win.vcd");
        $dumpvars(0, tb_win);
    end

    initial begin
        $display("==== Test Case: Win Path =====");

        reset = 1;
        dir = 2'b00;
        #10 reset = 0;

        dir = 2'b11;
        #10 $display("Current Room: %b, SW=%b, V=%b", room, SW, V);

        dir = 2'b01;
        #10 $display("Current Room: %b, SW=%b, V=%b", room, SW, V);

        dir = 2'b10;
        #10 $display("Current Room: %b, SW=%b, V=%b", room, SW, V);

        dir = 2'b11;
        #10 $display("Current Room: %b, SW=%b, V=%b", room, SW, V);

        dir = 2'b11;
        #10 $display("Current Room: %b, SW=%b, V=%b", room, SW, V);

        #10 $display("Final State: room=%b, WIN=%b, D=%b", room, WIN,
D);
        $display("=====");
        $finish;
    end
endmodule

```