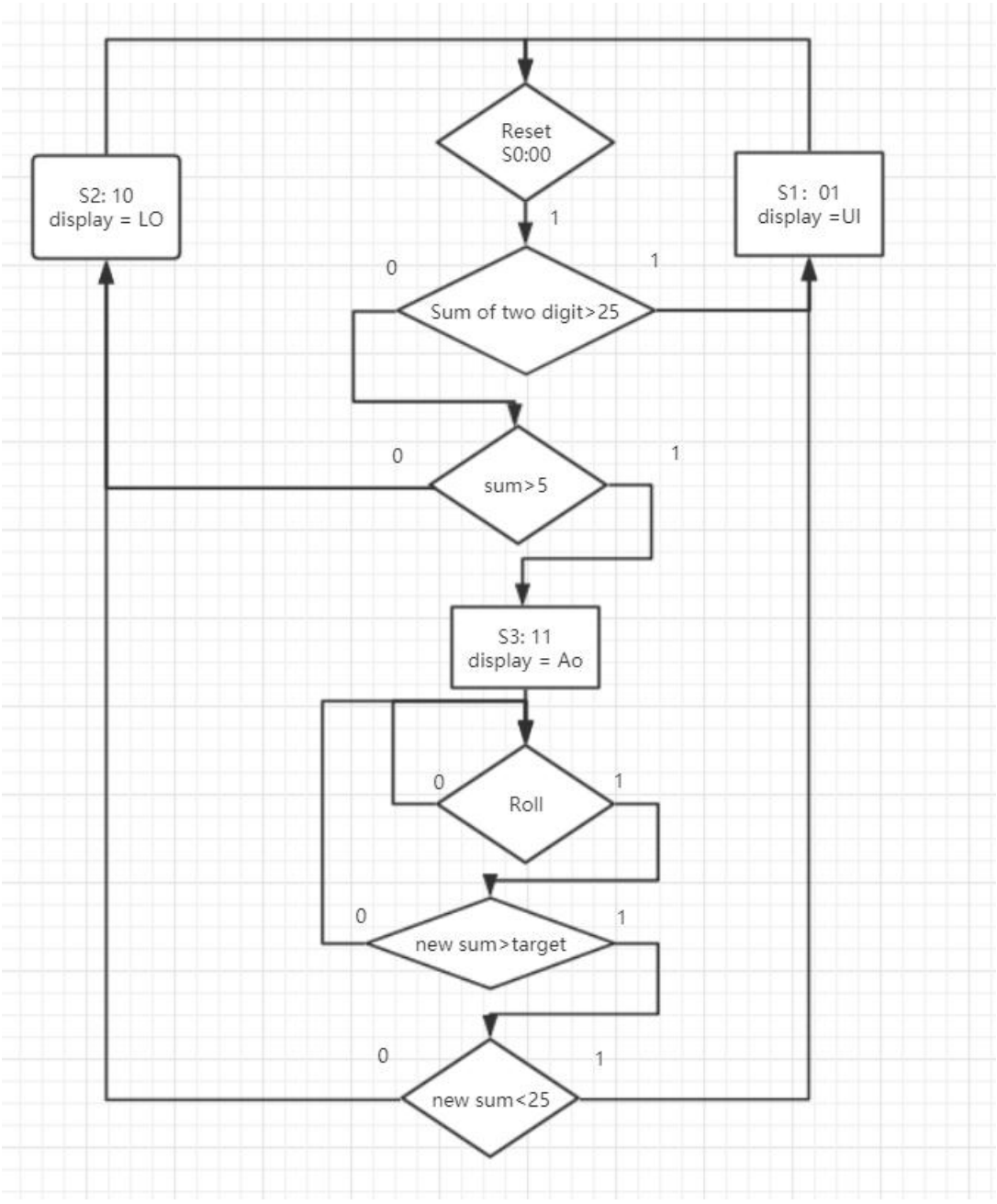
Lab5 Notebook

--Jiajun

1) Comment on my FSM design

For this game, I define totally 4 states. The first state was the reset state, I define it as S0 with 2-bit representation 00. Depend on the target after roll, there would be three different states. Win: state S1, denoted as 01. Lose: state S2, denoted as 10. Medium state: S3, denoted as 11. The state machine chart is shown as followed:



In my code design, I used two always block method to construct the FSM. The first always block was always @(*), with case (state). Depending on the target, roll, or sum condition, it will update the next_state value. Naturally, the next always block is used to iterate “state” to “nextstate”. The FSM part will be executed well with timeConst 50, and I apply a “flag” register, which can prevent it detect the roll signal twice in one press. Besides these always block, we can continue applying the always used in the lfsr display lab. With high frequency, the segment would display random number on the last two digits, display game result on the first two digits at the same time visually. The RTL file and the testbench file would be shown as followed.

2） RTL file

Initialization:

```
module lab4_top(


    output [7:0] q,


    output reg [6:0] seven_seg1,
    output reg [3:0] Activate,

    input clk,
    input roll,
    input start

    );


    reg [7:0] a = 8'b00000001;
    wire cout;   //Used to connect the clock_divider and LFSR
    wire cout1;
    wire cout2;
    wire [6:0] seven_display;
    reg [1:0] counter = 2'b00;
    reg [3:0] display = 3'b000;
    reg flag = 1;

    reg [4:0] target;
    parameter S0 = 2'b00, S1 = 2'b01,S2 = 2'b10, S3 = 2'b11;

    reg [1:0] state = 2'b00;
    reg [1:0] next_state = 2'b00;
    wire [4:0] sum;
```

Apply the module used before

```
clock_divider CDIV(cout,clk,80);
lfsr LFSR(q,a,start,cout);

clock_divider CDIV1(cout1,clk,10);

clock_divider CDIV2(cout2,clk,40);
```

First always block used for segment display

```
always @(posedge cout1)
begin
   if (counter > 2'b11) counter <= 2'b00;
   counter <= counter+1;
end

always @(*)
begin
    case(counter)
    2'b00: begin
        Activate = 4'b0111;
        // activate LED1 and Deactivate LED2, LED3, LED4

        case (state)
            S0:
                seven_seg1 = 7'b1111111;
            S1:
                seven_seg1 = 7'b1000001;
            S2:
                seven_seg1 = 7'b1110001;
            S3:
                seven_seg1 = 7'b0001000;

        endcase
        end
    2'b01: begin
        Activate = 4'b1011;
        // activate LED2 and Deactivate LED1, LED3, LED4

        case (state)
            S0:
                seven_seg1 = 7'b1111111;
            S1:
                seven_seg1 = 7'b1001111;
            S2:
                seven_seg1 = 7'b0000001;
            S3:
                seven_seg1 = 7'b1100010;
        endcase
        end
```

```
    2'b10: begin
        Activate = 4'b1101;
        // activate LED3 and Deactivate LED2, LED1, LED4
        display = q[7:4];
        seven_seg1 = seven_display;

        end
    2'b11: begin
        Activate = 4'b1110;
        // activate LED4 and Deactivate LED2, LED3, LED1
        display = q[3:0];
        seven_seg1 = seven_display;
        end

    endcase
end
```

Second always block used to determine the next state (Mealy machine)

```
led_segment LS(display,seven_display);
assign sum = q[7:4]+q[3:0];

always @(*) //q,state?
begin

    case(state)
        S0: begin

            if (roll==1) begin target = q[7:4]+q[3:0]; end

            if ( target >= 8'b00011001 & roll ==1) begin
                next_state = S1;
            end

            else if (target <= 8'b00000101 & roll ==1) begin
                next_state = S2;
            end

            else begin
                if(roll ==1)
                next_state = S3;
            end
        end

        S1:
            next_state = S1;

        S2:
            next_state = S2;
```

```
        S3: begin
        if (flag == 1) begin
            if ( sum <=target & roll == 1) begin
                next_state = S3;
                flag = 0;
            end
            else if (sum> target & sum<25 &roll == 1) begin
                next_state = S1;
                flag = 0;
            end
            else if (sum>target & sum>=25 &roll == 1) begin
                next_state = S2;
                flag = 0;
            end

        end

            else next_state = next_state;

        end
    endcase

    if (roll ==0) flag = 1;

end
```

Last always statement used to switch the state to next state

```
    always @ (posedge cout2)
    begin
        if (start == 1'b1) begin
            state = S0;
        end

        else begin

        state = next_state;;
        end

    end

endmodule
```

2b) Testbench file

```
module lab4_toptb( );
wire [7:0] q;
reg clk;
reg roll,start;
wire [3:0] Activate;
wire [6:0] seven_seg1;


always #0.5 clk = ~clk;

initial begin

clk = 0;
start = 0;
start = #80 1;
start = #160 0;

roll = 0;
roll = #200 1;
roll = #220 0;
roll = #320 1;
roll = #340 0;

end

lab4_top utb(q,seven_seg1,Activate,clk,roll,start);
//led_segment LS(display,seven_seg1);



endmodule
```

3) Constraint file

```
set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]

set_property PACKAGE_PIN U16 [get_ports {q[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[0]}]
set_property PACKAGE_PIN E19 [get_ports {q[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[1]}]
set_property PACKAGE_PIN U19 [get_ports {q[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[2]}]
set_property PACKAGE_PIN V19 [get_ports {q[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[3]}]
set_property PACKAGE_PIN W18 [get_ports {q[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[4]}]
set_property PACKAGE_PIN U15 [get_ports {q[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[5]}]
set_property PACKAGE_PIN U14 [get_ports {q[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[6]}]
set_property PACKAGE_PIN V14 [get_ports {q[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[7]}]
set_property PACKAGE_PIN W7 [get_ports {seven_seg1[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[6]}]
set_property PACKAGE_PIN W6 [get_ports {seven_seg1[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[5]}]
set_property PACKAGE_PIN U8 [get_ports {seven_seg1[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[4]}]
set_property PACKAGE_PIN V8 [get_ports {seven_seg1[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[3]}]
set_property PACKAGE_PIN U5 [get_ports {seven_seg1[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[2]}]
set_property PACKAGE_PIN V5 [get_ports {seven_seg1[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[1]}]
set_property PACKAGE_PIN U7 [get_ports {seven_seg1[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_seg1[0]}]
set_property PACKAGE_PIN U2 [get_ports {Activate[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[0]}]
set_property PACKAGE_PIN U4 [get_ports {Activate[1]}]
set_property PACKAGE_PIN V4 [get_ports {Activate[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[2]}]
set_property PACKAGE_PIN W4 [get_ports {Activate[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[3]}]

#Buttons

set_property PACKAGE_PIN U18 [get_ports roll]
set_property IOSTANDARD LVCMOS33 [get_ports roll]
set_property PACKAGE_PIN T18 [get_ports start]
set_property IOSTANDARD LVCMOS33 [get_ports start]

create_clock -period 20.000 -name clk -waveform {0.000 10.000}
set_input_delay -clock [get_clocks clk] 0.000 [get_ports -filter { NAME =~ "*" && DIRECTION == "IN" }]
set_output_delay -clock [get_clocks clk] 0.000 [get_ports -filter { NAME =~ "*" && DIRECTION == "OUT" }]
```

4) Report a) Utilization

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	159	0	20800	0.76
LUT as Logic	159	0	20800	0.76
LUT as Memory	0	0	9600	0.00
Slice Registers	414	0	41600	1.00
Register as Flip Flop	402	0	41600	0.97
Register as Latch	12	0	41600	0.03
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

2. Memory

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	0	0	50	0.00
RAMB36/FIFO*	0	0	50	0.00
RAMB18	0	0	100	0.00

* Note: Each Block RAM Tile only has one FIFO logic available and

3. DSP

Site Type	Used	Fixed	Available	Util%
DSPs	0	0	90	0.00

5. Clocking

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	1	0	32	3.13
BUFIO	0	0	20	0.00
MMCME2_ADV	0	0	5	0.00
PLLE2_ADV	0	0	5	0.00
BUFMRCE	0	0	10	0.00
BUFHCE	0	0	72	0.00
BUFR	0	0	20	0.00

b) Power report

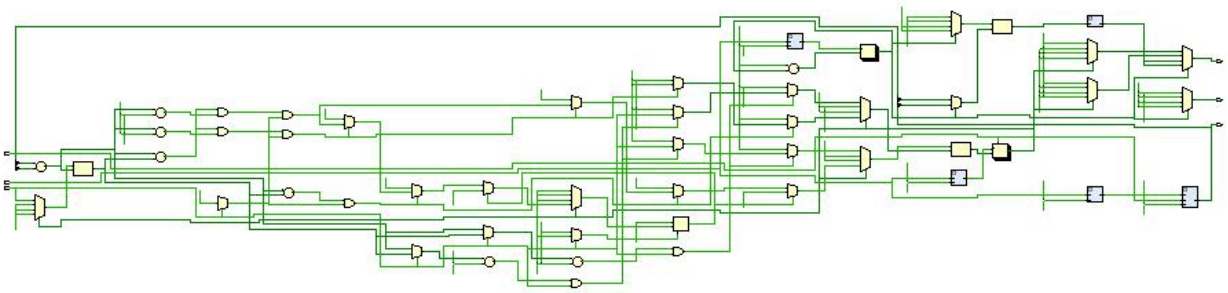
```
1. Summary
----------

+--------------------------+--------+
| Total On-Chip Power (W)  | 0.104  |
| Dynamic (W)              | 0.032  |
| Device Static (W)        | 0.072  |
| Effective TJA (C/W)      | 5.0    |
| Max Ambient (C)          | 84.5   |
| Junction Temperature (C) | 25.5   |
| Confidence Level         | Medium |
| Setting File             | ---    |
| Simulation Activity File | ---    |
| Design Nets Matched      | NA     |
+--------------------------+--------+
```

C) Timing-report

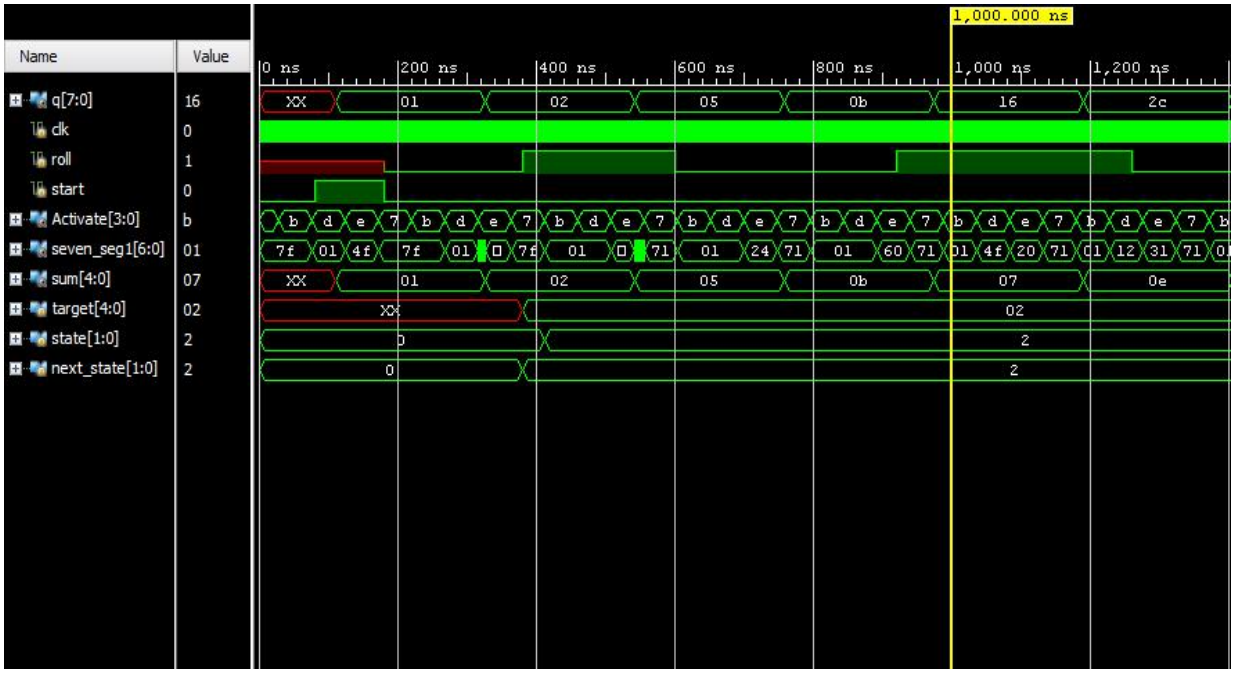
```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)
    -------      -------  ---------------------  -------------------      -------      -------
      3.547        0.000                      0                 1062        0.133        0.000


 THS Failing Endpoints  THS Total Endpoints     WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoint
 ---------------------  -------------------     --------     --------  ----------------------  -------------------
                     0                 1062        4.500        0.000                       0                   39


Max Delay Paths
--------------------------------------------------------------------------------
Slack (MET) :             3.547ns  (required time - arrival time)
  Source:                 CDIV/count0_reg[29]/C
                            (rising edge-triggered cell FDRE clocked by sys_clk_pin  {rise@0.000ns fall@5.000ns period=10.000ns})
  Destination:            CDIV/count3_reg[0]/R
                            (rising edge-triggered cell FDRE clocked by sys_clk_pin  {rise@0.000ns fall@5.000ns period=10.000ns})
  Path Group:             sys_clk_pin
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            10.000ns  (sys_clk_pin rise@10.000ns - sys_clk_pin rise@0.000ns)
  Data Path Delay:        5.885ns  (logic 0.952ns (16.178%)  route 4.933ns (83.822%))
  Logic Levels:           4  (LUT4=2 LUT5=2)
  Clock Path Skew:        -0.009ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    4.789ns = ( 14.789 - 10.000 )
    Source Clock Delay      (SCD):    5.072ns
    Clock Pessimism Removal (CPR):    0.274ns
  Clock Uncertainty:      0.035ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.000ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns
```

	(clock sys_clk_pin rise edge)				
		0.000	0.000	r	
W5		0.000	0.000	r	clk (IN)
	net (fo=0)	0.000	0.000		clk
W5	IBUF (Prop_ibuf_I_O)	1.458	1.458	r	clk_IBUF_inst/O
	net (fo=1, routed)	1.967	3.425		clk_IBUF
BUFGCTRL_X0Y0	BUFG (Prop_bufg_I_O)	0.096	3.521	r	clk_IBUF_BUFG_inst/O
	net (fo=390, routed)	1.551	5.072		CDIV/clk_IBUF_BUFG
SLICE_X15Y23	FDRE			r	CDIV/count0_reg[29]/C
SLICE_X15Y23	FDRE (Prop_fdre_C_Q)	0.456	5.528	r	CDIV/count0_reg[29]/Q
	net (fo=2, routed)	1.102	6.631		CDIV/count0_reg[29]
SLICE_X14Y19	LUT4 (Prop_lut4_I2_O)	0.124	6.755	r	CDIV/count0[0]_i_12__1/O
	net (fo=1, routed)	0.454	7.209		CDIV/count0[0]_i_12__1_n_0
SLICE_X13Y19	LUT5 (Prop_lut5_I4_O)	0.124	7.333	f	CDIV/count0[0]_i_4__1/O
	net (fo=1, routed)	0.807	8.140		CDIV/count0[0]_i_4__1_n_0
SLICE_X13Y19	LUT4 (Prop_lut4_I1_O)	0.124	8.264	f	CDIV/count0[0]_i_1__1/O
	net (fo=68, routed)	1.618	9.882		CDIV/count0[0]_i_1__1_n_0
SLICE_X13Y14	LUT5 (Prop_lut5_I1_O)	0.124	10.006	r	CDIV/count3[0]_i_1__1/O
	net (fo=32, routed)	0.951	10.957		CDIV/clear
SLICE_X14Y9	FDRE			r	CDIV/count3_reg[0]/R
	(clock sys_clk_pin rise edge)				
		10.000	10.000	r	
W5		0.000	10.000	r	clk (IN)
	net (fo=0)	0.000	10.000		clk
W5	IBUF (Prop_ibuf_I_O)	1.388	11.388	r	clk_IBUF_inst/O
	net (fo=1, routed)	1.862	13.250		clk_IBUF
BUFGCTRL_X0Y0	BUFG (Prop_bufg_I_O)	0.091	13.341	r	clk_IBUF_BUFG_inst/O
	net (fo=390, routed)	1.448	14.789		CDIV/clk_IBUF_BUFG
SLICE_X14Y9	FDRE			r	CDIV/count3_reg[0]/C
	clock pessimism	0.274	15.063		
	clock uncertainty	-0.035	15.028		
SLICE_X14Y9	FDRE (Setup_fdre_C_R)	-0.524	14.504		CDIV/count3_reg[0]
	required time		14.504		

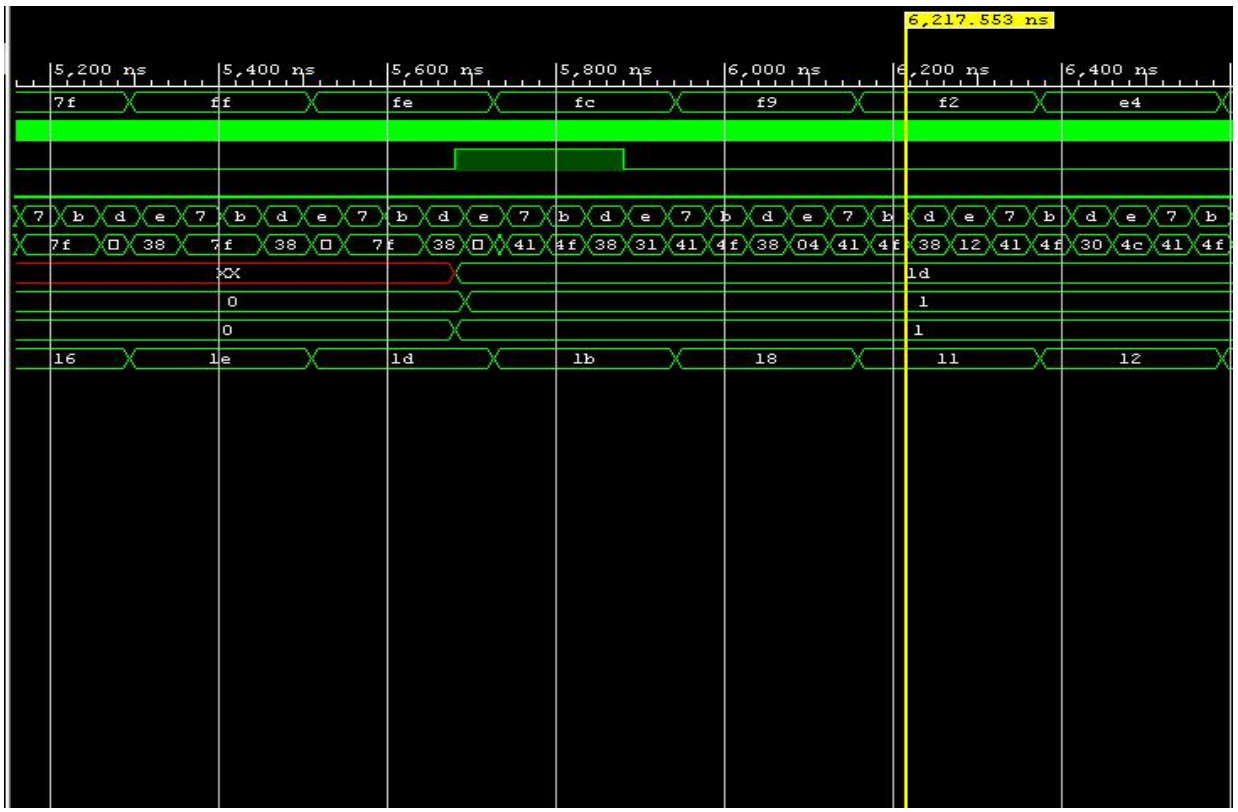
D) Schematic



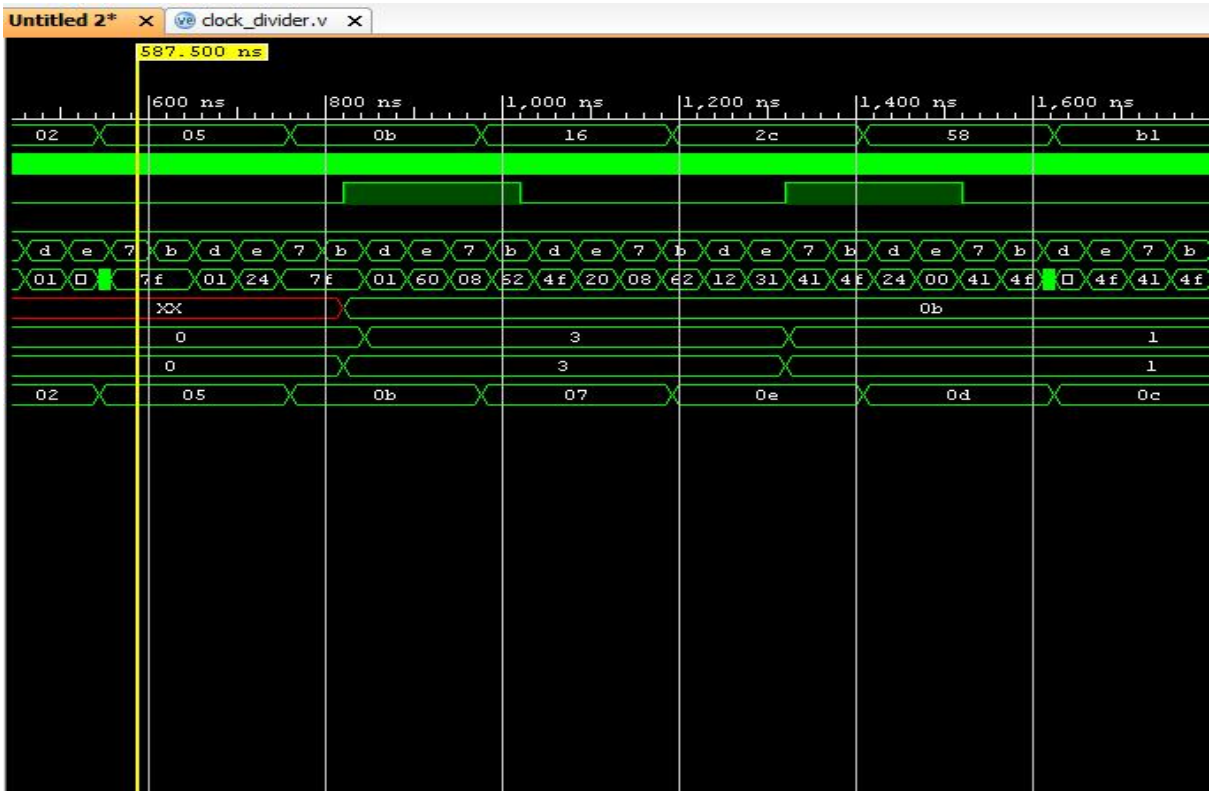
E)Waveform



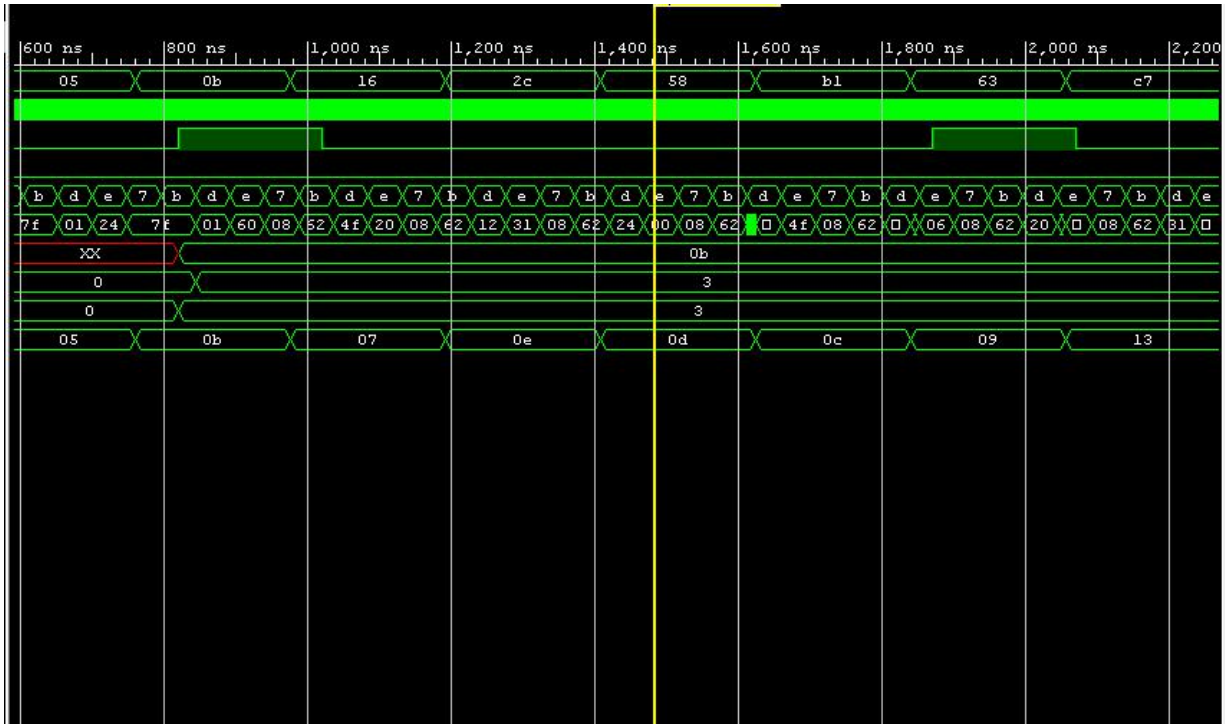
Roll was pressed first, since target is 2<5, then state and next state are 2, which is the lose case



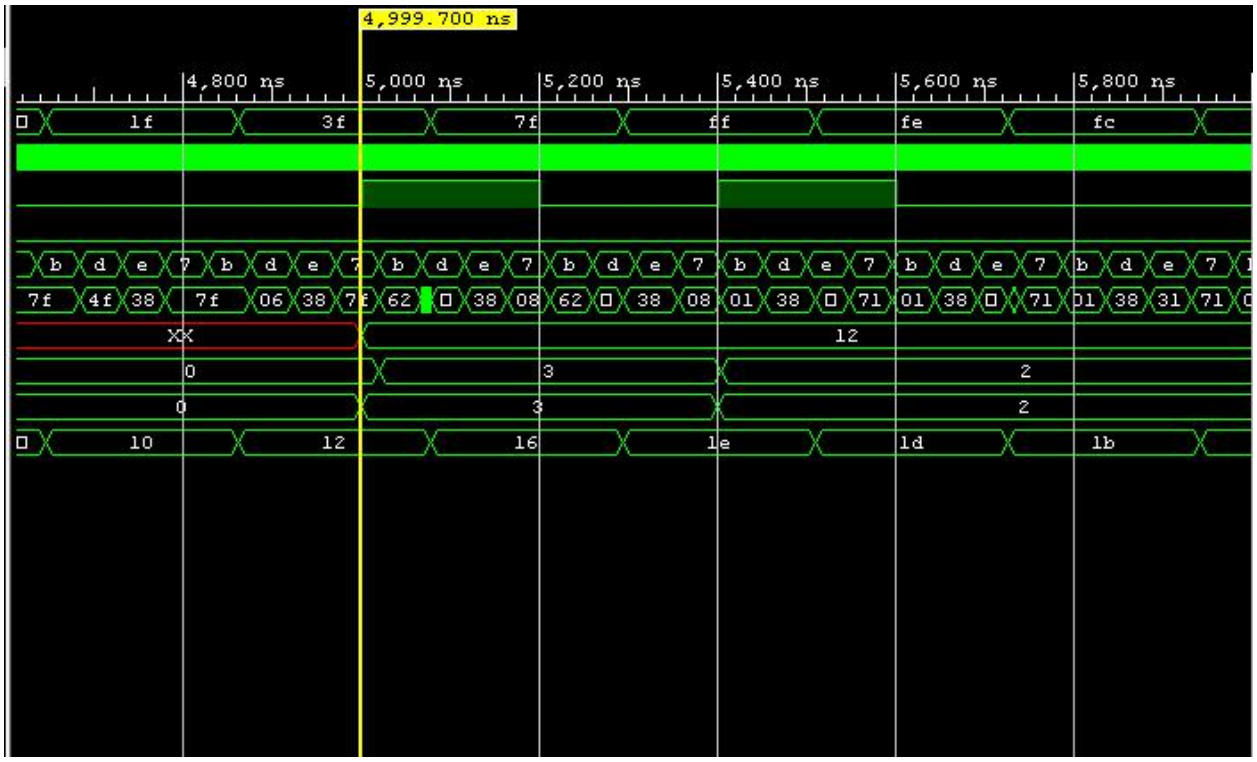
First roll was pressed. Since target = f+e > 25, the player win directly and enter the S1 (Win state)



After the first roll, the target is 0b, it enter the S3 (Ao state). When the second roll was pressed the sum is 0e. Since 0b<0e<25, the player win.



After the first roll, the target is 0b. It enter the S3 (Ao) state. When the second roll was pressed, the sum is 9. Since 9<0b, the state will stay at S3



After the first roll, the target was 3+f = 12 , it enter the S3(Ao state). After the second roll the target was f+f = 1e. Since 1e>25, the player lose, and it enter the lose state (S2).