

EXPERIMENT-05

AIM: To simulate and synthesis finite state machine using Xilinx ISE.

APPARATUS REQUIRED: Xilinx 14.7 Spartan6 FPGA

PROCEDURE:

STEP:1 Start the Xilinx navigator, Select and Name the New project.

STEP:2 Select the device family, device, package and speed.

STEP:3 Select new source in the New Project and select Verilog Module as the Source type.

STEP:4 Type the File Name and Click Next and then finish button. Type the code and save it.

STEP:5 Select the Behavioral Simulation in the Source Window and click the check syntax.

STEP:6 Click the simulation to simulate the program and give the inputs and verify the outputs as per the truth table.

STEP:7 Select the Implementation in the Sources Window and select the required file in the Processes Window.

STEP:8 Select Check Syntax from the Synthesize XST Process. Double Click in the Floorplan Area/IO/Logic-Post Synthesis process in the User Constraints process group. UCF(User constraint File) is obtained.

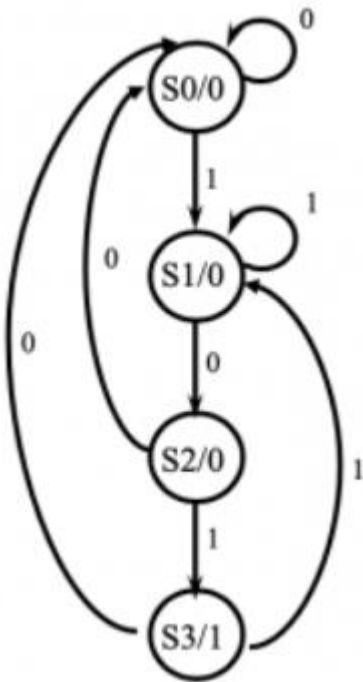
STEP:9 In the Design Object List Window, enter the pin location for each pin in the Loc column Select save from the File menu.

STEP:10 Double click on the Implement Design and double click on the Generate Programming File to create a bitstream of the design.(.v) file is converted into .bit file here.

STEP:11 On the board, by giving required input, the LEDs starts to glow light, indicating the output.

STEP:12 Load the Bit file into the SPARTAN 6 FPGA

LOGIC DIAGRAM:



VERILOG CODE:

```
module finitestatic(clk,res,din,dout);
input clk,res,din;
output reg dout;
parameter s0=2'b00,
          s1=2'b01,
          s2=2'b10,
          s3=2'b11;
reg [1:0]state;
always@ (posedge clk or posedge res)
begin
if(res) begin
dout <= 1'b0;
state <= s0;
end
else begin
```

```
case(state)
s0:begin
dout <=1'b0;
if(din)
state <=s1;
else
state <=s0;
end
s1:begin
dout <=1'b0;
if(~din)
state <=s2;
else
state <=s1;
end
s2:begin
dout <=1'b0;
if(din)
state <=s3;
else
state <=s0;
end
s3:begin
dout <=1'b1;
if(din)
state <=s1;
else
state <=s0;
```

endmodule

The screenshot displays the Vivado IDE interface for a project named 'finitestatic.v'. The left pane shows the variable declaration table, and the right pane shows the timing diagram.

Name	Value
clk	0
res	0
din	1
dout	1
> state[1:0]	1
> s0[1:0]	0
> s1[1:0]	1
> s2[1:0]	2
> s3[1:0]	3

The timing diagram on the right shows the waveforms for the variables. The horizontal axis represents time in nanoseconds (ns), ranging from 0 to 3,500.000 ns. The vertical axis lists the variables: clk, res, din, dout, state[1:0], s0[1:0], s1[1:0], s2[1:0], and s3[1:0]. The waveforms show the digital signals for each variable over time. A yellow vertical line marks the 1,500.000 ns point.

Thus ,the given finite state machine are simulated and synthesis are executed successfully.