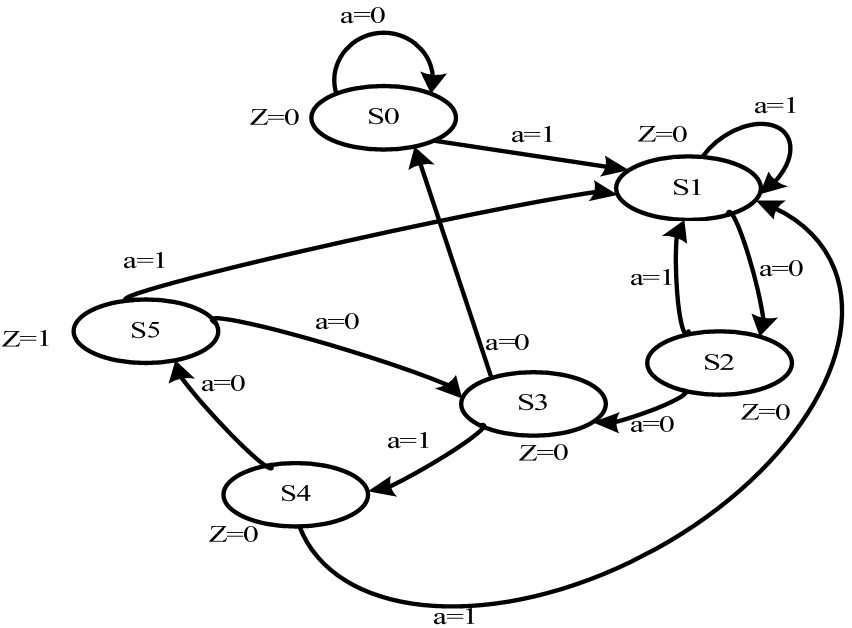
**Moore型verilog源代码：FSM实现10010串的检测**

**Moore状态转移图**



**module moorefsm(clk,rst,a,z);**

**input clk,rst;**

**input a;**

**output z;**

**reg z;**

**reg [3:0] currentstate,nextstate;**

**parameter S0 = 4'b0000;**

**parameter S1 = 4'b0001;**

**parameter S2 = 4'b0010;**

**parameter S3 = 4'b0011;**

**parameter S4 = 4'b0100;**

**parameter S5 = 4'b0101;**

**always@(posedge clk or negedge rst)**

**begin**

**if(!rst)**

**currentstate <= S0;**

**else**

**currentstate <= nextstate;**

**end**

**always@(currentstate or a or rst)**

**begin**

**if(!rst)**

**nextstate = S0;**

**else**

**case(currentstate)**

**S0: nextstate = (a==1)?S1:S0;**

**S1: nextstate = (a==0)?S2:S1;**

**S2: nextstate = (a==0)?S3:S1;**

**S3: nextstate = (a==1)?S4:S0;**

**S4: nextstate = (a==0)?S5:S1;**

**S5: nextstate = (a==0)?S3:S1;**

**default: nextstate = S0;**

**endcase**

**end**

**always@(rst or currentstate)**

**begin**

**if(!rst)**

**z = 0;**

**else**

**case(currentstate)**

**S0: z = 0;S1: z = 0;S2: z = 0;**

**S3: z = 0;S4: z = 0;S5: z = 1;**

**default: z = 0;**

**endcase**

**end**

**endmodule**

**moorefsm测试模块testbench**

**module tb\_fsm;**

**reg clk,rst;**

**reg a;**

**wire z;**

**moorefsm**

**fsm(.clk(clk),.rst(rst),.a(a),.z(z));**

**initial**

**begin**

**clk = 0;**

**rst = 1;**

**#5 rst = 0;**

**#3 rst = 1;**

**#20 a = 1;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

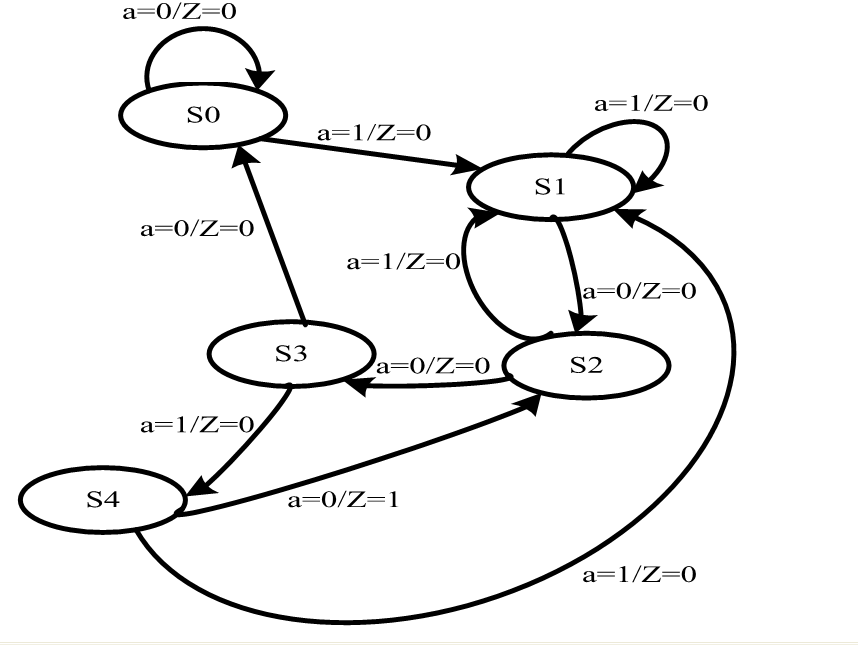
**end**

**always #50 clk = ~clk;**

**endmodule**

**Mealy型verilog源代码：FSM实现10010串的检测**

**Mealy状态转移图**



**module mealyfsm(clk,rst,a,z);**

**input clk;**

**input rst;**

**input a;**

**output z;**

**reg z;**

**reg [3:0] temp\_z;**

**reg [3:0] currentstate,nextstate;**

**parameter S0 = 4'b0000;**

**parameter S1 = 4'b0001;**

**parameter S2 = 4'b0010;**

**parameter S3 = 4'b0011;**

**parameter S4 = 4'b0100;**

**always@(posedge clk or negedge rst)**

**if(!rst)**

**currentstate <= S0;**

**else**

**currentstate <= nextstate;**

**always@(currentstate or a or rst)**

**if(!rst)**

**nextstate = S0;**

**else**

**case(currentstate)**

**S0: nextstate = (a == 1)? S1 : S0;**

**S1: nextstate = (a == 0)? S2 : S1;**

**S2: nextstate = (a == 0)? S3 : S1;**

**S3: nextstate = (a == 1)? S4 : S0;**

**S4: nextstate = (a == 0)? S2 : S0;**

**default:nextstate = S0;**

**endcase**

**always@(rst or currentstate or a)**

**if(!rst)**

**temp\_z = 0;**

**else**

**case(currentstate)**

**S0: temp\_z = 0;**

**S1: temp\_z = 0;**

**S2: temp\_z = 0;**

**S3: temp\_z = 0;**

**S4: temp\_z = (a == 0)? 1 : 0;**

**default:temp\_z = 0;**

**endcase**

**always@(posedge clk or negedge rst)**

**if(!rst)**

**z <= 0;**

**else**

**begin**

**if((temp\_z == 1)&&(nextstate == S2))**

**z <= 1;**

**else**

**z <= 0;**

**end**

**endmodule**

**mealyfsm测试模块testbench**

**module tb\_fsm;**

**reg clk,rst;**

**reg a;**

**wire z;**

**mealyfsm**

**fsm(.clk(clk),.rst(rst),.a(a),.z(z));**

**initial**

**begin**

**clk = 0;**

**rst = 1;**

**#5 rst = 0;**

**#3 rst = 1;**

**#20 a = 1;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**#100 a = 1;**

**#100 a = 0;**

**end**

**always #50 clk = ~clk;**

**endmodule**