Hacettepe University

Computer Engineering Department

BM233 Logic Design Lab - 2020 Fall

Lab Experiment 5

December 26, 2020

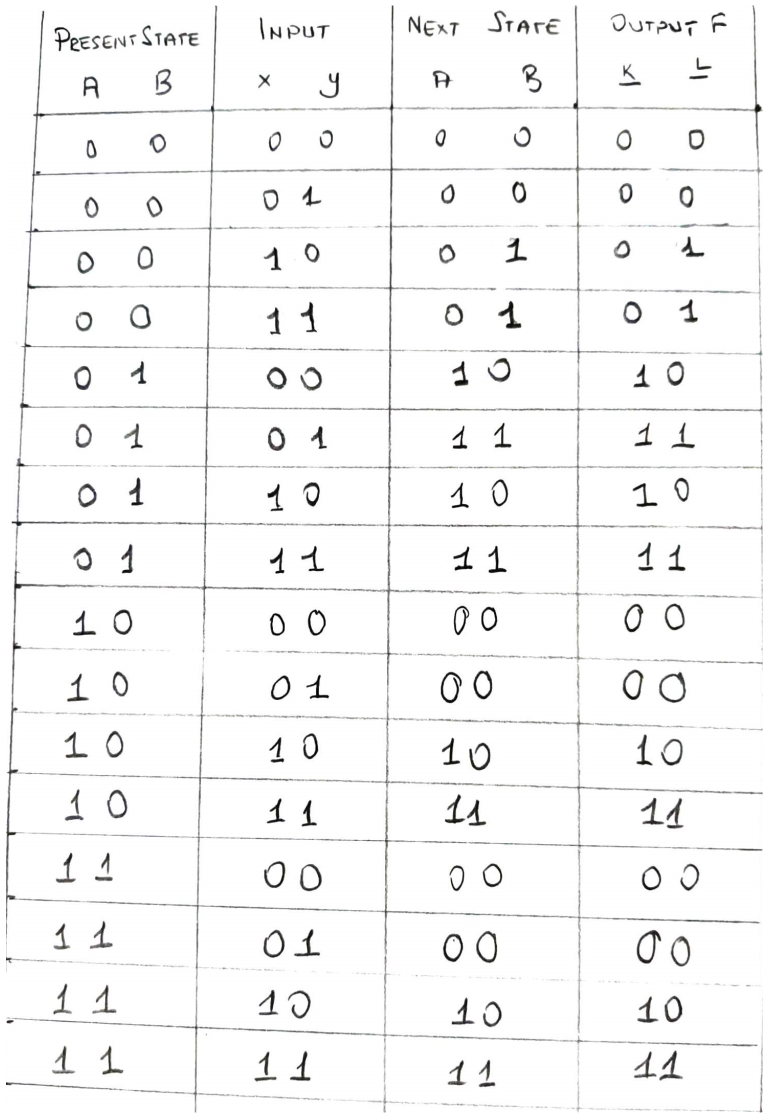
Student name:
Şevval Atmaca

Student Number:
b21827115

1 Problem Statement

In this experiment, I designed and simulated sequential circuits in Verilog. Firstly, I converted the given state transition diagram into a state transition table.Then we determine the number of flip-flops needed. I drew k-maps and found their equations by their k-map. I drew sequential circuit.With using this all we found we created a verilog.

2 State Transition Table



PRESENT STATE		INPUT		NEXT STATE		OUTPUT F	
A	B	x	y	A	B	K	L
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	0	1
0	1	0	0	1	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	1

Figure 1: State Transition Table

3 K-map

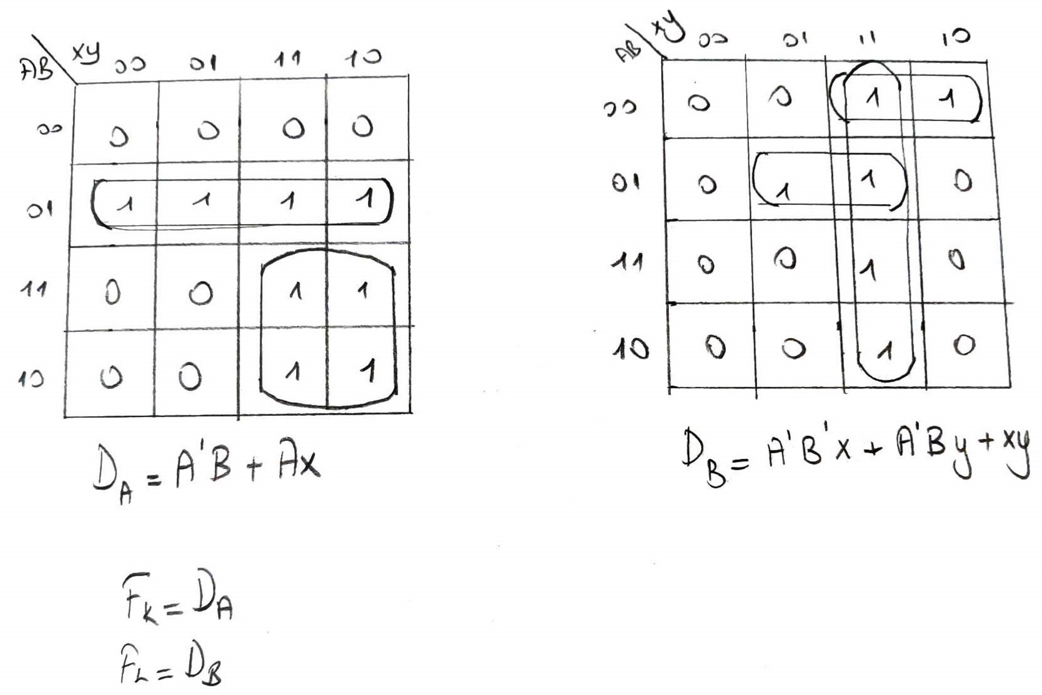


Figure 2: K-map

4 Circuit

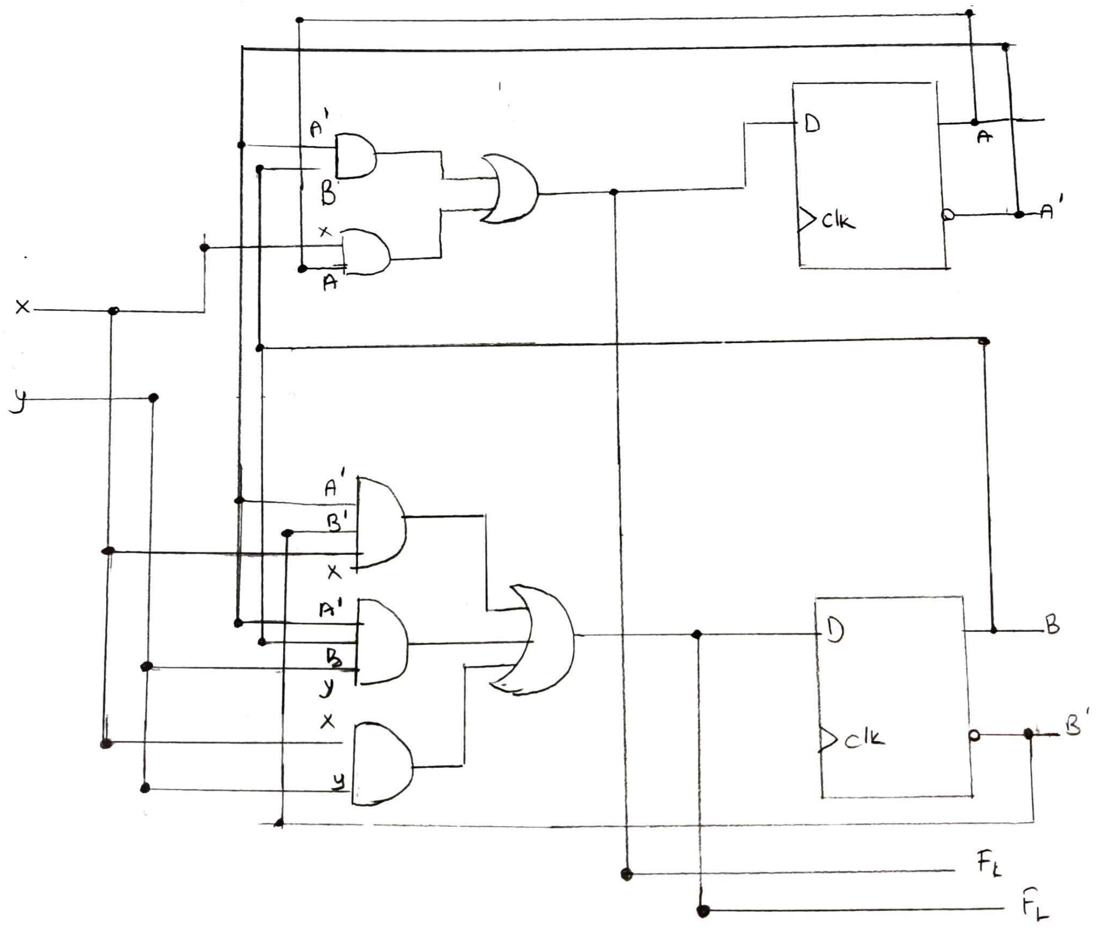


Figure 3: Circuit

5 Codes

```verilog
`timescale 1ns / 1ps


module Dflipflop(D, Clk,  Q);

    output reg  Q; //next
    input  D, Clk;

    always @(posedge Clk)
        begin
            Q <= D;
        end
endmodule
```

```verilog
`timescale 1ns / 1ps

module controller_tb;
    reg[1:0] M;//input
    reg clock;
    reg reset;
    wire[1:0] out;

    controller UUT(.M(M), .clock(clock), .reset(reset), .out(out));

    initial begin
        clock = 0;
        forever begin
            #10;//clock is changing in every 10 ns
            clock = ~clock;
        end
    end

    initial begin
        reset   = 1'b1;
        #50;
        reset = 1'b0;
        #100;
    end


    initial begin
        M= 2'b00;
        #50 M= 2'b00;
        #50; M=2'b10;
        #50 M= 2'b00;
        #20; M=2'b01;
        #50; M=2'b11;
        #50 M= 2'b10;
        #50; M=2'b11;
        #50; M=2'b01;
    end

endmodule
```

```
`timescale 1ns / 1ps
`include "Dflipflop.v"

module controller(
    input[1:0] M,//two inputs (x and y)
    input clock,
    input reset,
    output[1:0] out
    );

    reg[1:0] state = 2'b00;
    wire[1:0] next_state;
    wire[1:0] state1;
    wire t, t1;
    wire t2, t3, t4;

    and G1(t, ~state[1], state[0]);
    and G2(t1, state[1], M[1]);
    or G3(state1[1], t, t1);//state1

    and G4(t2, state[0], ~state[1], M[0]);
    and G5(t3, ~state[0], ~state[1], M[1]);
    and G6(t4, M[1], M[0]);
    or G7(state1[0] , t2, t3, t4);//state0

    Dflipflop D1(.D(state1[1]) ,.Clk(clock), .Q(next_state[1]));
    Dflipflop D0(.D(state1[0]) ,.Clk(clock), .Q(next_state[0]));

    always@ (reset or next_state) begin
        if(reset) begin state <= 2'b00; end
        else begin state<= next_state;end
    end

    or G8(out[1], t, t1);//output1
    or G9(out[0] , t2, t3, t4);//output0


endmodule
```

6 Waveform

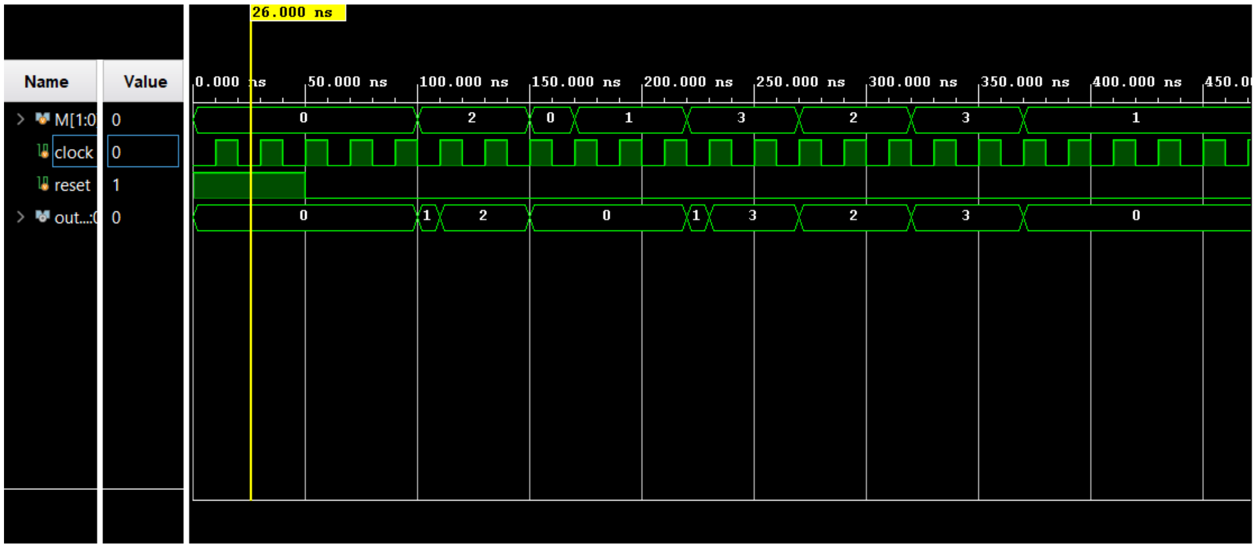


Figure 4: Circuit

7 Result

In my obtained result, I determined the initial state and then gave inputs and got outputs. I noticed that the previous output is affecting the next output. and also in the clock edge-responses output is changing. I understood that my results are correct by checking from the state transition diagram.

References

- lesson slide