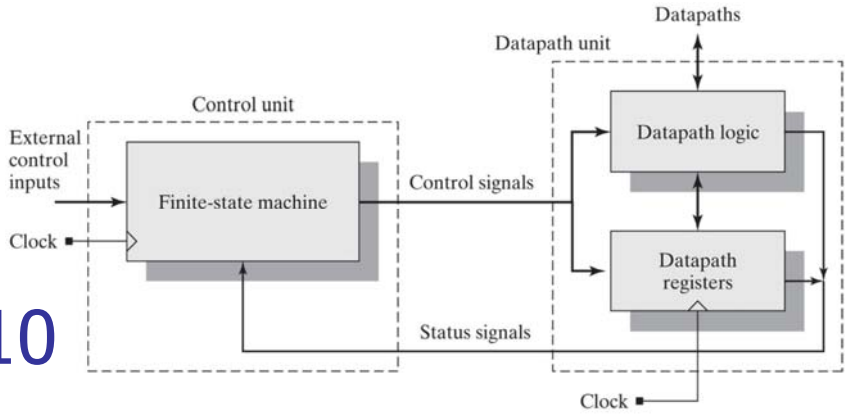


Topic 10

Finite State Machines & (ASM Charts)

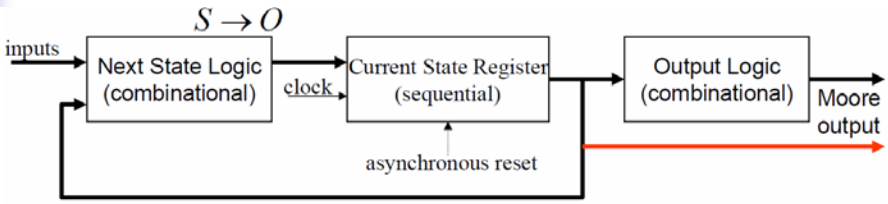
國立高雄大學電機工程學系

陳春僥

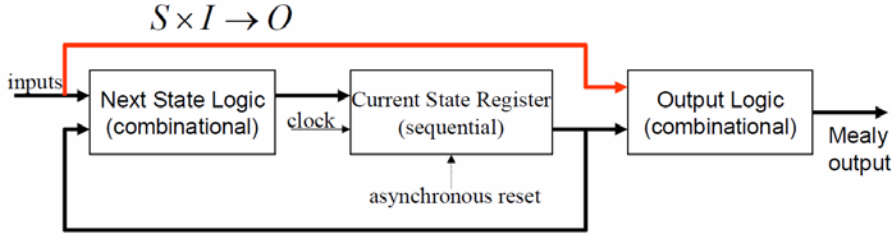
cychen@ieee.org

June 10, 2021

Finite State Machines

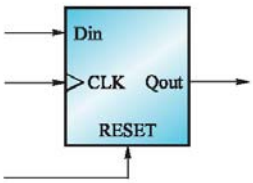


Moore Machine (state-based machine)

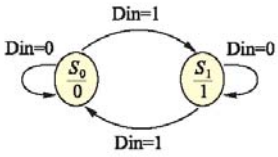


Mealy Machine (input-based machine)

Odd-Parity Checker (Moore)



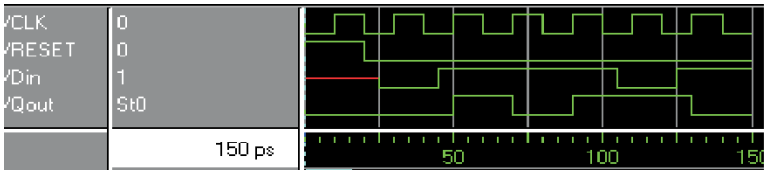
(a) 方塊圖



(b) 狀態圖

目前狀態	下一個狀態		輸出
	Din=0	Din=1	Qout
S_0=0	S_0	S_1	0
S_1=1	S_1	S_0	1

(c) 狀態表



Odd-Parity Checker (Moore)

```
module MOORE_BIN2(CLK, Din, RESET, Qout);
input CLK, RESET;
input Din;
output Qout;

reg Qout;

//Declare the value for all states
parameter
  S0 = 1'b0,
  S1 = 1'b1;

  //Declare current state and next state variables
  reg CS;
  reg NS;

always @ (posedge CLK or posedge RESET)
    begin
      if (RESET == 1'b1)
        CS = S0;    //Initial state
      else
        CS = NS;
    end
```

1

```
always @ (CS or Din)
  begin
    case (CS)
      S0 : begin
           Qout = 1'b0;
           if (Din == 1'b0)
           NS = S0;
           else
           NS = S1;
           end
      S1 : begin
           Qout = 1'b1;
           if (Din == 1'b0)
           NS = S1;
           else
           NS = S0;
           end
    endcase
  end
endmodule
```

2

Odd-Parity Checker (Moore)

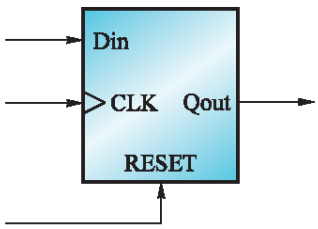
```
module testbench();

    // Inputs
        reg CLK;
        reg RESET;
        reg Din;
        wire Qout;

    // Instantiate the UUT
        MOORE_BIN2 uut (.CLK(CLK), .RESET(RESET), .Din(Din), .Qout(Qout));

    initial
        $monitor($time, " CLK=%b  RESET=%b Din=%b Qout=%b", CLK, RESET,
Din, Qout);
```

1

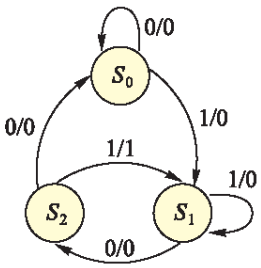
```
initial begin
    RESET = 1'b1;
    #20 RESET = 1'b0;
    #5 Din =0;
    #20 Din =1;
    #60 Din =0;
    #20 Din =1;
end

// Set up the clock to toggle every 10 time units
initial CLK = 1'b0;
always #10 CLK=~CLK;

//Finish the simulation at time 150
initial begin
    #150 $finish;
end
endmodule
```

2

Sequence (101) Detector (Mealy)



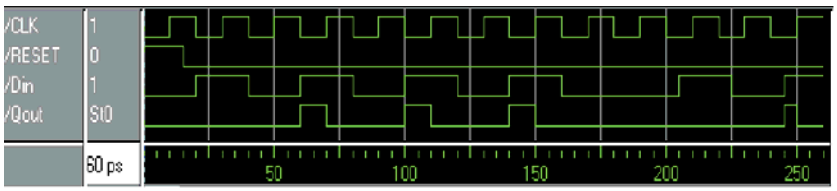
(a) 方塊圖



(b) 狀態圖

目前狀態	下一個狀態	
	Din=0	Din=1
S_0=00	S_0,0	S_1,0
S_1=01	S_2,0	S_1,0
S_2=11	S_0,0	S_1,1

(c) 狀態表



Sequence (101) Detector (Mealy)


```
module MEALY_GRY1(CLK, Din, RESET, Qout);
input CLK, RESET;
input Din;
output Qout;

reg Qout;

//Declare the value for all states
parameter [1:0]
  S0 = 2'b00,
  S1 = 2'b01,
  S2 = 2'b11;
```

1

```
//Declare current state and next state variables
reg [1:0] CS;
reg [1:0] NS;

always @ (posedge CLK or posedge RESET)
  begin
    if (RESET == 1'b1)
      CS = S0;     //Initial state
    else
      CS = NS;
  end
```

2

```
always @ (CS or Din)
  begin
    case (CS)
      S0 : begin
             if (Din == 1'b0)
               begin
                 NS = S0;
                 Qout = 1'b0;
               end
             else
               begin
                 NS = S1;
                 Qout = 1'b0;
               end
           end
      S1 : begin
             if (Din == 1'b0)
               begin
                 NS = S2;
                 Qout = 1'b0;
               end
             else
               begin
                 NS = S1;
                 Qout = 1'b0;
               end
           end
      S2 : begin
             if (Din == 1'b0)
               begin
                 NS = S0;
                 Qout = 1'b0;
               end
             else
               begin
                 NS = S1;
                 Qout = 1'b1;
               end
           end
    endcase
  end
endmodule
```

3

4

Sequence (101) Detector (Mealy)


```
module testbench();

// Inputs
    reg CLK;
    reg RESET;
    reg Din;

// Outputs
    wire Qout;

// Instantiate the UUT
    MEALY_GRY1 uut (.CLK(CLK), .RESET(RESET), .Din(Din), .Qout(Qout));
    initial
        $monitor($time, " CLK=%b  RESET=%b Din=%b Qout=%b", CLK, RESET,
Din, Qout);
```

1

```
initial begin
    #0 begin   RESET = 1'b1; Din=0; end
    #15 RESET = 1'b0;
    #5 Din =1;
    #20 Din =0;
    #20 Din =1;
    #20 Din =0;
    #20 Din =1;
    #20 Din =0;
    #20 Din =1;
    #20 Din =0;
    #45 Din =1;
    #20 Din =0;
    #20 Din =1;
    #20 Din =0;
    #20 Din =1;
end

// Set up the clock to toggle every 10 time units
initial CLK = 1'b0;
always #10 CLK=~CLK;

//Finish the simulation at time 290
initial begin
    #290 $finish;
end
endmodule
```

2

3