WINTER SEMESTER 2016

DIGITAL LOGIC AND DESIGN PROJECT

PROJECT TOPIC:FASTEST FINGER FIRST



GROUP MEMBERS:

SHASHANK PATHAK (15BCE1287)

OSHO AGYEYA (15BCE1326)

AIM:

To design a fastest finger first circuit to find which player pressed the switch first by utilization of basic gates and flip flops made by these gates itself.

ABSTRACT:

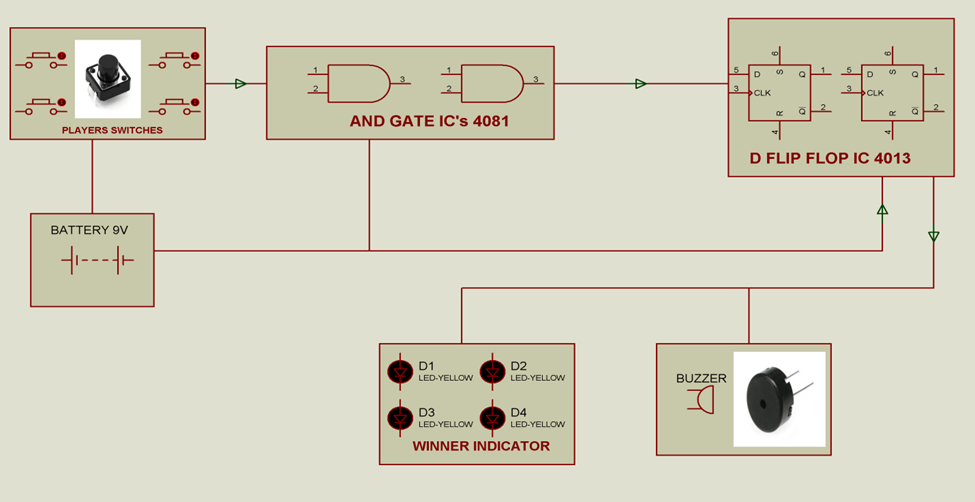
Quiz competitions are held in schools, colleges and also in Television channels as they carry out special programs, various other organizations. The process of identifying a participant whose option button pressed is very critical in timings as they might differ in seconds or milliseconds.

This process of carrying the tedious work is made simple by this Circuit. This Electronic Quiz buzzer enables to identify a particular participant as whom have pressed first amongst all. This Circuit is made up of a simple AND gate IC’s and a couple of SR type flip Flop IC’s are used in order to set and reset the device. An LED for Each participant is provided such that, it will indicate the person who has pressed the option switch first amongst all and also a beep sound indicator is made available when any switch is pressed. This is made of simple components using an AND gate , Flip Flop IC , LED’s and buzzer hence the circuit is cost effective.

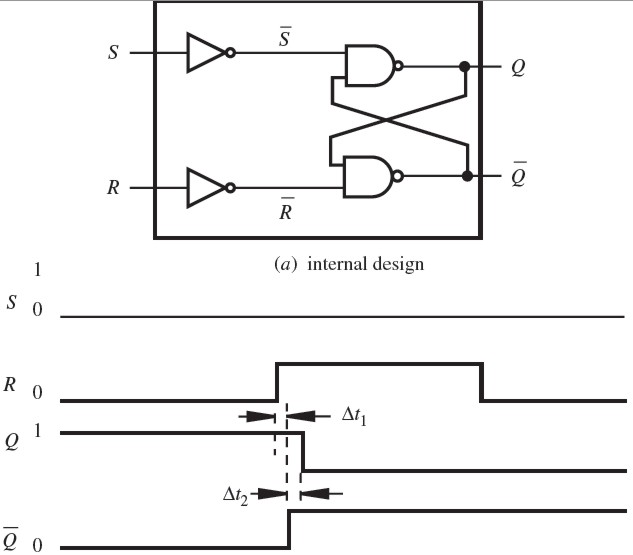
COMPONENTS REQUIRED:

1. Bread board
2. Wires
3. Switches
4. AND,NAND,NOT,OR gates
5. Battery 9V
6. Winner indicators LED’s
7. Buzzer
8. SR Flip Flop using basic gates
9. Resistors 1k ohm

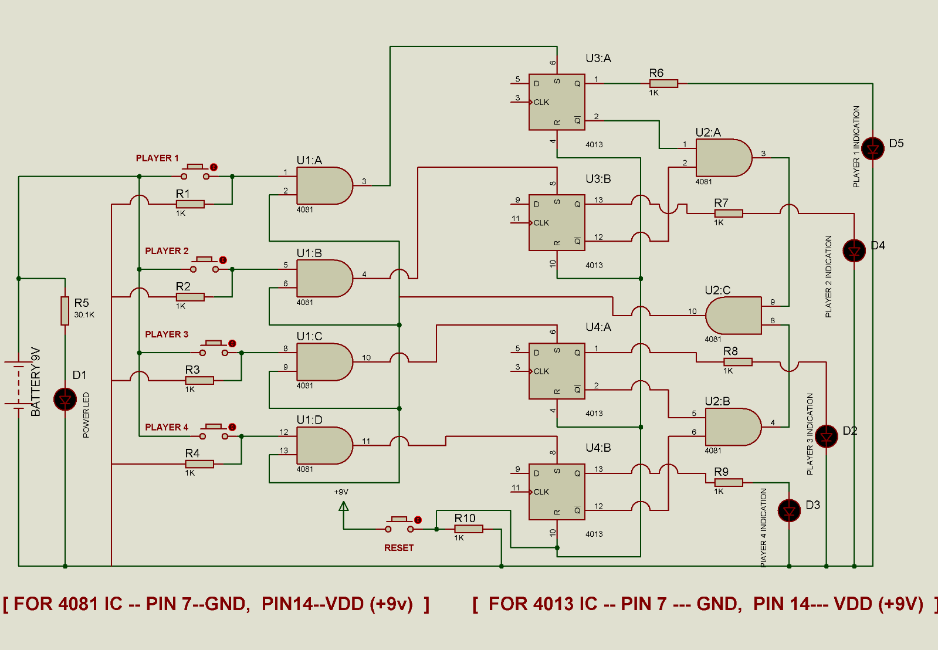
BLOCK DIAGRAM:



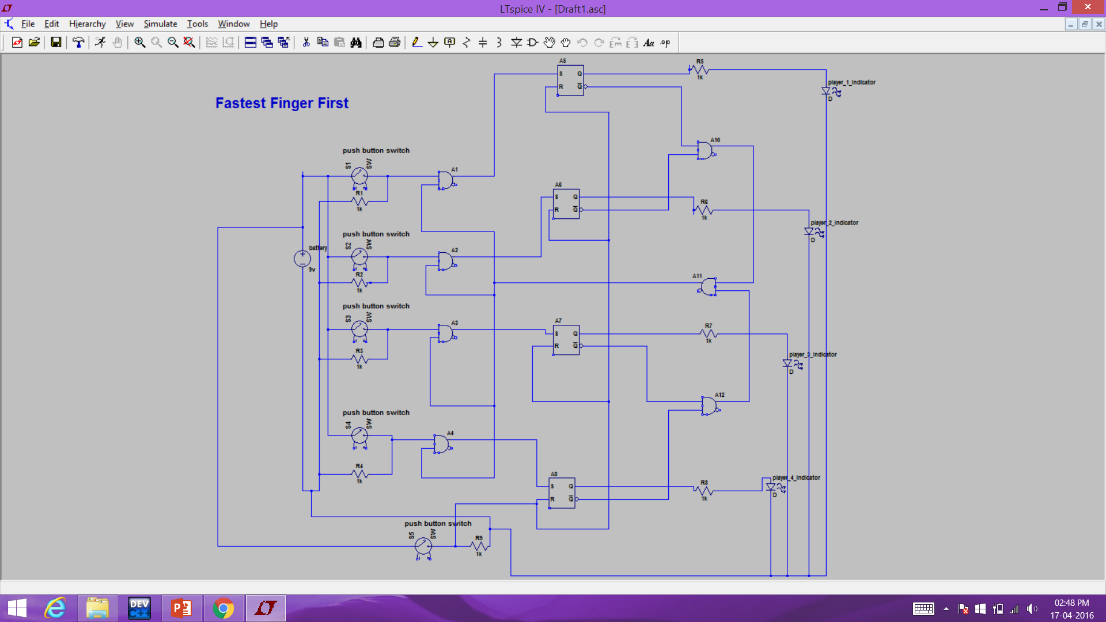
SR FLIP FLOP BY BASIC GATES



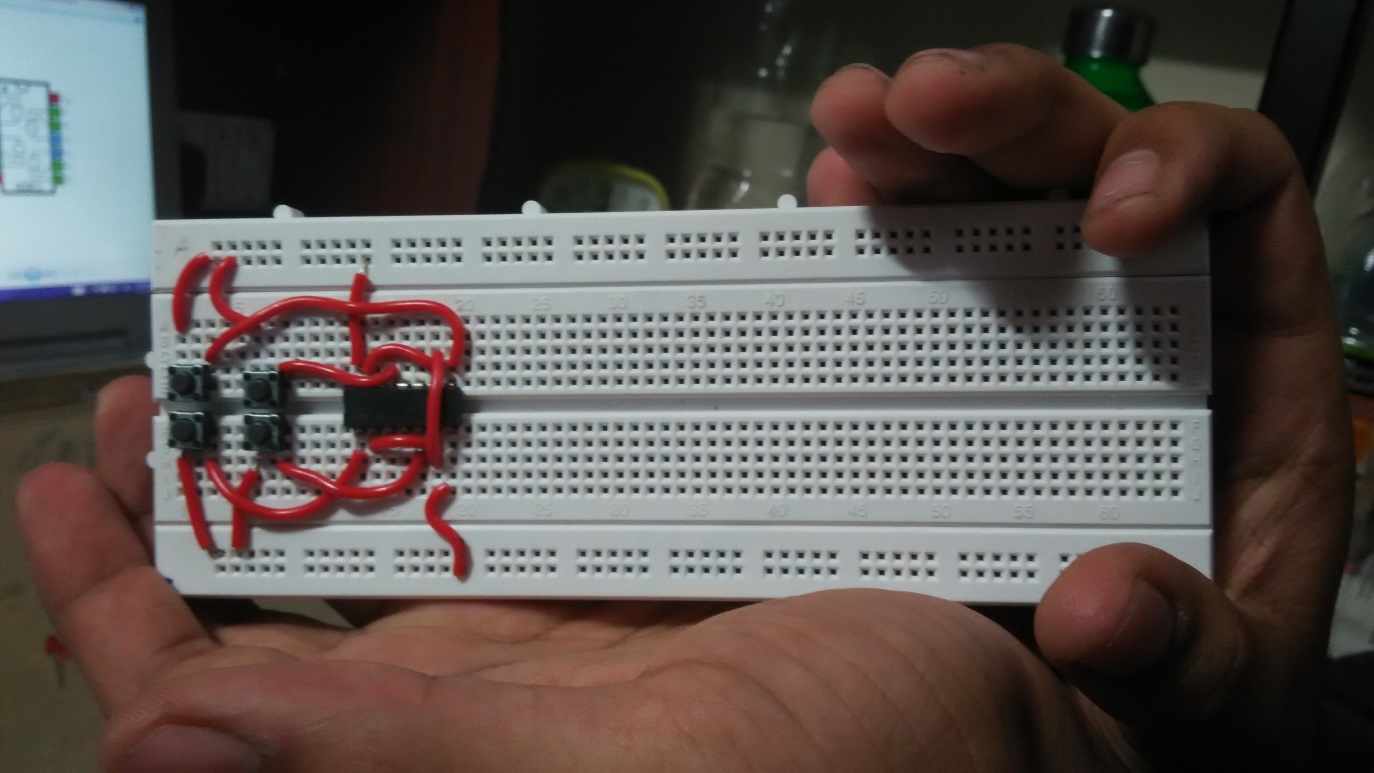
CIRCUIT DIAGRAM:

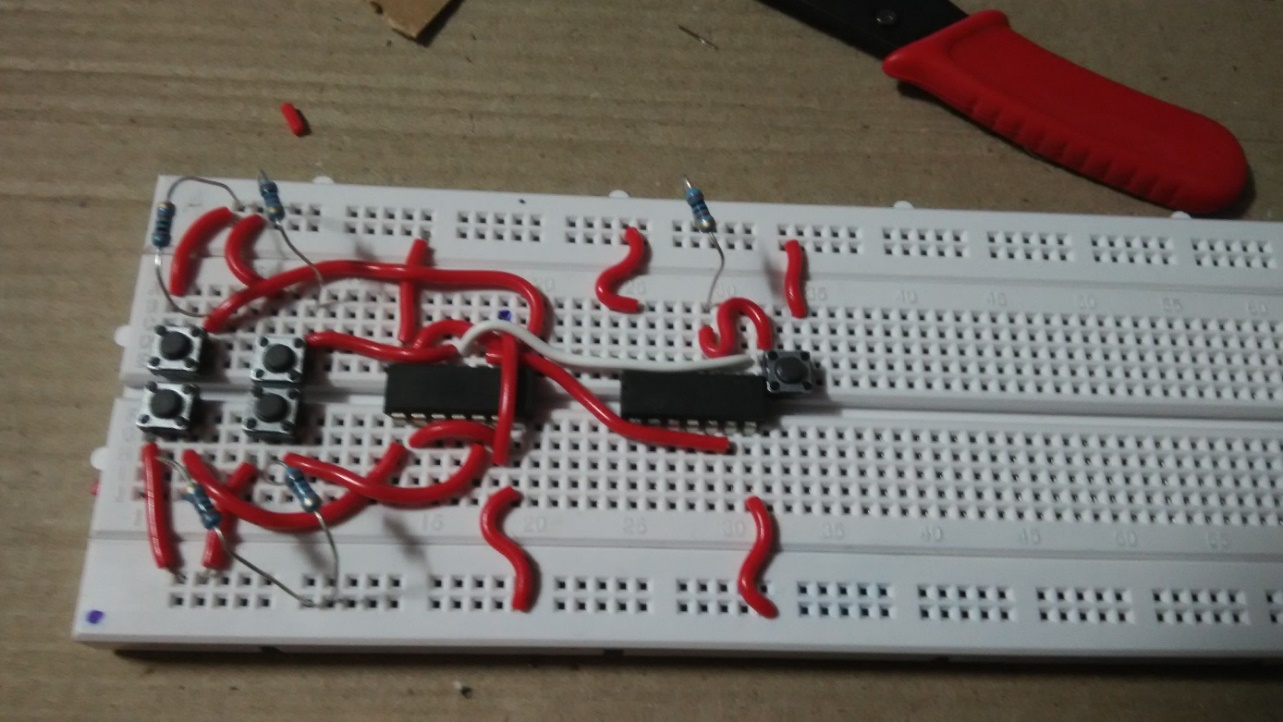


LTSpice DIAGRAM:

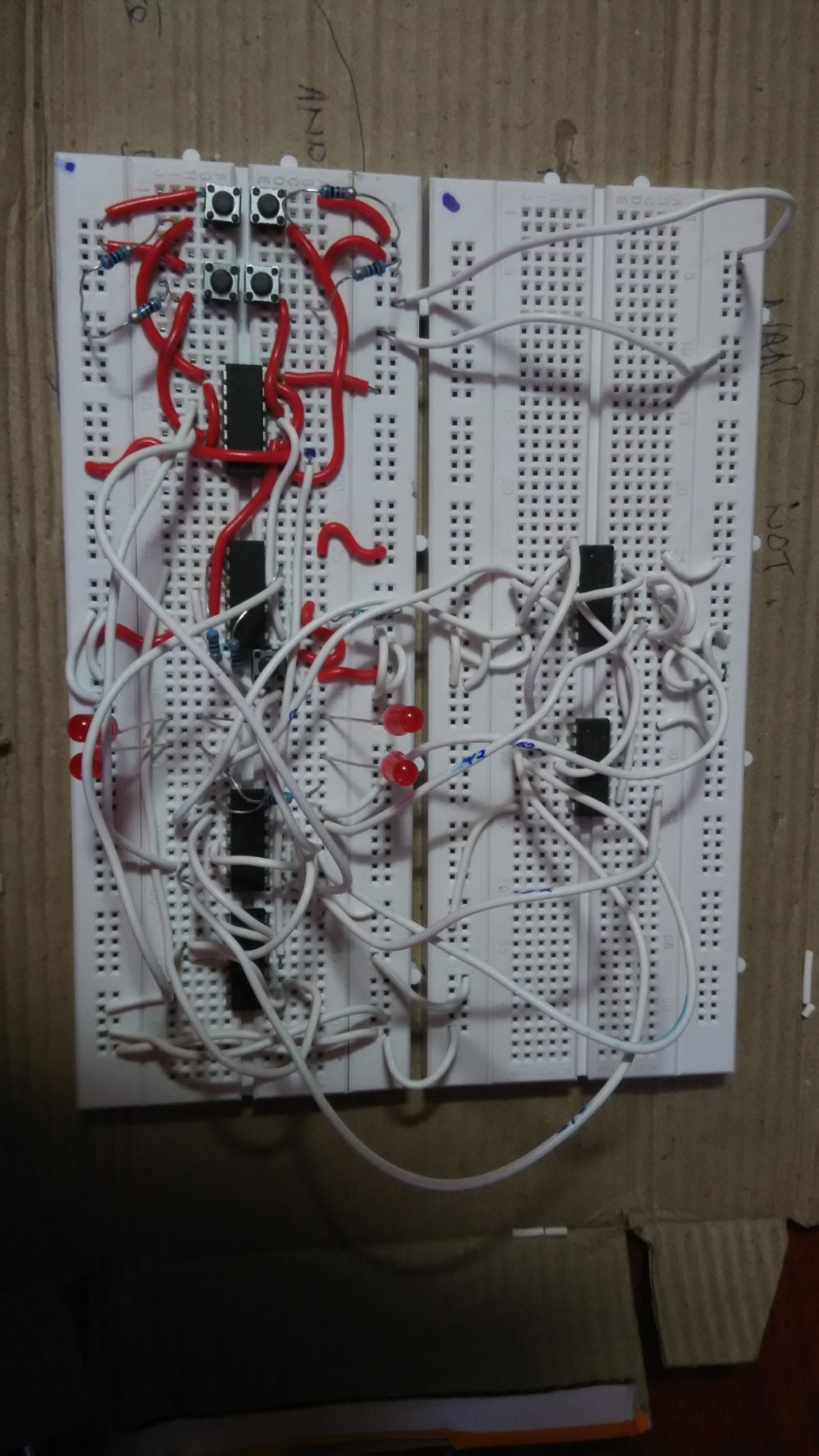


CIRCUIT UNDER CONSTRUCTION:





FINAL PROJECT:



VHDL CODE:

->NAND GATE:

--import std\_logic from the IEEE library

library ieee;

use ieee.std\_logic\_1164.all;

--ENTITY DECLARATION: name, inputs, outputs

entity nandGate is

port( A, B : in std\_logic;

F : out std\_logic);

end nandGate;

--FUNCTIONAL DESCRIPTION: how the NAND Gate works

architecture func of nandGate is

begin

F <= A nand B;

end func;

WORKING:

*The working of these codes is as follows:Firstly, the standard libraries are imported. The entity to be declared is NAND gate. The ports signify the input and outputs. Here, A and B are inputs while F is the output. The NAND gate function assigns A nand B to F. The assignment is done according to the respective truth table.*

|  |  |  |
| --- | --- | --- |
| *A* | *B* | *F* |
| *0* | *0* | *1* |
| *0* | *1* | *1* |
| *1* | *0* | *1* |
| *1* | *1* | *0* |

->SR FLIP FLOP:

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity SR\_FF is

PORT( S,R,CLOCK: in std\_logic;

Q, QBAR: out std\_logic);

end SR\_FF;

Architecture behavioral of SR\_FF is

begin

PROCESS(CLOCK)

variable tmp: std\_logic;

begin

if(CLOCK='1' and CLOCK'EVENT) then

if(S='0' and R='0')then

tmp:=tmp;

elsif(S='1' and R='1')then

tmp:='Z';

elsif(S='0' and R='1')then

tmp:='0';

else

tmp:='1';

end if;

end if;

Q <= tmp;

QBAR <= not tmp;

end PROCESS;

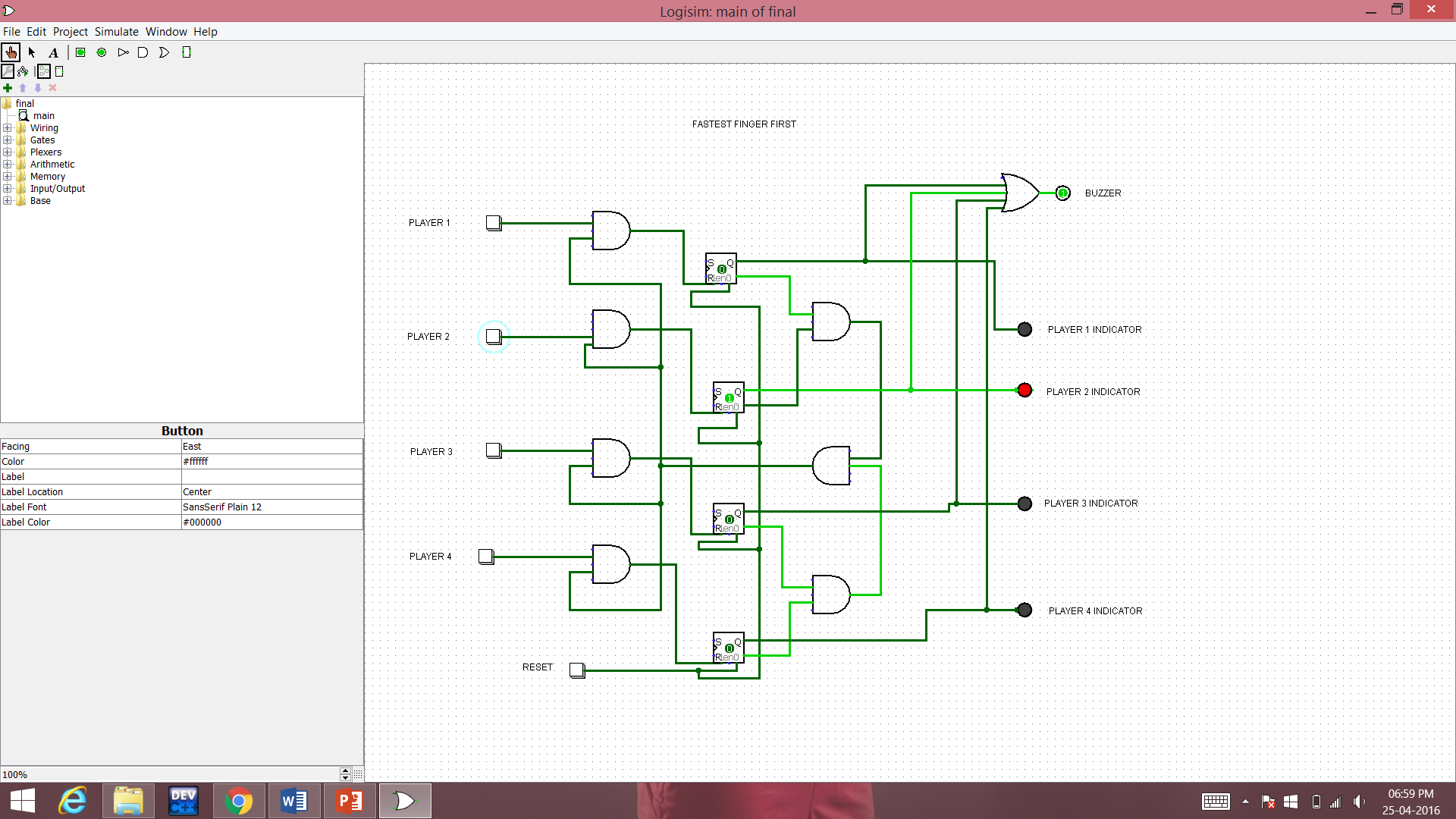
end behavioral;

WORKING:

*The working of these codes is as follows: Firstly, the standard libraries are imported. The entity to be declared is SR FLIP FLOP. The ports signify the input and outputs. Here, S,R AND CLOCK are inputs while Q AND QBAR are the outputs. The SR flip flop calculates tmp. The calculation is done according to the respective truth table. Finally, tmp is assigned to Q while not tmp is assigned to QBAR.*

|  |  |  |
| --- | --- | --- |
| *S* | *R* | *Q* |
| *0* | *0* | *PREVIOUS STATE* |
| *0* | *1* | *0* |
| *1* | *0* | *1* |
| *1* | *1* | *Z(INVALID)* |

SIMULATION USING LOGISM:



The End