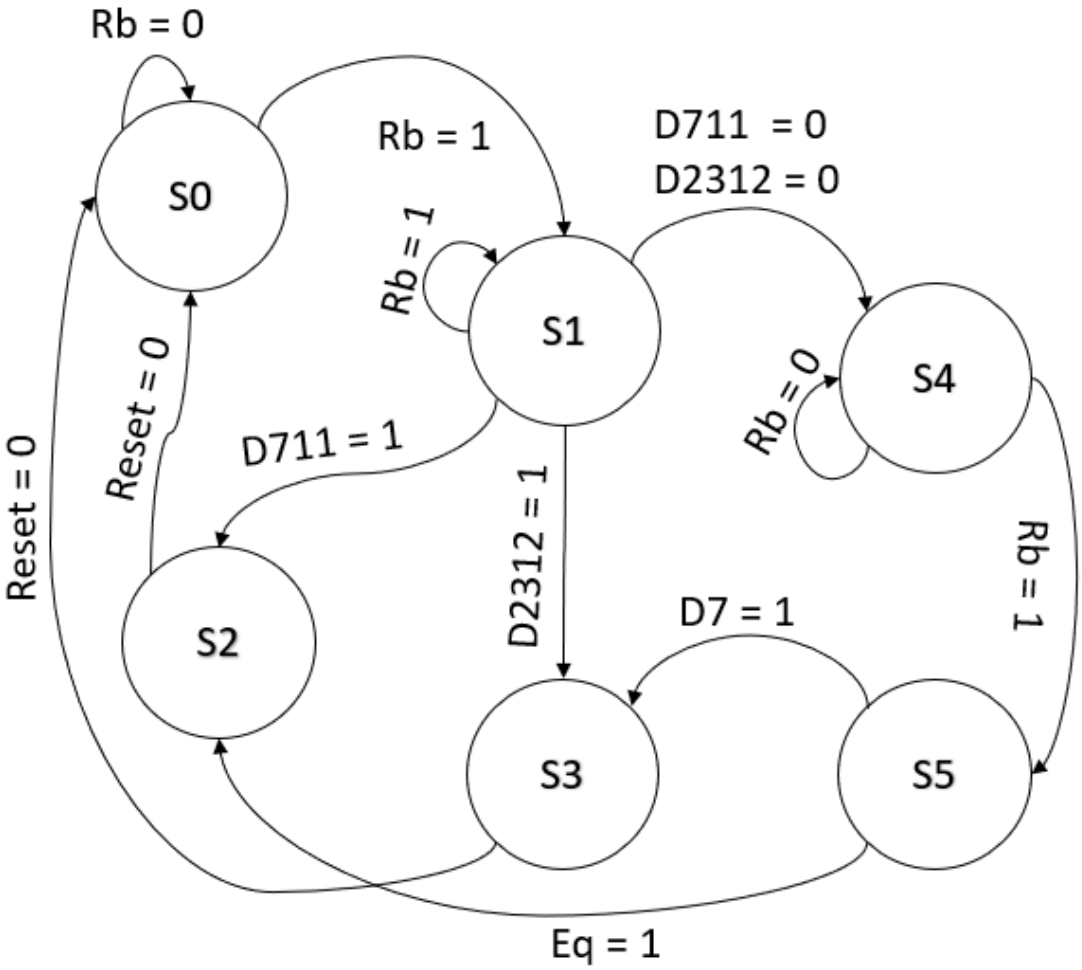
Report for Dice Game

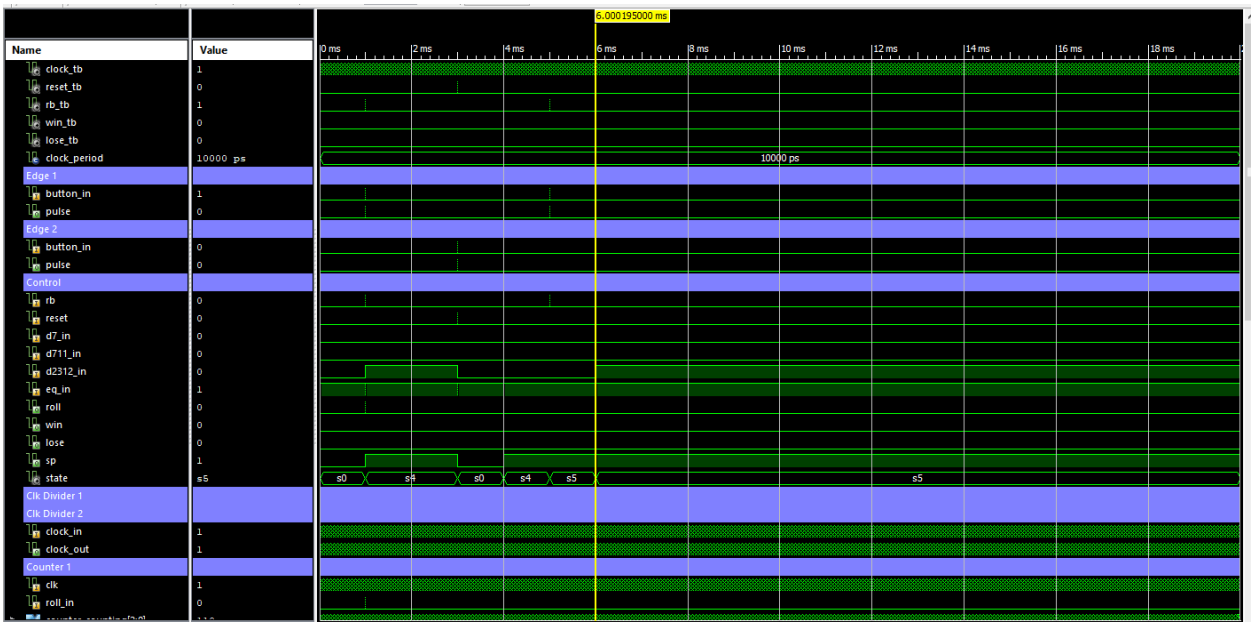
Question1:

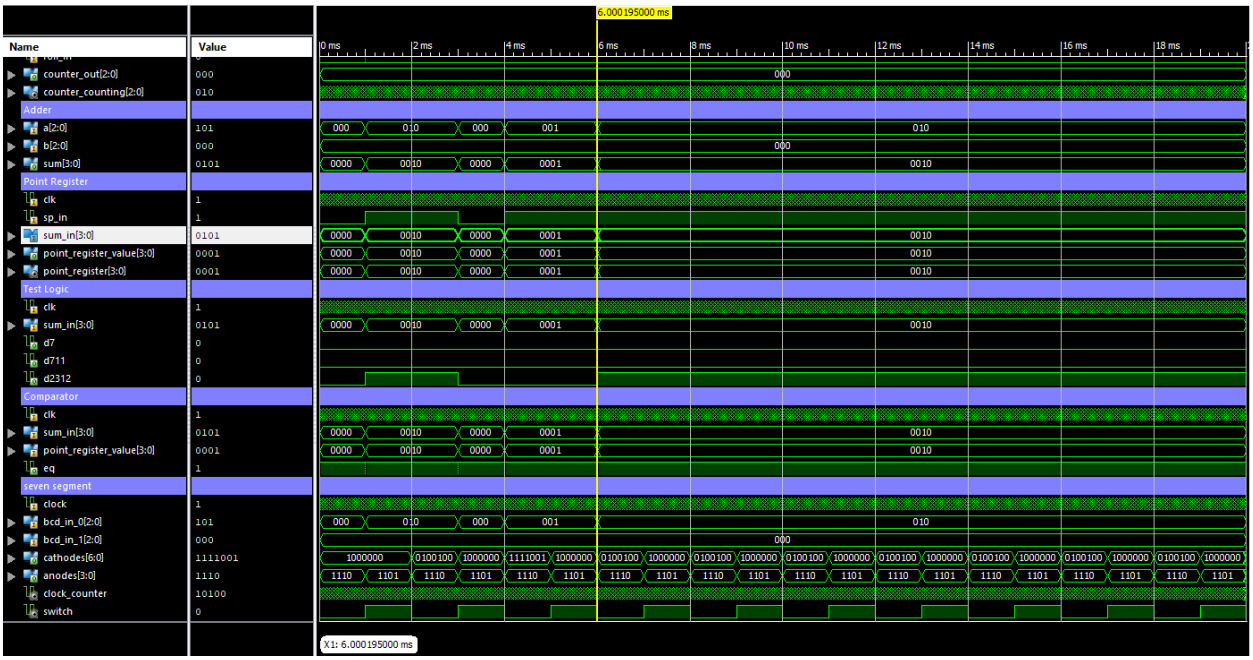
The FSM of the flow chart is as follows



Question 2:

The Simulation are as follows:


6.000195000 ms
Name
Value
clock_tb
reset_tb
rb_tb
win_tb
lose_tb
clock_period
10000 ps
Edge 1
button_in
pulse
Edge 2
button_in
pulse
Control
rb
reset
d7_in
d711_in
d2312_in
eq_in
roll
win
lose
sp
state
s5
Clk Divider 1
Clk Divider 2
clock_in
clock_out
Counter 1
clk
roll_in


6.000195000 ms
Name
Value
counter_out[2:0]
counter_counting[2:0]
Adder
a[2:0]
b[2:0]
sum[3:0]
Point Register
clk
sp_in
sum_in[3:0]
point_register_value[3:0]
point_register[3:0]
Test Logic
clk
sum_in[3:0]
d7
d711
d2312
Comparator
clk
sum_in[3:0]
point_register_value[3:0]
eq
seven segment
clock
bcd_in_0[2:0]
bcd_in_1[2:0]
cathodes[6:0]
anodes[3:0]
clock_counter
switch
X1: 6.000195000 ms

Win and Lose conditions

The Top Module is as follows:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:    21:32:06 10/25/2024
-- Design Name:
-- Module Name:    top - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
```

```
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity top is
	Port(
		clock : in std_logic;
		Reset : in std_logic;
		Rb : in std_logic;
		Win : out std_logic := '0';
		Lose : out std_logic := '0';
		cathodes_out : out std_logic_vector(6 downto 0) := (others => '0');
		anodes_out : out std_logic_vector(3 downto 0) := (others => '1')
	);
end top;

architecture Behavioral of top is

	component Edge_Detector
	  Port(
		 CLK_SYS   : in std_logic;
		 BUTTON_IN : in std_logic;
		 PULSE     : out std_logic
	  );
	end component;

	component Control is
	Port(
		clk     : in std_logic;
		Rb      : in std_logic;
		Reset   : in std_logic;
		D7_in   : in std_logic;
		D711_in  : in std_logic;
		D2312_in : in std_logic;
		Eq_in   : in std_logic;
		Roll    : out std_logic;
		Win     : out std_logic;
		Lose    : out std_logic;
		Sp      : out std_logic
	);
	end component;

	component onetosix_counter
	Port(
```

```
                clk : in std_logic;
                reset : in std_logic;
                Roll_in : in std_logic;
                counter_out : out std_logic_vector(2 downto 0)
        );
        end component;

        component clock_divider is
generic(
                FPGA_CLK_FREQUENCY : integer := 100000000;
                REQUIRED_FREQUENCY : integer := 50000000
);
Port (
                clock_in    : in STD_LOGIC;
   clock_out : out STD_LOGIC
        );
        end component;

        component Adder
Port (
   A : in  STD_LOGIC_VECTOR(2 downto 0);  -- 4-bit input A
   B : in  STD_LOGIC_VECTOR(2 downto 0);  -- 4-bit input B
   SUM : out STD_LOGIC_VECTOR(3 downto 0)  -- 5-bit Sum Output
);
        end component;

        component point_register
        Port(
                clk : in STD_LOGIC;
                Sp_in : in STD_LOGIC;
                SUM_IN : in std_logic_vector(3 downto 0);
                point_register_value : out std_logic_vector(3 downto 0)
        );
        end component;

        component comparator
        Port(
                clk : in std_logic;
                sum_in : in std_logic_vector(3 downto 0);
                point_register_value : in std_logic_vector(3 downto 0);
                Eq : out std_logic
        );
        end component;

        component Test_Logic
        Port(
                clk   : in std_logic;
                sum_in   : in std_logic_vector(3 downto 0);
```

```
                D7    : out std_logic;
                D711  : out std_logic;
                D2312 : out std_logic
        );
        end component;

        component bcdto7seg
         Port (
                        clock : in std_logic;
                         bcd_in_0   : in std_logic_vector (2 downto 0);
                         bcd_in_1   : in std_logic_vector (2 downto 0);
                         cathodes : out std_logic_vector (6 downto 0);
                        anodes : out std_logic_vector(3 downto 0)
         );
        end component;

        signal start_button : std_logic := '0';
        signal reset_button : std_logic := '0';
        signal Roll_out : std_logic := '0';
        signal Win_out : std_logic := '0';
        signal Lose_out : std_logic         := '0';
        signal counter_one_out : std_logic_vector(2 downto 0) := (others => '0');
        signal counter_two_out : std_logic_vector(2 downto 0) := (others => '0');
        signal clock_divided_1 : std_logic := '0';
        signal clock_divided_2 : std_logic := '0';
        signal Sum_out : std_logic_vector(3 downto 0) := (others => '0');
        signal point_register_val : std_logic_vector(3 downto 0) := (others => '0');
        signal Eq_out : std_logic := '0';
        signal D7_out : std_logic := '0';
        signal D711_out : std_logic := '0';
        signal D2312_out : std_logic := '0';
        signal Sp_out : std_logic := '0';
begin

        uut : Edge_Detector
        port map(
                CLK_SYS => clock,
                BUTTON_IN => Rb,
                PULSE => start_button
        );

        uut2 : Edge_Detector
        port map(
                CLK_SYS => clock,
                BUTTON_IN => Reset,
                PULSE => reset_button
        );
```

```
	uut3 : Control
	port map(
		clk => clock,
		Rb => start_button,
		Reset => reset_button,
		D7_in => D7_out,
		D711_in => D711_out,
		D2312_in => D2312_out,
		Eq_in => Eq_out,
		Roll => Roll_out,
		Win => Win,
		Lose => Lose,
		Sp => Sp_out

	);

	--uut4 : clock_divider
	--generic map(
	--		FPGA_CLK_FREQUENCY => 100000000,
	--		REQUIRED_FREQUENCY => 100000000
	--)
	--port map(
	--		clock_in => clock,
	--		clock_out => clock_divided_1
	--);

	uut5 : onetosix_counter
	port map(
		clk => clock,
		reset => reset_button,
		Roll_in => Roll_out,
		counter_out => counter_one_out
	);

	uut6 : clock_divider
	generic map(
		FPGA_CLK_FREQUENCY => 100000000,
		REQUIRED_FREQUENCY => 50000000
	)
	port map(
		clock_in => clock,
		clock_out => clock_divided_2
	);

	uut7 : onetosix_counter
	port map(
		clk => clock_divided_2,
		reset => reset_button,
```

```
                Roll_in => Roll_out,
                counter_out => counter_two_out
        );

        uut8 : Adder
        port map(
                A => counter_one_out,
                B => counter_two_out,
                SUM => Sum_out
        );

        uut9 : point_register
        port map(
                clk => clock,
                Sp_in => Sp_out,
                SUM_IN => Sum_out,
                point_register_value => point_register_val
        );

        uut10 : comparator
        port map(
                clk => clock,
                sum_in => Sum_out,
                point_register_value => point_register_val,
                Eq => Eq_out
        );

        uut11 : Test_Logic
        Port map(
                clk   => clock,
                sum_in   => Sum_out,
                D7    => D7_out,
                D711  => D711_out,
                D2312 => D2312_out
        );

        uut12 : bcdto7seg
         Port map (
                        clock => clock,
                        bcd_in_0  => counter_one_out,
                        bcd_in_1  => counter_two_out,
                        cathodes => cathodes_out,
                        anodes => anodes_out
         );


end Behavioral;
```

Question 3:

The Constraint Requirements for the FPGA of Nexys 3 are as follows:

```
## This file is a general .ucf for Nexys3 rev B board
## To use it in a project:
## - remove or comment the lines corresponding to unused pins
## - rename the used signals according to the project

##Clock signal
Net "clock" LOC=V10 | IOSTANDARD=LVCMOS33;


## 7 segment display
Net "cathodes_out(0)" LOC = T17 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L51P_M1DQ12, Sch name = CA
Net "cathodes_out(1)" LOC = T18 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L51N_M1DQ13, Sch name = CB
Net "cathodes_out(2)" LOC = U17 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L52P_M1DQ14, Sch name = CC
Net "cathodes_out(3)" LOC = U18 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L52N_M1DQ15, Sch name = CD
Net "cathodes_out(4)" LOC = M14 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name = IO_L53P, Sch
name = CE
Net "cathodes_out(5)" LOC = N14 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name = IO_L53N_VREF,
Sch name = CF
Net "cathodes_out(6)" LOC = L14 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name = IO_L61P, Sch
name = CG
#Net "cathodes_out(7)" LOC = M13 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name = IO_L61N, Sch
name = DP

Net "anodes_out(0)" LOC = N16 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L50N_M1UDQSN, Sch name = AN0
Net "anodes_out(1)" LOC = N15 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L50P_M1UDQS, Sch name = AN1
Net "anodes_out(2)" LOC = P18 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L49N_M1DQ11, Sch name = AN2
Net "anodes_out(3)" LOC = P17 | IOSTANDARD = LVCMOS33; #Bank = 1, pin name =
IO_L49P_M1DQ10, Sch name = AN3

## Leds
Net "Win" LOC = U16 | IOSTANDARD = LVCMOS33; #Bank = 2, pin name = IO_L2P_CMPCLK, Sch name
= LD0
Net "Lose" LOC = V16 |  IOSTANDARD = LVCMOS33; #Bank = 2, pin name = IO_L2N_CMPMOSI, Sch
name = LD1


## Switches
```

```
Net "Rb" LOC = T10 | IOSTANDARD = LVCMOS33; #Bank = 2, pin name = IO_L29N_GCLK2, Sch name = SW0
Net "Reset" LOC = T9 | IOSTANDARD = LVCMOS33; #Bank = 2, pin name = IO_L32P_GCLK29, Sch name = SW1
```

These includes 2 switches, 2LED and 2 seven segments which are refreshing at 1ms response time.