Homework 04

Ren-Der Chen (陳仁德)
Department of Computer Science and
Information Engineering
National Changhua University of Education
E-mail: rdchen@cc.ncue.edu.tw
Fall, 2019

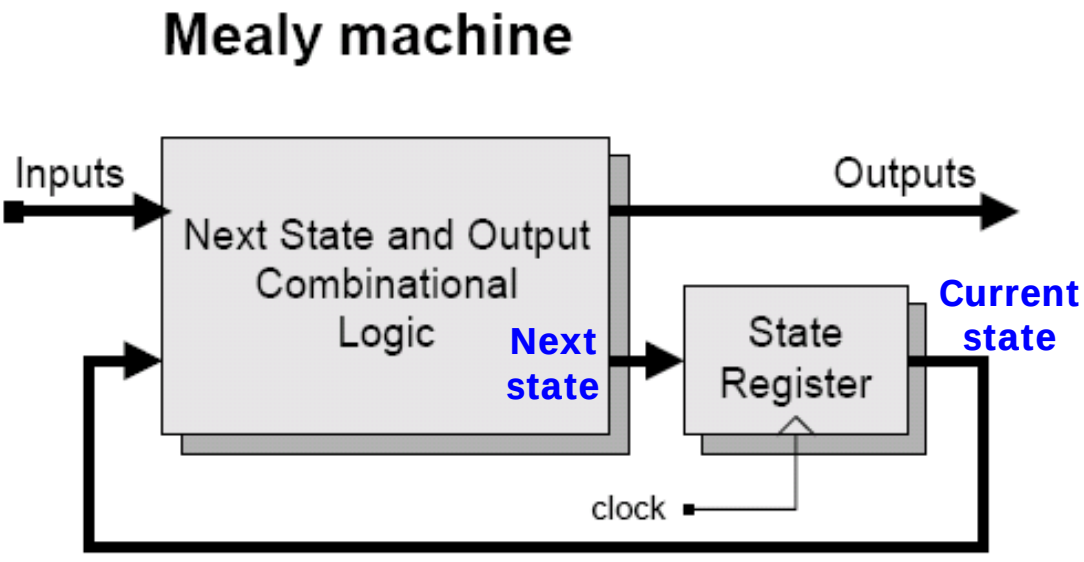
Sequence Detector

- A sequence detector accepts a string of bits (either 0 or 1) as input.
- Its output goes to 1 whenever a target sequence is detected.
- There are two types of detection: overlapping and non-overlapping.
- For a detector that allows overlapping, the final bits of one sequence can be the start of another sequence.
- Overlapping type will be adopted here.

Example

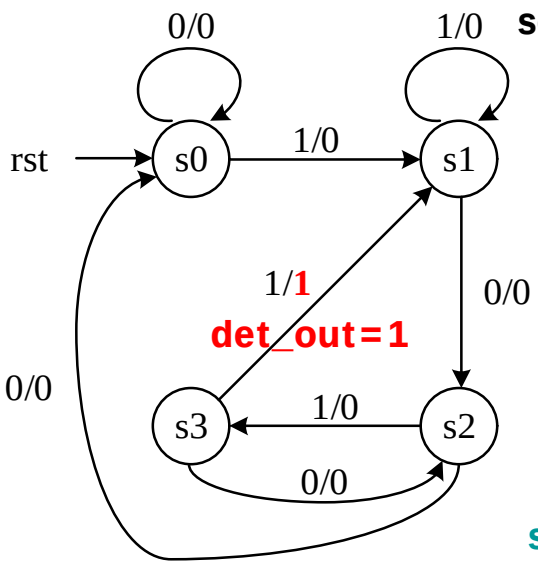
- Target sequence: **1011**
- Input sequence: **1 1 0 1 1 0 1 1 0**
- Detection output: **0 0 0 0 1 0 0 1 0**

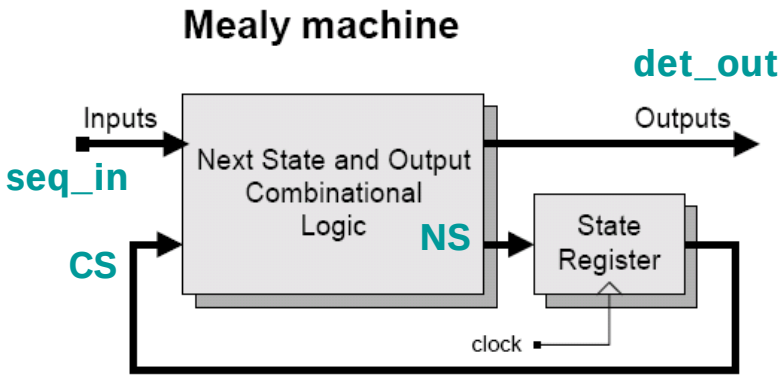
Design 1: Mealy Machine



Mealy Machine State Diagram

- Target sequence: **1011**

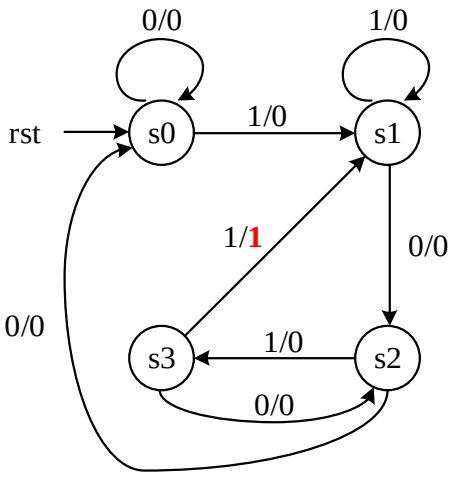




Verilog Code (1/4)

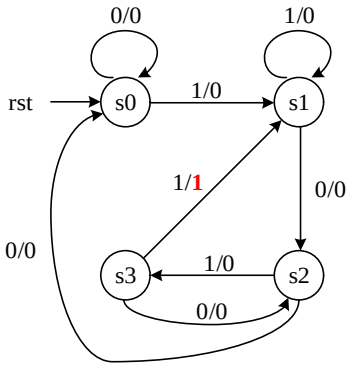
```
module sd_mealy(clk, rst, seq_in, det_out, state_out);
    input clk, rst, seq_in;
    output reg det_out;
    output [1:0] state_out;

    parameter
        s0=2'b00,    // no valid seq
        s1=2'b01,    // valid seq: 1
        s2=2'b10,    // valid seq: 10
        s3=2'b11;    // valid seq: 101
```

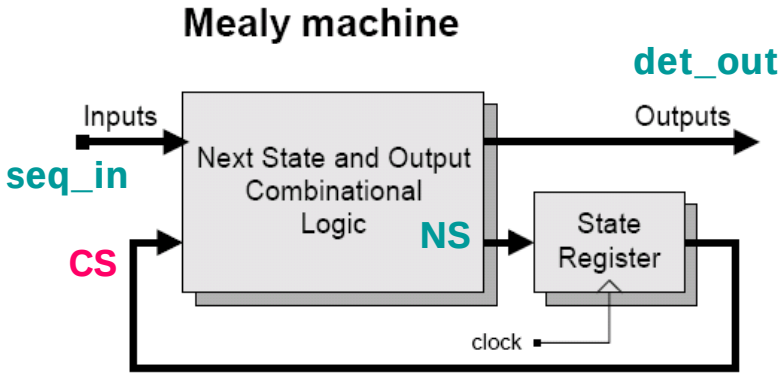
Four states



Verilog Code (2/4)

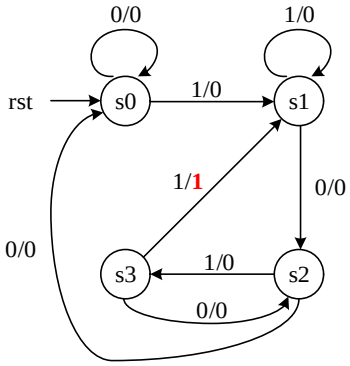
```
reg [1:0] CS, NS;
// CS: current state
// NS: next state

assign state_out = CS;

// sequential part of the Mealy FSM
always @(posedge clk, posedge rst) begin
    #2
    if (rst==1)
        CS = s0;
    else
        CS = NS;
end
```

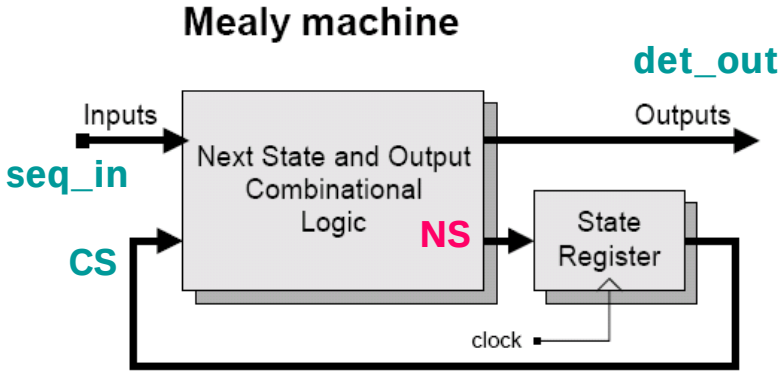




Verilog Code (3/4)

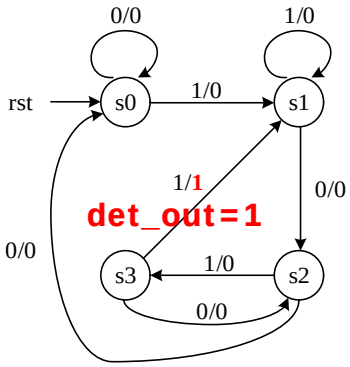
```
// combinational part of the Mealy FSM
// determine the next state
always @(CS, seq_in) begin
    #2
    case (CS)
        s0: NS = (seq_in==1) ? s1 : s0;
        s1: NS = (seq_in==1) ? s1 : s2;
        s2: NS = (seq_in==1) ? s3 : s0;
        s3: NS = (seq_in==1) ? s1 : s2;
        default: NS = s0;
    endcase
end
```

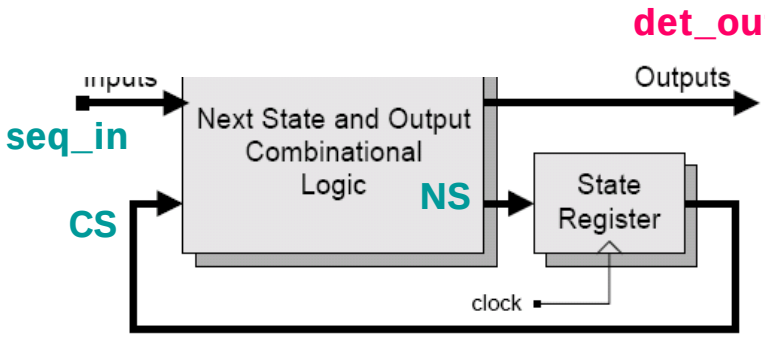




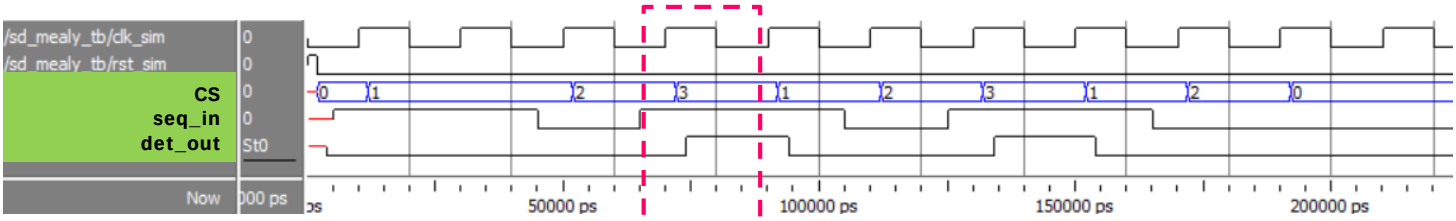
Verilog Code (4/4)

```
    // combinational part of the Mealy FSM
    // determine the detection output
    always @(CS, seq_in) begin
        #2
        case (CS)
            s0: det_out = 0;
            s1: det_out = 0;
            s2: det_out = 0;
            s3: det_out = (seq_in==1) ? 1 : 0;
            default: det_out = 0;
        endcase
    end
endmodule
```



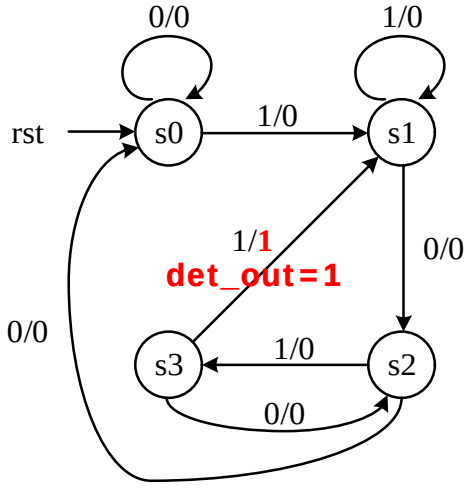


Simulation Result



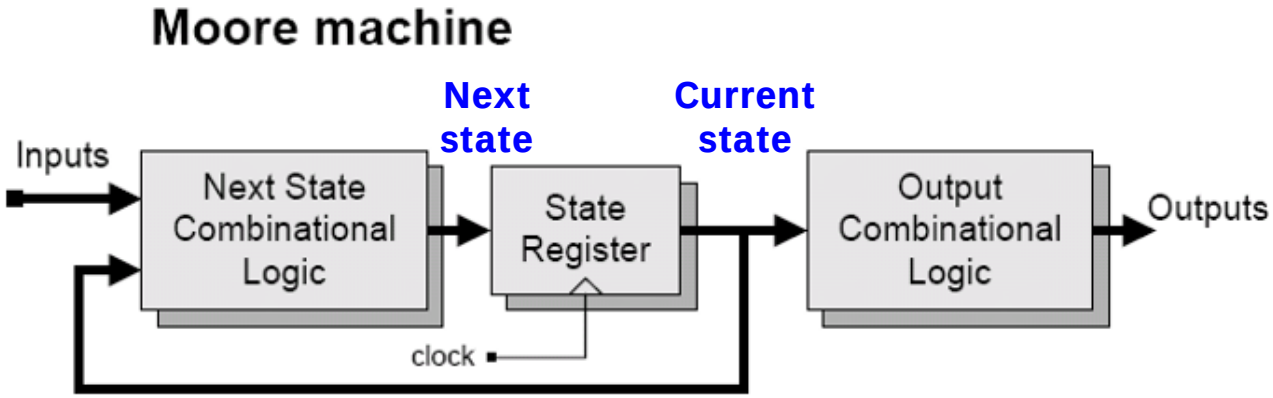
seq_in: 1 1 0 1 1 0 1 1 0

det_out: 1 1



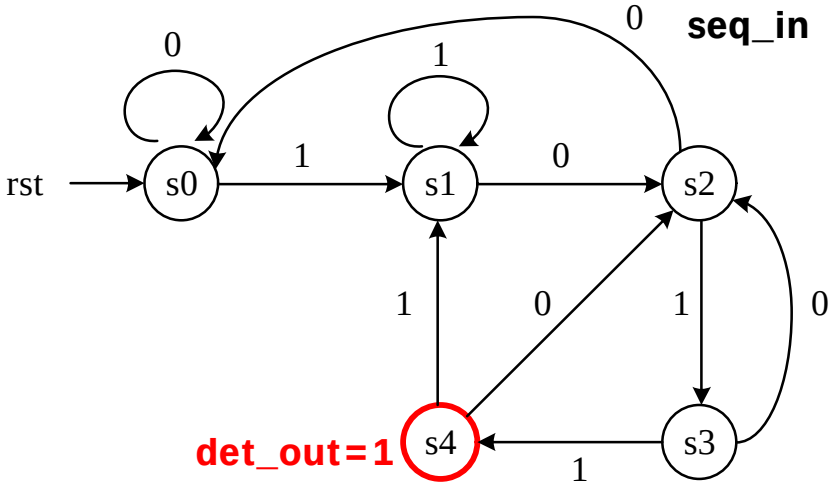
進入state S3時，此時seq_in為1
因此det_out立即變為1，與clk不同步

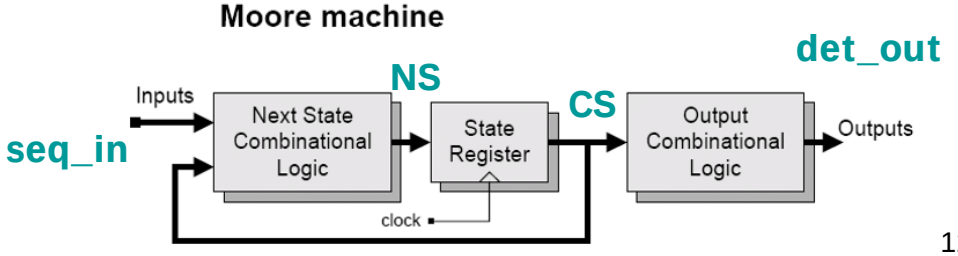
Design 2: Moore Machine



Moore Machine State Diagram

- Target sequence: **1011**

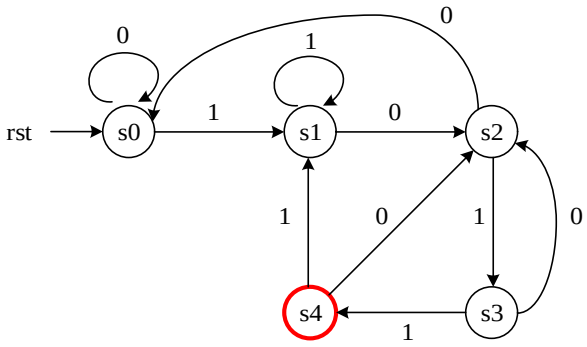




Verilog Code (1/4)

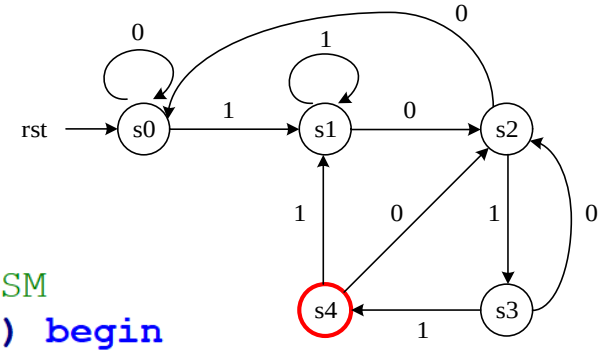
```
module sd_moore(clk, rst, seq_in, det_out, state_out);
    input clk, rst, seq_in;
    output reg det_out;
    output [2:0] state_out;

    parameter
        s0=3'b000,  // no valid bit
        s1=3'b001,  // valid seq: 1
        s2=3'b010,  // valid seq: 10
        s3=3'b011,  // valid seq: 101
        s4=3'b100;  // valid seq: 1011
```

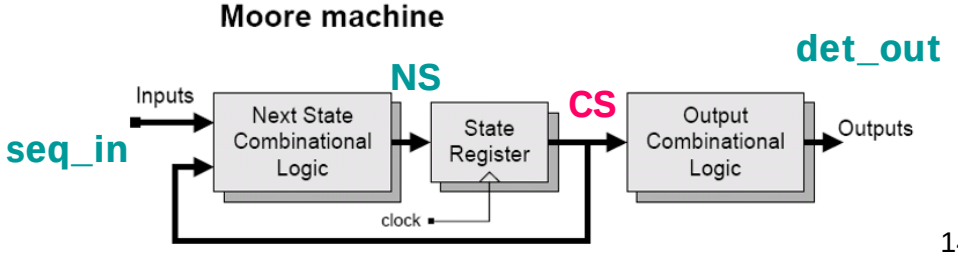
Five states



Verilog Code (2/4)

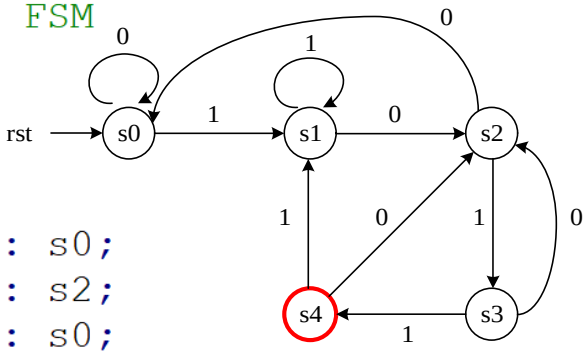
```
reg [2:0] CS, NS;
// CS: current state
// NS: next state

assign state_out = CS;

// sequential part of the Moore FSM
always @(posedge clk, posedge rst) begin
    #2
    if (rst==1)
        CS = s0;
    else
        CS = NS;
end
```

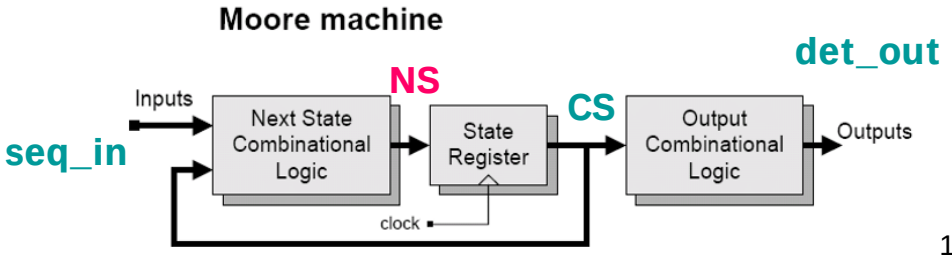




Verilog Code (3/4)

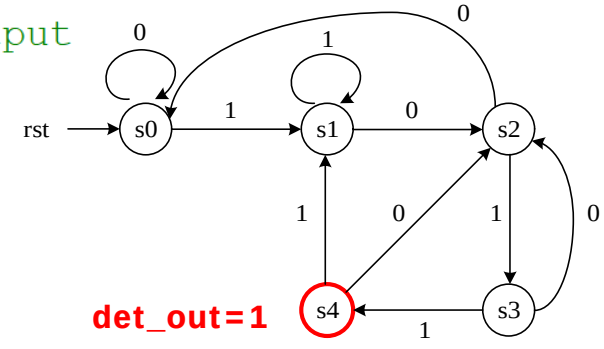
```
// combinational part of the Moore FSM
// determine the next state
always @(CS, seq_in) begin
    #2
    case (CS)
        s0: NS = (seq_in==1) ? s1 : s0;
        s1: NS = (seq_in==1) ? s1 : s2;
        s2: NS = (seq_in==1) ? s3 : s0;
        s3: NS = (seq_in==1) ? s4 : s2;
        s4: NS = (seq_in==1) ? s1 : s2;
        default: NS = s0;
    endcase
end
```

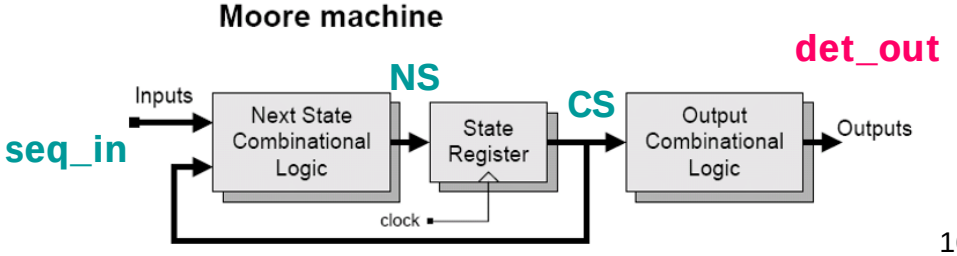




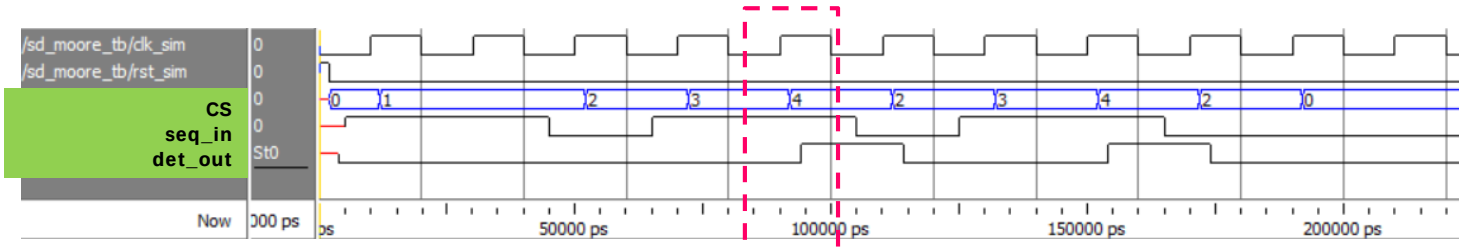
Verilog Code (4/4)

```
    // combinational part of the Moore FSM
    // determine the detection output
    always @(CS) begin
        #2
        case (CS)
            s0: det_out = 0;
            s1: det_out = 0;
            s2: det_out = 0;
            s3: det_out = 0;
            s4: det_out = 1;
            default: det_out = 0;
        endcase
    end
endmodule
```



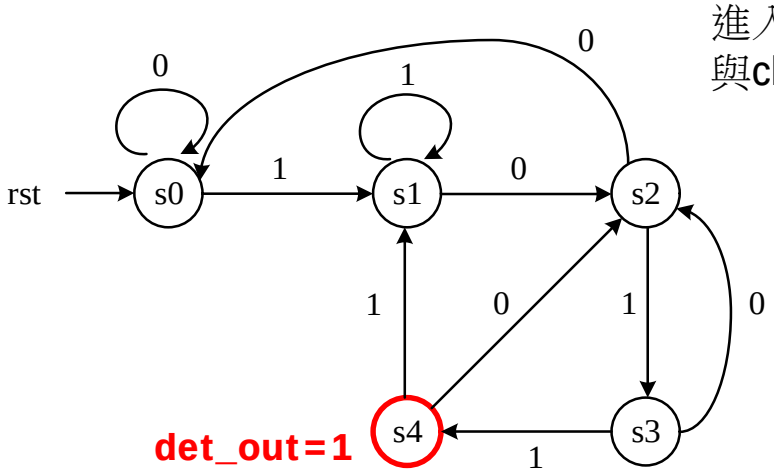


Simulation Result



seq_in:	1	1	0	1	1	0	1	1	0	
det_out:					**1**			**1**		

進入state S4時，det_out變為1，與clk同步



作業描述

- 請利用Mealy及Moore兩種FSM，設計兩個sequence detector電路
- Target sequence: **10111**
- Input sequence: **111011110111011101110111000**
- Detection output: **000000100001000100010001000**

作業繳交

- 繳交之作業檔案請以”您的學號_04”(ex. **S0754000_04**)命名，包含
  - Verilog原始檔 (請全部壓縮成一個**S0754000_04**.zip)
  - 作業報告檔 (請參考範本 **S0754000_04**.ppt)
- 繳交方式：彰化師大雲端學院，課程作業區，以附檔作答
- 繳交期限：**2019/12/25 (三) 12:00**
- 有關課程及作業相關問題，可於FB社團 (**2019_硬體描述語言**) 提出討論