**WELCOME**DEPARTMENT OF ECE

COMPUTER AIDED DIGITAL DESIGN

UE21EC252A

PROF.Y J PAVITHRA

PROJECT: TIC-TAC-TOE GAME

NAME : AKASH RAVI BHAT

SRN : PES1UG21EC025

SEC : A

TIC-TAC-TOE GAME USING FSM

AIM: Simulation and FPGA Implementation of FSM of TIC-TAC-TOE GAME using Xilinx Vivado tool.

PROCEDURE:

Step1: Create a Vivado Project using IDE sourcing verilog HDL model and targeting a specific FPGA device located on the Basys3 (Xilinx Artix-7 FPGA: XC7A35T-1CPG236C)

Step2 : Simulate the design using Vivado Simulator (Test Bench Module).

Step3 : Synthesize the design and observe the Schematic.

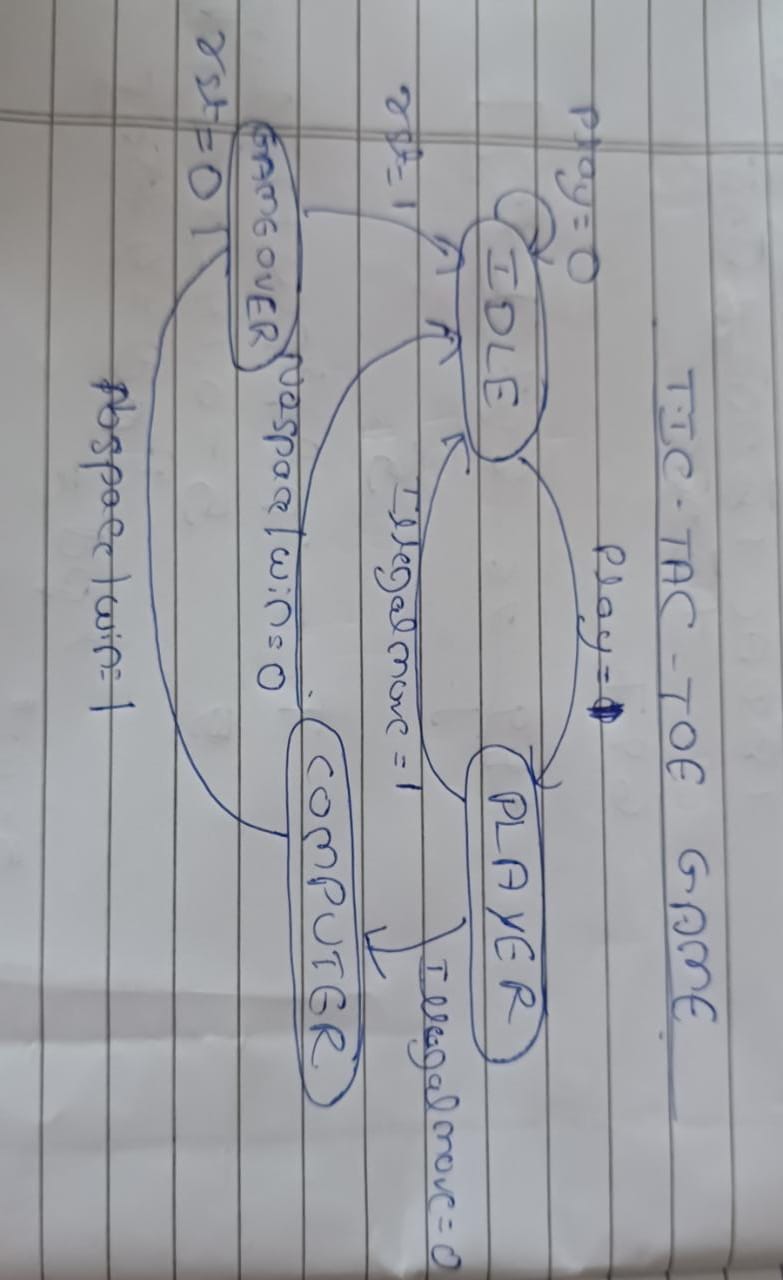
Step4 : Implement the design.

Step5 : Write Design Constraint (XDC) file to constrain the pin locations.

Step6 : Generate the bitstream.

Step7 : Configure the FPGA using the generated bitstream and verify the functionality in hardware.

FSM State diagram



DESIGN SOURCE

module rk(clk,rst,play,illegal\_move,no\_space,win,qout,seq,clk\_div);

input logic clk,rst;

input logic play;

input logic illegal\_move;

input logic no\_space;

input logic win;

output logic seq;

output logic clk\_div;

output logic [2:0]qout;

parameter idle=3'b000,

player = 3'b001,

computer = 3'b010,

game\_over = 3'b011;

reg[2:0]present,next;

logic [26:0] count=0;

always@(posedge clk)

begin

count<=count+1;

/\*if(count[26]==1)

begin

clk\_div<=count[26];

count<=0;

end \*/

end

assign clk\_div=count[26];

always@(posedge count[26])

begin

if(rst)

present<=idle;

else

present<=next;

end

always@(present , play,illegal\_move,no\_space,win)

begin

next = idle;

case(present)

idle:begin if(play) next =player; end

player: begin if(illegal\_move) next = idle;

else next = computer;end

computer: begin if(no\_space | win ) next = game\_over ;

else next = idle ; end

game\_over: begin if(rst) next = idle ;

else next = game\_over ; end

endcase

end

assign qout= (present === idle )? 3'b000 : (present === player )? 3'b001: (present === computer )? 3'b010 : (present === game\_over )? 3'b011: 3'b111;

assign seq= (present === game\_over )? 1 :0;

endmodule

SIMULATION SOURCE

module rk\_tb();

reg clk,rst;

reg play,illegal\_move,no\_space,win;

wire[2:0] qout;

wire seq;

wire clk\_div;

rk dut(.clk(clk),.rst(rst),.play(play),.illegal\_move(illegal\_move),.no\_space(no\_space),.win(win),.qout(qout),.seq(seq),.clk\_div(clk\_div));

always #5 clk=~clk;

initial begin

play=1;

illegal\_move=0;

no\_space=0;

win=0;

rst=1;

clk=0;

play=0;

#4rst=0;

#6play=1;

#10illegal\_move=0;

#10no\_space=0;

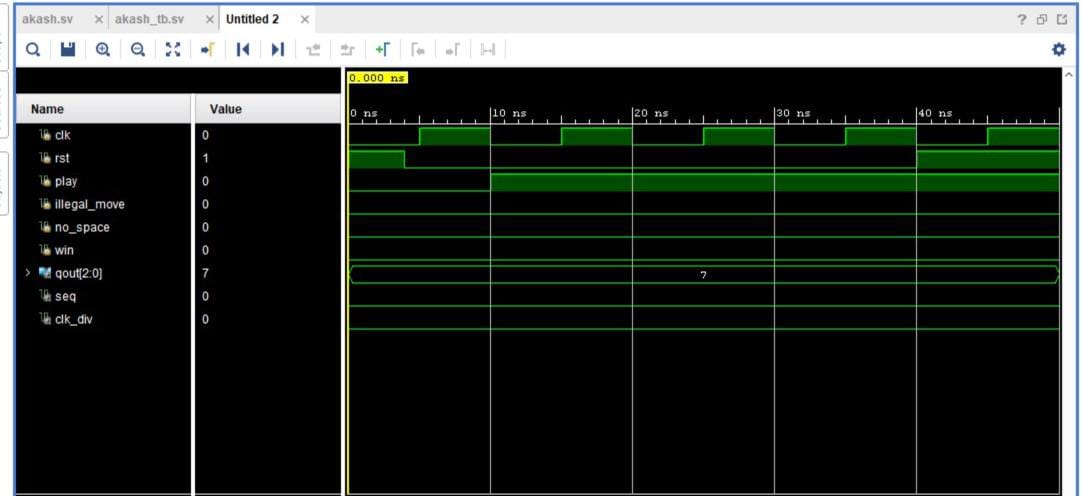
#10rst=1;

#10$finish;

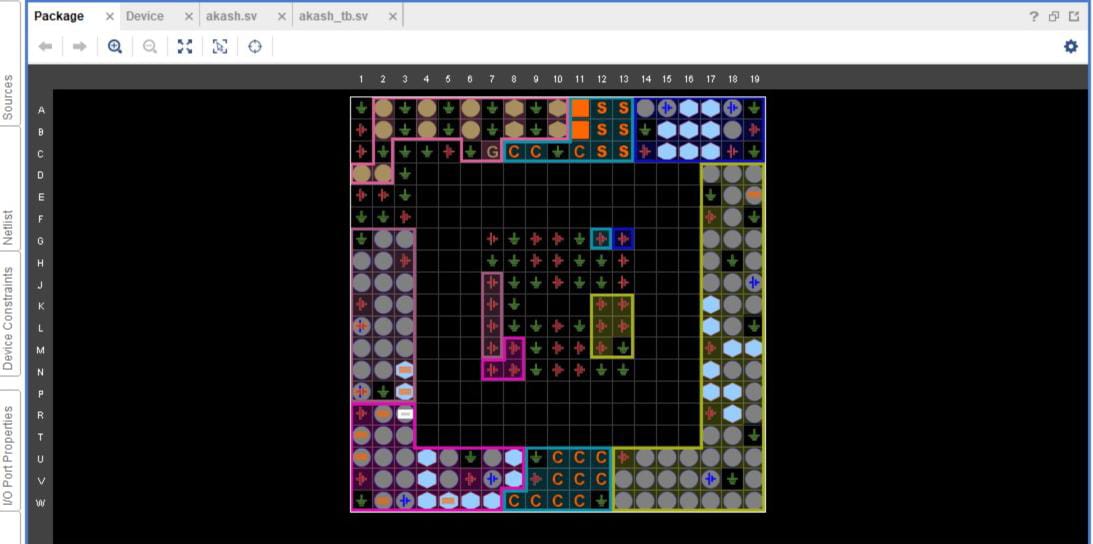
end

endmodule

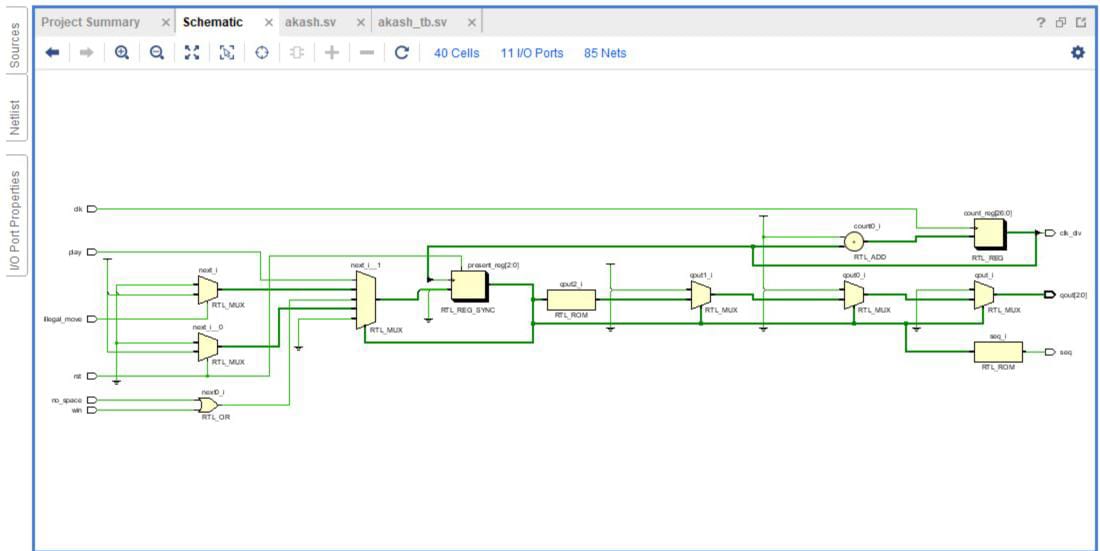
OUTPUT WAVEFORM:



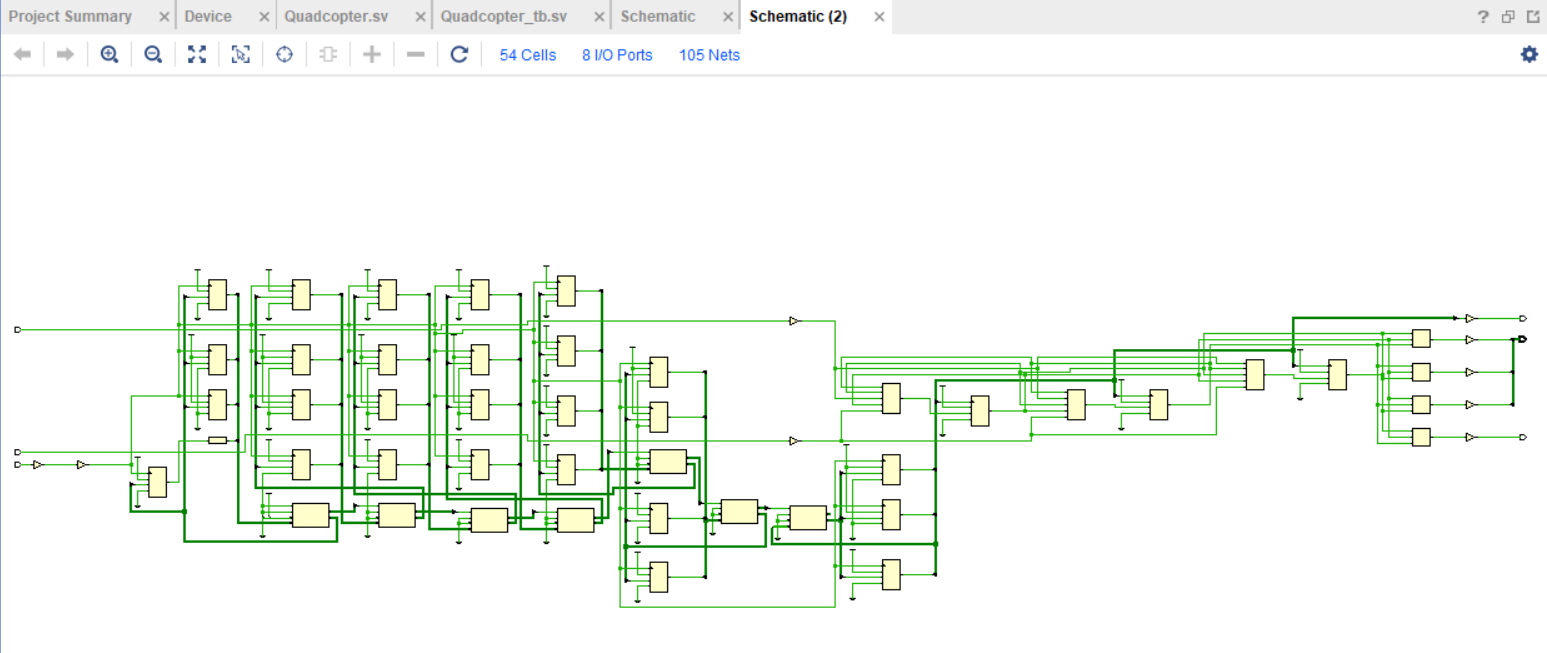
IMPLEMENTED DESIGN



ELABORATED DESIGN



SYNTHESISED DESIGN



RESULT

TIC-TAC-TOE GAME FSM was successfully simulated and implemented