DIGITAL DESIGN

ASSIGNMENTREPORT

ASSIGNMENT ID : 04

Student Name: Weibao Fu

Student ID: 11812202

PART 1: DIGITAL DESIGN THEORY

1.

JK filp-flop

J	K	Q	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristic equation: $Q_{t+1} = JQ' + K'Q$

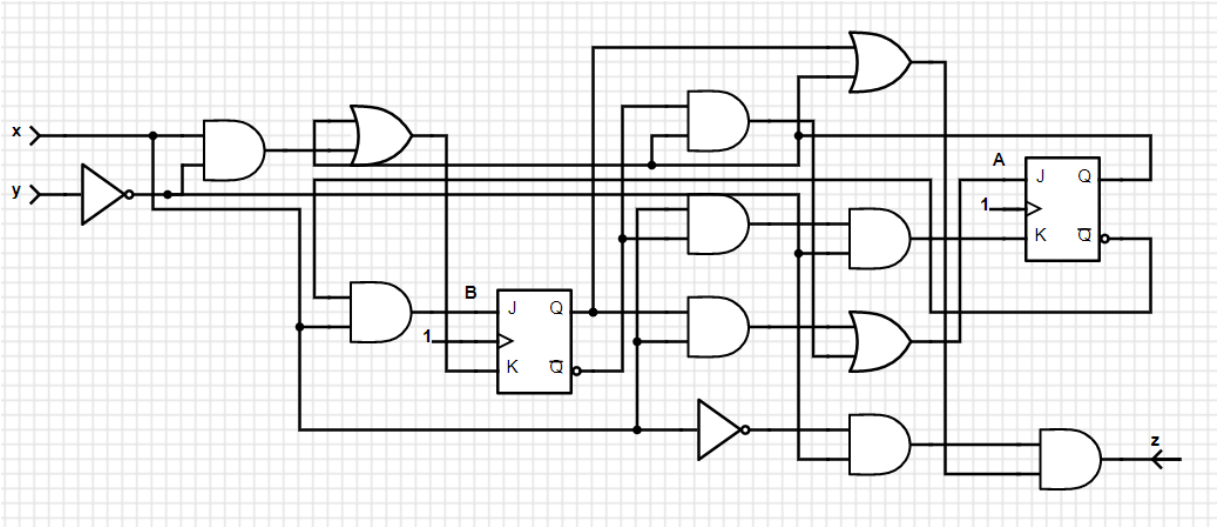
T flip-flop

T	Q	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic equation: $Q_{t+1} = TQ' + T'Q$

2.

Circuit diagram



State table

x	y	A	B	J_A	K_A	J_B	K_B	z
0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	1	1
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	0
1	0	0	1	1	0	1	1	0
1	0	1	0	1	1	0	1	0
1	0	1	1	1	0	0	1	0
1	1	0	0	0	0	1	0	0
1	1	0	1	1	0	1	0	0

1	1	1	0	0	0	0	1	0
1	1	1	1	1	0	0	1	0

State equation

From the state table, we can get $A_{t+1} = J_A A_t' + K_A' A_t$

$= (B_t x + B_t' y') A_t' + (B_t' x y')' A_t$

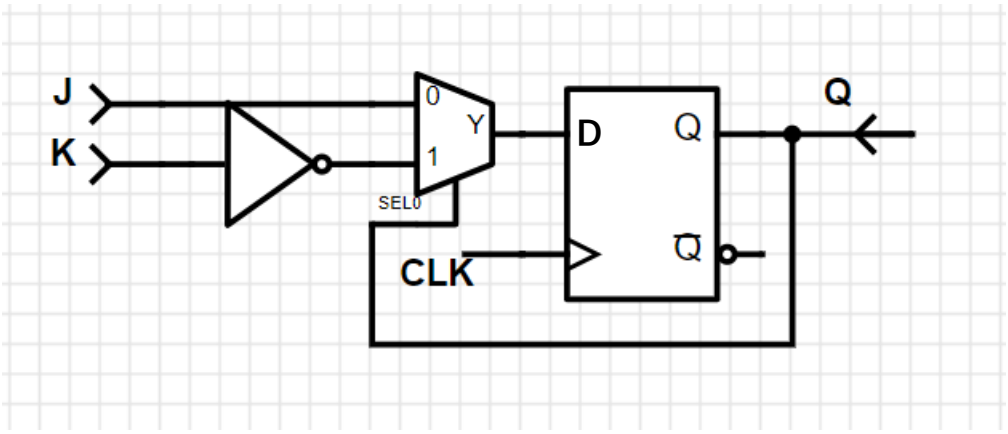
$= A_t' B_t' y' + B_t x + A_t x' + A_t y$

$B_{t+1} = J_B B_t' + K_B' B_t$

$= (A_t' x) B_t' + (A_t + x y')' B_t$

$= A_t' B_t' x + A_t' B_t (x' + y)$

3.



Since the state equation of JK flip-flop is $Q = JQ' + K'Q$

Then we can use multiplexer, we can set Q0 = J, Q1 = K to implement

$Y = JQ_0' + K'Q_1$, and we use a D flip-flop to finish the JK flip-flop.

4.

State table

A_t	B_t	T_A	T_B	A_{t+1}	B_{t+1}
0	0	0	1	0	1
0	1	1	1	1	0

1	0	1	0	0	0
1	1	1	1	0	0

State equation

$T_A = A + B$

$T_B = A' + B$

$A_{t+1} = T_A A' + T_A' A$

$= (A + B)A' + (A+B)'A$

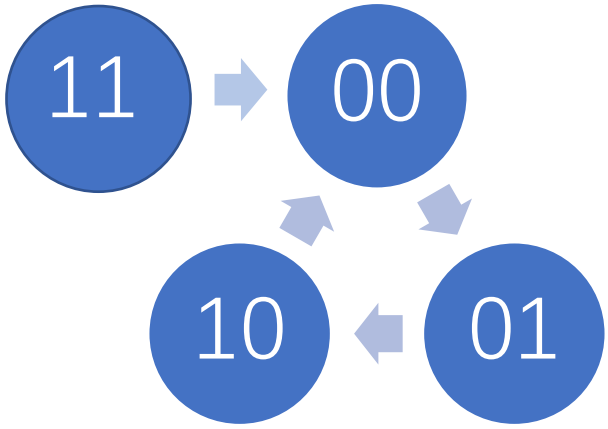
$= A'B$

$B_{t+1} = T_B B' + T_B' B$

$= (A' + B)B' + (A' + B)'B$

$= A'B'$

State transfer graph



-

From the state transfer graph and state table, we can get that the circuit is a ternary counter. It will loop in every 3 periods.

PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

Verilog Code

```
`timescale 1ns / 1ps
module T_flipflop(T,Clk,rst,Q,Qn);
input   T,Clk,rst;
output Q,Qn;
wire q1,qn,temp1,temp2,temp3,temp4;
nand G7(temp1,~Clk,T,Qn,rst);
nand G8(temp2,~Clk,T,Q);
nand G5(q1,temp1,qn);
nand G6(qn,temp2,q1,rst);
nand G3(temp3,q1,Clk);
nand G4(temp4,qn,Clk);
nand G1(Q,temp3,Qn);
nand G2(Qn,temp4,Q,rst);
endmodule
```

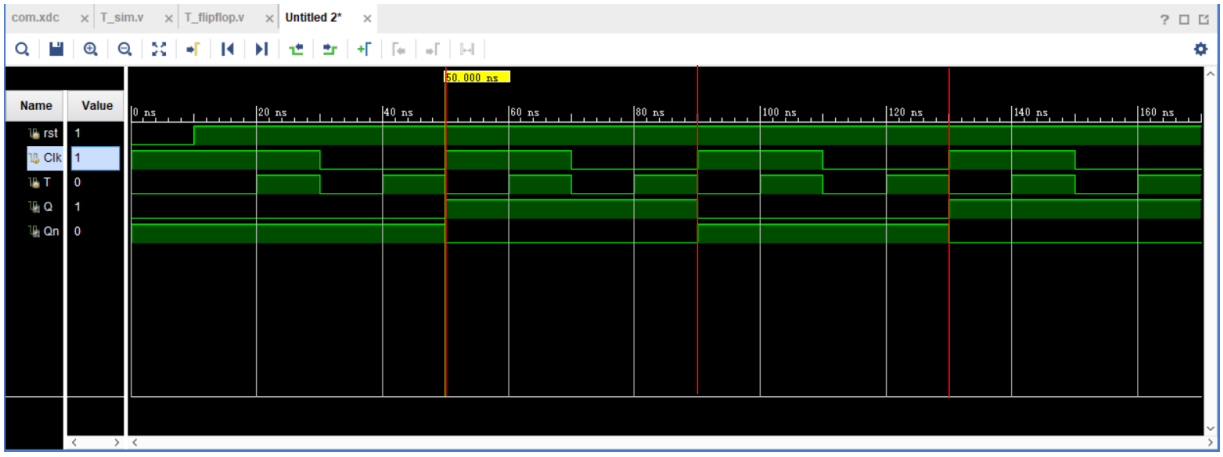
SIMULATION

Verilog Code (Simulation)

```
`timescale 1ns / 1ps
module T_sim( );
reg Clk,T,rst;
wire Q,Qn;
T_flipflop usim(T,Clk,rst,Q,Qn);
initial
begin
   {rst,Clk,T} = 3'b010;
    #10 rst = 1'b1;
    repeat(16)
   #10 {Clk,T} = {Clk,T}+1;
    $finish;
```

```
end
endmodule
```

Wave Form (rst is low level effective)



When 0-10ns, rst = 0, then Q=0, Qn=1

When 10-170ns, rst =1:

When 50ns, T is 1, and Clk is in posedge, $Q_t = 0$, so $Q_{t+1} = 1$ and $Q_t = 0$

When 90ns, T is 1, and Clk is in posedge, $Q_t = 1$, so $Q_{t+1} = 0$ and $Q_t = 1$

When 130ns, T is 1, and Clk is in posedge, $Q_t = 0$, so $Q_{t+1} = 1$ and $Q_t = 0$

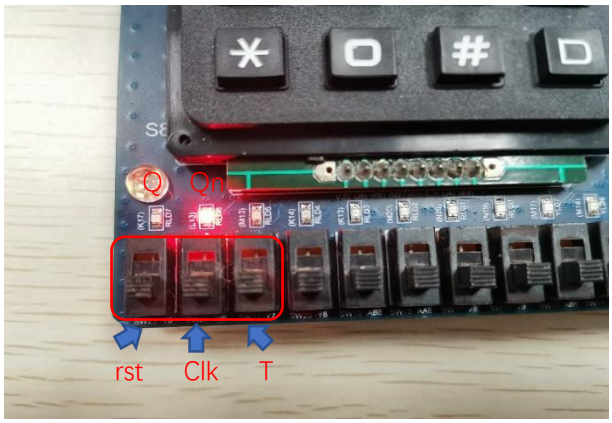
From the wave form, we can get when Clk is in posedge and T=1, the state of Q will change, else the state of Q will not change. The result is satisfied T filp-flop.

CONSTRAINT FILE AND THE TESTING

Constraint File

```
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets <Q_OBUF_inst_i_2_n_0>]
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets <Q_OBUF_inst_i_2>]
set_property SEVERITY {Warning}  [get_drc_checks LUTLP-1]
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property IOSTANDARD LVCMOS33 [get_ports Clk]
set_property IOSTANDARD LVCMOS33 [get_ports Q]
set_property IOSTANDARD LVCMOS33 [get_ports Qn]
set_property IOSTANDARD LVCMOS33 [get_ports rst]
set_property IOSTANDARD LVCMOS33 [get_ports T]
set_property PACKAGE_PIN Y9 [get_ports rst]
set_property PACKAGE_PIN W9 [get_ports Clk]
set_property PACKAGE_PIN Y7 [get_ports T]
set_property PACKAGE_PIN K17 [get_ports Q]
set_property PACKAGE_PIN L13 [get_ports Qn]
```

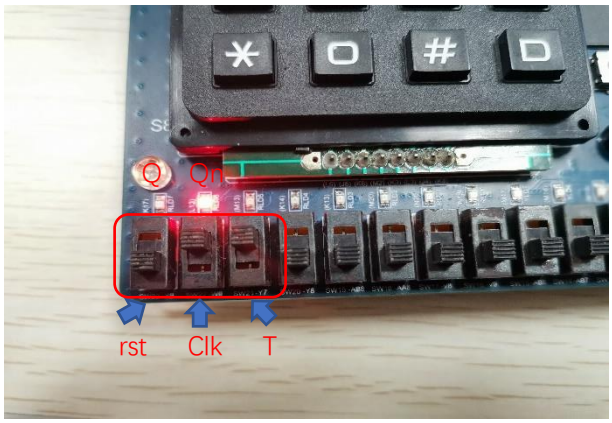
Analysis



Testcase #1

When rst is 0,

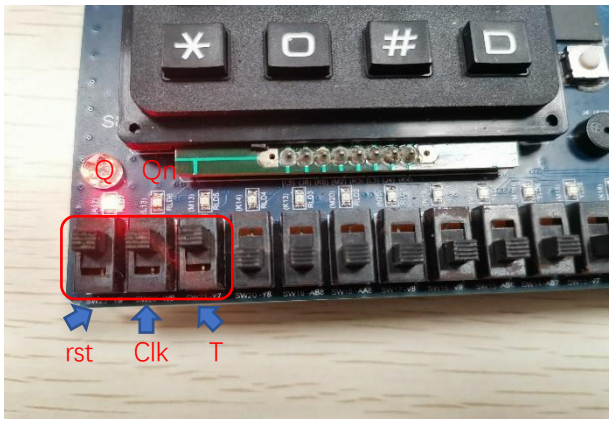
Then Q = 0, Qn = 1



Testcase #2

Although Clk and T are 1,

Q = 0, Qn = 1 since rst = 0

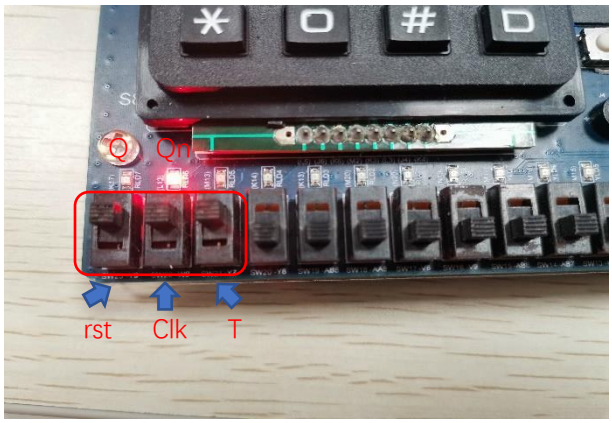


Testcase #3

When rst is 1, Clk and T are 1

Then the state will change,

Q = 1, Qn = 0

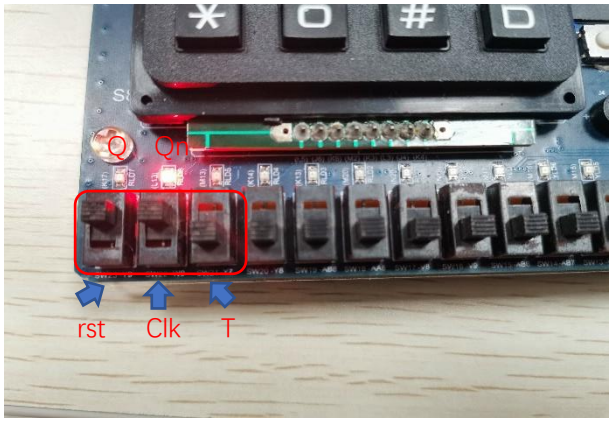


Testcase #4

When rst is 1, Clk and T are 1

Then the state will change,

Q = 0, Qn = 1

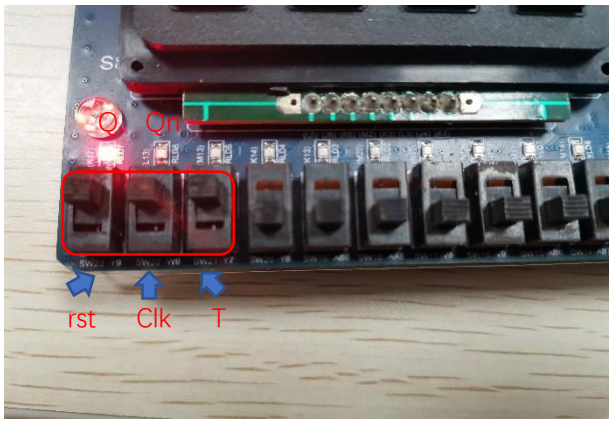


Testcase #5

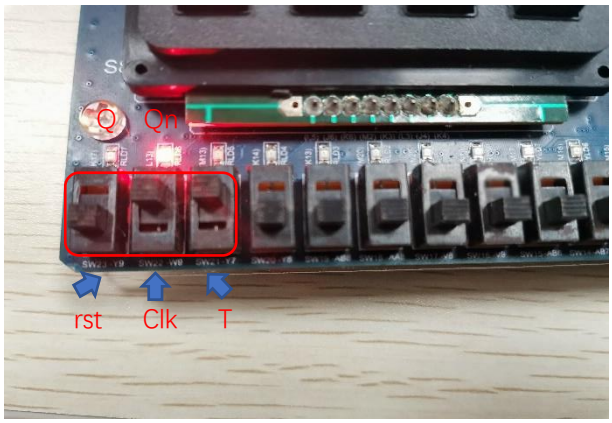
When rst is 1, Clk is 1,

Then the state will not change

since T=0, Q = 0, Qn = 1



Testcase #6
When rst is 1, Clk and T are 1
Then the state will change,
Q = 1, Qn = 0



Testcase #7
Although Clk and T are 1,
the state will reach original state since rst = 0, Q = 0, Qn = 1

PART 2: DIGITAL DESIGN LAB (TASK2)

DESIGN

Verilog Code

```
`timescale 1ns / 1ps
module task2(Clk,rst,x_in,y_out);
input Clk,rst;
input [4:0]x_in;
output reg[2:0]y_out;
reg [2:0]y_next;
reg sum=0;
parameter A=3'b001,B=3'b010,C=3'b100;
always @(posedge Clk,negedge rst) begin
```

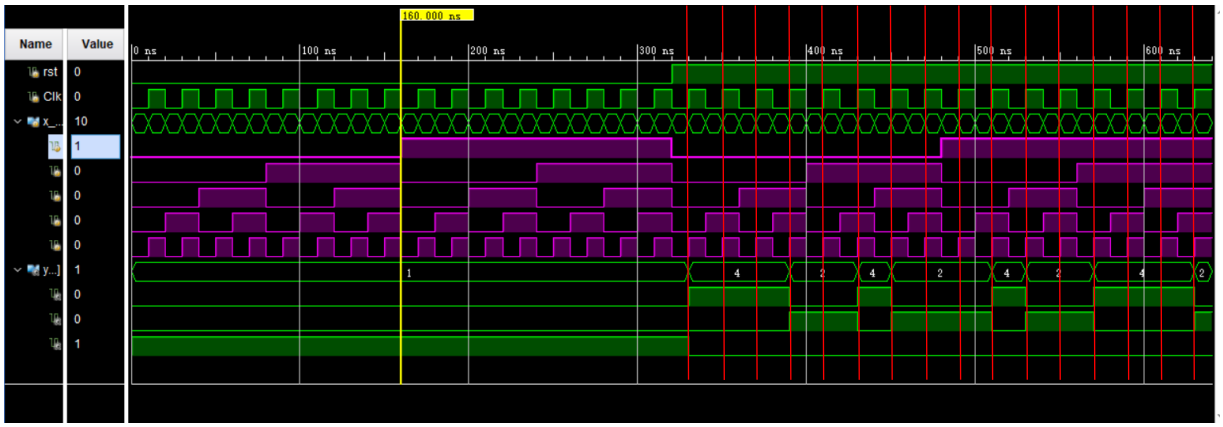
```
    if(~rst) y_next <= A;
    else begin
        sum = sum^x_in[0]^x_in[1]^x_in[2]^x_in[3]^x_in[4];
        if(sum)   y_next <= C;
        else y_next <= B;
    end
end
always @(y_next) begin
    y_out <= y_next;
end
endmodule
```

SIMULATION

Verilog Code (Simulation)

```
`timescale 1ns / 1ps
module task2_sim( );
reg Clk,rst;
reg [4:0]x_in;
wire [2:0]y_out;
task2 usim(Clk,rst,x_in,y_out);
initial begin
    {Clk,rst,x_in}=7'b0000000;
    while({rst,x_in}<6'b111111)
    #10 {rst,x_in} = {rst,x_in}+1;
    #10 $finish;
end
always #10 Clk=~Clk;
endmodule
```

Wave Form (rst is low level effective)



From the wave form:

<u>When 0-320ns</u>, rst = 0, so y_out is state A(001)

<u>When 320-640ns,</u> rst=1:

When 330ns, the clock is in posedge, x_in is (00001), the sum of 1 is 1, so the state is C(100)

When 350ns, the clock is in posedge, x_in is (00011), the sum of 1 is 3, so the state is C(100)

When 370ns, the clock is in posedge, x_in is (00101), the sum of 1 is 5, so the state is C(100)

When 390ns, the clock is in posedge, x_in is (00111), the sum of 1 is 8, so the state is B(010)

When 410ns, the clock is in posedge, x_in is (01001), the sum of 1 is 10, so the state is B(010)

When 430ns, the clock is in posedge, x_in is (01011), the sum of 1 is 13, so the state is C(100)

……(Others are ignored)

From the wave form, we can conclude that if rst is effective, then the output state is A. Else we read the information of in_x, and count the total number of 1, when the count is odd, the output state is C, else the output state is B. Such that, the result is satisfied.

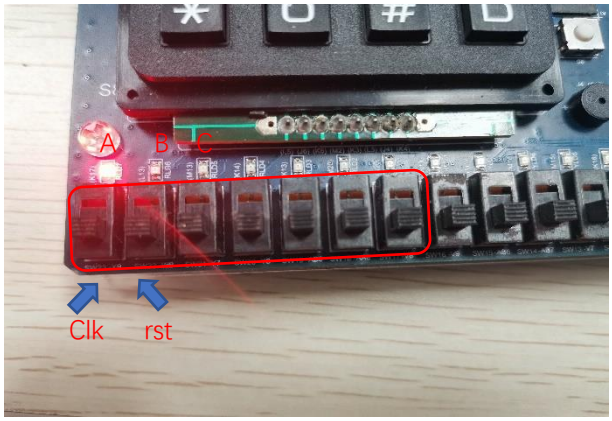
CONSTRAINT FILE AND THE TESTING

Constraint File

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets Clk]
set_property IOSTANDARD LVCMOS33 [get_ports {y_out[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y_out[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports Clk]
set_property IOSTANDARD LVCMOS33 [get_ports rst]
set_property IOSTANDARD LVCMOS33 [get_ports {x_in[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {x_in[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {x_in[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {x_in[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {x_in[0]}]
set_property PACKAGE_PIN K17 [get_ports {y_out[0]}]
set_property PACKAGE_PIN L13 [get_ports {y_out[1]}]
set_property PACKAGE_PIN M13 [get_ports {y_out[2]}]
set_property PACKAGE_PIN Y9 [get_ports Clk]
set_property PACKAGE_PIN W9 [get_ports rst]
set_property PACKAGE_PIN Y7 [get_ports {x_in[4]}]
set_property PACKAGE_PIN Y8 [get_ports {x_in[3]}]
set_property PACKAGE_PIN AB8 [get_ports {x_in[2]}]
set_property PACKAGE_PIN AA8 [get_ports {x_in[1]}]
set_property PACKAGE_PIN V8 [get_ports {x_in[0]}]
```

Analysis

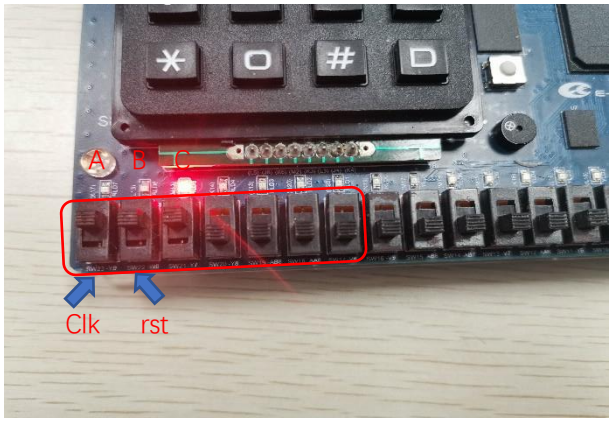
(In order to make the result looks more comfortable, I swap the order of (A,B,C) as below. Which means I link the higher level port with “M13”, lower with “K17”)



Testcase #1:

When rst=0, the sum of 1 is 0

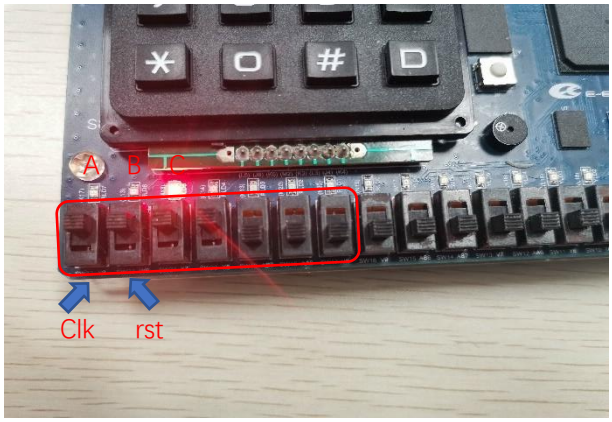
The state is A (001)



Testcase #2:

When rst=1, and Clk is in posedge, x_in(10000), the sum of 1 is 1

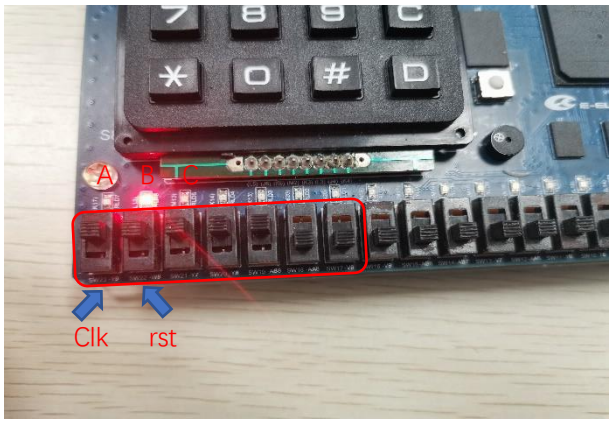
The state is C (100)



Testcase #3:

When rst=1, and Clk is in posedge, x_in(11000), the sum of 1 is 3

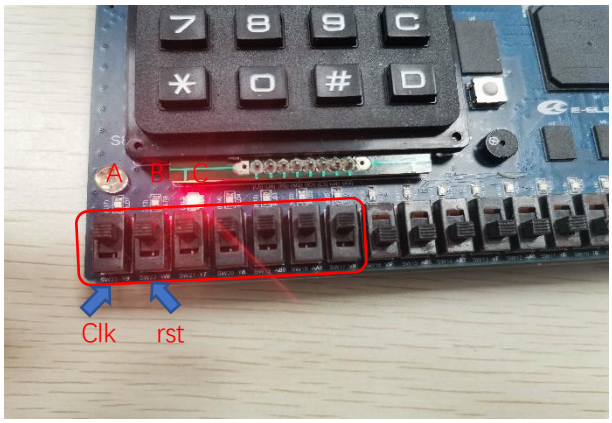
The state is C (100)



Testcase #4:

When rst=1, and Clk is in posedge, x_in(11100), the sum of 1 is 6

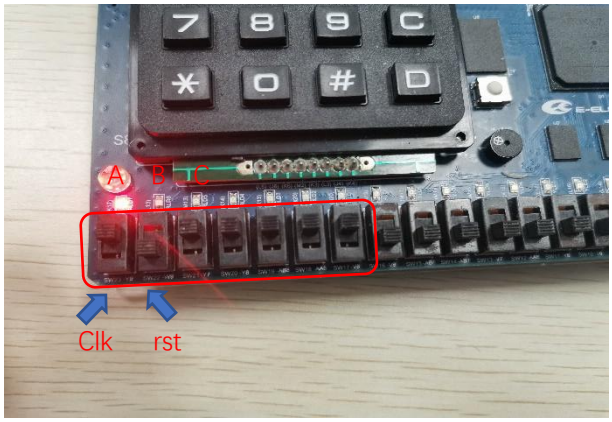
The state is B (010)



Testcase #5:

When rst=1, and Clk is in posedge, x_in(11111), the sum of 1 is 11

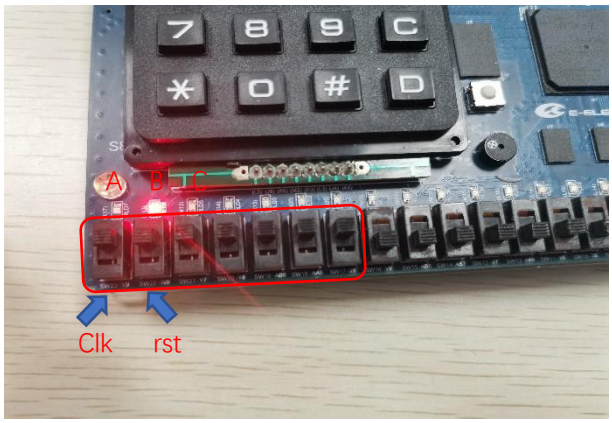
The state is C (100)



Testcase #6:

When rst=0, and Clk is in posedge, Although x_in(11111), the sum of 1 is still 11 since rst = 0

The state is A (001)



Testcase #7:

When rst=1, and Clk is in posedge, x_in(11111), the sum of 1 is 16

The state is B (010)

THE DESCRIPTION OF OPERATION

No problem