同济大学计算机系

数字逻辑课程实验报告

学　　号 **2253156**

姓　　名 闫浩扬

专　　业 计算机科学与技术（精英班）

授课老师 郭玉臣

一、 实验内容

- **实验介绍**

  在本次实验中，我们将使用 Verilog HDL 语言实现 D 触发器、JK 触发器以及 PC 寄存器的设计和仿真。

- **实验目标**

  深入了解 D 触发器、JK 触发器、D 触发器构成的 PC 寄存器的原理。

  学习使用 Verilog HDL 行为描述语言设计 D 触发器、JK 触发器、D 触发器构成的 PC 寄存器。

二、 硬件逻辑与原理

(一) 触发器

触发器是一种同步双稳态器件，用来记忆一位二进制数。所谓同步，是指触发器的记忆 状态按时钟（CLK）规定的启动指示点（脉冲边沿）来改变。

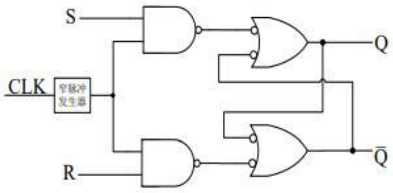
同步复位，指的是同步复位信号就是指复位信号只在所需时钟边沿到来时才有效。 所谓的异步复位，即无论时钟边沿到来与否，只要复位信号有效输出就会被复位。

下面将介绍几种触发器原理：

1) SR 触发器

下图给出了 SR 触发器的逻辑图和其功能表。其中×表示无关，↑表示时钟信号由低到高。数据输入端 S 和 R 称为同步输入，因为这个两个输入的数据只会在时钟脉冲上升沿时被传送到触发器。

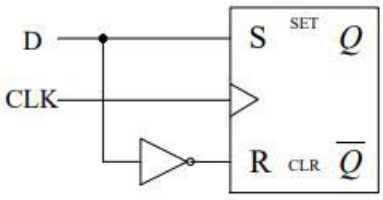
- 当 S 为高 R 为低时，Q 输出在时钟脉冲上升沿时变高，触发器置 1。
- 当 S 为低 R 位高时，Q 输出在时钟脉冲上升沿时变低，触发器置 0。
- 当 S 和 R 两者都为低时，Q 输出状态不会发生变化（保持）。
- 当 S 和 R 两者都为高时，Q 输出状态是不稳定的。



输入			输出		说明
S	R	CLK	Q	$\bar{Q}$	
0	0	×	Q^n	$\bar{Q}^n$	保持
0	1	↑	0	1	置 0
1	0	↑	1	0	置 1
1	1	↑	?	?	不稳

2) D 触发器

D 触发器是在 SR 触发器基础上构建的。即在一个 SR 触发器上增加一个非门。该触发器只有一个数据输入端 D，经反相器反相后，变成互补数据输入，送到 SR 触发器，从而避免了 SR 触发器存在的不稳态问题。当数 据输入 D=1，且在时钟脉冲上升沿，触发器置位（1 状态）；当数据输入 D=0，且在时钟脉 冲上升沿，触发器复位（0 状态）。

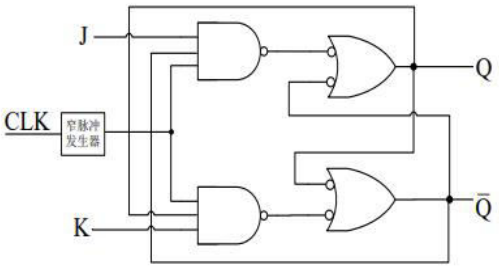


输入		输出		说明
D	CLK	Q	$\bar{Q}$	
1	↑	1	0	置位（存 1）
0	↑	0	1	复位（存 0）

3) JK 触发器

JK 触发器是一种广泛应用的触发器类型。字母 J 和 K 只表示它们是两个数据输入端符 号，没有什么特别含义。JK 触发器与 SR 触发器在置位、复位方面的功能是相同的，不同之 处在于，JK 触发器改进了 SR 触发器存在的不稳定状态。当 J=1，K=1 时，对每一个连续 的时钟脉冲，触发器可改变成相反状态或计数状态，这种工作方式称为交替操

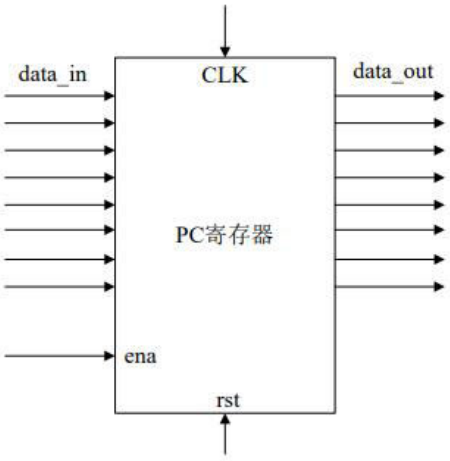
作。下图为 JK 触发器的内部逻辑及其功能表。



输入			输出		说明
J	K	CLK	Q	$\bar{Q}$	
0	0	↑	Q^n	$\overline{Q^n}$	保持
0	1	↑	0	1	置 0
1	0	↑	1	0	置 1
1	1	↑	Q^n	$\overline{Q^n}$	交替

(二) PC 寄存器

PC 寄存器（program counter）是组成 CPU 的基本部件，用来存放当前正在执行的指令，包括指令的操作码和地址信息。我们知道，D 触发器可以用于存储比特信号。如果 D 为 1，那么在时钟的上升沿，D 触发器的输出 Q 将变为 1。如果 D 为 0，那么在时钟的上升沿，D 触发器的输出 Q 将变为 0。在实时数字系统中，通常 D 触发器的时钟输入端始终有时钟信号输入。这就意味着在每个时钟的上升沿，当前 D 值都将被锁存在 Q 中。在本实验主要利用 Verilog HDL 模块实例化基于 D 触发器模块实现一个 PC 寄存器。下图给出了所要建模的 PC 寄存器其功能图。



三、 模块建模

(一) 触发器

1) 同步复位 D 触发器

接口定义

```
module Synchronous D FF(
   input CLK,//时钟信号，上升沿有效
   input D,//输入信号 D
   input RST_n,  //复位信号，低电平有效
   output reg Q1，//输出信号 Q
   output reg Q2//输出信号 0
);
```

Verilog 代码描述

```
`timescale 1ns / 1ps
module Synchronous_D_FF(
    input CLK,
    input D,
    input RST_n,
    output reg Q1,
    output reg Q2
    );
    always @ (posedge CLK) begin
        if(!RST_n) begin
            Q1 = 1'b0;
            Q2 = 1'b1;
        end
        else begin
            Q1 <= D;
            Q2 <= ~D;
        end
    end
endmodule
```

- 功能描述

Synchronous_D_FF 使用了时序逻辑和行为型描述，实现了一个同步复位 D 触发器。它在时钟上升沿将输入 D 传送给 Q1,如果复位信号 RST_n 为低电平，那么 Q1 将复位。

这种结构可以用于实现寄存器，计数器，移位寄存器等数字电路。

XDC 约束

变量	CLK	D	RST_n	Q1	Q2
N4 板上的管脚	BTNR (M17)	SW0 (J15)	BTNU (M18)	LD1 (H17)	LD2 (K15)

2）异步复位 D 触发器

接口定义

```
module Asynchronous_D_FF(
    input CLK, //时钟信号，上升沿有效
    input D,   //输入信号 D
    input RST_n,   //复位信号，低电平有效
    output reg Q1, //输出信号 Q
    output reg Q2  //输出信号Q¯
);
```

Verilog 代码描述

```
module Asynchronous_D_FF(
    input CLK,
    input D,
    input RST_n,
    output reg Q1,
    output reg Q2
    );
    always @ (posedge CLK, negedge RST_n)  //异步与同步的区别
    begin
        if(!RST_n) begin
            Q1 = 1'b0;
```

```
            Q2 = 1'b1;
        end
        else begin
            Q1 <= D;
            Q2 <= ~D;
        end
    end
endmodule
```

功能描述

Asynchronous_D_FF 模块实现了异步 D 触发器。异步 D 触发器是一种时序逻辑电路，它的输出 Q1 和 Q2 只取决于输入 D 的值，而不受时钟 CLK 的影响。当复位信号 RST_n 为 0 时，Q1 和 Q2 会被异步地置为 0 和 1，即不需要等待时钟的上升沿。当 RST_n 为 1 时，Q1 和 Q2 会在时钟的上升沿同步地跟随 D 的值，即 Q1=D，Q2=~D。异步与同步的区别在于，异步操作不需要时钟信号，而同步操作需要时钟信号来触发状态的变化。

XDC 约束

变量	CLK	D	RST_n	Q1	Q2
N4 板上的管脚	BTNR (M17)	SW0 (J15)	BTNU (M18)	LD1 (H17)	LD2 (K15)

3）异步复位 JK 触发器

接口定义

```
module JK_FF(
  input CLK, //时钟信号，上升沿有效
  input J, //输入信号 J
  input K, //输入信号 K
  input RST_n, //复位信号，低电平有效
  output reg Q1,//输出信号 Q
  output reg Q2 //输出信号 Q_
);
```

Verilog 代码描述

```
`timescale 1ns / 1ps
module JK_FF(
    input CLK,
    input J,
    input K,
    input RST_n,
    output reg Q1,
    output reg Q2
    );
    initial begin
        Q1=0;
        Q2=1;
    end
    always @ (posedge CLK or negedge RST_n) begin
        if(!RST_n) begin
            Q1=0;
            Q2=1;
        end
```

```
        else begin
            Q1 <= (J&&Q2)||(~K&&Q1);
            Q2 <= ~((J && Q2) || (~K && Q1));
        end
    end
endmodule
```

功能描述

JK_FF 模块采用了时序逻辑，实现了一个 JK 触发器，JK 触发器可以用于数据存储，计数，分频等功能。JK 触发器有两个端 J 和 K，一个时钟输入端 CLK，一个复位输入端 RST_n，和两个输出端 Q1 和 Q2。

always @ (posedge CLK or negedge RST_n)表示每当时钟上升沿或复位下降沿到来时，执行后面的语句。if(!RST_n)表示如果复位端为低电平，即 0，那么 Q1 和 Q2 恢复初始状态。else 表示如果复位端为高电平，即 1，那么根据 J 和 K 的值，更新 Q1 和 Q2 的值。

XDC 约束

变量	CLK	J	K	RST_n	Q1	Q2
N4 板上的管脚	BTNR (M17)	SW0 (J15)	SW1 (L16)	BTNU (M18)	LD1 (H17)	LD2 (K15)

（二）PC 寄存器

接口定义

```
module pcreg(
    input clk, //1 位输入，寄存器时钟信号，上升沿时为 PC 寄存器赋值 input
    rst,         //1 位输入，异步重置信号，高电平时将 PC 寄存器清零
     //注：当 ena 信号无效时，rst 也可以重置寄存器
    input ena, //1 位输入,有效信号高电平时 PC 寄存器读入 data_in
     //的值，否则保持原有输出 input [31:0] data_in,
     //32 位输入，输入数据将被存入寄存器内部
    output reg [31:0] data_out //32 位输出，工作时始终输出 PC //寄存器内部存
    储的值
);
```

Verilog 代码描述

```
`timescale 1ns / 1ps

module pcreg(
    input clk,
    input rst,
    input ena,
    input [31:0] data_in,
    output reg [31:0] data_out
    );

    always @ (posedge clk or posedge rst) begin
        if (rst)
            data_out = 32'h0000_0000;
        else
            if (ena)
                data_out = data_in;
    end
endmodule
```

功能描述

pcreg 模块采用了时序逻辑，实现了一个 32 位 PC 寄存器。可以用于存储程序计数器。

工作原理：在每个时钟上升沿或复位信号上升沿时，检查复位信号和使能信号的状态，如果复位信号为高，清空寄存器的内容；如果使能信号为高，将输入数据写入寄存器。在其他情况下，寄存器的内容保持不变。

四、 测试模块建模

(一) 同步复位 D 触发器

```
`timescale 1ns / 1ps

module Synchronous_D_FF_tb;
    reg CLK;
    reg D;
    reg RST_n;

    wire Q1;
    wire Q2;

    Synchronous_D_FF uut (
        .CLK(CLK),
        .D(D),
        .RST_n(RST_n),
        .Q1(Q1),
        .Q2(Q2)
    );
    initial
        CLK=0;
```

```
 always
        #40 CLK = ~CLK;

   initial begin
        D = 0;
        RST_n = 0;
        #40 D=0;
        #1 RST_n =0;
        #40 D=1;
        #1 RST_n =0;
        #40 D=1;
        #1 RST_n =1;
        #40 D=0;
        #1 RST_n =0;
        #40 D=0;
        #1 RST_n =1;
        #40 D=1;
        #40 D=0;
    end
endmodule
```

(二) 异步复位 D 触发器

```
 `timescale 1ns / 1ps

module Asynchronous_D_FF_tb;
    reg CLK;
    reg D;
    reg RST_n;

    wire Q1;
    wire Q2;

    Asynchronous_D_FF uut (
        .CLK(CLK),
        .D(D),
        .RST_n(RST_n),
        .Q1(Q1),
        .Q2(Q2)
    );
    initial
        CLK=0;
```

```
always
        #40 CLK = ~CLK;

    initial begin
        D = 0;
        RST_n = 0;
        #40 D=0;
        #1 RST_n =0;
        #40 D=1;
        #1 RST_n =0;
        #40 D=1;
        #1 RST_n =1;
        #40 D=0;
        #1 RST_n =0;
        #40 D=0;
        #1 RST_n =1;
        #40 D=1;
        #40 D=0;
    end
endmodule
```

(三) 异步复位 JK 触发器

```
`timescale 1ns / 1ps

module JK_FF_tb();
    reg CLK, J, K, RST_n;
    wire Q1,Q2;

    JK_FF uul(CLK, J, K, RST_n, Q1, Q2);

    initial begin
        J = 0;
        K = 0;
        CLK = 0;
        RST_n = 1;
    end
    always
        #10 CLK = ~CLK;
```

```
initial begin
         J = 0;K = 0;
        #40 J = 0; K = 1;
        #40 J = 1; K = 0;
        #40 J = 1; K = 1;
        #40 RST_n = 0;
          J = 0;K = 0;
        #40 J = 0; K = 1;
        #40 J = 1; K = 0;
        #40 J = 1; K = 1;

        #40 RST_n = 1;
         J = 1; K = 0;
        #40 J = 1; K = 1;
        #40 J = 0; K = 0;
        #40 J = 0; K = 1;
    end
endmodule
```

(四) 32 位 PC 寄存器

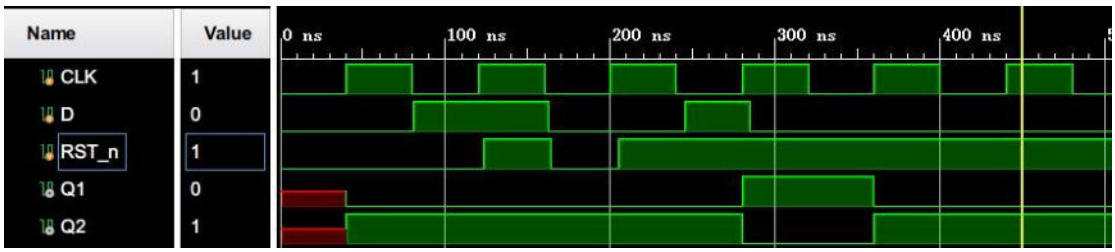
```
`timescale 1ns / 1ps
module pcreg_tb();
    reg clk;
    reg rst;
    reg ena;
    reg [31:0] data_in;
    wire [31:0] data_out;
    pcreg  uul(clk,  rst,  ena,  data_in,
data_out);
    initial begin
        clk = 0;
        rst = 0;
        ena = 1;
        data_in = 32'h0000_0000;
    end
    always
        #20 clk = ~clk;
    initial begin
```

```
 #20 data_in = 32'hfff0_1100;
        #10 data_in = 32'h11f1_1111;
        #30 ena = 0;
        #20 data_in = 32'h1100_1ff0;
        #10 data_in = 32'h1111_11f1;
        #20 rst = 1;
        #40 data_in = 32'h0101_f101;
        #10 ena = 1;
        #10 data_in = 32'h0101_010f;
        #10 rst = 0;
        #10 data_in = 32'h0111_1f10;
        #20 rst = 1;
    end
endmodule
```

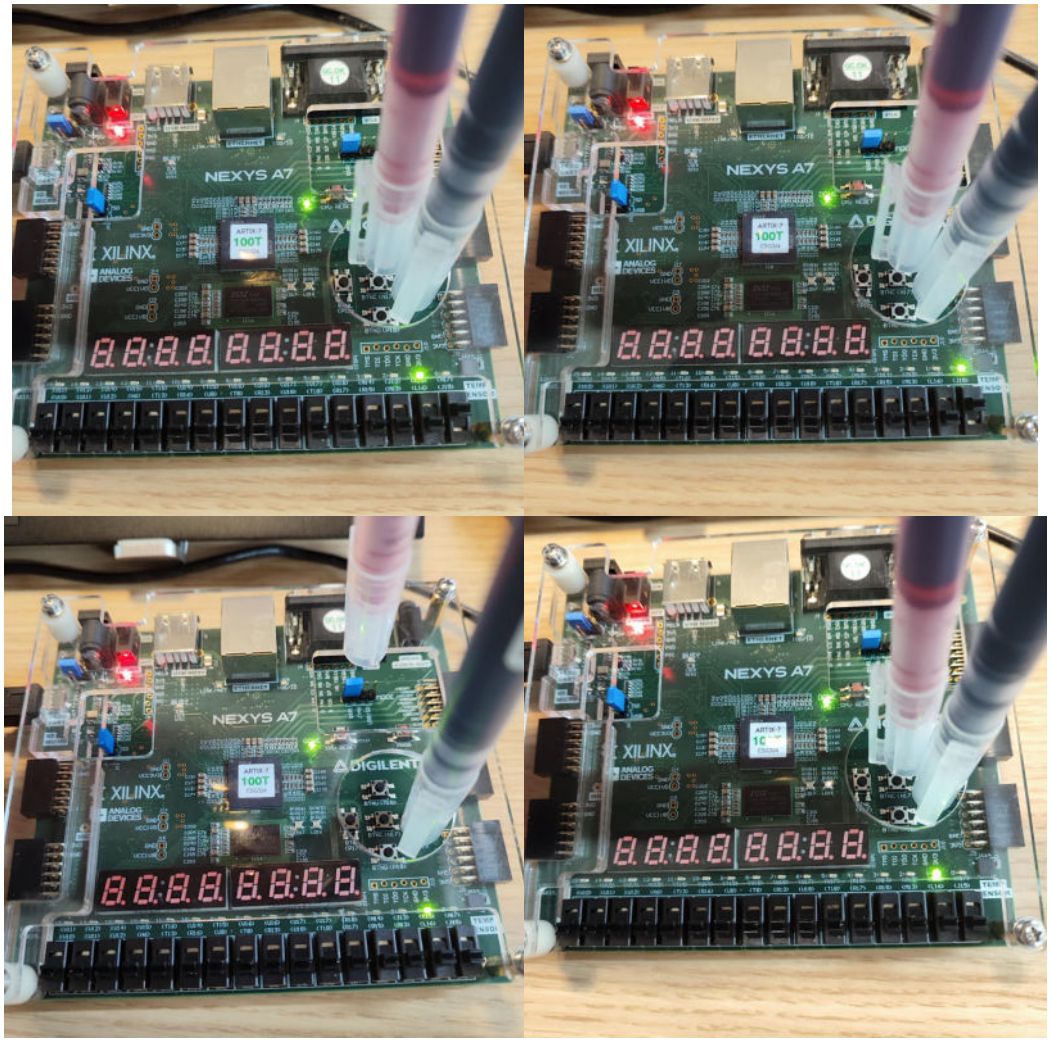
五、 实验结果

(一) 同步复位 D 触发器

- **modelsim 仿真**

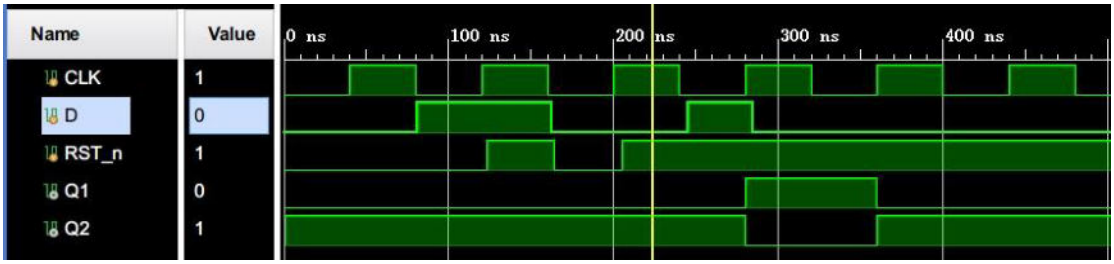


- **实验结果**

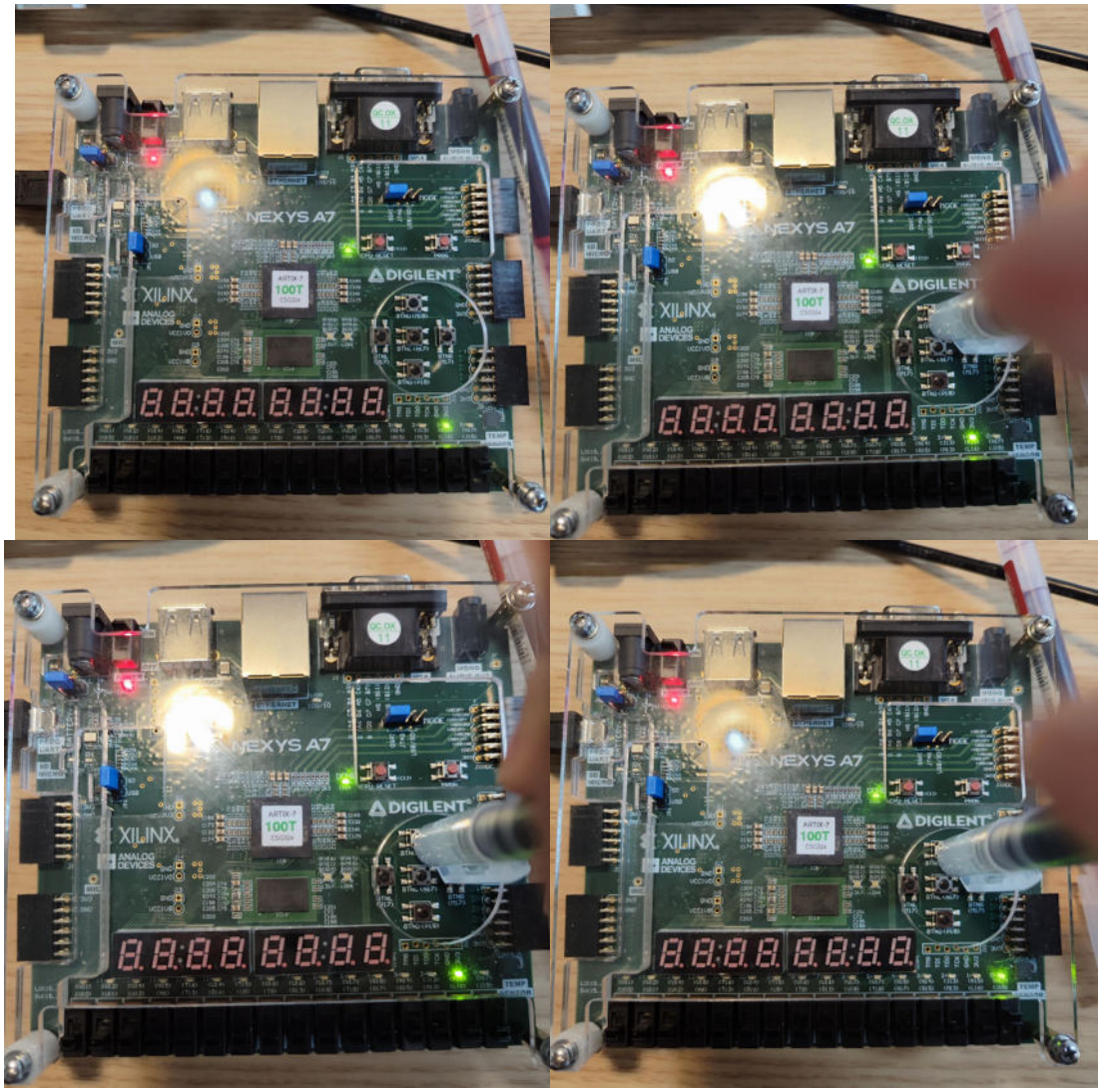


(二) 异步复位 D 触发器

- **modelsim 仿真**

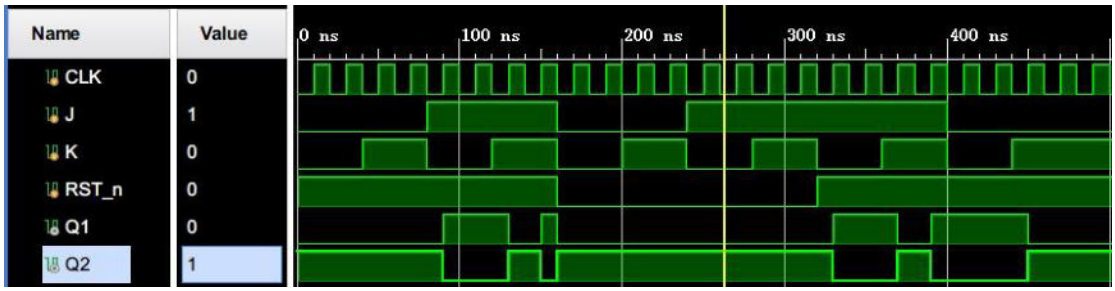


- **实验结果**

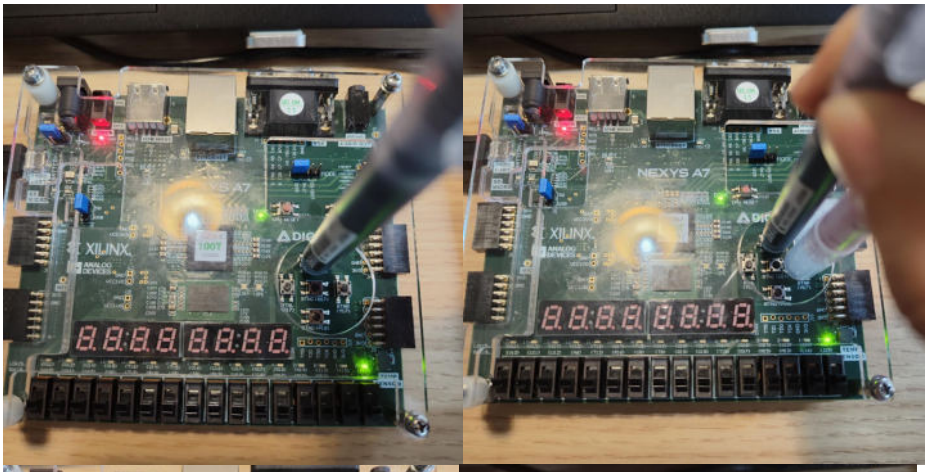
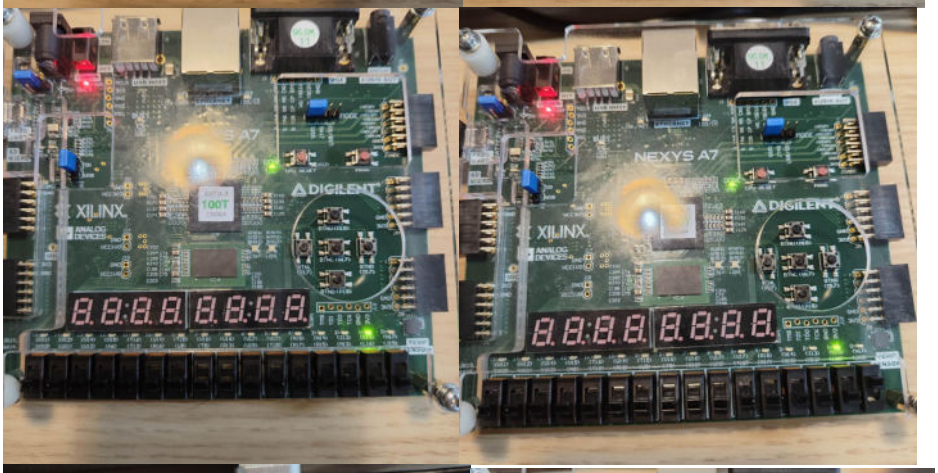
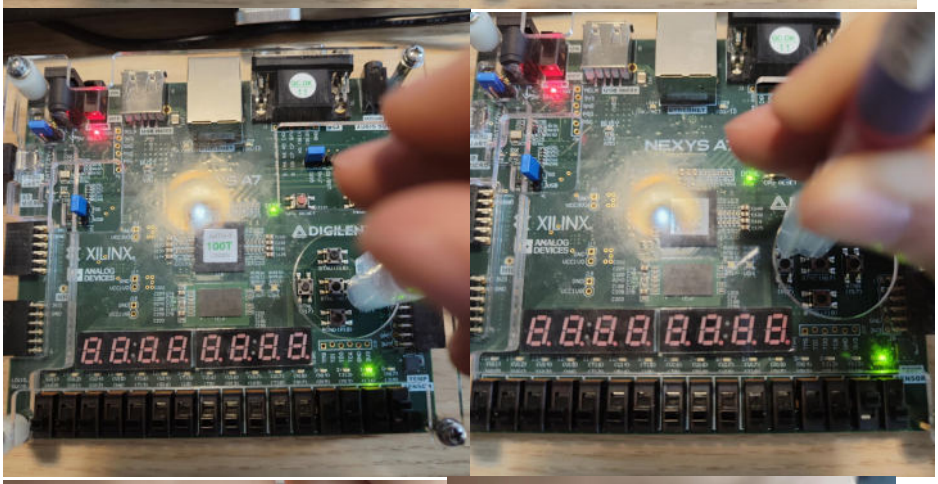
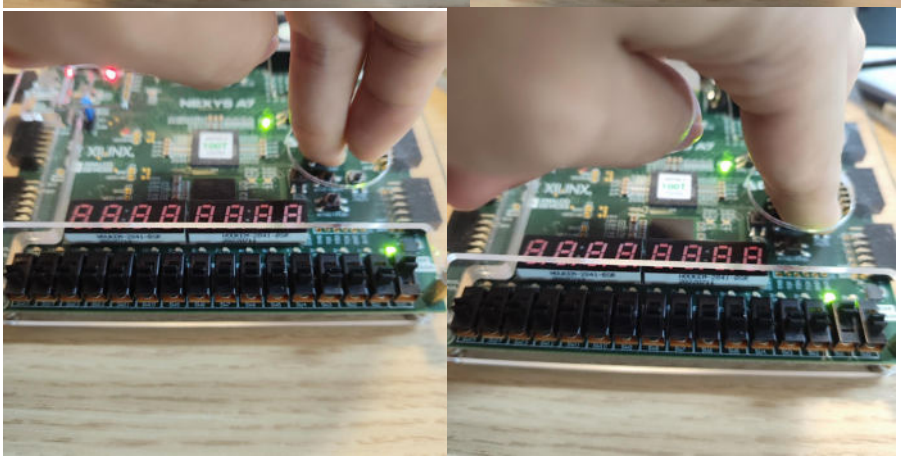


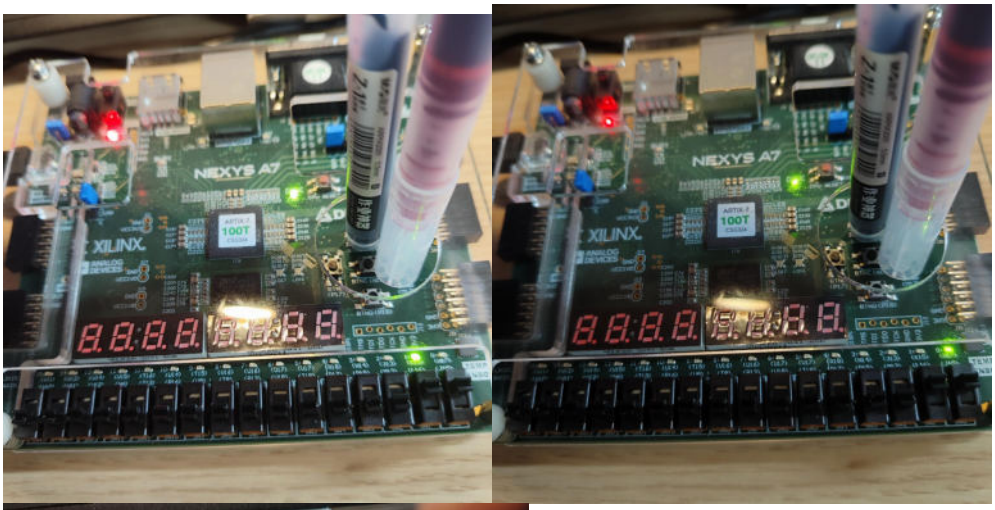
(三) 异步复位 JK 触发器

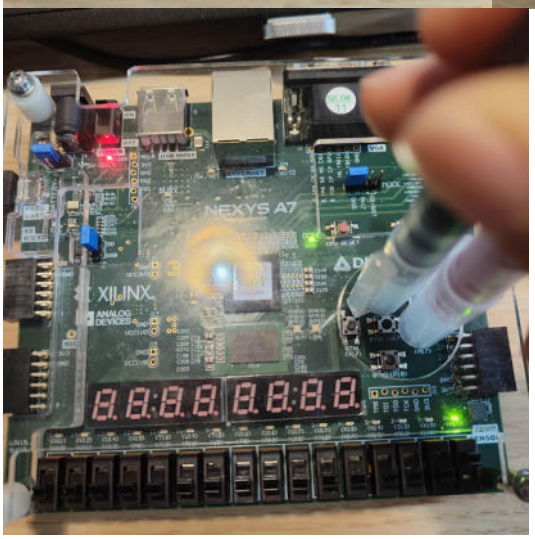
- modelsim 仿真波形图



- 实验结果





(四) 32 位 PC 寄存器

- **modelsim 仿真**