```
// Traffic Light Controller Module
module traffic_light_basic (
    input clk,
    input reset,
    input farm_car,
    output reg R1, Y1, G1, R2, Y2, G2
);

    reg [1:0] state;

    // State encoding
    parameter S0 = 2'b00; // Farm Green, Highway Red
    parameter S1 = 2'b01; // Farm Yellow, Highway Red
    parameter S2 = 2'b10; // Highway Green, Farm Red
    parameter S3 = 2'b11; // Highway Yellow, Farm Red

    always @(posedge clk or posedge reset) begin
        if (reset)
            state <= S2; // Start with highway green
        else begin
            case (state)
                S0: state <= S1;
                S1: state <= S2;
                S2: state <= S3;
                S3: state <= farm_car ? S0 : S2;
                default: state <= S2;
            endcase
```

```
        end
    end

    always @(*) begin
        // Turn off all lights by default
        R1 = 0; Y1 = 0; G1 = 0;
        R2 = 0; Y2 = 0; G2 = 0;

        case (state)
            S0: begin R1 = 1; G2 = 1; end
            S1: begin R1 = 1; Y2 = 1; end
            S2: begin R2 = 1; G1 = 1; end
            S3: begin R2 = 1; Y1 = 1; end
        endcase
    end
endmodule
```

// Test Bench for Traffic Light Controller

```
module tb_traffic_light_basic;

    reg clk;
    reg reset;
    reg farm_car;
    wire R1, Y1, G1, R2, Y2, G2;

    // Instantiate the traffic light controller
    traffic_light_basic uut (
```

```
    .clk(clk),
    .reset(reset),
    .farm_car(farm_car),
    .R1(R1), .Y1(Y1), .G1(G1),
    .R2(R2), .Y2(Y2), .G2(G2)
);

// Clock generation: 10ns period
initial begin
    clk = 0;
    forever #5 clk = ~clk;
end

// Stimulus process
initial begin
    // Initial reset
    reset = 1;
    farm_car = 0;
    #10 reset = 0;

    // Simulate farm car detection
    #40 farm_car = 1;
    #40 farm_car = 0;

    // Continue simulation
    #100 $finish;
end
```

```
    // Print header once at start
    initial begin
        $display("\nTime(ns) | farm_car |  R1  Y1  G1  |  R2  Y2  G2");
        $display("-----------------------------------------------");
    end

    // Print signal states at every positive clock edge
    always @(posedge clk) begin
        $display("%8t |    %b    | %b   %b   %b  | %b   %b   %b",
                 $time, farm_car, R1, Y1, G1, R2, Y2, G2);
    end

endmodule
```