

# 同济大学计算机系

## 数字逻辑课程实验报告



学 号 2151140

姓 名 王谦

专 业 信息安全

授课老师 郭玉臣

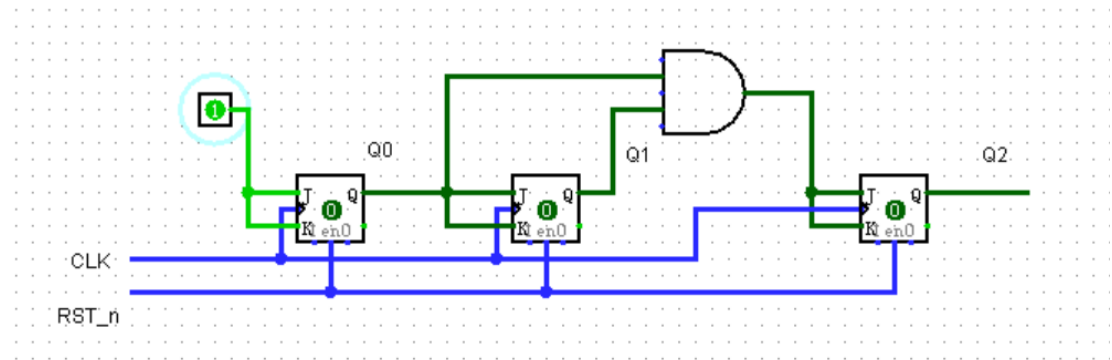
一、实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现计数器和分频器的设计和仿真。

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）

模 8 计数器：



三、模块建模

（该部分要求对实验中建模的所有模块进行功能描述，并列出各模块建模的 verilog 代码）

1.

1) 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，用  $M$  表示。构成计数器的核心元件是触发器。图 6.7.1 所示为 3 位同步模 8 计数器逻辑图，它由 3 个 JK 触发器组成。所有触发器的时钟都与同一个时钟脉冲源连接在一起，每一个触发器的状态变化都与时钟脉冲同步，计数器的模  $M=2^3=8$ 。（注：各触发器工作前要清 0）。表 6.7.1 是模 8 计数器的状态转移表。

表 6.7.1 模 8 计数器的状态转移表

时钟 个数	PS (现 态)			NS (次 态)			时钟 个数	PS (现态)			NS (次态)		
	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>		Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	0	0	0	0	0	1	5	1	0	0	1	0	1
2	0	0	1	0	1	0	6	1	0	1	1	1	0
3	0	1	0	0	1	1	7	1	1	0	1	1	1
4	0	1	1	1	0	0	8	1	1	1	0	0	0
							9 (循 环)	0	0	0	0	0	1

```

//模 8 计数器:
module Counter8(CLK,rst_n,oQ,oDisplay);
input CLK,rst_n;
output [2:0] oQ;
output [6:0] oDisplay;
reg cin = 1;
/*这是下板测试部分的代码，提交时注释掉
wire O_CLK;
Divider DIV(CLK,~rst_n,O_CLK);
JK_FF JK0(O_CLK,cin,cin,rst_n,oQ[0],);
JK_FF JK1(O_CLK,oQ[0],oQ[0],rst_n,oQ[1],);
JK_FF JK2(O_CLK,oQ[0] && oQ[1],oQ[0] && oQ[1],rst_n,oQ[2],);
display7 DIS({1'b0,oQ},{oDisplay});*/
//下方是非下板测试时的正常代码
JK_FF JK0(CLK,cin,cin,rst_n,oQ[0],);
JK_FF JK1(CLK,oQ[0],oQ[0],rst_n,oQ[1],);
JK_FF JK2(CLK,oQ[0] && oQ[1],oQ[0] && oQ[1],rst_n,oQ[2],);
display7 DIS({1'b0,oQ},{oDisplay});
endmodule
/*****/

//JK_FF:
module JK_FF(CLK,J,K,RST_n,Q1,Q2);
input CLK;
input J,K,RST_n;
output reg Q1,Q2;
always@(posedge CLK or posedge RST_n)
begin
if(RST_n)
begin Q1 = (J && ~Q1 || ~K && Q1);
Q2 = ~Q1; end
else
begin Q1 = 1'b0;
Q2 = ~Q1; end
end
endmodule
/*****/

//display_7:
module display7(iData,oData);
input [3:0] iData;//四位输入 D3~D0
output reg [6:0] oData; //七位译码输出 g~a
always@(iData)
begin
case(iData)
4'b0000:

```

```

        begin oData=7'b1000000;    end    //0
4'b0001:
        begin oData=7'b1111001;    end    //1
4'b0010:
        begin oData=7'b0100100;    end    //2
4'b0011:
        begin oData=7'b0110000;    end    //3
4'b0100:
        begin oData=7'b0011001;    end    //4
4'b0101:
        begin oData=7'b0010010;    end    //5
4'b0110:
        begin oData=7'b0000010;    end    //6
4'b0111:
        begin oData=7'b1111000;    end    //7
4'b1000:
        begin oData=7'b0000000;    end    //8
4'b1001:
        begin oData=7'b0010000;    end    //9
default:;
endcase
end
endmodule

```

2.

## 2) 分频器

每一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, 因此我们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。在本实验中, 要求同学采用 Verilog HDL 行为描述方法设计一个分频器。

### ● 接口定义:

```

module Divider (
    input I_CLK,    //输入时钟信号, 上升沿有效
    input Rst,      //复位信号, 高电平有效
    output O_CLK    //输出时钟
);

```

注: 在 module 中使用 parameter 语句, 使该分频器的默认分频倍数为 20。

```

module Divider(I_CLK,rst,O_CLK);
input I_CLK,rst;
output reg O_CLK;
reg [31:0] cnt;
parameter N=20;
//parameter N=100000000;
always @ (posedge I_CLK or negedge rst)
begin
    if(rst)

```

```

        begin
            cnt <= 32'd0;
            O_CLK<= 0;
        end
    else if(cnt < N/2-1)
        begin
            cnt <= cnt + 1'b1;
        end
    else
        begin
            cnt <= 32'b0;
            O_CLK <= ~O_CLK;
        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. 模 8 计数器：

```

`timescale 1ns / 1ns
module Counter8_tb;
    reg CLK,rst_n;
    wire [2:0] oQ;
    wire [6:0] oDisplay;
    initial CLK = 0;
    always #2 CLK = ~CLK;
    initial
    begin
        rst_n = 0;
        #13
        rst_n = 1;
        #13
        rst_n = 0;
        #13
        rst_n = 1;
        #13
        rst_n = 0;
        #13
        rst_n = 1;
        #113
        rst_n = 0;
        #113
    end
endmodule

```

```

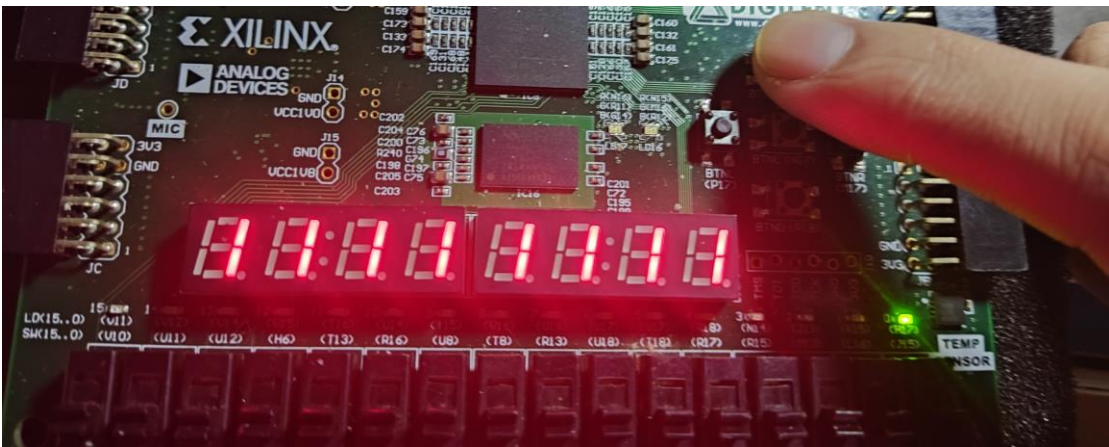
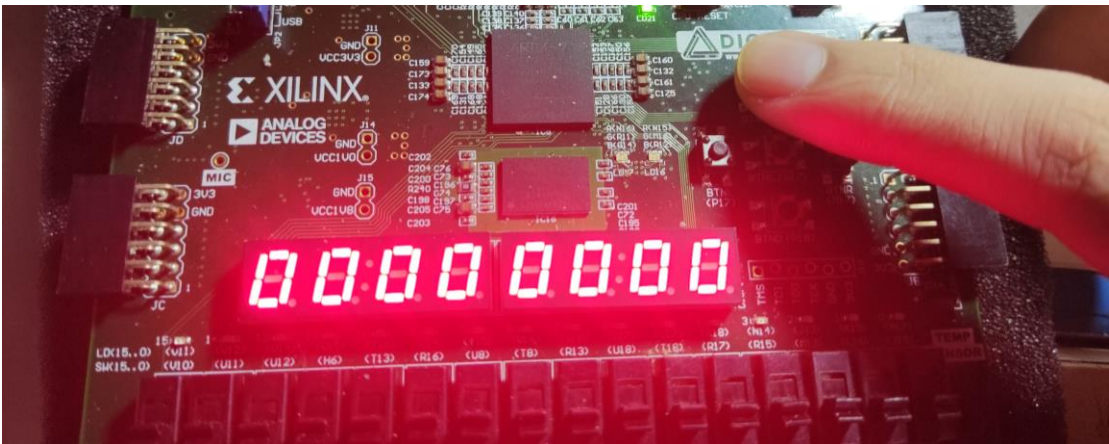
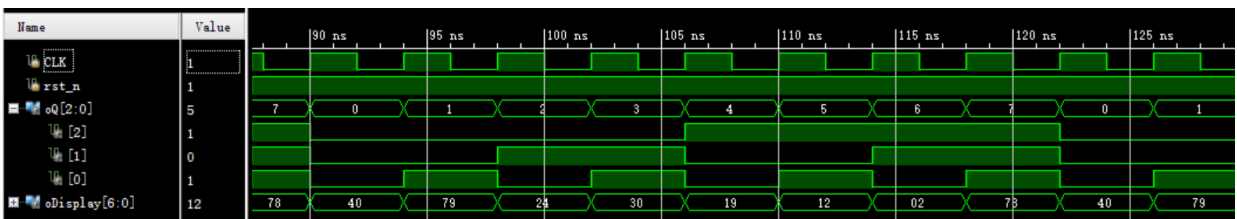
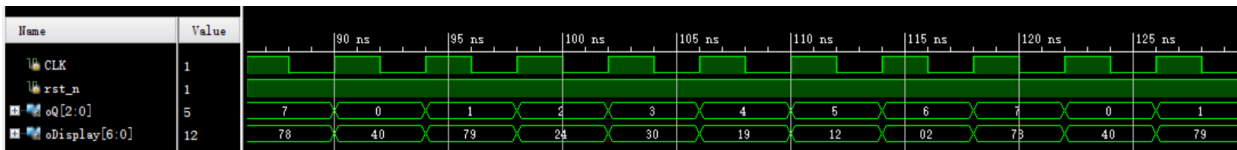
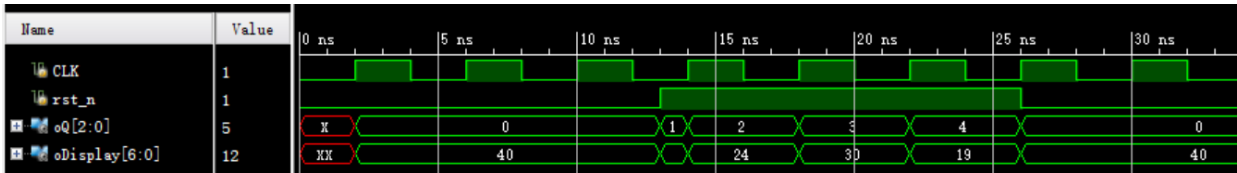
        rst_n = 1;
        #113
        $stop;
    end
Counter8
Counter8_init(
    .CLK(CLK),
    .rst_n(rst_n),
    .oQ(oQ),
    .oDisplay(oDisplay)
);
Endmodule
2.
`timescale 1ns / 1ns
module Divider_tb;
    reg I_CLK;
    reg rst;
    // Outputs
    wire O_CLK;
    // Instantiate the Unit Under Test (UUT)
    always #5 I_CLK = ~I_CLK;
    initial begin
        // Initialize Inputs
        I_CLK = 0;
        rst = 0;
        #50;
        rst = 1;
        #50;
        rst = 0;
        #500
        rst = 1;
        #400
        $stop;
    end
end
Divider
Divider_init(
    .I_CLK(I_CLK),
    .rst(rst),
    .O_CLK(O_CLK)
);
Endmodule

```

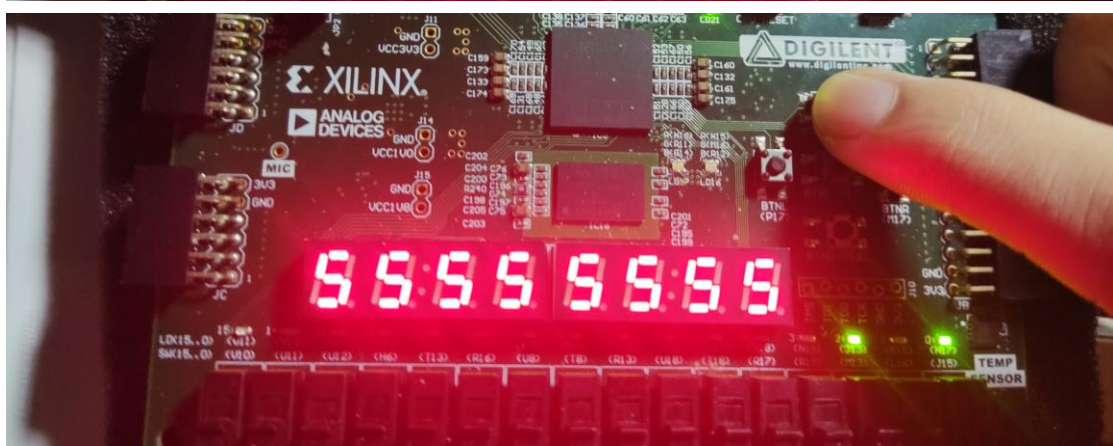
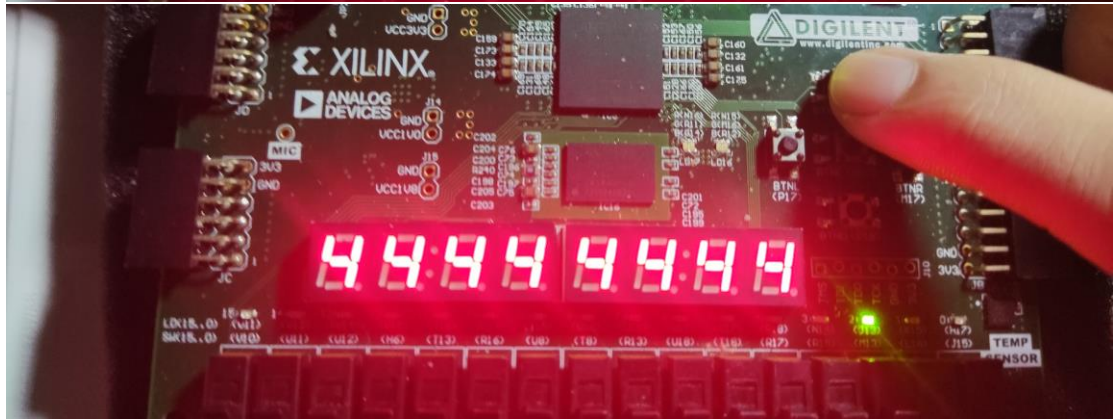
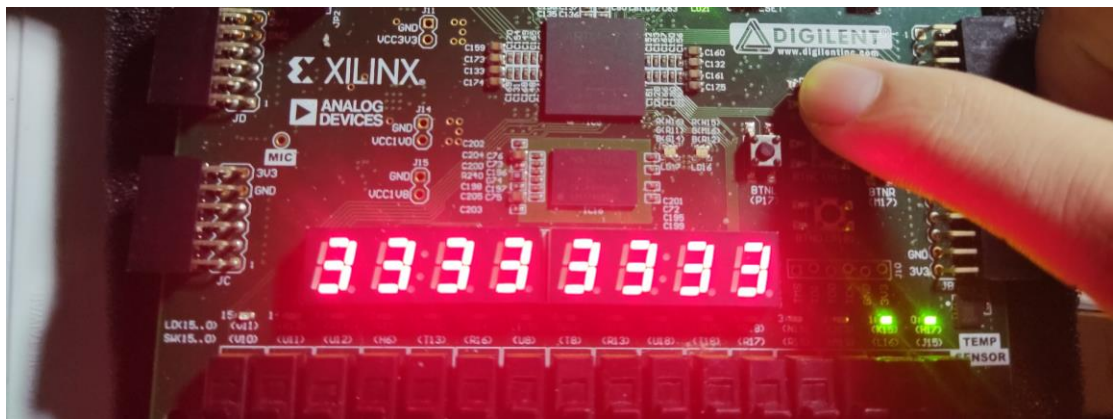
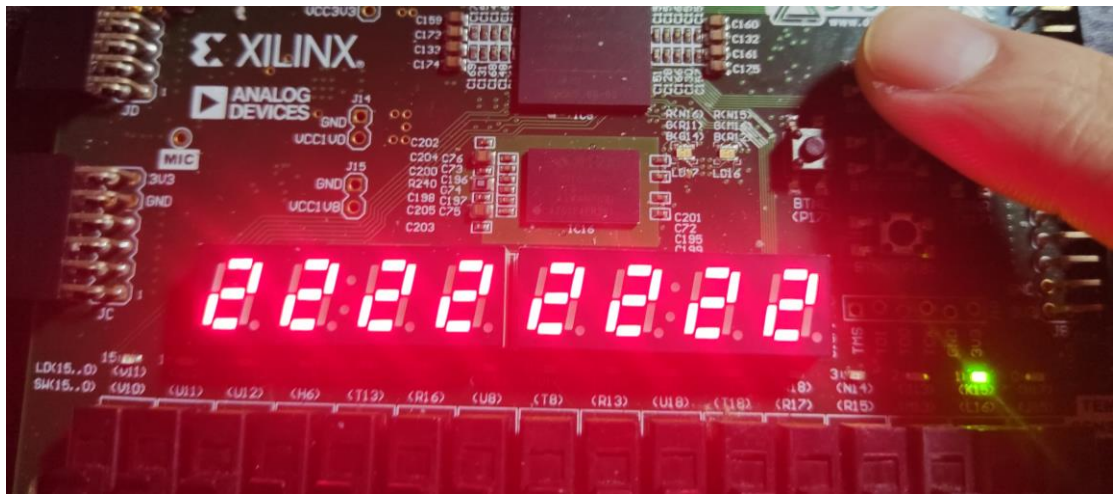
五、实验结果

（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））

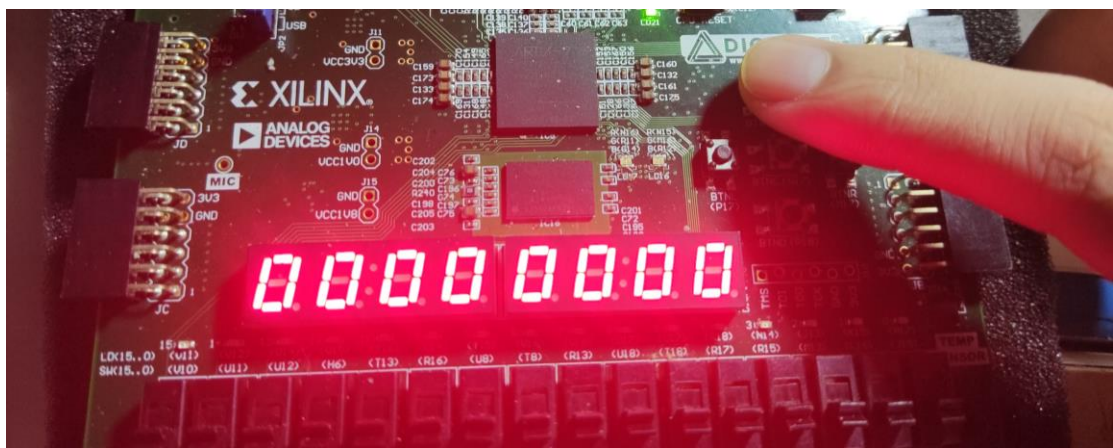
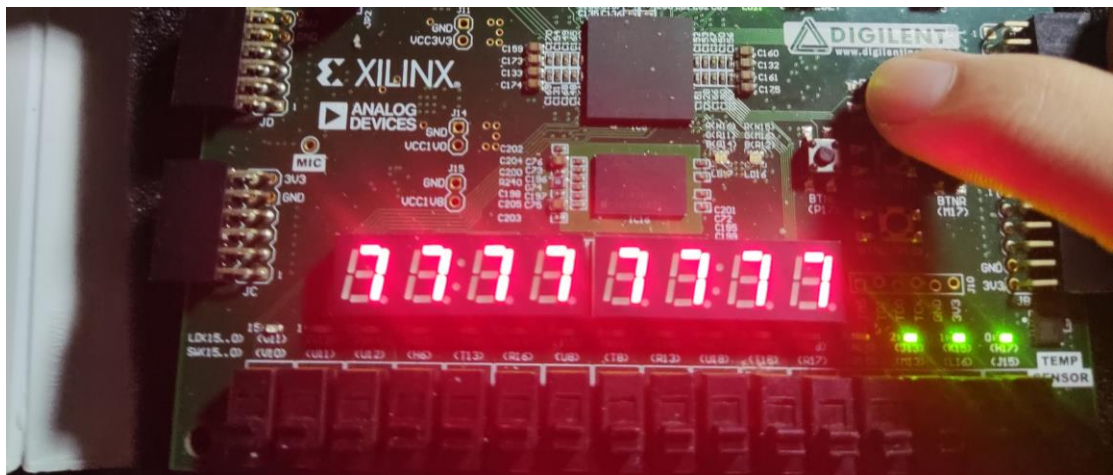
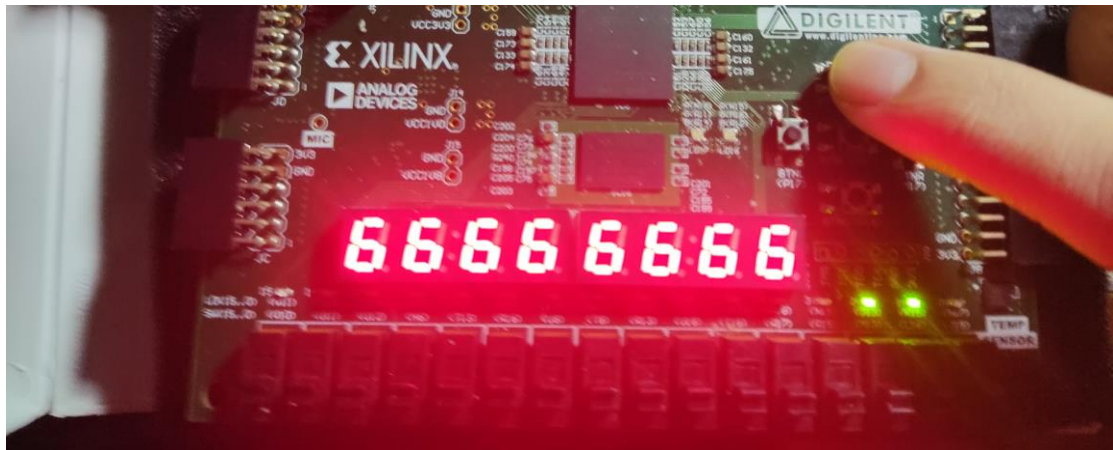
1.模 8 计数器





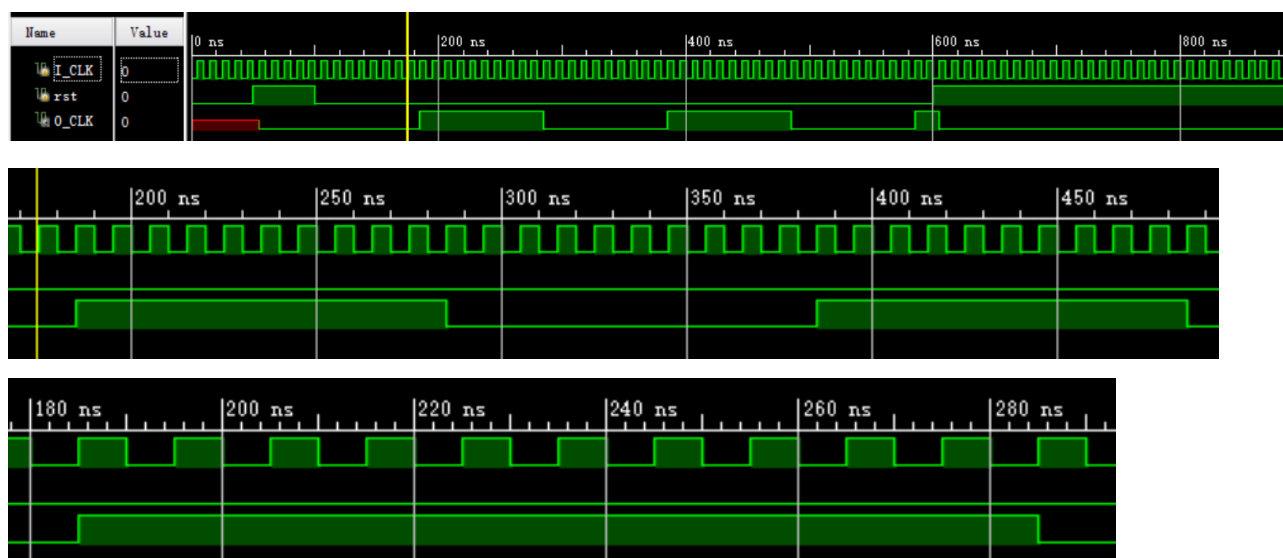




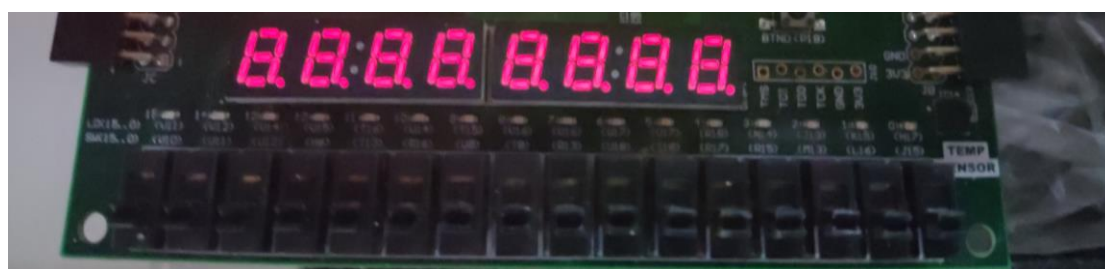
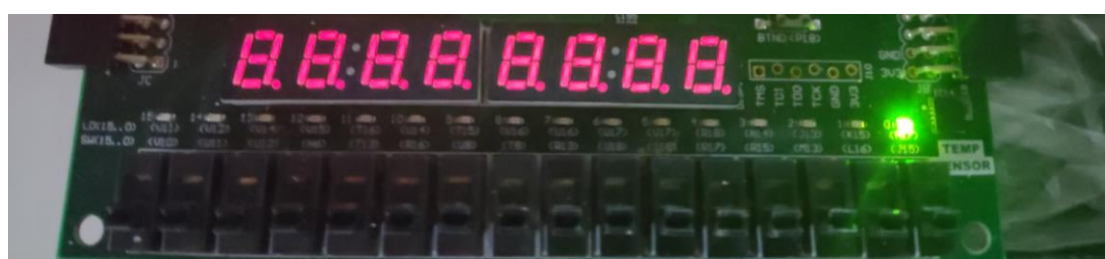
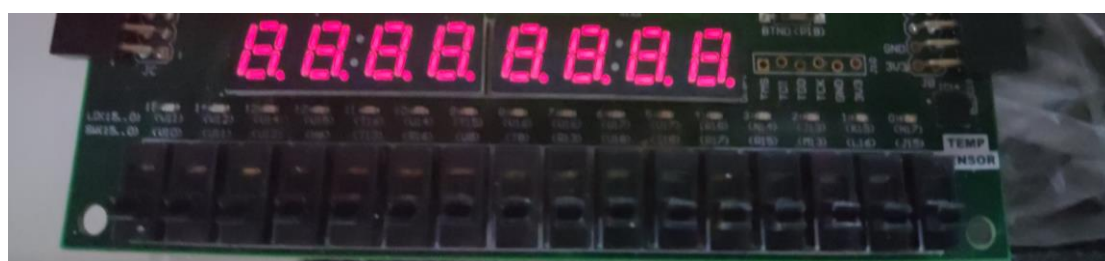
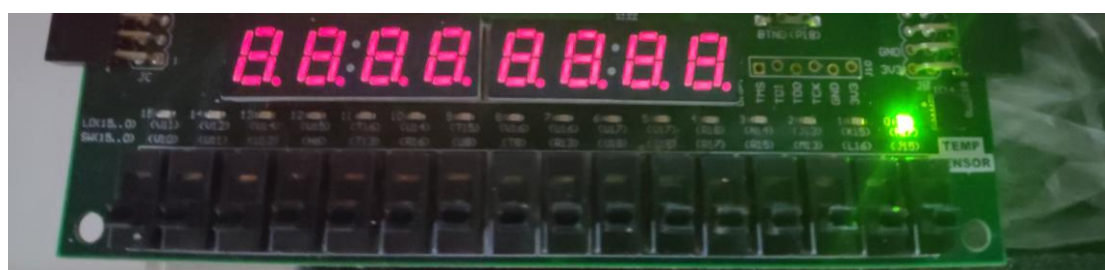


按此规律循环。

## 2.分频器



可以看到分频后为原频率的二十分之一。



按此规律重复。