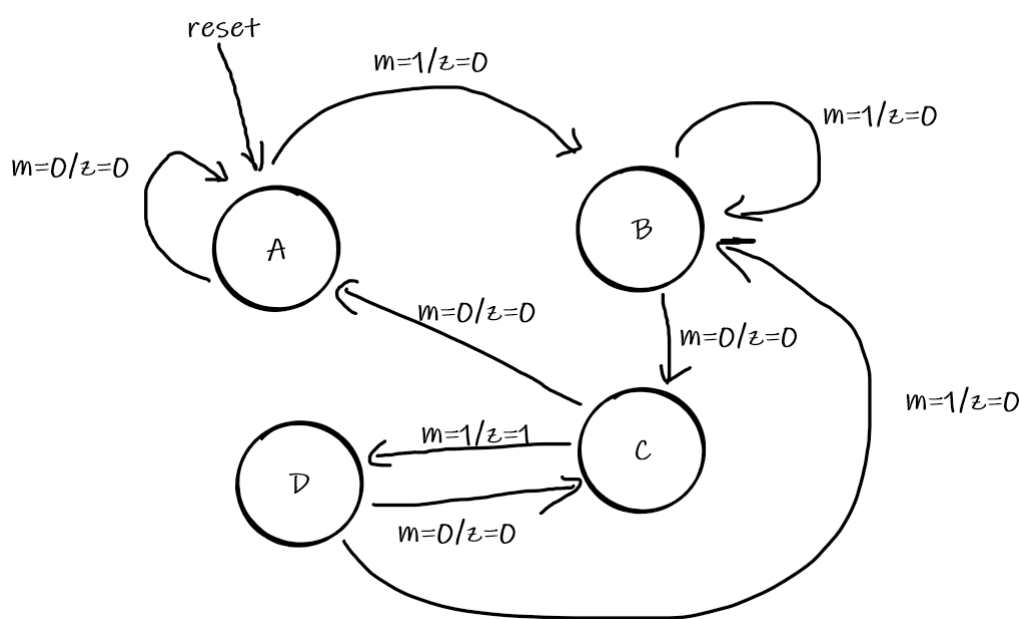


作业6

一、设计可重叠101序列检测器，同时用verilog开发该模块，并开发测试程序。

1、原始状态图（Mealy型输出）：

- 状态说明：
- A：未检测到可能的101序列；
 - B：检测到“1”；
 - C：检测到“10”；
 - D：检测到“101”。



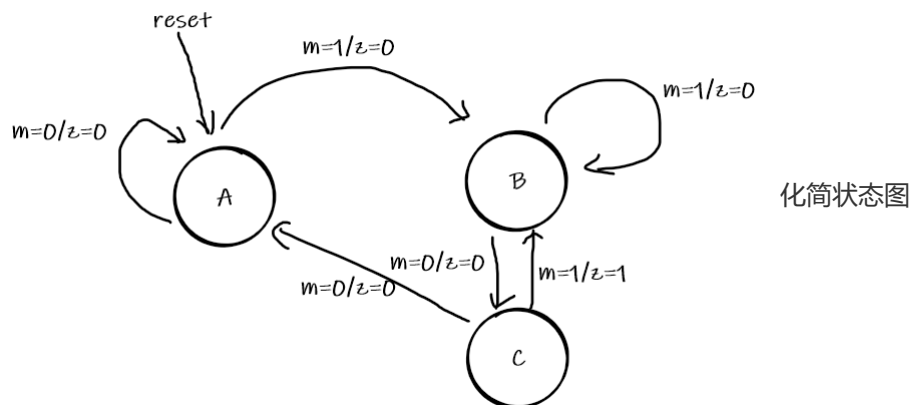
2、作状态表，由表可知，状态B和状态D等效，状态表可化简为图2

| 现态 | 次态/输出 | |
|----|-------|-----|
| | m=0 | m=1 |
| A | A/0 | B/0 |
| B | C/0 | B/0 |
| C | A/0 | D/1 |
| D | C/0 | B/0 |

图1

| 现态 | 次态/输出 | |
|----|-------|-----|
| | m=0 | m=1 |
| A | A/0 | B/0 |
| B | C/0 | B/0 |
| C | A/0 | B/1 |

图2



3、作状态分配表:

| 现态 y_2y_1 | 次态/输出 | |
|----------------|------------|------------|
| | $m=0$ | $m=1$ |
| | Y_2Y_1/Z | Y_2Y_1/Z |
| 00 | 00/0 | 01/0 |
| 01 | 10/0 | 01/0 |
| 10 | 00/0 | 01/1 |
| 11 | dd/0 | dd/0 |

4、分析次态函数表达式:

| y_2y_1 | | 00 | 01 | 11 | 10 |
|----------|---|----|----|----|----|
| m | 0 | 0 | 1 | d | 0 |
| | 1 | 0 | 0 | d | 0 |

Y2 卡诺图

| y_2y_1 | | 00 | 01 | 11 | 10 |
|----------|---|----|----|----|----|
| m | 0 | 0 | 0 | d | 0 |
| | 1 | 1 | 1 | d | 1 |

Y1 卡诺图

| y_2y_1 | | 00 | 01 | 11 | 10 |
|----------|---|----|----|----|----|
| m | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 |

Z 卡诺图

$$\begin{aligned}
 Y_1 &= m \\
 Y_2 &= \overline{m}y_1 \\
 Z &= my_2\overline{y_1}
 \end{aligned}$$

5、verilog代码:

```

`timescale 1ns / 1ps
module sequence(clk,reset,m,z);
    input  clk,reset,m;
    output reg z;

```

```

reg [2:1]y,Y;
parameter [2:1]A=2'b00,B=2'b01,C=2'b10;
always@(m,y)
begin
    if(reset == 0)
        begin
            z=0;Y=A;
        end
    else
        case(y)
            A:if(m)
                begin
                    z=0;Y=B;
                end
            else
                begin
                    z=0;Y=A;
                end
            B:if(m)
                begin
                    z=0;Y=B;
                end
            else
                begin
                    z=0;Y=C;
                end
            C:if(m)
                begin
                    z=1;Y=B;
                end
            else
                begin
                    z=0;Y=A;
                end
            default:
                begin
                    z=0;Y=2'bxx;
                end
        endcase
    end
    always@(negedge reset,posedge clk)
        if(reset == 0) y<=A;
        else y<=Y;
endmodule

```

6、测试程序

```

`timescale 1ns / 1ps
module sequence_testbench;
    reg clk,reset,m;
    wire z;
    sequence test(.clk(clk),.reset(reset),.m(m),.z(z));
    parameter list=18'b0101_0010_1010_1001_01; // 测试序列，可以更改
    reg [4:0]i; //计数
    initial begin
        clk=0;
        reset=0;
    end
endmodule

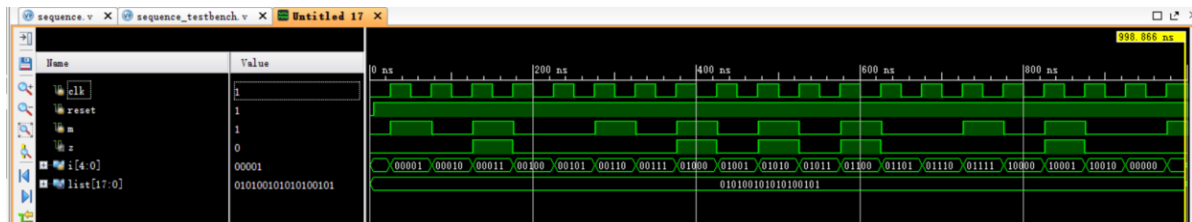
```

```

m=0;
#5 reset=1;
end
always #25 clk<=~clk; //一个周期25*2=50个，共展示1000ns内的，一个20个周期
always@(negedge reset,posedge clk)
begin
if(reset==0)
i<=5'b0;
else if(i<=5'd17)
begin
#1 m<=list[i];
i<=i+5'd1;
end
else
i<=5'b0;
end
endmodule

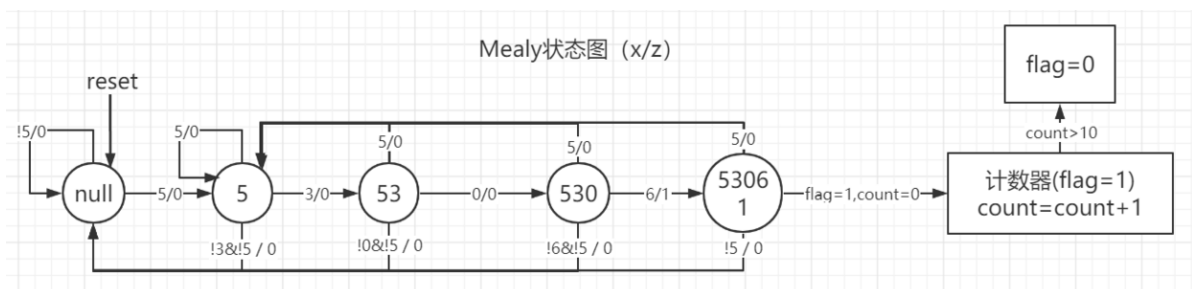
```

7、结果，符合可重叠101序列检测器的预期功能。



二、假设有一个数字系统，它通过识别输入的数据命令字5306来发动一次关键命令数据传输。关键命令数据有效生命时间为100us，请画出该数字系统的状态图。

(只做状态图，提示：100us时间是通过计数器产生，即计数器的时钟信号CLK每10 us产生一次，要保证100计时，需要计数器接收10次CLK。)



状态说明：

null: 检测初始状态，或未检测到可能的5306序列

5: 检测到一个“5”

53: 检测到一个“53”

530: 检测到一个“530”

5306: 检测到一个“5306”

功能说明：

检测到5306后，向计数器传达一个flag=1,count=0的命令，并继续接收下一个数字，并判断。
(flag=1: 一个命令信号，以发动一次关键命令数据传输；count=0: 计数器置零信号)

