CS 223

SECTION 3

LAB 3

BERAT BİÇER

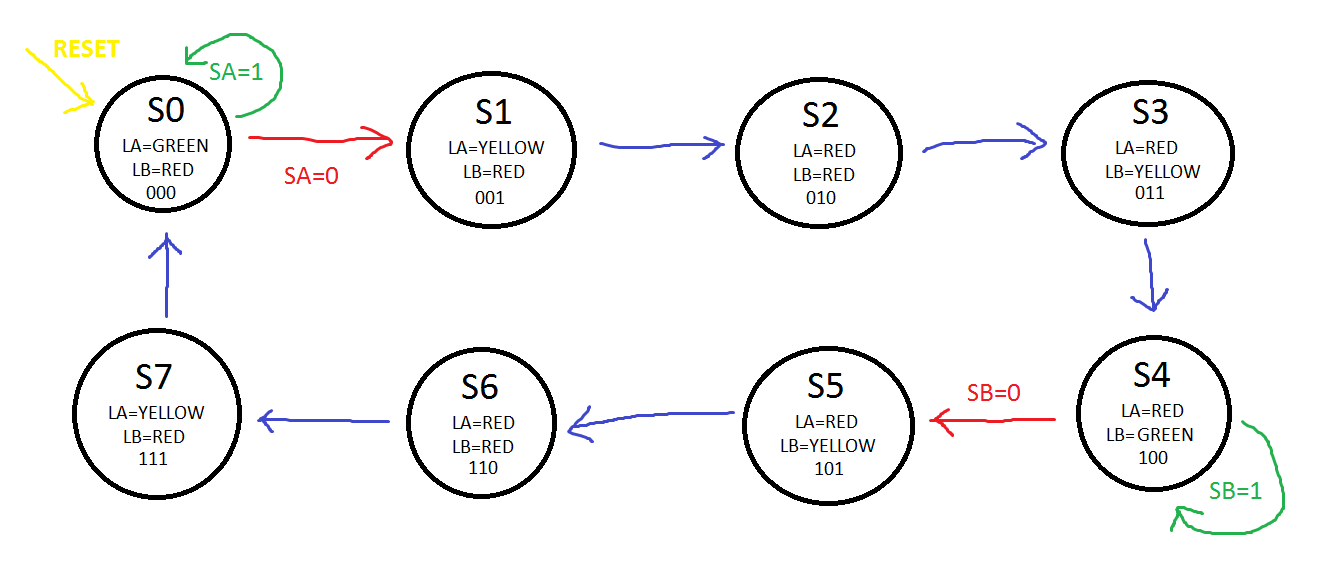
21503050

06.11.2016

BOX 7

**MOORE DIAGRAM**

**STATE ENCODING**



|  |  |  |  |
| --- | --- | --- | --- |
| STATE NAME | S2 | S1 | S0 |
| S0 | 0 | 0 | 0 |
| S1 | 0 | 0 | 1 |
| S2 | 0 | 1 | 0 |
| S3 | 0 | 1 | 1 |
| S4 | 1 | 0 | 0 |
| S5 | 1 | 0 | 1 |
| S6 | 1 | 1 | 0 |
| S7 | 1 | 1 | 1 |

**CURRENT-NEXT-OUTPUT TABLE**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CURRENT STATE | | | INPUTS | | NEXT STATE | | | OUTPUTS | | | |
| D2 | D1 | D0 | S1 | S0 | D2’ | D1’ | D0’ | LA1 | LA0 | LB1 | LB0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

**LOGIC EQUATIONS:**

D2’ = (D0 & D1) # D2

D1’ = (D1 # D0)

D0’ = (~D0) & (D1 + (~D2 & ~SA) + (D2 & ~SB))

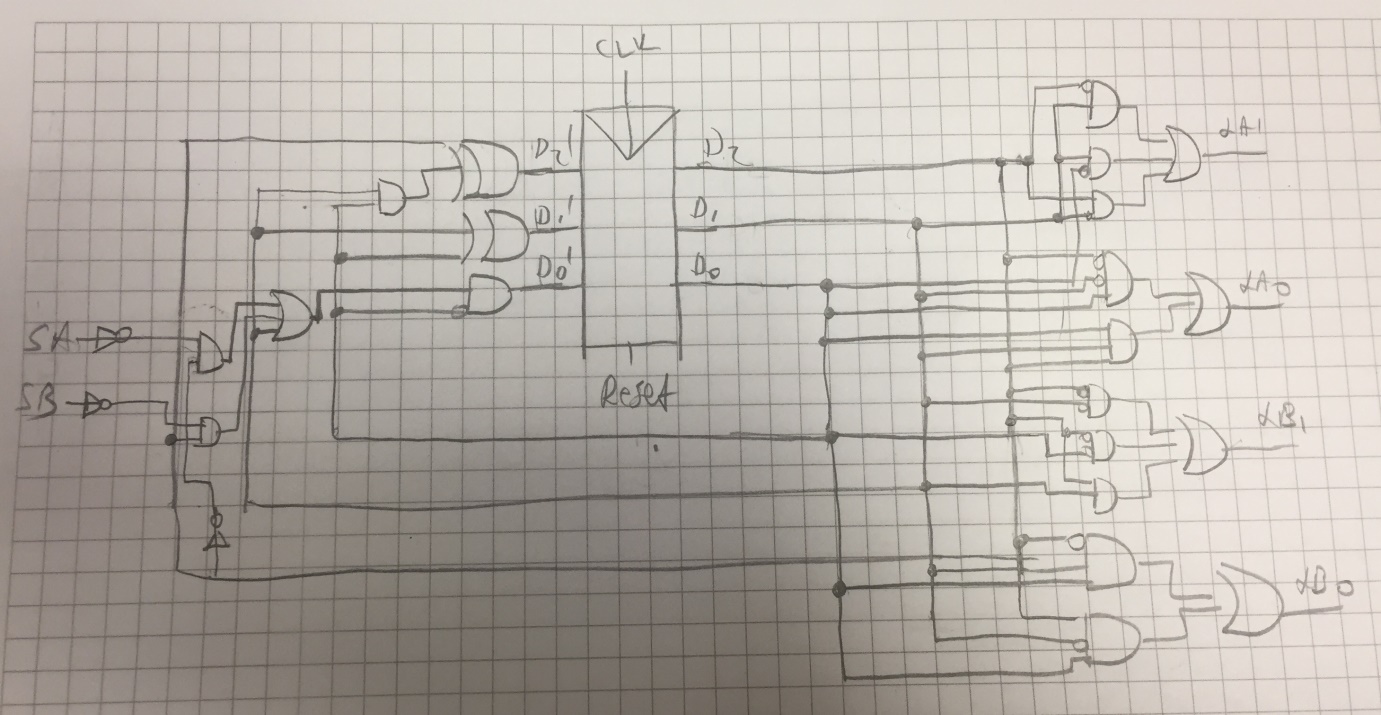
LA1 = (~D2 & D1) + (D1 & ~D0) + (D2 & ~D1)

LA0 = (~D2 & ~D1 & D0) + (D2 & D1 & D0)

LB1 = (~D2 & ~D1) + (~D2 & ~D0) + (D2 & D1)

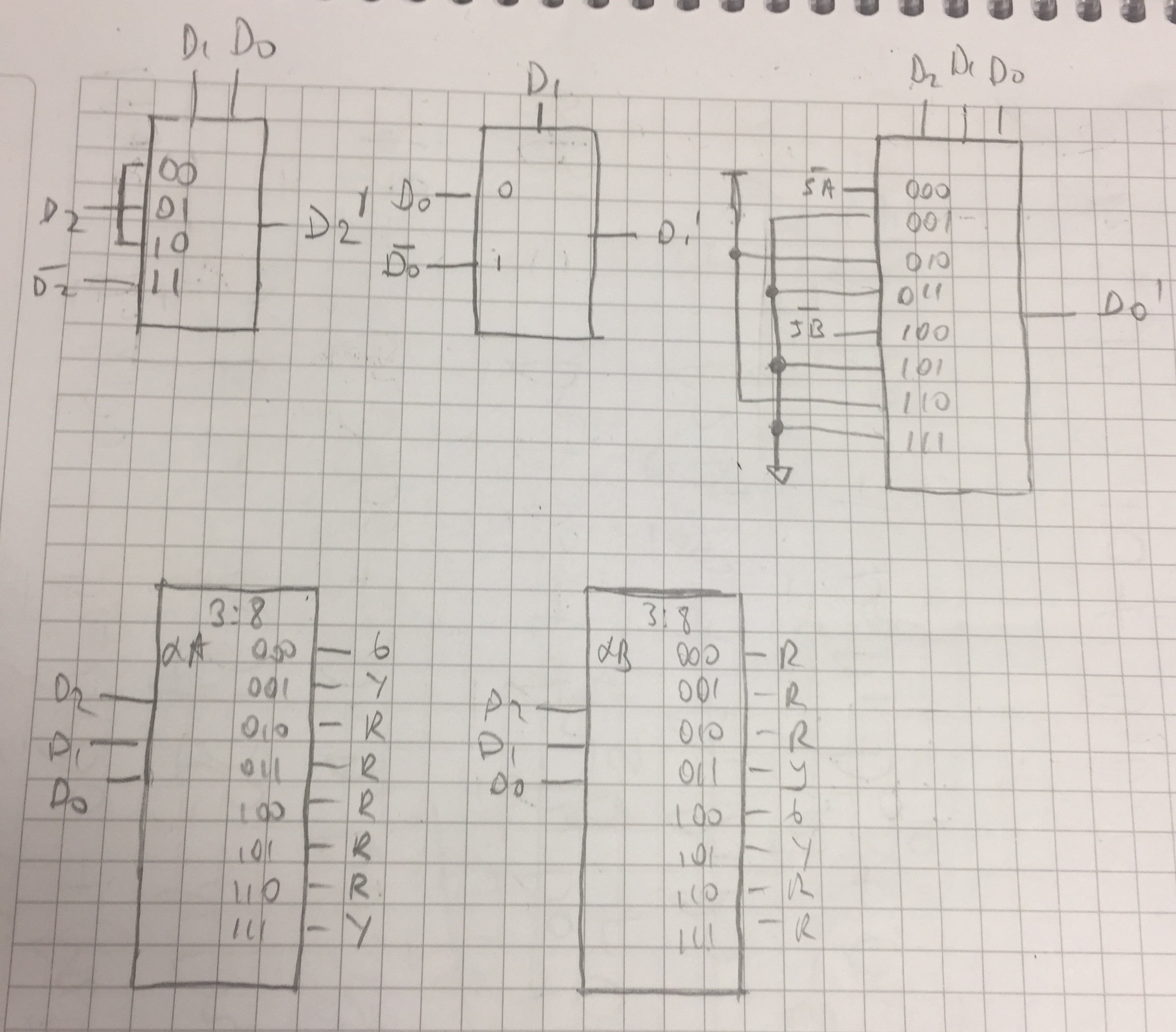
LB0 = (~D2 & D1 & D0) + (D2 & ~D1 & D0)

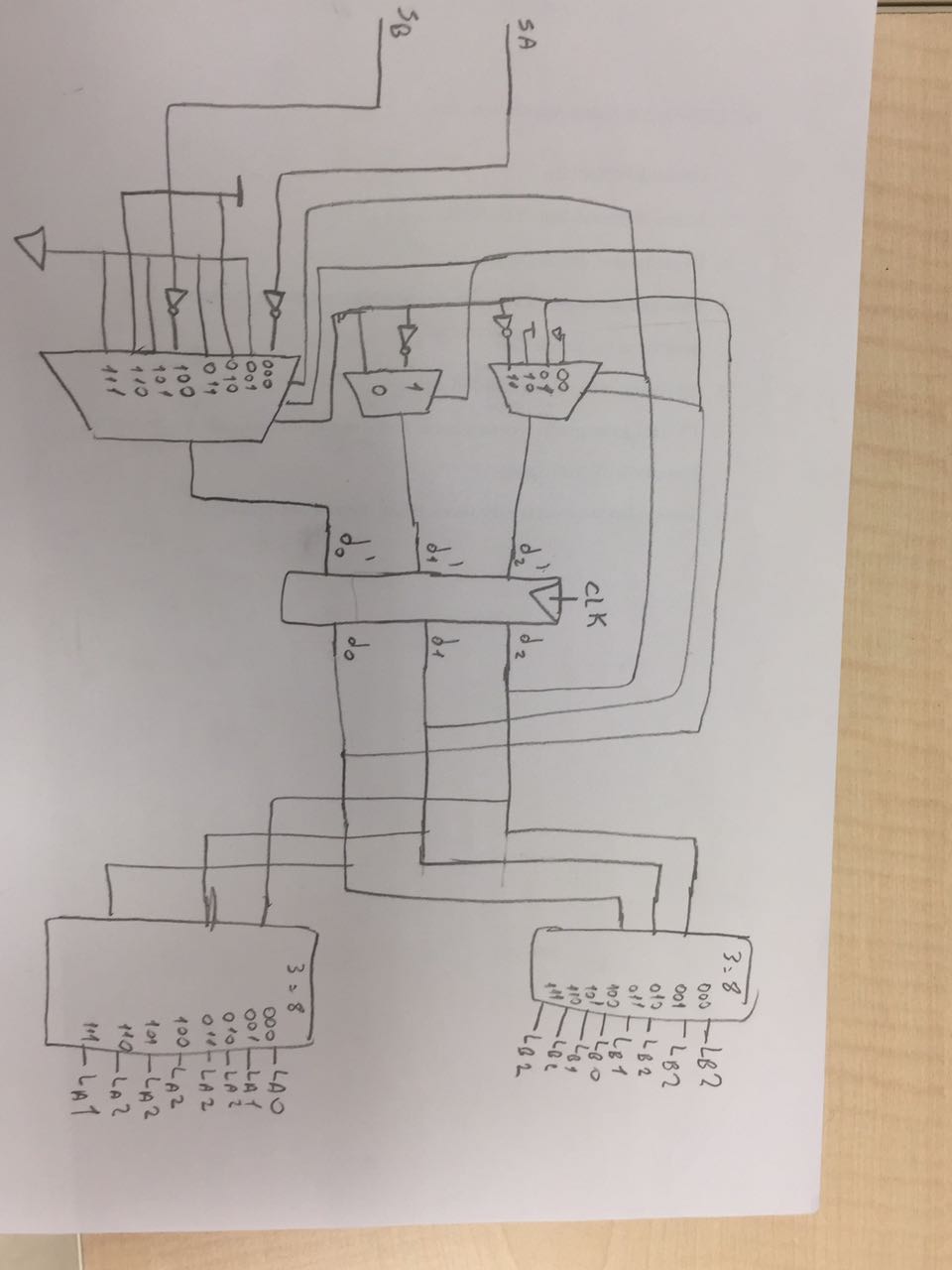
**SCHEMATIC:**

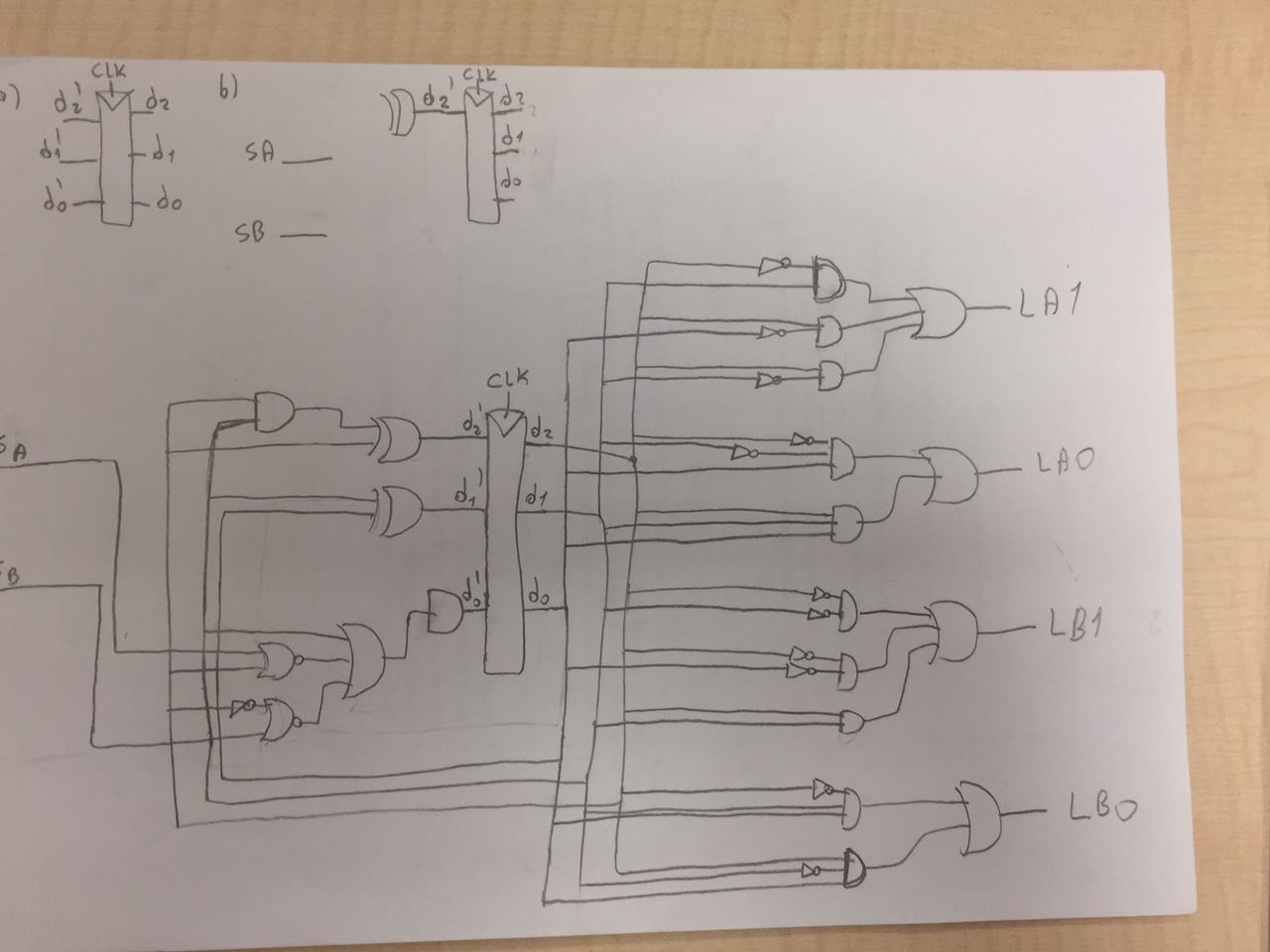


B-) We need 3 flip-flops since we have 3-bit states.

C-)







Next State:

`timescale 1ns / 1ps

module NextStateMux( input logic [2:0] state,

input logic sa,sb,

output logic [2:0] next );

logic [1:0] logic1; // For S2'

assign logic1 = {state[1],state[0]};

always\_comb

case(logic1)

2'b00: next[2] = state[2];

2'b01: next[2] = state[2];

2'b10: next[2] = state[2];

2'b11: next[2] = ~state[2];

default: next[2] = 1'b0;

endcase

assign next[1] = state[1] ? state[0] : ~state[0]; // For S1'

always\_comb // For S0'

case(state)

3'b000: next[0] = ~sa;

3'b001: next[0] = 0;

3'b010: next[0] = 1;

3'b011: next[0] = 0;

3'b100: next[0] = ~sb;

3'b101: next[0] = 0;

3'b110: next[0] = 1;

3'b111: next[0] = 0;

default: next[0] = 1'b0;

endcase

endmodule

Output State:

`timescale 1ns / 1ps

module OutputDecoder( input logic [2:0] currentState,

output logic [5:0] fsmOutput );

always\_comb

case(currentState)

3'b000: fsmOutput[2:0] = 3'b001;

3'b001: fsmOutput[2:0] = 3'b010;

3'b010: fsmOutput[2:0] = 3'b100;

3'b011: fsmOutput[2:0] = 3'b100;

3'b100: fsmOutput[2:0] = 3'b100;

3'b101: fsmOutput[2:0] = 3'b100;

3'b110: fsmOutput[2:0] = 3'b100;

3'b111: fsmOutput[2:0] = 3'b010;

default: fsmOutput[2:0] = 3'b000;

endcase

always\_comb

case(currentState)

3'b000: fsmOutput[5:3] = 3'b100;

3'b001: fsmOutput[5:3] = 3'b100;

3'b010: fsmOutput[5:3] = 3'b100;

3'b011: fsmOutput[5:3] = 3'b010;

3'b100: fsmOutput[5:3] = 3'b001;

3'b101: fsmOutput[5:3] = 3'b010;

3'b110: fsmOutput[5:3] = 3'b100;

3'b111: fsmOutput[5:3] = 3'b100;

default: fsmOutput[5:3] = 3'b000;

endcase

endmodule

FSM:

`timescale 1ns / 1ps

module FSM( input logic clk,reset,sa,sb,

output logic [5:0] led );

typedef enum logic [2:0] {S0,S1,S2,S3,S4,S5,S6,S7} statetype;

statetype [2:0] currentState, nextState;

// Counter

logic [24:0] counter;

always@ ( posedge clk )

counter <= counter + 1;

// Next State Logic

logic [2:0] muxOutput;

NextStateMux nextStateLogic( currentState, sa, sb, muxOutput );

// Output Logic

logic [5:0] fsmOutput;

OutputDecoder outputLogic( currentState, fsmOutput );

//State register

always\_ff@( posedge clk, posedge reset )

if (reset)

currentState <=3'b000;

else if( counter[24] )

currentState<=nextState;

else

currentState<=currentState ;

assign nextState = muxOutput;

assign currentState = nextState;

assign led = fsmOutput;

endmodule

Simulation:

`timescale 1ns / 1ps

module sim( input logic clk,reset,sa,sb,

output logic [5:0] led );

FSM fsm(clk,reset,sa,sb,led);

initial

begin

reset = 1; #10;

reset = 0;

end

always\_ff

begin

clk = 1; #2;

clk = 0; #2;

end

always @(posedge clk)

begin

#20;

sa = 0; sb = 0; #200;

sb = 1; #200;

sa = 1; sb = 0; #200;

sb = 1; #200;

end

endmodule

Constraint:

# Clock signal

set\_property PACKAGE\_PIN W5 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]

# Switches

set\_property PACKAGE\_PIN V17 [get\_ports {reset}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {reset}]

set\_property PACKAGE\_PIN V16 [get\_ports {sa}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sa}]

set\_property PACKAGE\_PIN W16 [get\_ports {sb}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {sb}]

# LEDs

set\_property PACKAGE\_PIN U16 [get\_ports {led[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[0]}]

set\_property PACKAGE\_PIN E19 [get\_ports {led[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[1]}]

set\_property PACKAGE\_PIN U19 [get\_ports {led[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[2]}]

set\_property PACKAGE\_PIN V19 [get\_ports {led[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[3]}]

set\_property PACKAGE\_PIN W18 [get\_ports {led[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[4]}]

set\_property PACKAGE\_PIN U15 [get\_ports {led[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {led[5]}]