```
module top_module(
    input clk,
    input areset,    // Freshly brainwashed Lemmings walk left.
    input bump_left,
    input bump_right,
    input ground,
    input dig,
    output walk_left,
    output walk_right,
    output aaah,
    output digging
);

    localparam [2:0]
        walk_l    = 3'b000,
        walk_r    = 3'b001,
        fall_l    = 3'b010,
        fall_r    = 3'b011,
        dig_l     = 3'b100,
        dig_r     = 3'b101,
        splatter  = 3'b110;

    reg [2:0] state, next;
    reg [6:0] count;

    // Sequential block
    always @(posedge clk or posedge areset) begin
        if (areset) begin
            state <= walk_l;
            count <= 0;
        end else begin
            state <= next;
            if (state == fall_l || state == fall_r)
                count <= count + 1;
            else
                count <= 0;
        end
    end

    // Next-state logic
    always @(*) begin
        case (state)
            walk_l: begin
                if (!ground) next = fall_l;
                else if (dig) next = dig_l;
                else if (bump_left) next = walk_r;
                else next = walk_l;
            end
            walk_r: begin
                if (!ground) next = fall_r;
                else if (dig) next = dig_r;
                else if (bump_right) next = walk_l;
                else next = walk_r;
            end
            fall_l: begin
                if (ground)
                    next = (count > 19) ? splatter : walk_l;
                else
                    next = fall_l;
            end
            fall_r: begin
                if (ground)
                    next = (count > 19) ? splatter : walk_r;
                else
                    next = fall_r;
            end
            dig_l: next = ground ? dig_l : fall_l;
            dig_r: next = ground ? dig_r : fall_r;
            splatter: next = splatter;
            default: next = walk_l;
        endcase
    end

    // Output logic
    assign walk_left  = (state == walk_l);
    assign walk_right = (state == walk_r);
    assign aaah       = (state == fall_l || state == fall_r);
    assign digging    = (state == dig_l || state == dig_r);

endmodule
```