İSTANBUL TEKNİK ÜNİVERSİTESİ

ELEKTRİK ELEKTRONİK FAKÜLTESİ

DIGITAL SYSTEM DESIGN APPLICATIONS

(EHB 436E)

Fsm Experiments Report

Hasan Emre AYDEMİR

1. CIRCUIT THAT DETECTS FOUR CONSECUTIVE 1 OR 0

1.1 Finite State Machine Encoding Methods

When we design a finite state machine, every symbolic state (A, B, C…) must be represented by a pattern of bits. This is called state encoding. The choice of encoding affects the speed, area, power, and complexity of the FSM.

Finite state machines can be implemented using several state-encoding methods, including Binary encoding, One-Hot encoding, Gray encoding, One-Cold encoding, Johnson (Ring Counter) encoding, Output-Encoded (Hybrid) encoding, and Tool-Optimized (Automatic) encoding. The three most commonly used ones are Binary, Gray, and One-Hot encoding.

a. Binary Encoding

Binary encoding assigns each state a unique binary number and uses the minimum possible number of flip-flops, requiring only $\log_2(N)$ bits for N states—for example, 8 states can be represented with just 3 bits. This method is highly area-efficient because it uses very few registers, which also helps reduce power consumption since fewer flip-flops toggle during transitions. It is widely supported by synthesis tools and works well for small or straightforward FSMs. However, binary encoding needs more combinational logic to decode state values, which increases the depth of the logic and slows down the critical path. For this reason, it is often not the best choice for high-speed designs where timing is tight.

b. Gray Code Encoding

Gray encoding arranges the states so that consecutive states differ by only one bit, such as the sequence 00 → 01 → 11 → 10, ensuring a Hamming distance of one. Because only a single bit changes at each step, switching activity is minimized, which reduces dynamic power consumption and helps prevent glitches in the next-state logic. This makes Gray code ideal for counters, rotary encoders, and FSMs that follow a strictly ordered or ring-like sequence. However, its benefits weaken in complex machines with many non-sequential transitions, where the one-bit change rule cannot be maintained. Additionally, synthesis tools may find Gray encoding harder to optimize, and it typically does not reach the same speed performance as one-hot encoding.

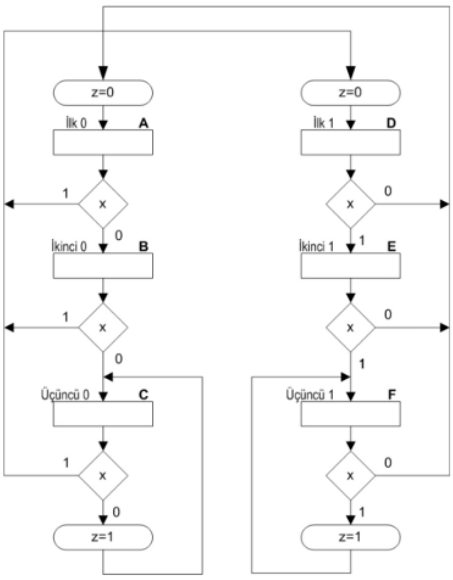
c. One-Hot Encoding

One-Hot encoding uses exactly one flip-flop per state, with only one bit set to ‘1’ while all others remain ‘0’, such as A=0001, B=0010, C=0100, and D=1000 for a four-state machine. This method is extremely fast because almost no decoding logic is required—the active state is directly represented by the flip-flop output. It is especially suitable for FPGA implementations, where LUT and register resources are plentiful, and it simplifies both design and debugging. The next-state logic is shallow and efficient, often providing the highest achievable clock frequency. However, One-Hot encoding consumes significantly more flip-flops since N states require N registers, increasing static power and making it less practical for ASICs or power-sensitive systems with many states.

1.2 State Encoding For The State Diagram

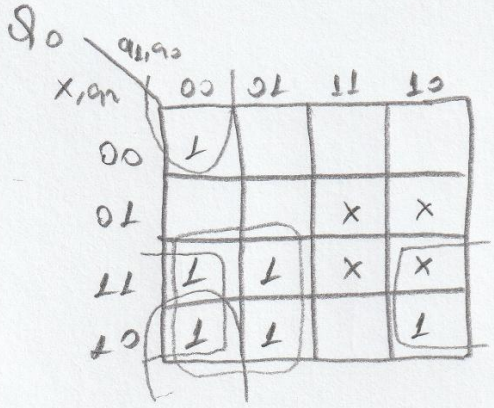
state	binary code
A	000
B	001
C	010
D	011
E	100
F	101

Figure 3: State Encoding Example

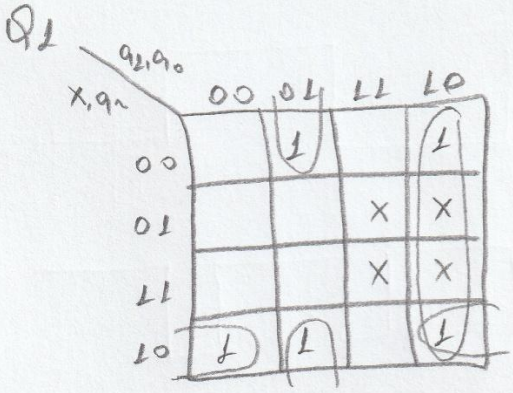


1.3 Reduction Results of Output Functions

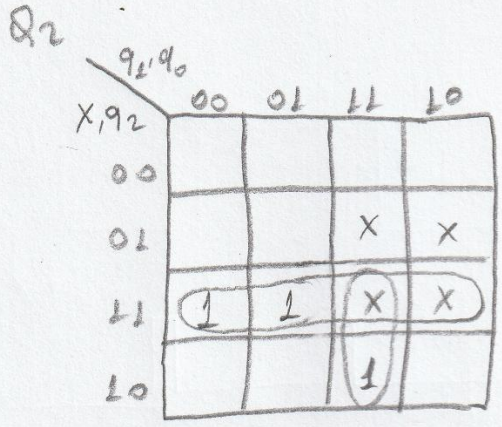
x	q2	q1	q0	Q2	Q1	Q0	z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	k	k	k	k
0	1	1	1	k	k	k	k
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	k	k	k	k
1	1	1	1	k	k	k	k



$$Q_0 = \bar{q_2}\,\bar{q_1}\,\bar{q_0} + x\,\bar{q_1} + x\,\bar{q_0}$$

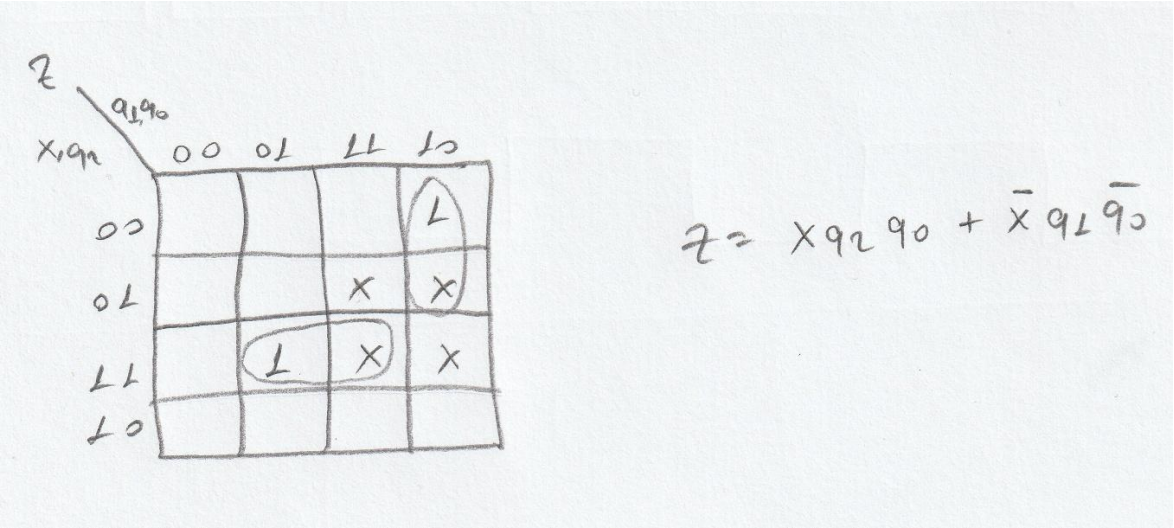


$$Q_1 = x\,\bar{q_2}\,\bar{q_0} + \bar{q_2}\,\bar{q_1}\,q_0 + q_1\,\bar{q_0}$$



$$Q_2 = x\,q_2 + x\,q_1\,q_0$$

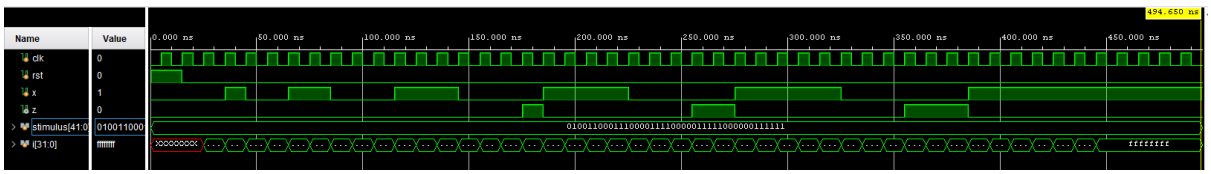
$$Q_2 = x\,(q_2 + q_1\,q_0)$$



1.4 Verilog Code, Testbench Code and Simulation Results

```
module fsm1_mealy_4same(
    input clk,
    input rst,
    input x,
    output z
);
   reg q2, q1, q0;

   wire Q2, Q1, Q0;

   assign Q2 = (q2 & x) | (q0 & q1 & x);

   assign Q1 = (q1 & ~q0) | (q0 & ~q1 & ~q2) | (x & ~q1 & ~q2);

   assign Q0 = (x & (~q0 | ~q1)) | (~q0 & ~q1 & ~q2);

   assign z = (q0 & q2 & x) | (q1 & ~q0 & ~x);

   always @(posedge clk or posedge rst) begin
    if (rst) begin
       q2 <= 0;
       q1 <= 0;
       q0 <= 0;

        end

       else begin

    q2 <= Q2;
    q1 <= Q1;
    q0 <= Q0;
    end
   end
endmodule
```

```
`timescale 1ns / 1ps

module fsm1_mealy_4same_tb;

    reg clk;
    reg rst;
    reg x;
    wire z;

FSM1 DUT(
     .clk(clk),
     .rst(rst),
     .x(x),
     .z(z)
);

    initial begin
     clk = 0;
     forever #5 clk = ~clk;
    end

    initial begin
     rst = 1;
     #20;
     rst = 0;
    end

    reg[41:0] seq = 42'b010011000111000011110000011111000000111111;
     integer i;

    initial begin
     x = 0;
     #25;
     for (i = 41; i >= 0; i = i - 1) begin
        x = seq[i];
        #10;     // one clock cycle
     end
     #50;

     $stop;

    end

endmodule
```



1.5 Faulty Outputs and Causes

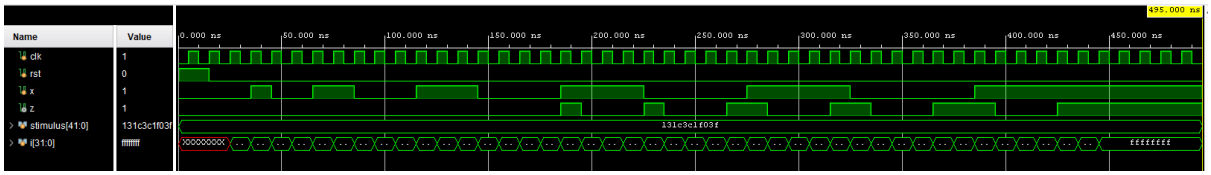
In the Mealy implementation, z is combinational, therefore short faulty 0/1 pulses may occur due to path delays. Timing simulation may show these glitches. In the Moore implementation, adding a DFF removes all glitches. In the Mealy implementation the output z is a purely combinational function of the current state and the input x. Due to unequal propagation delays in the FPGA, short glitches (faulty 0/1 pulses) may appear on z when x or the state changes. In the Moore implementation, z is registered with a D flip-flop and only updated on the rising edge of the clock, so all combinational glitches are filtered out and no faulty outputs occur.

1.6 How to Add DFF to z Output

```
reg z_reg;
   assign z = z_reg;

   always @(posedge clk or posedge rst) begin
       if (rst) begin
           state  <= S0;
           z_reg  <= 1'b0;
       end else begin
           state  <= state_next;
           z_reg  <= z_mealy;   // Mealy → DFF → Moore
       end
   end
```

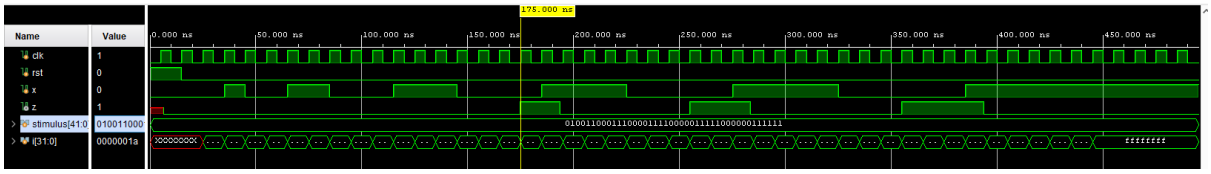
A D flip-flop was inserted at the output. This makes the output synchronous and removes faulty transitions.

1.7 Simulation Results After Changing Machine Type

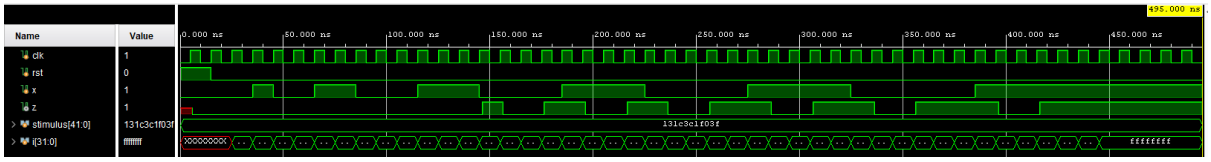


z output is delayed by one clock cycle compared to the Mealy design

1.8 Post-Implementation Timing Simulation



Mealy Post-Implementation Timing Simulation



Moore Post-Implementation Timing Simulation

In the Post-Implementation Timing Simulation, no distinct 0/1 glitch was observed at the Mealy FSM output. However, because the Mealy output is purely combinational, small transient shifts appeared on the z signal due to unequal propagation delays along different logic paths. In contrast, the Moore design registers the z output with a D flip-flop that updates the output only on the rising edge of the clock, completely eliminating all combinational glitching.

1.9 Utilization and Timing Summaries

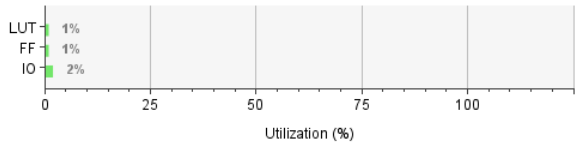
Primitives

Ref Name	Used	Functional Category
LUT4	3	LUT
IBUF	3	IO
FDCE	3	Flop & Latch
OBUF	1	IO
LUT3	1	LUT
BUFG	1	Clock

FSM1 Mealy Primitives

Summary

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
FF	3	65200	0.00
IO	4	210	1.90



FSM1 Mealy Utilization

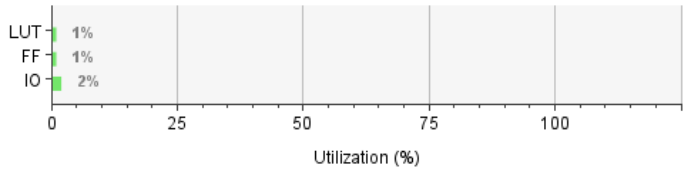
Primitives

Ref Name	Used	Functional Category
FDCE	4	Flop & Latch
LUT4	3	LUT
IBUF	3	IO
OBUF	1	IO
LUT3	1	LUT
BUFG	1	Clock

FSM1 Moore Primitives

Summary

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
FF	4	65200	0.01
IO	4	210	1.90



FSM1 Moore Utilization

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	8.487	1	1	4	q0_reg/C	q1_reg/D	1.524	0.606	0.918	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	8.745	1	1	4	q1_reg/C	q0_reg/D	1.249	0.580	0.669	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	8.763	1	1	4	q1_reg/C	q2_reg/D	1.277	0.608	0.669	10.0	sys_clk_pin	sys_clk_pin		0.035

Intra-Clock Paths - sys_clk_pin - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 4	0.262	1	1	4	q0_reg/C	q2_reg/D	0.369	0.183	0.186	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 5	0.281	1	1	4	q0_reg/C	q0_reg/D	0.372	0.186	0.186	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 6	0.300	1	1	4	q2_reg/C	q1_reg/D	0.404	0.231	0.173	0.0	sys_clk_pin	sys_clk_pin		0.000

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 7	∞	3	2	4	x	z	8.600	5.139	3.461	∞	input port clock			0.000

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 8	∞	3	2	4	x	z	2.536	1.529	1.007	-∞	input port clock			0.000

Unconstrained Paths - NONE - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 9	∞	1	1	3	rst	q0_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 10	∞	1	1	3	rst	q1_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 11	∞	1	1	3	rst	q2_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 12	∞	2	1	4	x	q2_reg/D	3.037	1.643	1.394	∞	input port clock	sys_clk_pin		0.025
Path 13	∞	2	1	4	x	q1_reg/D	3.030	1.643	1.387	∞	input port clock	sys_clk_pin		0.025
Path 14	∞	2	1	4	x	q0_reg/D	3.009	1.615	1.394	∞	input port clock	sys_clk_pin		0.025

Unconstrained Paths - NONE - sys_clk_pin - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 15	∞	2	1	4	x	q1_reg/D	0.814	0.304	0.510	-∞	input port clock	sys_clk_pin		0.000
Path 16	∞	2	1	4	x	q2_reg/D	0.823	0.303	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 17	∞	2	1	4	x	q0_reg/D	0.824	0.304	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 18	∞	1	1	3	rst	q0_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 19	∞	1	1	3	rst	q1_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 20	∞	1	1	3	rst	q2_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000

Unconstrained Paths - sys_clk_pin - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 21	∞	2	2	4	q0_reg/C	z	7.096	4.104	2.992	∞	sys_clk_pin			0.025

Unconstrained Paths - sys_clk_pin - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 22	∞	2	2	4	q2_reg/C	z	2.122	1.452	0.670	-∞	sys_clk_pin			0.025

Output Ports Clock-to-out

Reference Clock	Output Port	IO Reg Type	Delay Type	Max Clk to Port	Max Edge	Max Process Corner	Min Clk to Port	Min Edge	Min Process Corner	Internal Clock	Output Enable/Disable
clk100	z	FDCE		12.051	Rise	SLOW	3.562	Rise	FAST		

FSM1 Mealy Max-Min Port

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	8,226 ns	Worst Hold Slack (WHS):	0,290 ns	Worst Pulse Width Slack (WPWS):	4,500 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	3	Total Number of Endpoints:	3	Total Number of Endpoints:	4

All user specified timing constraints are met.

FSM1 Mealy Timing Summaries

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	8.482	1	1	3	state_reg[0]/C	state_reg[0]/D	1.511	0.580	0.931	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	8.502	1	1	3	state_reg[0]/C	state_reg[1]/D	1.537	0.606	0.931	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	8.514	1	1	3	state_reg[0]/C	z_reg_reg/D	1.526	0.606	0.920	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	8.744	1	1	4	state_reg[2]/C	state_reg[2]/D	1.252	0.580	0.672	10.0	sys_clk_pin	sys_clk_pin		0.035

Intra-Clock Paths - sys_clk_pin - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	0.295	1	1	4	state_reg[1]/C	state_reg[1]/D	0.402	0.230	0.172	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 6	0.297	1	1	4	state_reg[1]/C	z_reg_reg/D	0.404	0.231	0.173	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 7	0.308	1	1	4	state_reg[1]/C	state_reg[2]/D	0.400	0.227	0.173	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 8	0.308	1	1	4	state_reg[1]/C	state_reg[0]/D	0.399	0.227	0.172	0.0	sys_clk_pin	sys_clk_pin		0.000

Unconstrained Paths - NONE - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 9	∞	1	1	4	rst	state_reg[0]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 10	∞	1	1	4	rst	state_reg[1]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 11	∞	1	1	4	rst	state_reg[2]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 12	∞	1	1	4	rst	z_reg_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 13	∞	2	1	4	x	state_reg[1]/D	3.037	1.643	1.394	∞	input port clock	sys_clk_pin		0.025
Path 14	∞	2	1	4	x	z_reg_reg/D	3.030	1.643	1.387	∞	input port clock	sys_clk_pin		0.025
Path 15	∞	2	1	4	x	state_reg[0]/D	3.009	1.615	1.394	∞	input port clock	sys_clk_pin		0.025
Path 16	∞	2	1	4	x	state_reg[2]/D	3.002	1.615	1.387	∞	input port clock	sys_clk_pin		0.025

Unconstrained Paths - NONE - sys_clk_pin - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 17	∞	2	1	4	x	state_reg[2]/D	0.814	0.304	0.510	-∞	input port clock	sys_clk_pin		0.000
Path 18	∞	2	1	4	x	z_reg_reg/D	0.814	0.304	0.510	-∞	input port clock	sys_clk_pin		0.000
Path 19	∞	2	1	4	x	state_reg[1]/D	0.823	0.303	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 20	∞	2	1	4	x	state_reg[0]/D	0.824	0.304	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 21	∞	1	1	4	rst	state_reg[0]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 22	∞	1	1	4	rst	state_reg[1]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 23	∞	1	1	4	rst	state_reg[2]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 24	∞	1	1	4	rst	z_reg_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000

Unconstrained Paths - sys_clk_pin - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 25	∞	1	1	1	z_reg_reg/C	z	6.188	4.115	2.073	∞	sys_clk_pin			0.025

Unconstrained Paths - sys_clk_pin - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 26	∞	1	1	1	z_reg_reg/C	z	1.913	1.406	0.507	-∞	sys_clk_pin			0.025

Reference Clock	Output Port	IO Reg Type	Delay Type	Max Clk to Port	Max Edge	Max Process Corner	Min Clk to Port	Min Edge	Min Process Corner	Internal Clock	Output Enable/Disable
sys_clk_pin	z	FDCE		11.616	Rise	SLOW	3.663	Rise	FAST		

FSM1 Moore Max-Min Port

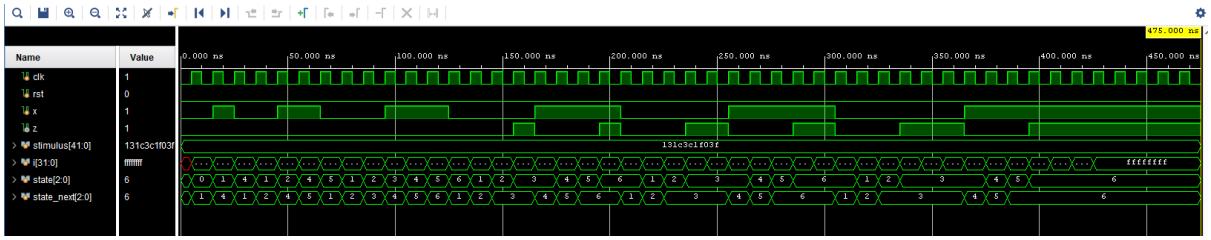
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	8,226 ns	Worst Hold Slack (WHS):	0,310 ns	Worst Pulse Width Slack (WPWS):	4,500 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	4	Total Number of Endpoints:	4	Total Number of Endpoints:	5

All user specified timing constraints are met.

FSM1 Moore Timing Summaries

1.10 Stucking in Arbitrary States or Not

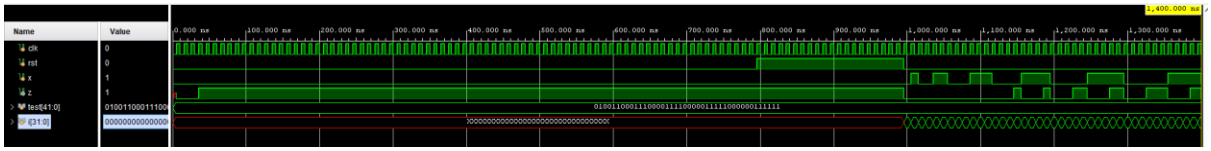


The lab manual asks to test the FSM by initializing it in arbitrary states such as ‘110’ and ‘111’. To test undesirable states, the state register was initialized to 3'b110 and 3'b111 and the same input sequence was applied. The simulation shows that on the first clock edge the FSM jumps from 111 to the reset state S0 (due to the default: state_next = S0; clause) and then continues in the normal set of states. Therefore, the circuit does not remain stuck in an arbitrary state; it is self-correcting and can recover from an illegal state in at most one clock cycle.

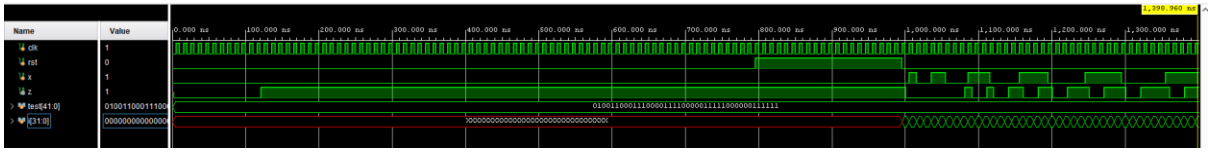
1.11 Comparison of the Behavioral Model with Your Design

The FSM1_Mealy_4same module is a structural design that implements all combinational logic using assign statements and updates state values only inside a clocked DFF, fully matching the requirements of a Mealy-style specification. Since the output z is purely combinational, the circuit can react within the same clock cycle, immediately producing z = 1 when four consecutive 0s or 1s are detected. In contrast, the fsm1_behav module combines state transitions and output assignments inside a single sequential always block, making it a truly behavioral implementation. Because the output is assigned inside a clocked block, z changes only on the rising edge of the clock, effectively behaving like a Moore-type output. Additionally, both versions produce z = 1 when four consecutive 0s or four consecutive 1s are observed. Therefore, the two designs differ in timing behavior, even though they detect the same pattern. In the behavioral FSM, the next-state logic is written using case and if structures, which causes Vivado to generate separate combinational logic blocks for each state. Since each block repeats the state decoding process independently, the total LUT usage increases.

1.12 Utilization and Timing Summaries of Behavioral Model



Behavioral Simulation



Post-Implementation Timing Simulation

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	8.357	1	4	FSM_sequential_state_reg[2]/C	z_reg/D	1.615	0.718	0.897	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	8.471	1	4	FSM_sequential_state_reg[2]/C	FSM_sequential_state_reg[0]/D	1.523	0.718	0.805	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	8.491	1	4	FSM_sequential_state_reg[2]/C	FSM_sequential_state_reg[2]/D	1.549	0.744	0.805	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	8.738	1	4	FSM_sequential_state_reg[1]/C	FSM_sequential_state_reg[1]/D	1.257	0.580	0.677	10.0	sys_clk_pin	sys_clk_pin		0.035

Intra-Clock Paths - sys_clk_pin - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	0.201	1	4	FSM_sequential_state_reg[0]/C	z_reg/D	0.305	0.186	0.119	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 6	0.218	1	4	FSM_sequential_state_reg[2]/C	FSM_sequential_state_reg[1]/D	0.310	0.227	0.083	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 7	0.254	1	4	FSM_sequential_state_reg[0]/C	FSM_sequential_state_reg[2]/D	0.361	0.183	0.178	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 8	0.273	1	4	FSM_sequential_state_reg[0]/C	FSM_sequential_state_reg[0]/D	0.364	0.186	0.178	0.0	sys_clk_pin	sys_clk_pin		0.000

Unconstrained Paths - NONE - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 9	∞	2	4	rst	z_reg/CE	4.351	1.588	2.763	∞	input port clock	sys_clk_pin		0.025
Path 10	∞	1	4	rst	FSM_sequential_state_reg[0]/CLR	3.353	1.464	1.888	∞	input port clock	sys_clk_pin		0.025
Path 11	∞	1	4	rst	FSM_sequential_state_reg[1]/CLR	3.353	1.464	1.888	∞	input port clock	sys_clk_pin		0.025
Path 12	∞	1	4	rst	FSM_sequential_state_reg[2]/CLR	3.353	1.464	1.888	∞	input port clock	sys_clk_pin		0.025
Path 13	∞	2	4	x	z_reg/D	2.710	1.615	1.095	∞	input port clock	sys_clk_pin		0.025
Path 14	∞	2	4	x	FSM_sequential_state_reg[1]/D	2.469	1.615	0.854	∞	input port clock	sys_clk_pin		0.025
Path 15	∞	2	4	x	FSM_sequential_state_reg[0]/D	2.465	1.615	0.850	∞	input port clock	sys_clk_pin		0.025
Path 16	∞	2	4	x	FSM_sequential_state_reg[2]/D	2.459	1.609	0.850	∞	input port clock	sys_clk_pin		0.025

Unconstrained Paths - NONE - sys_clk_pin - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 17	∞	2	4	x	FSM_sequential_state_reg[0]/D	0.649	0.304	0.345	-∞	input port clock	sys_clk_pin		0.000
Path 18	∞	2	4	x	FSM_sequential_state_reg[1]/D	0.650	0.304	0.346	-∞	input port clock	sys_clk_pin		0.000
Path 19	∞	2	4	x	FSM_sequential_state_reg[2]/D	0.652	0.307	0.345	-∞	input port clock	sys_clk_pin		0.000
Path 20	∞	2	4	x	z_reg/D	0.736	0.304	0.432	-∞	input port clock	sys_clk_pin		0.000
Path 21	∞	1	4	rst	FSM_sequential_state_reg[0]/CLR	0.991	0.232	0.758	-∞	input port clock	sys_clk_pin		0.000
Path 22	∞	1	4	rst	FSM_sequential_state_reg[1]/CLR	0.991	0.232	0.758	-∞	input port clock	sys_clk_pin		0.000
Path 23	∞	1	4	rst	FSM_sequential_state_reg[2]/CLR	0.991	0.232	0.758	-∞	input port clock	sys_clk_pin		0.000
Path 24	∞	2	4	rst	z_reg/CE	1.333	0.277	1.056	-∞	input port clock	sys_clk_pin		0.000

Unconstrained Paths - sys_clk_pin - NONE - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 25	∞	1	1	z_reg/C	z	6.044	3.980	2.064	∞	sys_clk_pin			0.025

Unconstrained Paths - sys_clk_pin - NONE - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 26	∞	1	1	z_reg/C	z	1.855	1.366	0.489	-∞	sys_clk_pin			0.025

Output Ports Clock-to-out

Reference Clock	Output Port	IO Reg Type	Delay Type	Max Clk to Port	Max Edge	Max Process Corner	Min Clk to Port	Min Edge	Min Process Corner	Internal Clock	Output Enable/Disable
clk100	z	FDRE		11.245	Rise	SLOW	3.335	Rise	FAST		

FSM1 Behavioral Max-Min Port

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	8,357 ns	Worst Hold Slack (WHS):	0,201 ns	Worst Pulse Width Slack (WPWS):	4,500 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	4	Total Number of Endpoints:	4	Total Number of Endpoints:	5

All user specified timing constraints are met.

FSM1 Behavioral Timing Summaries

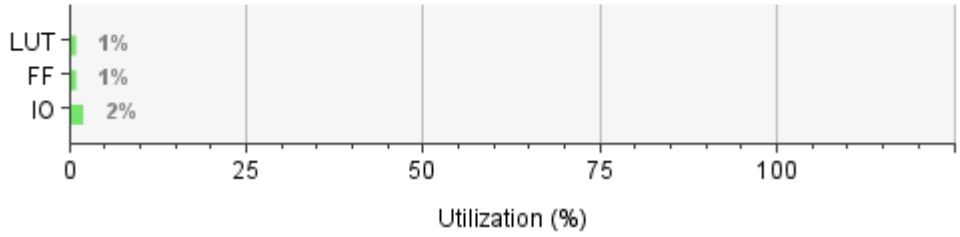
Primitives

Ref Name	Used	Functional Category
LUT4	4	LUT
IBUF	3	IO
FDCE	3	Flop & Latch
OBUF	1	IO
LUT1	1	LUT
FDRE	1	Flop & Latch
BUFG	1	Clock

FSM1 Behavioral Moore Utilization

Summary

Resource	Utilization	Available	Utilization %
LUT	4	32600	0.01
FF	4	65200	0.01
IO	4	210	1.90



FSM1 Behavioral Utilization

1.13 Comparison in terms of resource usage and path delays

The structural model (fsm1_mealy_4same) is more compact in terms of resource usage, requiring only 4 LUTs and 3 flip-flops. However, because its output *z* is not registered, the clock-to-out delay is slightly higher compared to the behavioral model. On the other hand, the behavioral model (fsm1_behav) uses one additional LUT and one extra flip-flop, but since the output is registered, it achieves a shorter clock-to-out delay and therefore provides a small timing advantage. In summary, the structural model is more resource-efficient and reflects true Mealy behavior, while the behavioral model offers a more stable registered output and slightly better timing performance.

The **Mealy FSM** provides the fastest response because its output can change immediately based on both the current state and the input. In this project, the structural Mealy design uses the least amount of hardware resources (minimal LUTs and flip-flops) and is highly optimized by the synthesis tool. Its advantages include low latency, reduced hardware cost, and high performance. The main drawbacks are the possibility of output glitches due to its combinational nature and a slightly larger clock-to-out delay compared to a Moore implementation. This model is preferred in FPGA designs where resource efficiency and fast reaction time are important, or when strict Mealy behavior is required in academic assignments.

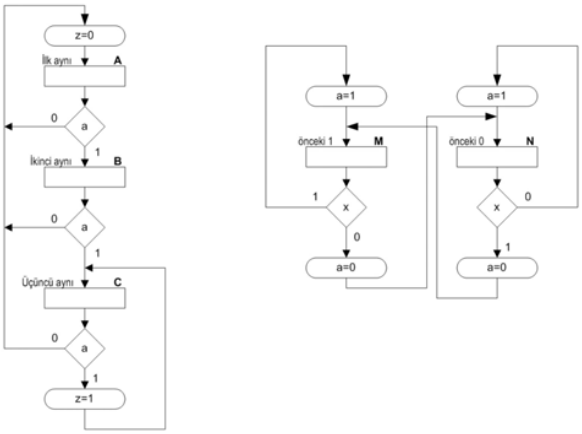
The **Moore FSM** produces its output entirely through a register, resulting in a more stable and glitch-free signal. Because the output changes only on the rising edge of the clock, timing analysis becomes more predictable and robust. Its advantages include reliable output behavior, better timing margins, and lower clock-to-out delay. The main disadvantages are the one-cycle delay in the output response and the use of an extra flip-flop compared to the Mealy version. The Moore model is preferred in designs where output stability is crucial, glitch-free signals are required, or tight timing constraints must be met.

The **Behavioral FSM** is the easiest to write because both state transitions and output logic are described within a single always block. It is advantageous for learning, simulation, and rapid prototyping. However, this style offers less control over synthesis optimization and may use slightly more resources than a structural Mealy or Moore design. In addition, the distinction between Mealy and Moore behavior is less explicit, making its hardware implementation somewhat less predictable. The behavioral model is mainly preferred during early design stages, educational environments, or when a quick functional FSM is required, but it is generally not used in timing-critical or resource-optimized FPGA designs.

2. DESIGN WITH DIVIDED STATE DIAGRAMS

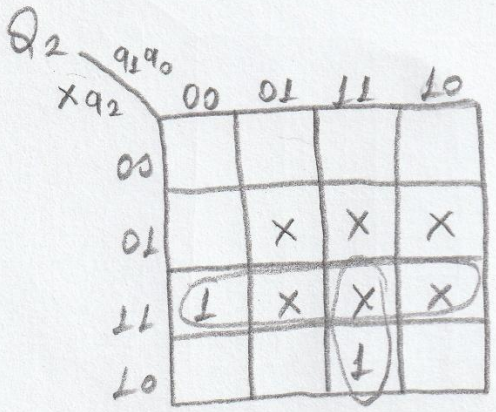
2.1 State Encoding For The State Diagram

state	binary code
A	000
B	001
C	010
M	011
N	100



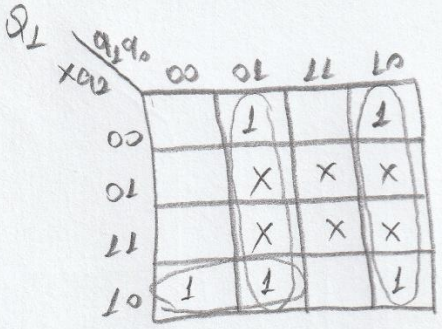
2.2 Reduction Results of Output Functions

x	q2	q1	q0	Q2	Q1	Q0	z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	k	k	k	k
0	1	1	0	k	k	k	k
0	1	1	1	k	k	k	k
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	0
1	1	0	1	k	k	k	k
1	1	1	0	k	k	k	k
1	1	1	1	k	k	k	k

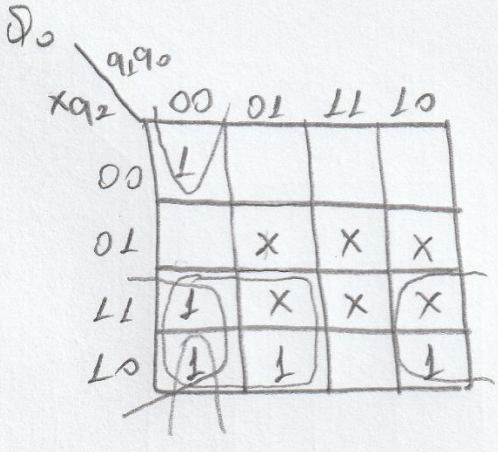


$$Q_2 = Xq_2 + Xq_1q_0$$

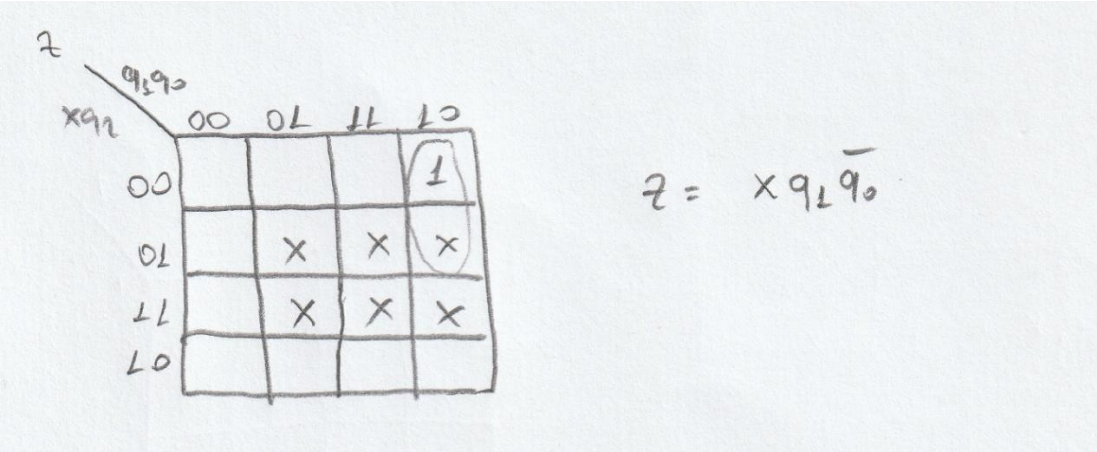
$$Q_2 = X(q_2 + q_1q_0)$$



$$Q_1 = \bar{q}_1q_0 + q_1\bar{q}_0 + X\bar{q}_2\bar{q}_1$$



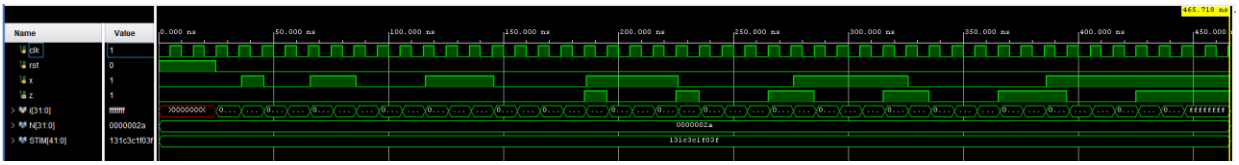
$$Q_0 = \bar{q}_2\bar{q}_1q_0 + X\bar{q}_1 + X\bar{q}_0$$



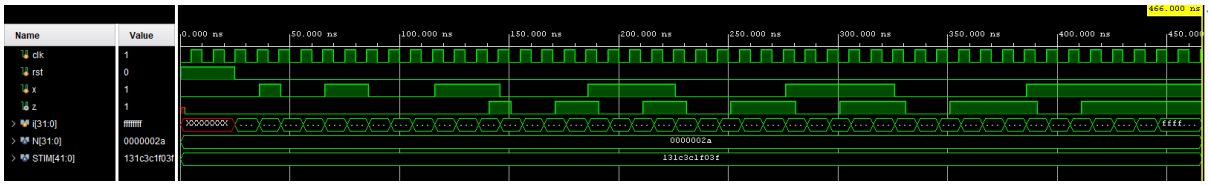
2.3 Verilog Code, Testbench Code and Simulation Results

```
`timescale 1ns / 1ps

module FSM2 (
    input  wire clk,
    input  wire rst,
    input  wire x,
    output wire z
);
    reg [2:0] Q;
    reg       z_reg;

    assign z = z_reg;

    wire [2:0] D;
    wire       a;
    wire       z_next;
    wire Q2 = Q[2];
    wire Q1 = Q[1];
    wire Q0 = Q[0];
    wire D2;
    wire D1;
    wire D0;

    assign D = {D2, D1, D0};
    assign a = ~(Q2 ^ x);

    assign D2 = x;
    assign D1 = a & (Q0 ^ Q1);
    assign D0 = a & (~Q1 & ~Q0);

    assign z_next = a & Q1 & ~Q0;

    always @(posedge clk or posedge rst) begin
        if (rst) begin
            Q     <= 3'b000;
            z_reg <= 1'b0;
        end else begin
            Q     <= D;
            z_reg <= z_next;
        end
    end
endmodule
```

```
 `timescale 1ns / 1ps

module FSM2_tb;

    reg  clk;
    reg  rst;
    reg  x;
    wire z;

    FSM2 dut (
        .clk (clk),
        .rst (rst),
        .x   (x),
        .z   (z)
    );
    initial clk = 0;
    always #5 clk = ~clk;

    localparam integer N = 42;
    localparam [N-1:0] STIM =
        42'b010011000111000011110000011111000000111111;
    integer i;

    initial begin

        $dumpfile("FSM2_tb.vcd");
        $dumpvars(0, FSM2_tb);

        rst = 1;
        x   = 0;
        repeat (3) @(posedge clk);
        rst = 0;
        $display("===== Reset bitti =====");
        $display("Giris dizisi:
010011000111000011110000011111000000111111");

        for (i = N-1; i >= 0; i = i - 1) begin
            x = STIM[i];
            @(posedge clk);
            #1;
            $display("t=%0t  idx=%0d  x=%b  z=%b",
                     $time, (N-1-i), x, z);
        end

        $display("=== Simulasyon bitti ===");
        #20;
        $finish;
    end

endmodule
```



2.4 Post-Implementation Timing Simulation



2.5 Utilization and Timing Summaries

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	8.582	1	3	Q_reg[2]/C	Q_reg[0]/D	1.412	0.715	0.697	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	8.591	1	3	Q_reg[2]/C	z_reg_reg/D	1.405	0.715	0.690	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	8.600	1	3	Q_reg[2]/C	Q_reg[1]/D	1.440	0.743	0.697	10.0	sys_clk_pin	sys_clk_pin		0.035

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 4	0.210	1	3	Q_reg[1]/C	z_reg_reg/D	0.302	0.227	0.075	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 5	0.260	1	3	Q_reg[0]/C	Q_reg[1]/D	0.367	0.183	0.184	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 6	0.279	1	3	Q_reg[0]/C	Q_reg[0]/D	0.370	0.186	0.184	0.0	sys_clk_pin	sys_clk_pin		0.000

Unconstrained Paths - NONE - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 7	∞	1	4	rst	Q_reg[0]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 8	∞	1	4	rst	Q_reg[1]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 9	∞	1	4	rst	Q_reg[2]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 10	∞	1	4	rst	z_reg_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 11	∞	2	4	x	Q_reg[1]/D	2.861	1.641	1.220	∞	input port clock	sys_clk_pin		0.025
Path 12	∞	2	4	x	Q_reg[0]/D	2.835	1.615	1.220	∞	input port clock	sys_clk_pin		0.025
Path 13	∞	2	4	x	z_reg_reg/D	2.824	1.615	1.209	∞	input port clock	sys_clk_pin		0.025
Path 14	∞	1	4	x	Q_reg[2]/D	2.457	1.491	0.966	∞	input port clock	sys_clk_pin		0.025

Unconstrained Paths - NONE - sys_clk_pin - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 15	∞	1	4	x	Q_reg[2]/D	0.636	0.259	0.377	-∞	input port clock	sys_clk_pin		0.000
Path 16	∞	2	4	x	z_reg_reg/D	0.761	0.304	0.458	-∞	input port clock	sys_clk_pin		0.000
Path 17	∞	2	4	x	Q_reg[0]/D	0.771	0.304	0.467	-∞	input port clock	sys_clk_pin		0.000
Path 18	∞	2	4	x	Q_reg[1]/D	0.772	0.305	0.467	-∞	input port clock	sys_clk_pin		0.000
Path 19	∞	1	4	rst	Q_reg[0]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 20	∞	1	4	rst	Q_reg[1]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 21	∞	1	4	rst	Q_reg[2]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 22	∞	1	4	rst	z_reg_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 23	∞	1	1	z_reg_reg/C	z	6.056	3.980	2.076	∞	sys_clk_pin			0.025

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 24	∞	1	1	z_reg_reg/C	z	1.865	1.366	0.499	-∞	sys_clk_pin			0.025

Output Ports Clock-to-out

Reference Clock	Output Port	IO Reg Type	Delay Type	Max Clk to Port	Max Edge	Max Process Corner	Min Clk to Port	Min Edge	Min Process Corner	Internal Clock	Output Enable/Disable
clk100	z	FDCE		11.258	Rise	SLOW	3.346	Rise	FAST		

FSM2 Max-Min Port

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	8,582 ns	Worst Hold Slack (WHS):	0,210 ns	Worst Pulse Width Slack (WPWS):	4,500 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	3	Total Number of Endpoints:	3	Total Number of Endpoints:	5

All user specified timing constraints are met.

FSM2 Timing Summaries

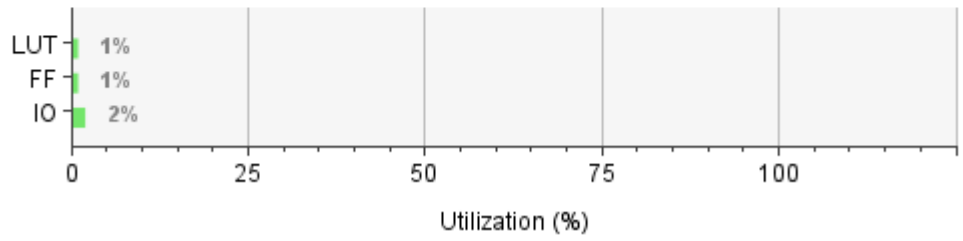
Primitives

Ref Name	Used	Functional Category
FDCE	4	Flop & Latch
LUT4	3	LUT
IBUF	3	IO
OBUF	1	IO
BUFG	1	Clock

FSM2 Primitives

Summary

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
FF	4	65200	0.01
IO	4	210	1.90



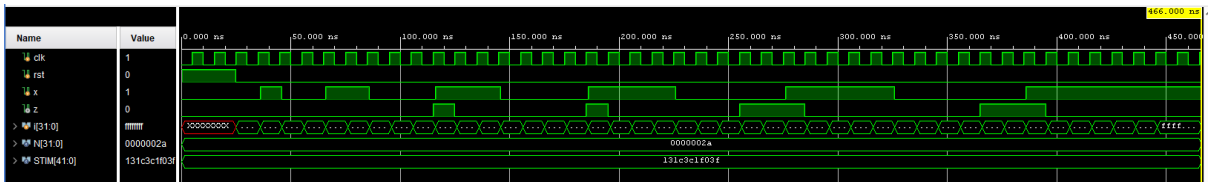
FSM2 Utilization

2.6 Verilog code, Testbench code and Simulation results of the Behavioral Model.

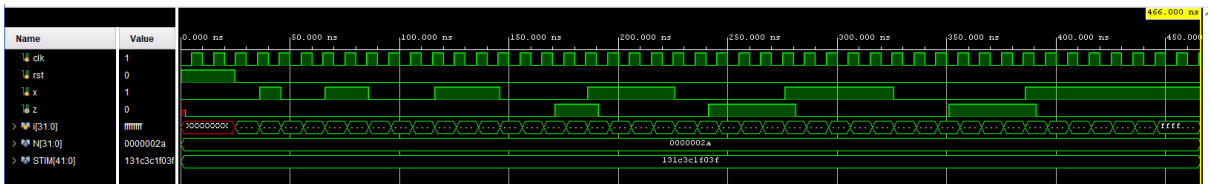
```
`timescale 1ns / 1ps

module fsm2_behav(
    input       clk,
    input       rst,
    input       x,
    output reg  z
);
    parameter A = 3'b000,
              B = 3'b001,
              C = 3'b010,
              M = 3'b011,
              N = 3'b100;
    reg [2:0] state;

    always @(posedge clk, posedge rst)
    begin
        if(rst)
        begin
            state <= A;
            z     <= 0;
        end
        else
        begin
            case(state)

                A : begin
                    if(x == 0)
                        state <= B;
                    else
                        state <= M;
                    z <= 0;
                end


                B : begin
                    if(x == 0)
                        state <= C;
                    else
                        state <= M;
                    z <= 0;
                end


                C : begin
                    if(x == 0)
                    begin
                        state <= C;
                        z     <= 1;
                    end
                    else
                    begin
                        state <= M;
                        z     <= 0;
                    end
                end
```

```
                M : begin
                    if(x == 1)
                        state <= N;
                    else
                        state <= A;
                    z <= 0;
                end

                N : begin
                    if(x == 0)
                        state <= B;
                    else
                        state <= M;
                    z <= 0;
                end

                default: begin
                    state <= A;
                    z     <= 0;
                end

            endcase
        end
    end
endmodule
```

```
`timescale 1ns / 1ps

module fsm2_behav_tb;

    reg  clk;
    reg  rst;
    reg  x;
    wire z;

    fsm2_behav dut (
        .clk (clk),
        .rst (rst),
        .x   (x),
        .z   (z)
    );
    initial clk = 0;
    always #5 clk = ~clk;


    localparam integer N = 42;
    localparam [N-1:0] STIM =
        42'b010011000111000011110000011111000000111111;

    integer i;

    initial begin

        rst = 1;
        x   = 0;
        repeat (3) @(posedge clk);
        rst = 0;

        $display("===== Reset Completed =====");
        $display("Input Sequence:
010011000111000011110000011111000000111111");

        for (i = N-1; i >= 0; i = i - 1) begin
            x = STIM[i];
            @(posedge clk);
            #1;
            $display("t=%0t | index=%0d | x=%b | z=%b",
                     $time, (N-1-i), x, z);
        end

        $display("===== Simulation Finished =====");
        #20;
        $finish;
    end

endmodule
```



2.7 Post-Implementation Timing Simulation of the Behavioral Model



2.8 Utilization and Timing Summaries of Behavioral Model

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	8.484	1	4	FSM_sequential_state_reg[0]/C	FSM_sequential_state_reg[0]/D	1.509	0.580	0.929	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 2	8.504	1	4	FSM_sequential_state_reg[0]/C	FSM_sequential_state_reg[2]/D	1.535	0.606	0.929	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 3	8.744	1	4	FSM_sequential_state_reg[1]/C	FSM_sequential_state_reg[1]/D	1.252	0.580	0.672	10.0	sys_clk_pin	sys_clk_pin		0.035
Path 4	8.760	1	4	FSM_sequential_state_reg[1]/C	z_reg/D	1.280	0.608	0.672	10.0	sys_clk_pin	sys_clk_pin		0.035

Intra-Clock Paths - sys_clk_pin - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 5	0.252	1	4	FSM_sequential_state_reg[0]/C	z_reg/D	0.359	0.184	0.175	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 6	0.269	1	4	FSM_sequential_state_reg[0]/C	FSM_sequential_state_reg[1]/D	0.361	0.186	0.175	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 7	0.290	1	4	FSM_sequential_state_reg[2]/C	FSM_sequential_state_reg[2]/D	0.397	0.230	0.167	0.0	sys_clk_pin	sys_clk_pin		0.000
Path 8	0.303	1	4	FSM_sequential_state_reg[2]/C	FSM_sequential_state_reg[0]/D	0.394	0.227	0.167	0.0	sys_clk_pin	sys_clk_pin		0.000

Unconstrained Paths - NONE - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 9	∞	1	4	rst	FSM_sequential_state_reg[0]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 10	∞	1	4	rst	FSM_sequential_state_reg[1]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 11	∞	1	4	rst	FSM_sequential_state_reg[2]/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 12	∞	1	4	rst	z_reg/CLR	3.197	1.464	1.733	∞	input port clock	sys_clk_pin		0.025
Path 13	∞	2	4	x	FSM_sequential_state_reg[2]/D	3.037	1.643	1.394	∞	input port clock	sys_clk_pin		0.025
Path 14	∞	2	4	x	z_reg/D	3.030	1.643	1.387	∞	input port clock	sys_clk_pin		0.025
Path 15	∞	2	4	x	FSM_sequential_state_reg[0]/D	3.009	1.615	1.394	∞	input port clock	sys_clk_pin		0.025
Path 16	∞	2	4	x	FSM_sequential_state_reg[1]/D	3.002	1.615	1.387	∞	input port clock	sys_clk_pin		0.025

Unconstrained Paths - NONE - sys_clk_pin - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 17	∞	2	4	x	FSM_sequential_state_reg[1]/D	0.814	0.304	0.510	-∞	input port clock	sys_clk_pin		0.000
Path 18	∞	2	4	x	z_reg/D	0.814	0.304	0.510	-∞	input port clock	sys_clk_pin		0.000
Path 19	∞	2	4	x	FSM_sequential_state_reg[2]/D	0.823	0.303	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 20	∞	2	4	x	FSM_sequential_state_reg[0]/D	0.824	0.304	0.520	-∞	input port clock	sys_clk_pin		0.000
Path 21	∞	1	4	rst	FSM_sequential_state_reg[0]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 22	∞	1	4	rst	FSM_sequential_state_reg[1]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 23	∞	1	4	rst	FSM_sequential_state_reg[2]/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000
Path 24	∞	1	4	rst	z_reg/CLR	0.920	0.232	0.688	-∞	input port clock	sys_clk_pin		0.000

Unconstrained Paths - sys_clk_pin - NONE - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 25	∞	1	1	z_reg/C	z	6.188	4.115	2.073	∞	sys_clk_pin			0.025

Unconstrained Paths - sys_clk_pin - NONE - Hold

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 26	∞	1	1	z_reg/C	z	1.913	1.406	0.507	-∞	sys_clk_pin			0.025

Output Ports Clock-to-out

Reference Clock	Output Port	IO Reg Type	Delay Type	Max Clk to Port	Max Edge	Max Process Corner	Min Clk to Port	Min Edge	Min Process Corner	Internal Clock	Output Enable/Disable
clk100	z	FDCE		11.390	Rise	SLOW	3.394	Rise	FAST		

FSM2 Behavioral Max-Min Port

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	8,582 ns	Worst Hold Slack (WHS):	0,210 ns	Worst Pulse Width Slack (WPWS):	4,500 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns	Total Pulse Width Negative Slack (TPWS):	0,000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	3	Total Number of Endpoints:	3	Total Number of Endpoints:	5

All user specified timing constraints are met.

FSM2 Behavioral Timing Summaries

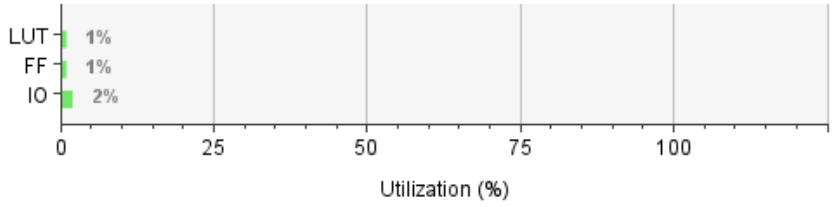
Primitives

Ref Name	Used	Functional Category
LUT4	4	LUT
FDCE	4	Flop & Latch
IBUF	3	IO
OBUF	1	IO
BUFG	1	Clock

FSM2 Behavioral Moore Utilization

Summary

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
FF	4	65200	0.01
IO	4	210	1.90



FSM2 Behavioral Utilization

2.9 Comparison in terms of resource usage and path delays

When comparing the structural FSM (FSM2) with the behavioral implementation (fsm2_behav), the results clearly show that the structural design is more optimized in both resource usage and timing performance. The structural FSM requires only 3 LUTs, whereas the behavioral model uses 4 LUTs—approximately a 33% increase in logic utilization. Timing analysis also indicates a slight advantage for the structural version, which achieves a clock-to-out delay of 11.258 ns compared to 11.390 ns in the behavioral model, corresponding to maximum operating frequencies of 88.8 MHz and 87.7 MHz respectively. These results demonstrate that the structural FSM is marginally faster and more area-efficient. Nevertheless, the behavioral FSM remains easier to read, modify, and maintain, making it more suitable for conceptual understanding, documentation, and higher-level design workflows.