

同济大学计算机系

数字逻辑课程综合实验报告



学 号 2352595

姓 名 张嘉麟

专 业 计算机科学与技术

授课老师 郭玉臣

一、实验内容

计数器：

计数器的功能是记忆脉冲的个数，它是数字系统中应用最广泛的基本时序逻辑构件。计数器所能记忆脉冲的最大数目称为该计数器的模。构成计数器的核心元件是触发器。本题中使用 logisim 画出同步模 8 电路原理图，实现由 3 个 JK 触发器组成的 3 位同步模 8 计数器，所有触发器的时钟都与同一个始终脉冲相连。本题通过实例化 JK 触发器与 7 段数码管来实现数字的显示；

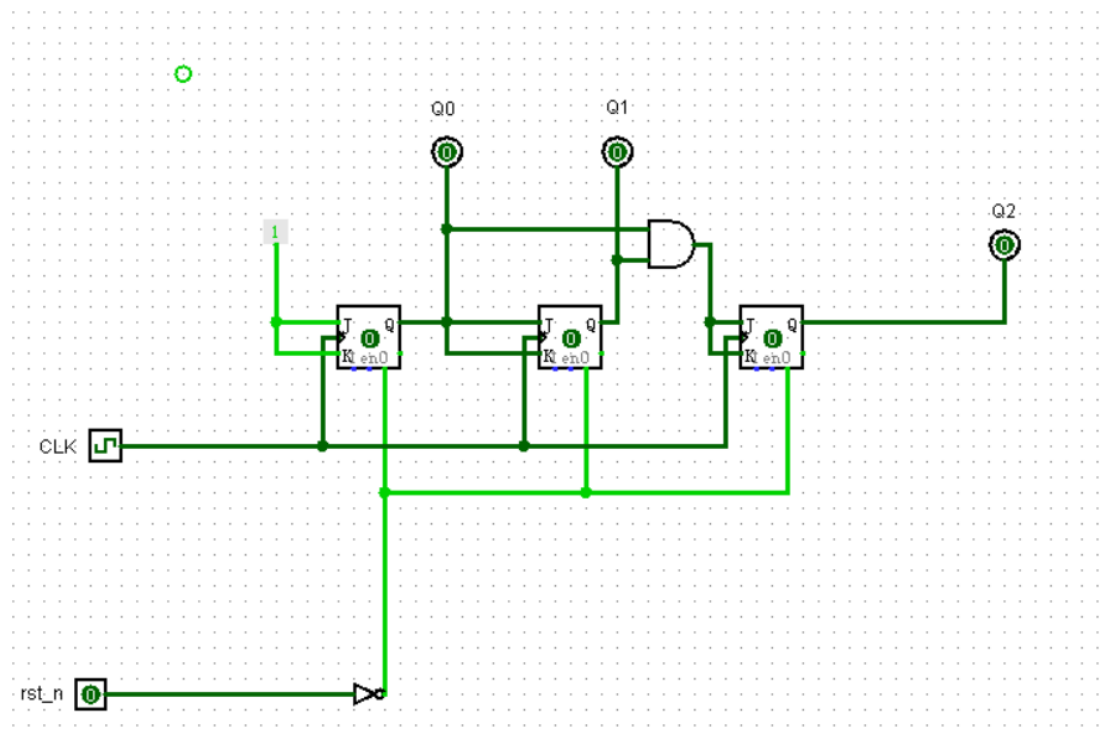
分频器：

每个计数器的脉冲输出频率等于其输入时钟频率除以计数模值，因此可以很容易地利用计数器由一个输入时钟信号获得分频后的时钟信号，这种应用称为分频。

二、硬件逻辑图

（实验步骤中要求用 logisim 画图的实验，在该部分给出 logisim 原理图，否则该部分在实验报告中不用写）

(1) Counter8




```

module SevenSegmentDisplay (
    input [2:0] binary_input,
    output reg [6:0] segments
);
    always @(*) begin
        case (binary_input)
            3'b000: segments = 7'b1000000; // 0
            3'b001: segments = 7'b1111001; // 1
            3'b010: segments = 7'b0100100; // 2
            3'b011: segments = 7'b0110000; // 3
            3'b100: segments = 7'b0011001; // 4
            3'b101: segments = 7'b0010010; // 5
            3'b110: segments = 7'b0000010; // 6
            3'b111: segments = 7'b1111000; // 7
            default: segments = 7'b0000000; // Default, all off
        endcase
    end
endmodule

```

```

module Counter8(
    input CLK, //时钟信号，上升沿有效
    input rst_n, //异步复位信号，低电平有效
    output [2:0] oQ, //二进制计数器输出
    output [6:0] oDisplay //七段数字显示管输出
);

```

```

    wire Q0, Q1, Q2;

```

```

    // Instantiate JK flip-flops for each bit

```

```

    JK_FF ff0 (
        .J(1'b1),
        .K(1'b1),
        .CLK(CLK),
        .rst_n(rst_n),
        .Q(Q0)
    );

```

```

    JK_FF ff1 (
        .J(Q0),
        .K(Q0),

```

```

        .CLK(CLK),
        .rst_n(rst_n),
        .Q(Q1)
    );

    JK_FF ff2 (
        .J(Q0 & Q1),
        .K(Q0 & Q1),
        .CLK(CLK),
        .rst_n(rst_n),
        .Q(Q2)
    );

    assign oQ = {Q2, Q1, Q0};

    // Instantiate the seven-segment display module
    SevenSegmentDisplay display (
        .binary_input(oQ),
        .segments(oDisplay)
    );
endmodule

```

(2) Divider

```

`timescale 1ns / 1ns

module Divider(
    input I_CLK,        // 输入时钟信号，上升沿有效
    input rst,          // 同步复位信号，高电平有效
    output reg O_CLK    // 输出时钟，先高电平再低电平
);
    parameter DIVIDE_Factor = 20; // 分频倍数

    // 计算计数上限
    localparam half_DIVIDE_Factor = DIVIDE_Factor / 2;

    initial begin
        O_CLK = 0; // 初始化 O_CLK 为 0
    end

    reg [$clog2(half_DIVIDE_Factor)-1:0] counter; // 计数器

    always @(posedge I_CLK) begin
        if (rst) begin
            // 同步复位：在时钟上升沿重置计数器和输出时钟

```

```

        counter <= 0;
        O_CLK <= 0;
    end else if (counter == half_DIVIDE_Factor - 1) begin
        // 当计数器达到上限时，翻转输出时钟，并重置计数器
        O_CLK <= ~O_CLK;
        counter <= 0;
    end else begin
        // 否则，计数器加 1
        counter <= counter + 1;
    end
end
endmodule

```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

(1) Counter8

```

`timescale 1ns / 1ns
module Counter8_tb;
    // Testbench signals
    reg CLK;
    reg rst_n;
    wire [2:0] oQ;
    wire [6:0] oDisplay;

    // Instantiate the Counter8 module
    Counter8 uut (
        .CLK(CLK),
        .rst_n(rst_n),
        .oQ(oQ),
        .oDisplay(oDisplay)
    );

    // Clock generation
    initial begin
        CLK = 0;
        forever #5 CLK = ~CLK; // 10 ns period clock
    end

    // Test sequence
    initial begin
        // Initialize signals
        rst_n = 0;
    end
endmodule

```

```

        // Apply reset
        #10 rst_n = 1; // Release reset after 10 ns

        // Run for 160ns to cover multiple counting cycles
        #160 $finish;
    end
endmodule

```

(2) Divider

```

`timescale 1ns / 1ns

module Divider_tb;

    reg I_CLK;
    reg rst;
    wire O_CLK;

    // 实例化待测模块
    Divider #(.DIVIDE_Factor(20)) uut (
        .I_CLK(I_CLK),
        .rst(rst),
        .O_CLK(O_CLK)
    );

    // 时钟生成
    initial begin
        I_CLK = 0;
        forever #5 I_CLK = ~I_CLK; // 10ns 周期，即 100 MHz 时钟
    end

    // 初始化和测试过程
    initial begin
        // 初始化
        rst = 1;

        // 释放复位
        #15 rst = 0;

        // 运行一段时间以观察行为。
        #200;

        // 停止仿真
        $finish;
    end
endmodule

```

```
end
```

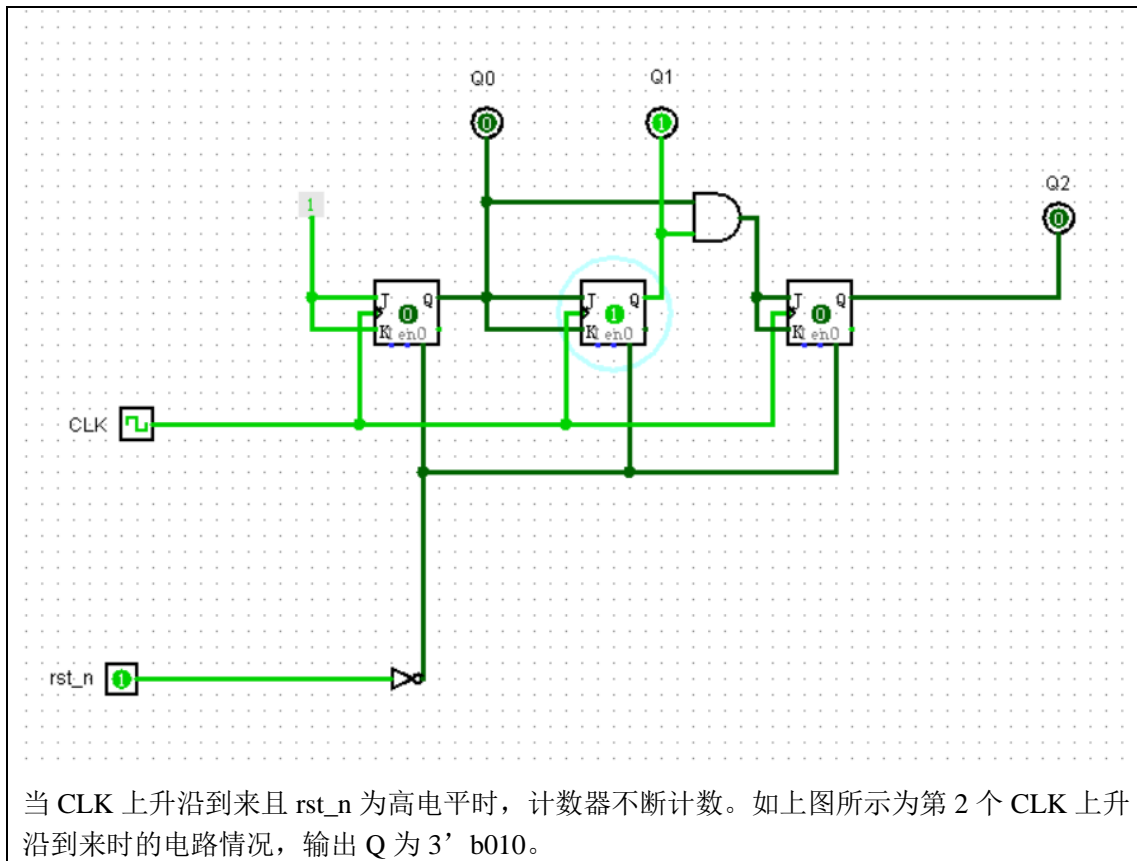
```
endmodule
```

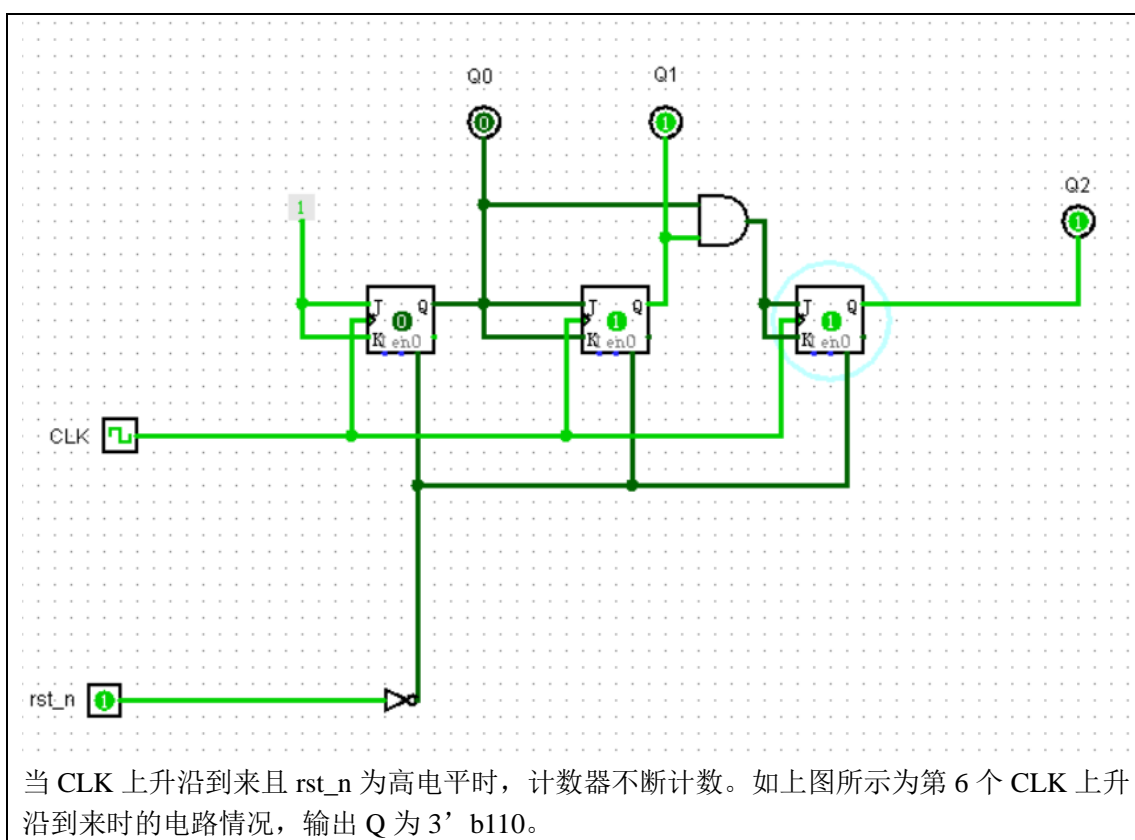
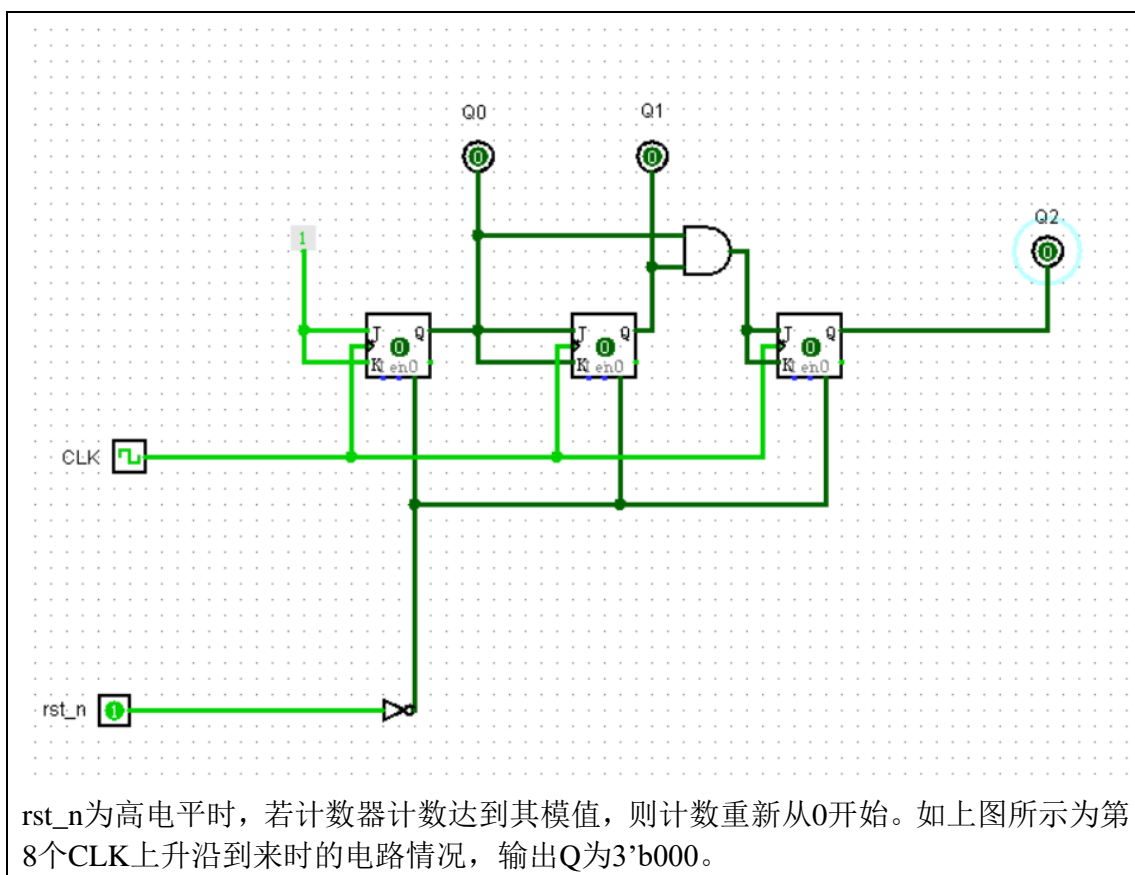
五、实验结果

（该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图））

1. logisim 逻辑验证图

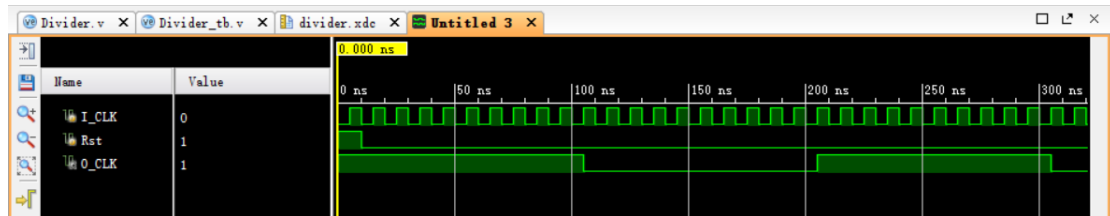
(1) Counter8







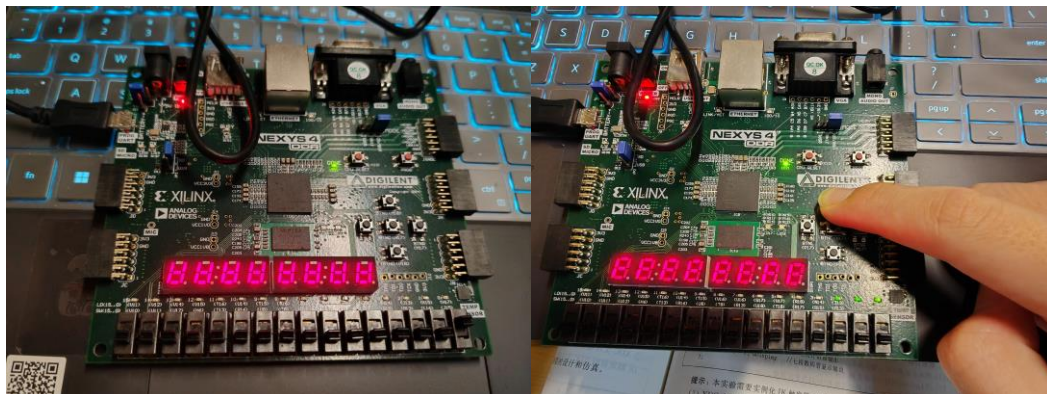
第二版：将 O_CLK 初始值设为 1



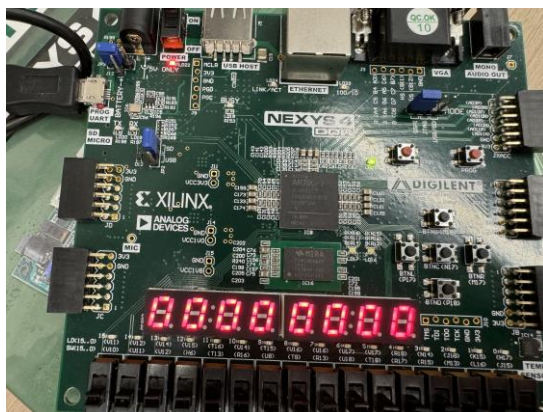
3.下板结果

(1) Counter8

首次失败尝试：



问题：数字展示反了、按下 M18 之后灯亮的 incorrect、不按 M18 的时候没有自动数字变化。



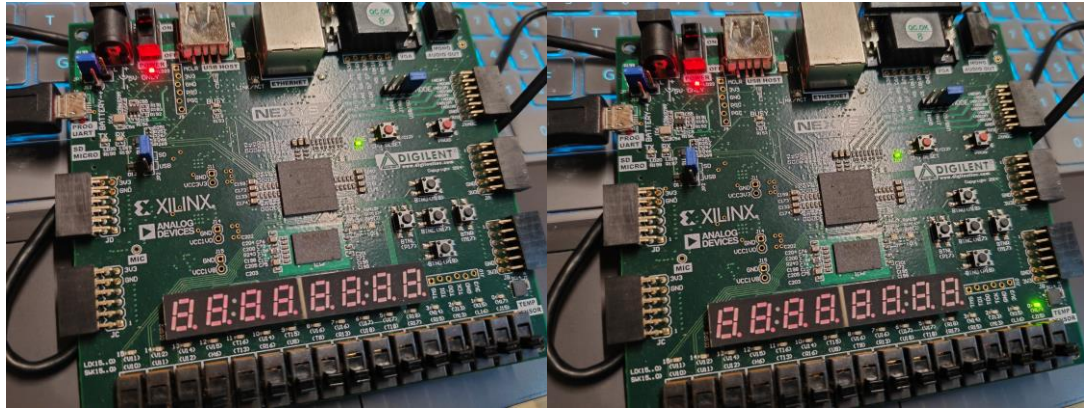
更改后结果正确

(2) Divider

测试的时候调成每个周期 1s:

```
// parameter DIVIDE_FACTOR = 20; // 分频倍数默认为20  
parameter DIVIDE_FACTOR = 50000000; // 分频倍数设置为50000000 即每秒一个周期  
integer count;
```

没有按下 M18 的时候 H17 闪烁:



按下 M18 之后 H17 保持常亮:

