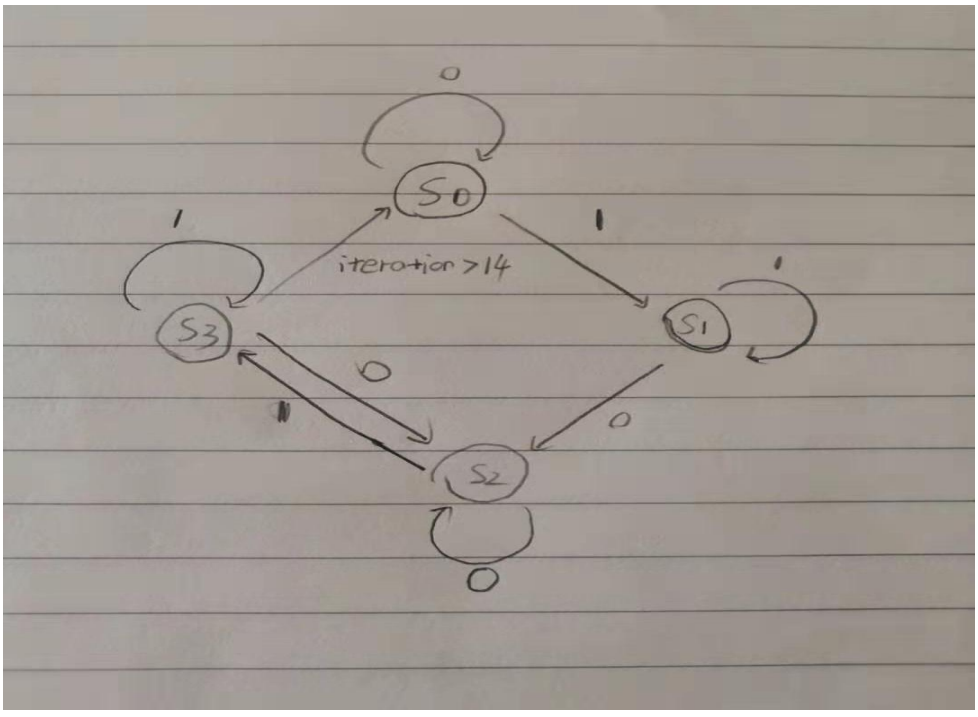
LAB 7 notebook

Jiajun, Guan

a). Comments on the storing and moving text design:

For the 16 bit CORDIC computer, the design is used to calculate the cosine and sine value of a given value. It will take as input a 16-bit signed binary fixed point number, corresponding to an angle in the range 0 to π/2. There are two methods available. The first method is calculated in parallel. It require at least 16 full adders at the same time, as well as a large shift register which do not map well into an FPGA. We used the bit-serial design with 3 sets of shift registers, serial adder-subtractor and two input multiplexers. X and y are coupled wire 13 input multiplexers to the serial adder-subtractor in the other register datapath. Z is input to the third serial adder-subtractor along with the corresponding value in TAN_ROM. In my program, the module cordic will be used to calculate the cosine and sine value given x0,y0,z0 in serial. Each time the cordic module will return a new value of x, and y, when the button is pressed, the new_x and new_y will be sent as the new input.

The top file will be in charge of the display and control. In my file, there are totally four states. The state diagram is shown as followed:



For the state S0, it is the initial state and do nothing. When pushed button is pressed, it entered state S1. For the S1, it will pass the initial value to the cordic module, in which x = 26dd, y = 0, z is read from the16 switches. After passing the value, the state machine enter S2 until the user release the button. In S2, cordic module will execute the calculation and return a new value of the x, y,z. It will be passed to the module again when enter S3 by pressing the button. When release the button, the state will enter S2 again, and the iteration number increase by 1. If there is more than 14 iteration, the state will finally enter S0 and start from beginning.

B) RTL file and testbench file 1.top file

```
module lab7_top(
        input clk,
        input pushed,
        input [15:0] initial_z,
        output [6:0] seven_segl,
        output [3:0] Activate
    );

    reg [15:0] x,y,z;
    wire [15:0] new_x,new_y,new_z;
    reg [3:0] iteration = 4'b0000;
    reg flag = 1'b1;

    parameter S0 = 2'b00, S1 = 2'b01,S2 = 2'b10, S3 = 2'b11;

    reg [2:0] state = 2'b00;
    reg [2:0] next_state = 2'b00;
    wire cout1;
    wire cout2;
    wire [3:0] Activate_display;
    wire [6:0] seven_display;

    clock_divider CDIV1(cout1,clk,30);
    clock_divider CDIV2(cout2,clk,10);


    led_segment LED7(cout2,new_x[15:12],new_x[11:8],new_x[7:4],new_x[3:0],seven_display,Activate_display);
    cordic COD(clk,new_x,new_y,new_z,x,y,z,iteration,~flag);
```

```
    always @ (posedge coutl)
    begin
        case(state)
        S0:begin
        end

        S1: begin

            if (flag==1)
                x <= 16'h26dd;
                y <= 0;
                z <= initial_z;
                iteration<=0;
                flag <= 0;
        end

        S2: begin
            flag <= 1;
        end

        S3: begin
            if (flag==1) begin
                x <= new_x;
                y <= new_y;
                z <= new_z;
                iteration <= iteration+1;
                flag <= 0;
            end
        end

        endcase

        if (iteration > 13)
        begin
            iteration <= 0;
            state <= S0;
        end

        else
            state <= next_state;
    end
```

```
    assign seven_segl = seven_display;
    assign Activate = Activate_display;

endmodule
```

2. The cordic module file:

```
`timescale 1ns/1ps

module cordic(clock,new_x,new_y,new_z,x0,y0,z0,iteration,start);

    input clock;
    input signed [15:0] x0,y0;
    input signed [15:0] z0;
    output signed [15:0] new_x,new_y,new_z;
    input [3:0] iteration;
    input start;
    reg [3:0] i = 4'b0000;
    reg signed [15:0] x,y,z;
    reg [15:0] tan_rom [15:0];
    reg addflag = 1;

    initial begin
    tan_rom[0] = 16'b0011001001000011;
    tan_rom[1] = 16'b0001110110101100;
    tan_rom[2] = 16'b0000111110101101;
    tan_rom[3] = 16'b0000011111110101;
    tan_rom[4] = 16'b0000001111111110;
    tan_rom[5] = 16'b0000000111111111;
    tan_rom[6] = 16'b0000000011111111;
    tan_rom[7] = 16'b0000000001111111;
    tan_rom[8] = 16'b0000000000111111;
    tan_rom[9] = 16'b0000000000011111;
    tan_rom[10] = 16'b0000000000001111;
    tan_rom[11] = 16'b0000000000000111;
    tan_rom[12] = 16'b0000000000000011;
    tan_rom[13] = 16'b0000000000000001;
    tan_rom[14] = 16'b0000000000000000;
    tan_rom[15] = 16'b0000000000000000;

    end
```

```
    wire z_signed;

    reg cx = 0;
    reg cy = 0;
    reg cz = 0;

    assign z_signed = z0[15];
```

```
always @(posedge clock)
begin
    if (start==1) begin
        if (addflag == 1)
            cx <= 0;
            cy <= 0;
            cz <= 0;
            addflag <= 0;end

    else if (start == 0)begin
        i <= 0;
        addflag <= 1; end

    if (addflag==0) begin
        if (i+iteration <16) begin
            x[i] <= x0[i]^y0[i+iteration]^cx;
            y[i] <= y0[i]^x0[i+iteration]^cy;
            z[i] <= z0[i]^(tan_rom[iteration][i])^cz;
```

cx <= (z_signed==0)? ((~x0[i])&y0[i+iteration])|(cx&y0[i+iteration])|(cx&(~x0[i])): (x0[i]&y0[i+iteration])|(cx&y0[i+iteration])|(cx&x0[i]);

cy <= (z_signed==0)? (y0[i]&x0[i+iteration])|(cy&x0[i+iteration])|(cy&y0[i]) :((~y0[i])&x0[i+iteration])|(cy&x0[i+iteration])|(cy&(~y0[i])) ;

cz<=(z_signed==0)?cz&(tan_rom[iteration][i]))|(cz&(~z0[i])):(z0[i]&(tan_rom[iteration][i]))|(cz&(tan_rom[iteration][i]))|(cz&z0[i]); end

```
        else begin
            x[i] <= x0[i]^y0[15]^cx;
            y[i] <= y0[i]^x0[15]^cy;
            z[i] <= z0[i]^(tan_rom[iteration][15])^cz;
```

cx <= (z_signed==0)? ((~x0[i])&y0[15])|(cx&y0[15])|(cx&~x0[i]) : (x0[i]&y0[15])|(cx&y0[15])|(cx&x0[i]) ;

cy <= (z_signed==0)? (y0[i]&x0[15])|(cy&x0[15])|(cy&y0[i]) :(~y0[i]&x0[15])|(cy&x0[15])|(cy&~y0[i]) ;

cz<=(z_signed==0)?(~z0[i]&(tan_rom[iteration][15]))|(cz&(tan_rom[iteration][15]))|(cz&~z0[i]):(z0[i]&(tan_rom[iteration][15]))|(cz&(tan_rom[iteration][15]))|(cz&z0[i]); end

```
            if (i<15) begin
                  i <= i + 1;  end
            else begin
                  i <= 15;     end
        end
    end

    assign new_x = x;
    assign new_y = y;
    assign new_z = z;

endmodule
```

3. Simulation testbench file

```
module lab7_test;

reg clk;
reg pushed;
reg [15:0] intial_z;
wire [6:0] seven_seg1;
wire [3:0] Activate;



initial
    begin
        clk = 0;
        forever
        #0.3 clk = ~clk;
    end


initial begin
    intial_z = 16'h0000;
    pushed = 0;
    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;
```

```
    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;

    pushed = #300 1;
    pushed = #200 0;
end

lab7_top LAB7(clk,pushed,intial_z,seven_segl,Activate);

endmodule
```

c) Constraint file

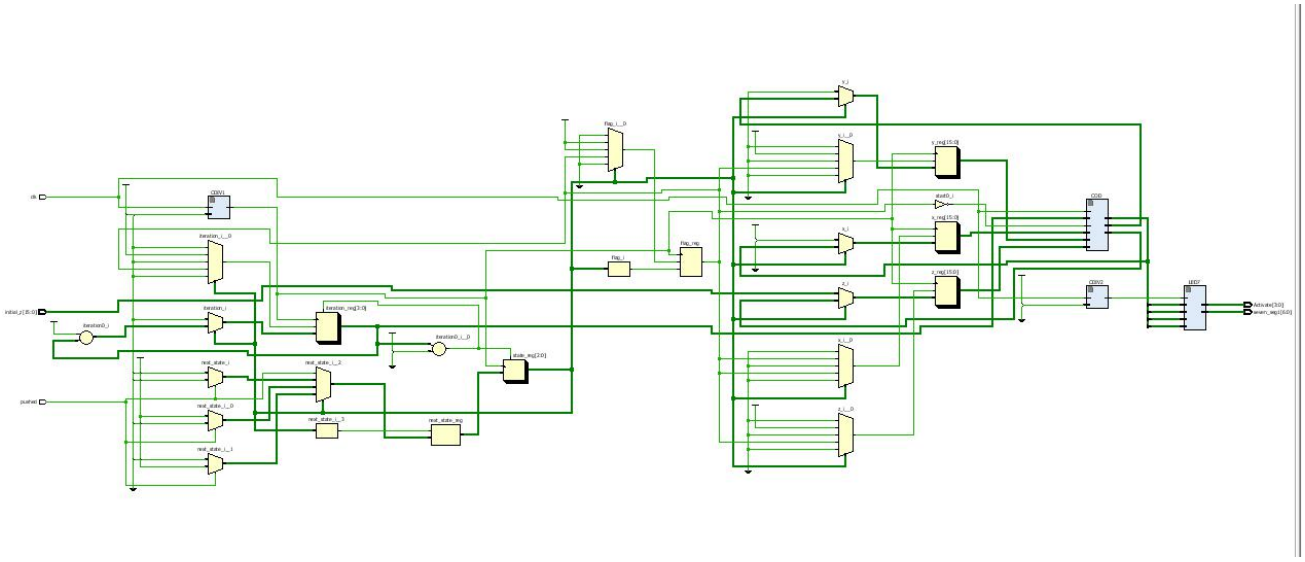
```
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
# leds
set_property PACKAGE_PIN V17 [get_ports {initial_z[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[0]}]
set_property PACKAGE_PIN V16 [get_ports {initial_z[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[1]}]
set_property PACKAGE_PIN W16 [get_ports {initial_z[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[2]}]
set_property PACKAGE_PIN W17 [get_ports {initial_z[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[3]}]
set_property PACKAGE_PIN W15 [get_ports {initial_z[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[4]}]
set_property PACKAGE_PIN V15 [get_ports {initial_z[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[5]}]
set_property PACKAGE_PIN W14 [get_ports {initial_z[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[6]}]
set_property PACKAGE_PIN W13 [get_ports {initial_z[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[7]}]
set_property PACKAGE_PIN V2 [get_ports {initial_z[8]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[8]}]
set_property PACKAGE_PIN T3 [get_ports {initial_z[9]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[9]}]
set_property PACKAGE_PIN T2 [get_ports {initial_z[10]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[10]}]
set_property PACKAGE_PIN R3 [get_ports {initial_z[11]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[11]}]
set_property PACKAGE_PIN W2 [get_ports {initial_z[12]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[12]}]
set_property PACKAGE_PIN U1 [get_ports {initial_z[13]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[13]}]
set_property PACKAGE_PIN T1 [get_ports {initial_z[14]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[14]}]
set_property PACKAGE_PIN R2 [get_ports {initial_z[15]}]
set_property IOSTANDARD LVCMOS33 [get_ports {initial_z[15]}]
```

```
set_property PACKAGE_PIN W7 [get_ports {seven_segl[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[6]}]
set_property PACKAGE_PIN W6 [get_ports {seven_segl[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[5]}]
set_property PACKAGE_PIN U8 [get_ports {seven_segl[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[4]}]
set_property PACKAGE_PIN V8 [get_ports {seven_segl[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[3]}]
set_property PACKAGE_PIN U5 [get_ports {seven_segl[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[2]}]
set_property PACKAGE_PIN V5 [get_ports {seven_segl[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[1]}]
set_property PACKAGE_PIN U7 [get_ports {seven_segl[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seven_segl[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]}]


#Buttons
set_property PACKAGE_PIN U18 [get_ports pushed]
set_property IOSTANDARD LVCMOS33 [get_ports pushed]



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" }]
```

D)

1.Schematic



2. Power report

Total On-Chip Power (W)	0.103
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	Low
Setting File	---
Simulation Activity File	---
Design Nets Matched	NA

3. Utilization report

1.1 Summary of Registers by Type

Total	Clock Enable	Synchronous	Asynchronous
0	_	-	-
0	_	-	Set
0	_	-	Reset
0	_	Set	-
0	_	Reset	-
0	Yes	-	-
0	Yes	-	Set
0	Yes	-	Reset
13	Yes	Set	-
360	Yes	Reset	-

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 a

3. DSP

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

5. Clocking

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	2	0	32	6.25
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

Design Timing Summary

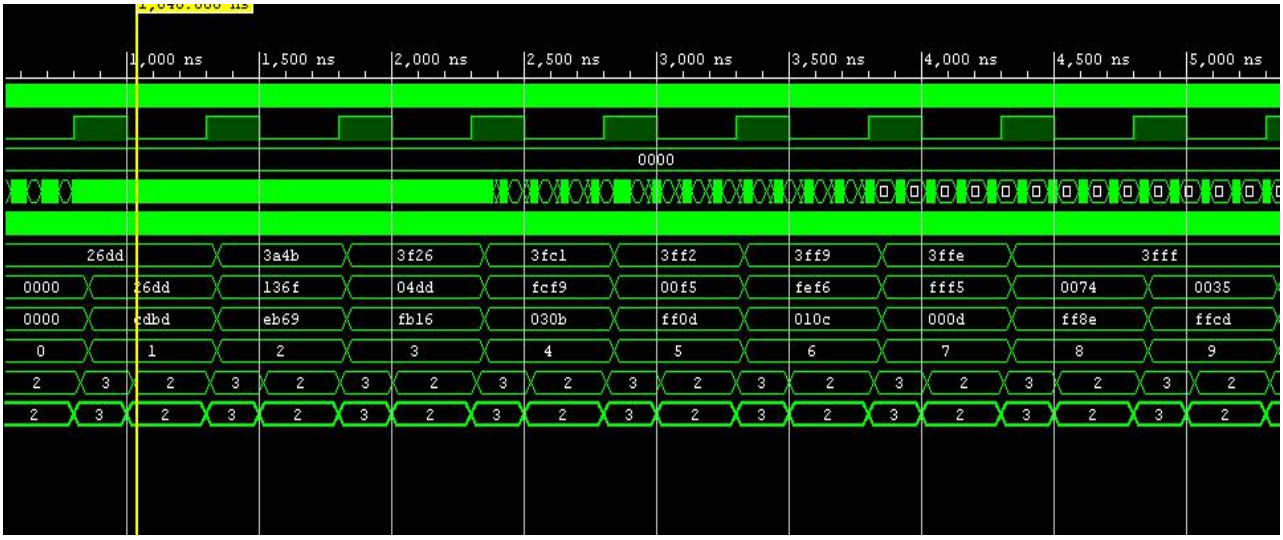
WNS(ns)	TNS(ns)	TNS Failing Endpoints	TNS Total Endpoints	WHS(ns)	THS(ns)	THS Failing Endpoints
-2.701	-18.318	7	778	0.131	0.000	0

THS Failing Endpoints	THS Total Endpoints	WPWS(ns)	TPWS(ns)	TPWS Failing Endpoints	TPWS Total Endpoints
0	778	4.500	0.000	0	317

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             3.781ns  (required time - arrival time)
  Source:                 CDIV1/count1_reg[1]/C
                            (rising edge-triggered cell FDRE clocked by sys_clk_pin  {rise@0.000ns fall@5.000ns period=10.000ns})
  Destination:            CDIV1/count3_reg[20]/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:        5.712ns  (logic 0.952ns (16.668%)  route 4.760ns (83.332%))
  Logic Levels:           4  (LUT4=2 LUT5=2)
  Clock Path Skew:        -0.043ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    4.764ns = ( 14.764 - 10.000 )
    Source Clock Delay      (SCD):    5.065ns
    Clock Pessimism Removal (CPR):    0.258ns
  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=316, routed)       1.544     5.065    CDIV1/clk_IBUF_BUFG
    SLICE_X47Y68         FDRE                                   r  CDIV1/count1_reg[1]/C
  -------------------------------------------------------------------    -------------------
    SLICE_X47Y68         FDRE (Prop_fdre_C_Q)       0.456     5.521 f  CDIV1/count1_reg[1]/Q
                         net (fo=2, routed)         1.098     6.619    CDIV1/count1_reg[1]
    SLICE_X45Y71         LUT4 (Prop_lut4_I0_O)      0.124     6.743 r  CDIV1/d_i_24/O
                         net (fo=1, routed)         0.665     7.408    CDIV1/d_i_24_n_0
    SLICE_X45Y71         LUT5 (Prop_lut5_I4_O)      0.124     7.532 r  CDIV1/d_i_14/O
                         net (fo=1, routed)         0.876     8.408    CDIV1/d_i_14_n_0
    SLICE_X44Y71         LUT4 (Prop_lut4_I0_O)      0.124     8.532 f  CDIV1/d_i_5/O
```

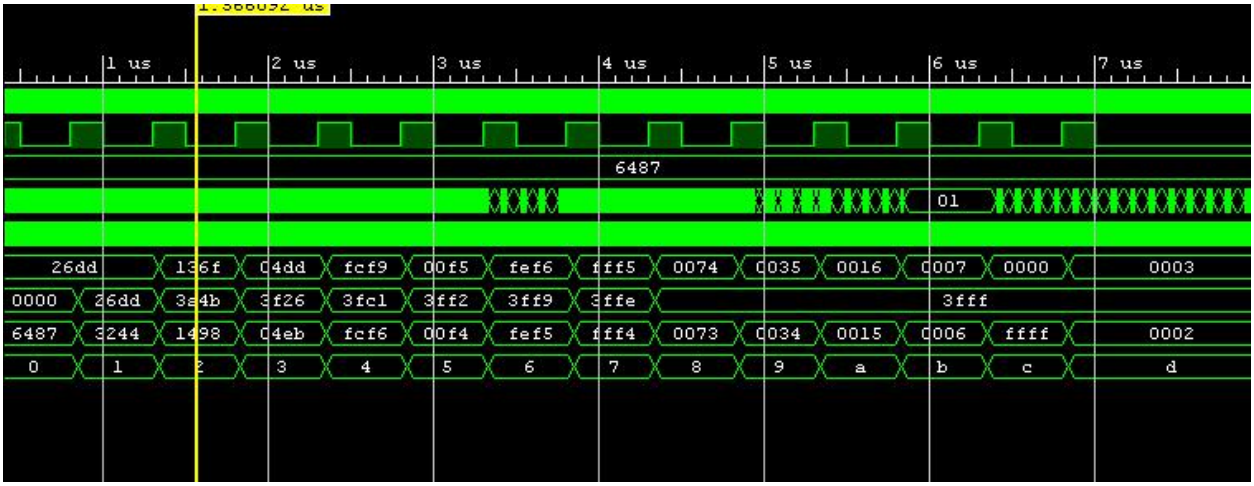
```
                         (clock sys_clk_pin rise edge)
                                                   10.000    10.000 r
    W5                                              0.000    10.000 r  clk (IN)
                         net (fo=0)                 0.000    10.000    clk
    W5                   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=316, routed)       1.423    14.764    CDIV1/clk_IBUF_BUFG
    SLICE_X41Y72         FDRE                                     r  CDIV1/count3_reg[20]/C
                         clock pessimism            0.258    15.022
                         clock uncertainty         -0.035    14.987
    SLICE_X41Y72         FDRE (Setup_fdre_C_R)     -0.429    14.558    CDIV1/count3_reg[20]
  -------------------------------------------------------------------
                         required time                       14.558
                         arrival time                       -10.776
  -------------------------------------------------------------------
                         slack                                3.781
```

E) Simulation waveform



The second line represent the push bottom, and from top to button the hexadecimal number represent the value of x,y,z. In the test case, the intial z is 0, and the x value of each step is

0x26dd,0x3a4b,0x3f26,0x3fc1,0x3ff2,0x3ff9,0x3ffe,0x3fff in 9 iterations. The result totally satisfy the expection(including y and z).



When the initial z is 0x6487 (corresponding to 90 degree)

Finally we have cos z = 0x0003, sinz = 0x3fff. This is highly closed to the accurate result

Cos 90 = 0, sin 90 = 1. The waveform demonstrate the program successfully.