```
1   module FSM_demo (Rst, Clk, RdestOut);
2      input Rst, Clk;
3
4      output [15:0] RdestOut;
5
6      reg [2:0] PS, NS;
7      reg [15:0] Imm;
8      reg [3:0] RdestRegLoc, RsrcRegLoc, OpCode, i;
9      reg en, Imm_s;
10
11     wire [4:0] Flags;
12
13     parameter S0 = 3'b000,
14               S1 = 3'b001,
15               S2 = 3'b010,
16               S3 = 3'b011,
17               S4 = 3'b100,
18               S5 = 3'b101,
19               S6 = 3'b110;
20
21     parameter ADD  = 4'b0000;
22
23
24     RegFile_Alu myRegAlu(
25        .RdestRegLoc(RdestRegLoc),
26        .RsrcRegLoc(RsrcRegLoc),
27        .Clk(Clk),
28        .En(en),
29        .Rst(Rst),
30        .Imm(Imm),
31        .Imm_s(Imm_s),
32        .OpCode(OpCode),
33        .RdestOut(RdestOut),
34        .Flags(Flags)
35     );
36
37        always@(negedge Rst, negedge Clk) begin
38        if(~Rst) begin
39           NS <= S0;
40        end
41        else begin
42           PS <= NS;
43           case(NS)
44              S0: NS <= S1;
45              S1: NS <= S2;
46              S2: NS <= S3;
47              S3: NS <= S3;
48
49              default: NS <= S0;
50           endcase
51        end
52     end
53
54     always@(PS) begin
55        case(PS)
56           S0: begin
57              RdestRegLoc = 4'b0000;
58              Imm_s = 1;
59              Imm = 1;
60              en = 1;
61              OpCode = 4'b0000;
62              RsrcRegLoc = 4'b0;
63           end
64
65           S1: begin
66              RdestRegLoc = 4'b0001;
67              Imm_s = Imm_s;
68              Imm = 2;
69              en = en;
70              OpCode = OpCode;
71              RsrcRegLoc = RsrcRegLoc;
72           end
73
```

```
74            S2: begin
75               RdestRegLoc = RdestRegLoc;
76               Imm_s = 0;
77               Imm = Imm;
78               en = en;
79               OpCode = OpCode;
80               RsrcRegLoc = RsrcRegLoc;
81            end
82
83            S3: begin
84               RdestRegLoc = RdestRegLoc;
85               Imm_s = Imm_s;
86               Imm = Imm;
87               en = 0;
88               OpCode = OpCode;
89               RsrcRegLoc = RsrcRegLoc;
90            end
91         endcase
92      end
93
94   endmodule
```