Lab4 Notebook

--Jiajun

1) Comment on LFSR & 7-segment LED display.

Using for flip-flop outputs XORed together and fed back to the input of the shift register, we could construct the Linear Feedback Shift Register. The LFSR is used to generate the pseudorandom number. In this experiment, for the output q with eight bits, the feedback input would be din = q[1]^q[2]^q[3]^q[7], the other bit just shift to left. What's more, we need a clock divider to set up the appropriate time. Instead of using the big counter(about 27 bit), here we could implement it with four small integer. The calculation would be

```
Output Frequency = 100000000/(2*(TIMECONST^4))
```

In order to make time make the module reuseable, we will set TIMECONST as a parameter which can be adjusted easily for different frequency clock.

For the seven segment design, we first need to convert the display number to the corresponding segment. So we first create a module, the module would transfer the 4-bit input into the 7 segment output. Since for the 7-segment display, the board could only show one digit each time, this would be controlled by the 4-bit activate, in which "1110" only enable the forth segment, "1101" would only enable the third segment and so on. However, actually human-eye would consider the high-frequency display as the continuous image, thus we just need to set up a propriate clock and change the activate number between "1101" and "1110", we could achieve the function. In my code, I add "1011" and "0111" for the further use. It would be shown on the testbench. The most impressive process in my debug is the usage of module. Different from the "function" we used in other program language, one module represents for the basic design block, it is not allowed to instantiate the module repeatedly in the always statement. The solution is, we could use more registers to store the corresponding variable, and instantiate it outside the always statement. Last but not least, for the 8 switches input, we just need to set the seed as input and modify the constrain file like the lab2.

2) (a) RTL file (Time_divider, LFSR, 7-segement display, top mpdule

```
module lfsr(
    output reg [7:0] q,
    input [7:0] seed,
    input rst,
    input clock
    );

    assign din = q[1]^q[2]^q[3]^q[7];

    always@(posedge clock)
    begin
        if (rst)
            q <= seed;

        else
            begin
                q[0] <= din;
                q[1] <= q[0];
                q[2] <= q[1];
                q[3] <= q[2];
                q[4] <= q[3];
                q[5] <= q[4];
                q[6] <= q[5];
                q[7] <= q[6];
            end
    end


endmodule
```

Figure 1 LFSR design

```
module clock_divider(cout, cin);
input cin;
output cout;
parameter timeconst = 60;//constant
   integer count0;
   integer count1;
   integer count2;
   integer count3;
   reg d;
   reg cout;
```

```
    initial begin
    count0=0;
    count1=0;
    count2=0;
    count3=0;
    d = 0;
    end
    always @ (posedge cin )
    begin
    count0 <= (count0 + 1); if
    ((count0 == timeconst))
    begin
    count0 <= 0;
    count1 <= (count1 + 1);
    end
    else if ((count1 == timeconst))
    begin
    count1 <= 0;
    count2 <= (count2 + 1);
    end
    else if ((count2 == timeconst))
    begin
    count2 <= 0;
    count3 <= (count3 + 1);
    end
    else if ((count3 == timeconst))
    begin
    count3 <= 0;
    d <= ~ (d);
    end
    cout <= d;
    end // end always
endmodule
```

```
//
//////////////////////////////////////////////////////////////////////////////////


module led_segment(
    input [3:0] digit,
    output reg [6:0] seven_segl
);

    always @(digit)
    begin
        //activate <= 4'b1110;
        case (digit)
             4'b0000 : seven_segl <= 7'b0000001;
             4'b0001 : seven_segl <= 7'b1001111;
             4'b0010 : seven_segl <= 7'b0010010;
             4'b0011 : seven_segl <= 7'b0000110;
             4'b0100 : seven_segl <= 7'b1001100;
             4'b0101 : seven_segl <= 7'b0100100;
             4'b0110 : seven_segl <= 7'b0100000;
             4'b0111 : seven_segl <= 7'b0001111;
             4'b1000 : seven_segl <= 7'b0000000;
             4'b1001 : seven_segl <= 7'b0000100;
             4'b1010 : seven_segl <= 7'b0000010;
             4'b1011 : seven_segl <= 7'b1100000;
             4'b1100 : seven_segl <= 7'b0110001;
             4'b1101 : seven_segl <= 7'b1000010;
             4'b1110 : seven_segl <= 7'b0110000;
             4'b1111 : seven_segl <= 7'b0111000;
        endcase
    end

endmodule
```

Figure 2 7-segment display module (Convert the 4-bit input to 7-bit output)

```
module lab4_top(
   output [7:0] q,
   input clk,
   input rst,
   input [7:0] a,
   output [6:0] seven_segl,
   output reg [3:0] Activate
    );

    wire cout;   //Used to connect the clock_divider and LFSR
    wire cout1;
    reg [1:0] counter = 2'b00;
    reg [3:0] display;


    clock_divider CDIV(cout,clk,65);
    lfsr LFSR(q,a,rst,cout);



    clock_divider CDIV1(cout1,clk,6);

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

        case(counter)
        2'b00: begin
            Activate = 4'b1111;
            // activate LED1 and Deactivate LED2, LED3, LED4
            end
        2'b01: begin
```

```
            end
        2'b10: begin
            Activate = 4'b1101;
            // activate LED3 and Deactivate LED2, LED1, LED4
            display = q[7:4];
            end
        2'b11: begin
            Activate = 4'b1110;
            // activate LED4 and Deactivate LED2, LED3, LED1
            display = q[3:0];
            end
        endcase
    end


    led_segment LS(display,seven_segl);


endmodule
```

Figure 3 lab4_top file (The main module in this lab)

(b) Testbench showing

C:/Users/guan0138/Desktop/project_4/project_4.srcs/sim_1/new/lab4_toptb.v

```
24  `timescale 1ns / 100ps
25  wire  [7:0] q;
26  reg clk;
27  reg clk_2;
28  reg[7:0] seed;
29  reg rst;
30  reg[1:0] counter = 0;
31  reg[3:0] Activate = 0;
32  reg[3:0] display = 0;
33  lfsr lf(q,seed,rst,clk);
34
35  always #30 clk = ~clk;
36
37  always #3 clk_2 = ~clk_2;
38
39  initial begin
40  clk = 0;
41  clk_2 = 0;
42  seed = 8'b00000001;
43  rst =  1;
44  rst = #40 0;
45  end
46
47      always @(posedge clk_2)
48      begin
49         if (counter > 2'b11) counter = 2'b00;
50         counter =counter+1;
51
52          case(counter)
53          2'b00: begin
54              Activate = 4'b0111;
55              // activate LED1 and Deactivate LED2, LED3, LED4
56              end
```

```
        2'b10: begin
            Activate = 4'b1101;
            // activate LED3 and Deactivate LED2, LED1, LED4
            display = q[7:4];
            end
        2'b11: begin
            Activate = 4'b1110;
            // activate LED4 and Deactivate LED2, LED3, LED1
            display = q[3:0];
            end
        endcase
    end

//led_segment LS(display,seven_seg1);


endmodule
```

Figure 4 Testbench file

3. New Constrain file

```
# contraints file (.xdc file) for lab 4
# Clock signal
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_port
# leds
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]}]
```

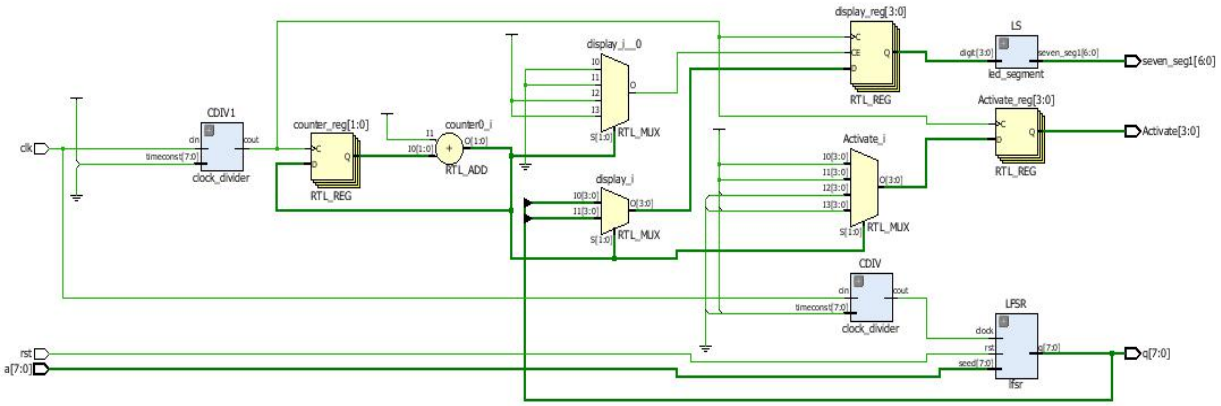
```
set_property PACKAGE_PIN U2 [get_ports {Activate[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[0]}]
set_property PACKAGE_PIN U4 [get_ports {Activate[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[1]}]
set_property PACKAGE_PIN V4 [get_ports {Activate[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[2]}]
set_property PACKAGE_PIN W4 [get_ports {Activate[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Activate[3]}]

set_property PACKAGE_PIN V17 [get_ports {a[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]
set_property PACKAGE_PIN V16 [get_ports {a[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]
set_property PACKAGE_PIN W16 [get_ports {a[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
set_property PACKAGE_PIN W17 [get_ports {a[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
set_property PACKAGE_PIN W15 [get_ports {a[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[4]}]
set_property PACKAGE_PIN V15 [get_ports {a[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[5]}]
set_property PACKAGE_PIN W14 [get_ports {a[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[6]}]
set_property PACKAGE_PIN W13 [get_ports {a[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[7]}]
#Buttons
set_property PACKAGE_PIN U18 [get_ports rst]
set_property IOSTANDARD LVCMOS33 [get_ports rst]



    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. New report a) schematic



b) Utilization report

2. Slice Logic Distribution

Site Type	Used	Fixed	Available	Util%
Slice	94	0	8150	1.15
SLICEL	82	0		
SLICEM	12	0		
LUT as Logic	100	0	20800	0.48
using O5 output only	0			
using O6 output only	91			
using O5 and O6	9			
LUT as Memory	0	0	9600	0.00
LUT as Distributed RAM	0	0		
LUT as Shift Register	0	0		
LUT Flip Flop Pairs	19	0	20800	0.09
fully used LUT-FF pairs	2			
LUT-FF pairs with one unused LUT	13			
LUT-FF pairs with one unused Flip Flop	14			
Unique Control Sets	12			

* Note: Review the Control Sets Report for more information regarding control sets.

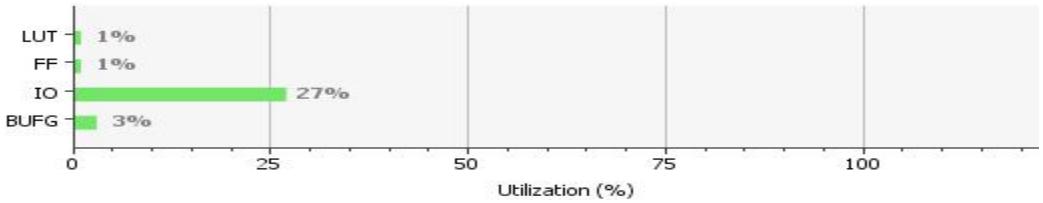
3. Memory

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 therefore can accommodate only one FIFO36E1 or

4. DSP

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



5. IO and GT Specific

Site Type	Used	Fixed	Available	Util%
Bonded IOB	29	29	106	27.36
IOB Master Pads	13			
IOB Slave Pads	15			
Bonded IPADs	0	0	10	0.00
Bonded OPADs	0	0	4	0.00
PHY_CONTROL	0	0	5	0.00
PHASER_REF	0	0	5	0.00
OUT_FIFO	0	0	20	0.00
IN_FIFO	0	0	20	0.00
IDELAYCTRL	0	0	5	0.00
IBUFDS	0	0	104	0.00
GTPE2_CHANNEL	0	0	2	0.00
PHASER_OUT/PHASER_OUT_PHY	0	0	20	0.00
PHASER_IN/PHASER_IN_PHY	0	0	20	0.00
IDELAYE2/IDELAYE2_FINEDELAY	0	0	250	0.00
IBUFDS_GTE2	0	0	2	0.00
ILOGIC	0	0	106	0.00
OLOGIC	0	0	106	0.00

C) Power report

1. Summary

Total On-Chip Power (W)	0.072
Dynamic (W)	0.000
Device Static (W)	0.072
Effective TJA (C/W)	5.0
Max Ambient (C)	84.6
Junction Temperature (C)	25.4
Confidence Level	Low
Setting File	---
Simulation Activity File	---
Design Nets Matched	NA

D) Timing report

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

WNS(ns)	TNS(ns)	TNS Failing Endpoints	TNS Total Endpoints	WHS(ns)	THS(ns)	THS Failing Endpoints	THS Total Endpoints	WPWS(ns)	TPWS(ns)	TPWS F
3.842	0.000	0	708	0.122	0.000	0	708	4.500	0.000	

Max Delay Paths

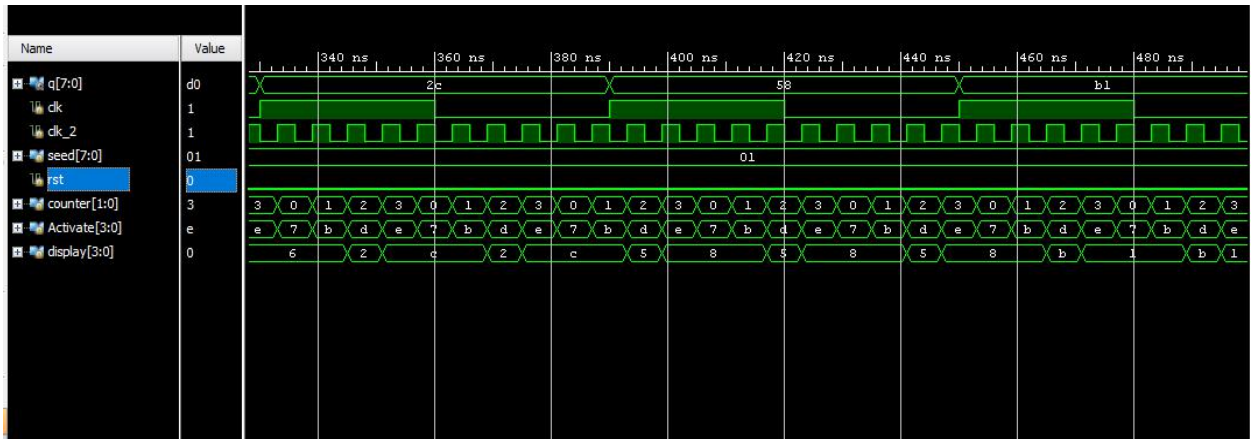
Slack (MET) : 3.842ns (required time - arrival time)
Source: CDIV1/count0_reg[2]/C
(rising edge-triggered cell FDRE clocked by sys_clk_pin {rise@0.000ns fall@5.000ns period=10.000ns})
Destination: CDIV1/count1_reg[0]/R
(rising edge-triggered cell FDRE clocked by sys_clk_pin {rise@0.000ns fall@5.000ns 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: 5.561ns (logic 0.952ns (17.119%) route 4.609ns (82.881%))
Logic Levels: 4 (LUT2=1 LUT4=2 LUT5=1)
Clock Path Skew: -0.038ns (DCD - SCD + CPR)
Destination Clock Delay (DCD): 4.792ns = (14.792 - 10.000)
Source Clock Delay (SCD): 5.090ns
Clock Pessimism Removal (CPR): 0.260ns
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

Location	Delay type	Incr(ns)	Path(ns)		Netlist Resource(s)
	(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=260, routed)	1.560	5.080		CDIV1/clk_IBUF_BUFG

SLICE_X9Y5	FDRE			r	CDIV1/count0_reg[2]/C
SLICE_X9Y5	FDRE (Prop_fdre_C_Q)	0.456	5.546	f	CDIV1/count0_reg[2]/Q
	net (fo=2, routed)	1.267	6.813		CDIV1/count0_reg[2]
SLICE_X7Y9	LUT4 (Prop_lut4_I0_O)	0.124	6.937	r	CDIV1/count0[0]_i_11__0/O
	net (fo=1, routed)	0.433	7.370		CDIV1/count0[0]_i_11__0_n_0
SLICE_X7Y9	LUT5 (Prop_lut5_I4_O)	0.124	7.494	r	CDIV1/count0[0]_i_3__0/O
	net (fo=1, routed)	0.809	8.303		CDIV1/count0[0]_i_3__0_n_0
SLICE_X6Y9	LUT4 (Prop_lut4_I0_O)	0.124	8.427	f	CDIV1/count0[0]_i_1__0/O
	net (fo=68, routed)	0.904	9.331		CDIV1/count0[0]_i_1__0_n_0
SLICE_X12Y9	LUT2 (Prop_lut2_I1_O)	0.124	9.455	r	CDIV1/count1[0]_i_1__0/O
	net (fo=64, routed)	1.196	10.651		CDIV1/count1[0]_i_1__0_n_0
SLICE_X10Y5	FDRE			r	CDIV1/count1_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=260, routed)	1.451	14.792		CDIV1/clk_IBUF_BUFG
SLICE_X10Y5	FDRE			r	CDIV1/count1_reg[0]/C
	clock pessimism	0.260	15.052		
	clock uncertainty	-0.035	15.017		
SLICE_X10Y5	FDRE (Setup_fdre_C_R)	-0.524	14.493		CDIV1/count1_reg[0]
	required time		14.493		
	arrival time		-10.651		
	slack		3.842		

5) New workspace

a) Behaviour simulation



b) Post-implementation simulation

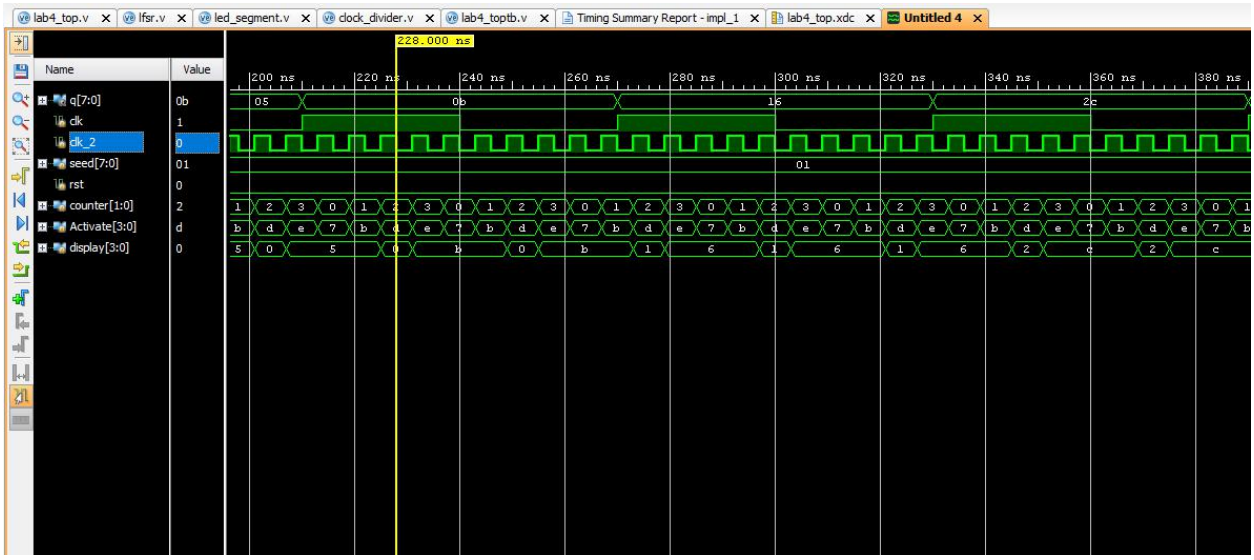


Figure 5 Post-simulation simulation waveform

6) Comparison table

	Led Design	7-segment design
Utilization	| Site Type | Used | Fixed | Available | Util% | | Slice LUTs* | 0 | 0 | 50 | 0.04 | | LUT as Logic | 8 | 0 | 20800 | 0.04 | The Memory, DSP ,h IO and GT Specific Clocking and Specific Feature are exactly the same	| Site Type | Used | Fixed | Available | Util% | | Slice LUTs* | 16 | 0 | 50| 0.08 | | LUT as Logic | 16 | 0 | 50| 0.08 |
Power	| Total On-Chip Power (W) | 0.074 | | Dynamic (W) | 0.008 | | Device Static (W) | 0.072 | The other attribute are the same, that is, the LED showing	| Total On-Chip Power (W) | 0.072 | | Dynamic (W) | 0.009 | | Device Static (W) | 0.072 |
Timing report	WNS(ns):3.021 WHS(ns): 2.277 The clock summary is the same Data Path Delay: 14.257ns (logic 6.079ns (43.815%) route 11.238ns (56.185%)) required time 14.257 arrival time -11.238	WNS(ns): 3.842 which is also the Slack MET (required time - arrival time) WHS(ns): 2.317 Data Path Delay: 14.493ns (logic 5.452ns (42.230%) route 10.651ns (57.770%)) required time 14.493 arrival time -10.651