```
module StateMachine(
    input Up,
    input Down,
    input Center,
    input SEC2,
    input Moving,
    input HIT,
    input clk,
    output Rungame,
    output INITSTATE,
    output TimerStart2Sec,
    output Frog_Up,
    output Frog_Down,
    output Frog_Blink,
    output Reset
    );
    wire up, down, center;
    wire INIT, DELAY2SEC, FORWARD, UP, DOWN, END_DELAY;
    wire Next_INIT, Next_DELAY2SEC, Next_FORWARD, Next_UP, Next_DOWN, Next_END_DELAY;
    FDRE #(.INIT(1'b0)) sync1 (.C(clk), .R(1'b0), .CE(1'b1), .D(Up), .Q(up));
    FDRE #(.INIT(1'b0)) sync2 (.C(clk), .R(1'b0), .CE(1'b1), .D(Down), .Q(down));
    FDRE #(.INIT(1'b0)) sync3 (.C(clk), .R(1'b0), .CE(1'b1), .D(Center), .Q(center));


    assign Next_INIT = INIT & !center;
    FDRE #(.INIT(1'b1)) init     (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_INIT),
.Q(INIT));

    assign Next_DELAY2SEC = (DELAY2SEC & !SEC2) | (INIT & center) | (END_DELAY &
center);
    FDRE #(.INIT(1'b0)) del2sec  (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_DELAY2SEC),
.Q(DELAY2SEC));

    assign Next_FORWARD = (FORWARD & (up ~^ down) & !HIT) | (DELAY2SEC & SEC2) | (UP
& !Moving) | (DOWN & !Moving);
    FDRE #(.INIT(1'b0)) forward  (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_FORWARD),
.Q(FORWARD));

    assign Next_UP = (UP & Moving & !HIT) | (FORWARD & up & !down) | (UP & !Moving &
up & !down);
    FDRE #(.INIT(1'b0)) upff     (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_UP),
.Q(UP));

    assign Next_DOWN = (DOWN & Moving & !HIT) | (FORWARD & !up & down) | (DOWN &
!Moving & !up & down);
    FDRE #(.INIT(1'b0)) downff   (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_DOWN),
.Q(DOWN));

    assign Next_END_DELAY = (END_DELAY & !center) | (FORWARD & HIT) | (UP & HIT) |
(DOWN & HIT);
    FDRE #(.INIT(1'b0)) endDelay (.C(clk), .R(1'b0), .CE(1'b1), .D(Next_END_DELAY),
.Q(END_DELAY));

    assign INITSTATE = INIT;
    assign TimerStart2Sec = DELAY2SEC;
    assign Frog_Up = UP;
    assign Frog_Down = DOWN;
    assign Frog_Blink = DELAY2SEC | END_DELAY;
    assign Rungame = !(INIT | DELAY2SEC | END_DELAY);
    assign Reset = (END_DELAY & center);

endmodule
```