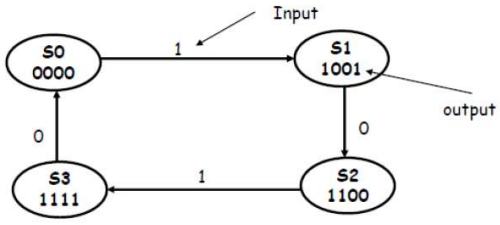
ECE 581 Homework 2

Fall 2021

Chuck Faber

Problem 1.

1) Write the SV code for the following Moore FSM. Write a testbench for it. Test your code with a simulator. Give all the codes and the simulated results.



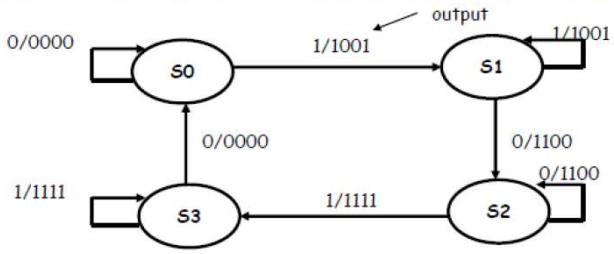
Moore FSM Code

```
module mooreFSM (
    input in, rst, clk,
    output logic [3:0] out
    );
    typedef enum logic [1:0] {S0, S1, S2, S3} states_t;
    states_t cs, ns;

    // Sequential Logic
    always_ff @(posedge clk, posedge rst) begin: set_ns
        if (rst) cs <= S0;
        else cs <= ns;
    end: set_ns

    // Next State Logic
    always_comb begin: ns_logic
        ns = cs;
        unique case (cs)
            S0: begin
                if (in) ns = S1;
                if (!in) ns = S0;
            end
            S1: begin
                if (in) ns = S1;
                if (!in) ns = S2;
            end
            S2: begin
                if (in) ns = S3;
                if (!in) ns = S2;
            end
            S3: begin
                if (in) ns = S3;
                if (!in) ns = S0;
            end
            default: ns = cs;
        endcase
    end: ns_logic

    // Output Logic
```

```
    always_comb begin: output_logic
        unique case (cs)
            S0: out = 4'b0000;
            S1: out = 4'b1001;
            S2: out = 4'b1100;
            S3: out = 4'b1111;
            default: out = 'x;
        endcase
    end: output_logic

endmodule
```

* Testbench and Transcript for both FSMs included with part 2 of this problem below.

2) Write the SV code for the following Mealy FSM. Write a testbench for it. Test your code with a simulator. Give all the codes and the simulated results.



Mealy FSM Code

```
module mealyFSM (
    input in, rst, clk,
    output logic [3:0] out
    );
    typedef enum logic [1:0] {S0, S1, S2, S3} states_t;
    states_t cs, ns;

    // Sequential Logic
    always_ff @(posedge clk, posedge rst) begin: set_ns
        if (rst) cs <= S0;
        else cs <= ns;
    end: set_ns

    // Next State Logic
    always_comb begin: ns_logic
        ns = cs;
        unique case (cs)
            S0: begin
                if (in) ns = S1;
                if (!in) ns = S0;
            end
            S1: begin
                if (in) ns = S1;
                if (!in) ns = S2;
            end
            S2: begin
                if (in) ns = S3;
                if (!in) ns = S2;
            end
            S3: begin
```

```
                if (in) ns = S3;
                if (!in) ns = S0;
            end
            default: ns = cs;
        endcase
    end: ns_logic

    // Output Logic
    always_comb begin: output_logic
        out = 'x;
        unique case (cs)
            S0: begin
                if (in) out = 4'b1001;
                if (!in) out = 4'b0000;
            end
            S1: begin
                if (in) out = 4'b1001;
                if (!in) out = 4'b1100;
            end
            S2: begin
                if (in) out = 4'b1111;
                if (!in) out = 4'b1100;
            end
            S3: begin
                if (in) out = 4'b1111;
                if (!in) out = 4'b0000;
            end
            default: out = 'x;
        endcase
    end: output_logic

endmodule
```

Moore and Mealy Testbench

```
module top();

    logic in, rst, clk;
    logic [3:0] trace;
    logic [3:0] out_moore, out_mealy;


    initial begin
        clk = 1'b1;
        forever #50 clk = ~clk;
    end

    mooreFSM moore (
        .out(out_moore),
        .* );

    mealyFSM mealy (
        .out(out_mealy),
        .* );

    task resetFSM();
        in = 1'bx;
```

```
        @(negedge clk) rst = 1'b1;
        @(negedge clk) rst = 1'b0;
        $display("\nResetting FSM.");
    endtask

    initial begin
        for (int i = 0; i < 16; i++) begin
            resetFSM();
            trace = i & 4'hF;
            $display($time, " Trace: %04b\tInput: %01b\tMoore State: %s\tMoore Output:%04b\tMealy State:
%s\tMealy Output: %04b", trace, in, moore.cs.name(), out_moore, mealy.cs.name(), out_mealy);
            for (int j = 0; j < 4; j++) begin
                repeat (1) @(negedge clk);
                in = trace[j];
                repeat (1) @(negedge clk);
                $display($time, " Trace: %04b\tInput: %01b\tMoore State: %s\tMoore Output:%04b\tMealy
State: %s\tMealy Output: %04b", trace, in, moore.cs.name(), out_moore, mealy.cs.name(), out_mealy);
            end
        end
        $finish;
    end

endmodule
```

Moore and Mealy Transcript

```
$ vsim -c top
Reading pref.tcl

# 2020.1

# vsim -c top
# Start time: 23:21:10 on Oct 27,2021
# Loading sv_std.std
# Loading work.top
# Loading work.mooreFSM
# Loading work.mealyFSM
VSIM 1> run -all
#
# Resetting FSM.
#                  150 Trace: 0000      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                  350 Trace: 0000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                  550 Trace: 0000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                  750 Trace: 0000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                  950 Trace: 0000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#
# Resetting FSM.
#                 1150 Trace: 0001      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                 1350 Trace: 0001      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                 1550 Trace: 0001      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                 1750 Trace: 0001      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                 1950 Trace: 0001      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#
# Resetting FSM.
#                 2150 Trace: 0010      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                 2350 Trace: 0010      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                 2550 Trace: 0010      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                 2750 Trace: 0010      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                 2950 Trace: 0010      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#
# Resetting FSM.
#                 3150 Trace: 0011      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                 3350 Trace: 0011      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                 3550 Trace: 0011      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                 3750 Trace: 0011      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                 3950 Trace: 0011      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
```

```
#
# Resetting FSM.
#                4150 Trace: 0100      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                4350 Trace: 0100      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                4550 Trace: 0100      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                4750 Trace: 0100      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                4950 Trace: 0100      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#
# Resetting FSM.
#                5150 Trace: 0101      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                5350 Trace: 0101      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                5550 Trace: 0101      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                5750 Trace: 0101      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#                5950 Trace: 0101      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#
# Resetting FSM.
#                6150 Trace: 0110      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                6350 Trace: 0110      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                6550 Trace: 0110      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                6750 Trace: 0110      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                6950 Trace: 0110      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#
# Resetting FSM.
#                7150 Trace: 0111      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                7350 Trace: 0111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                7550 Trace: 0111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                7750 Trace: 0111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                7950 Trace: 0111      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#
# Resetting FSM.
#                8150 Trace: 1000      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                8350 Trace: 1000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                8550 Trace: 1000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                8750 Trace: 1000      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#                8950 Trace: 1000      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#
# Resetting FSM.
#                9150 Trace: 1001      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                9350 Trace: 1001      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                9550 Trace: 1001      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                9750 Trace: 1001      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#                9950 Trace: 1001      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#
# Resetting FSM.
#               10150 Trace: 1010      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#               10350 Trace: 1010      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#               10550 Trace: 1010      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#               10750 Trace: 1010      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#               10950 Trace: 1010      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#
# Resetting FSM.
#               11150 Trace: 1011      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#               11350 Trace: 1011      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#               11550 Trace: 1011      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#               11750 Trace: 1011      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#               11950 Trace: 1011      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#
# Resetting FSM.
#               12150 Trace: 1100      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#               12350 Trace: 1100      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#               12550 Trace: 1100      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#               12750 Trace: 1100      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#               12950 Trace: 1100      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#
# Resetting FSM.
#               13150 Trace: 1101      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#               13350 Trace: 1101      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#               13550 Trace: 1101      Input: 0        Moore State: S2 Moore Output:1100       Mealy State: S2 Mealy Output: 1100
#               13750 Trace: 1101      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#               13950 Trace: 1101      Input: 1        Moore State: S3 Moore Output:1111       Mealy State: S3 Mealy Output: 1111
#
# Resetting FSM.
#               14150 Trace: 1110      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#               14350 Trace: 1110      Input: 0        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: 0000
#               14550 Trace: 1110      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
```

```
#                14750 Trace: 1110      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                14950 Trace: 1110      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#
# Resetting FSM.
#                15150 Trace: 1111      Input: x        Moore State: S0 Moore Output:0000       Mealy State: S0 Mealy Output: xxxx
#                15350 Trace: 1111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                15550 Trace: 1111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                15750 Trace: 1111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
#                15950 Trace: 1111      Input: 1        Moore State: S1 Moore Output:1001       Mealy State: S1 Mealy Output: 1001
# ** Note: $finish    : problem1.sv(156)
#    Time: 15950 ps  Iteration: 1  Instance: /top
# End time: 23:21:10 on Oct 27,2021, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
```

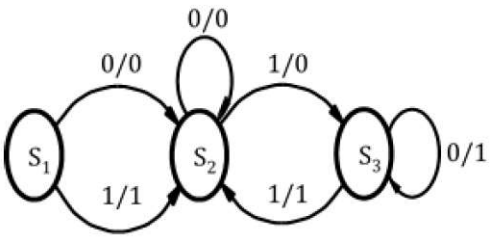
Discussion: In my testbench, I test all combinations of 4 bit long traces, resetting both FSMs at the conclusion of each trace. I tested both FSMs simultaneously with one testbench.

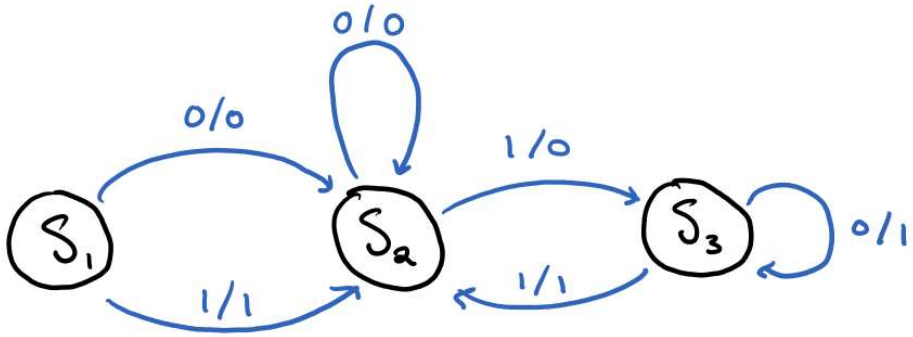
In my case statements, particularly in the Mealy output logic and the next state logic for both FSMs, I check for ‘in’ and ‘!in’ rather than using an if/else or ternary operator, because it is possible for ‘in’ to be neither ‘1’ nor ‘0’ (i.e. ‘x’ or ‘z’). An if/else would catch if ‘in’ were ‘1’ and in all other cases execute the ‘else’ statement which would not be correct behavior.

I adjust for this by initializing certain variables before using the case statement. For example, with the ‘out’ variable in the Mealy machine, it is initialized to ‘x’s before assigning to it using the unique case statement. Similarly I pre-assign the ‘ns’ (next state) variable with the ‘cs’ (current state) before assigning to it.

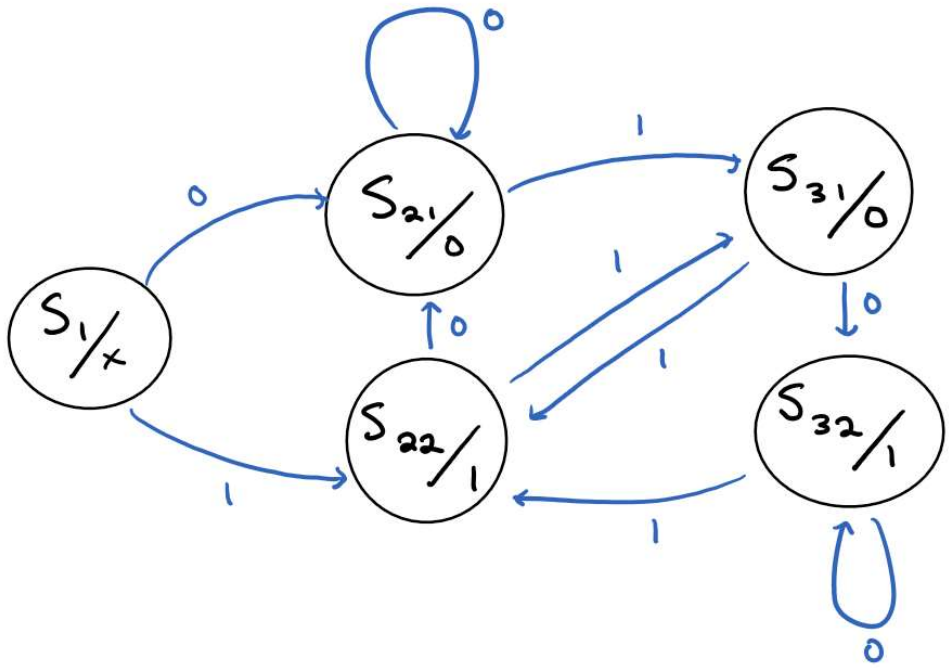
In addition, in every case I also provide a default case to avoid warnings when simulating when the case expression doesn’t match one of the case statements (i.e. if it is ‘x’).

Problem 2. Convert the following Mealy Machine into an equivalent Moore Machine.



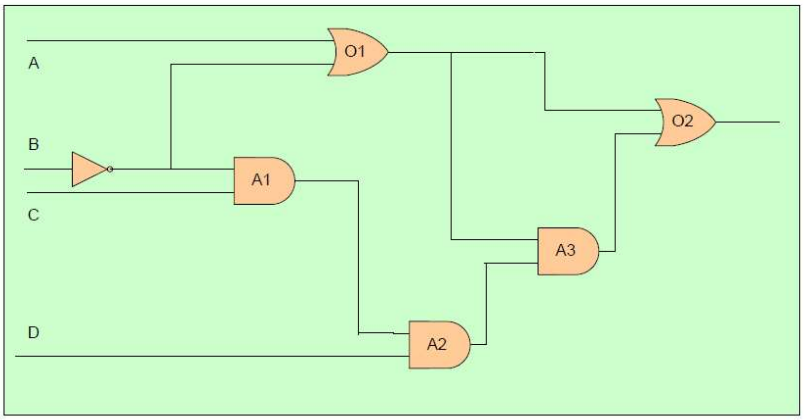


Mealy

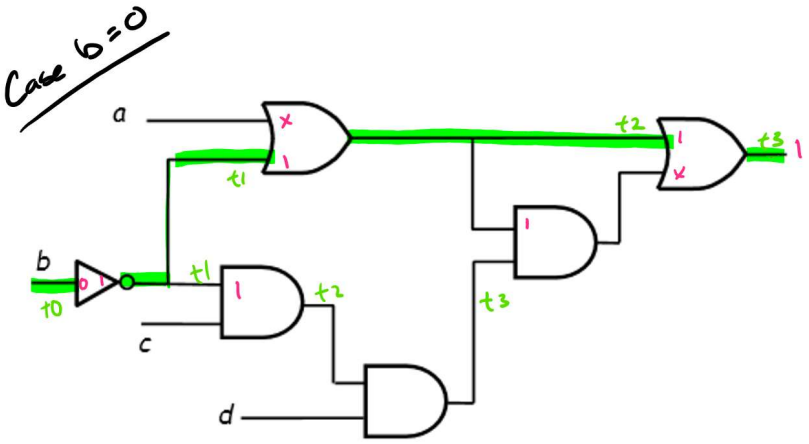


Moore

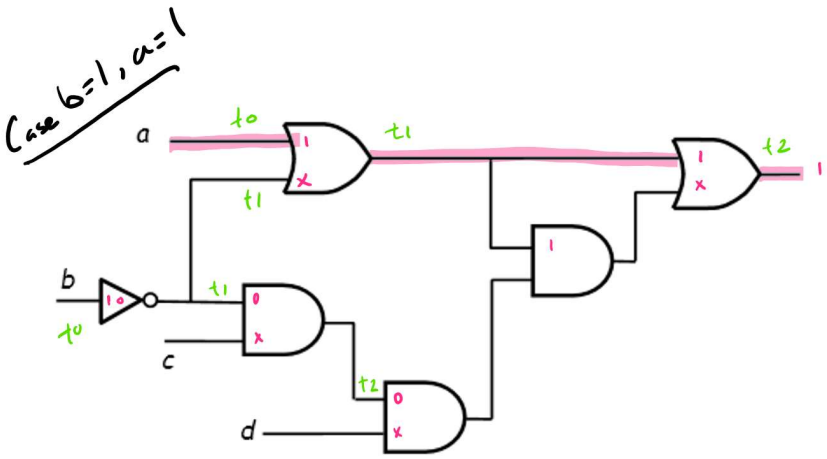
Problem 3. Assume that each gate has a delay t. Find the maximal propagation delay of the circuit. Justify your result.



The maximal propagation delay is 4t. The critical path is determined by the 'b' input. In the case of 'b' being 0, this would make the first OR gate output '1' regardless of what 'a' is. So theoretically, we wouldn't have to wait for the input of 'a' (though in this case, the input of 'b' happens at t1, so 'a' would already have a valid input at the gate by that time. If 'a' was 0, we would need to wait on the output of the inverter coming from 'b' to see what the output of the OR gate is. Since the first OR gate is a '1' this necessitates that the second OR gate is also a '1' while we don't care about the results from the bottom half of this circuit, so it only takes 3t to propagate the signal.



In the case of 'b'='1' we have to consider what happens with 'a'. If 'a' is '1' at t0, then it doesn't need to wait on the output from the inverter, and can begin to propagate a '1' all the way to the end within 2t.



However in the case where ‘a’ is ‘0’, the OR gate at the top must wait on the output of the inverter to determine if the output is a ‘0’ or a ‘1’ . Since we said ‘b’ is ‘1’, the output of the inverter will be 0, and the output of the first OR gate will be 0. The second OR gate must wait on the output of the last AND gate at t2. By t2, the bottom branch is only at the second AND gate, however because the last AND gate gets a ‘0’ at t2, the output of the final AND is locked to be ‘0’ and we don’t need to wait on the output from the second AND gate to propagate this change. So the final OR gate gets two ‘0’s at t3, and can finally output ‘0’ by t4. This is why the propagation delay is 4t.