

同济大学计算机系

数字逻辑课程实验报告



学 号 1652286

姓 名 李辉

专 业 计算机

授课老师 张冬冬

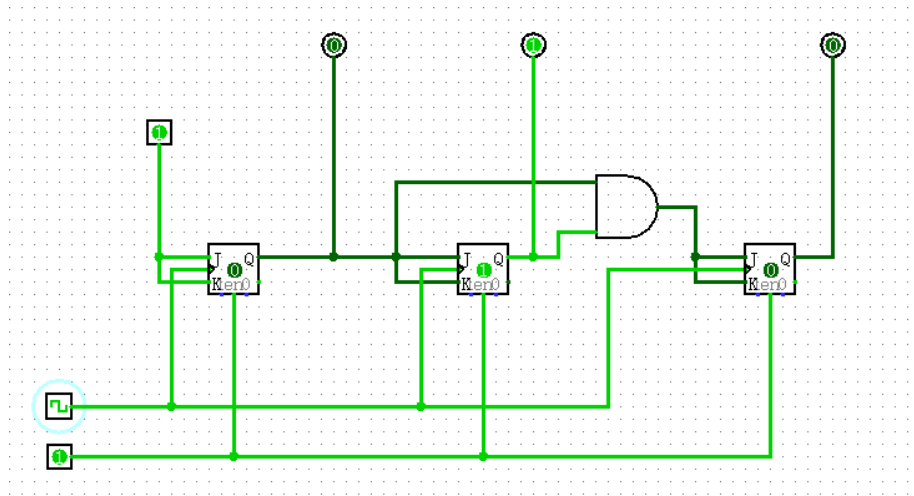
一、实验内容

1、实现同步模 8 计数器（计数器需要通过 logisim 验证逻辑，并通过实例化分频器、七段译码显示管、JK 触发器后下板验证，使数字译码管通过数码管显示）；

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) 6.7_1 Counter8 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) 6.7_1 Counter8 模 8 计数器

```
`timescale 1ns / 1ns
module Counter8(
    input CLK,
    input rst_n,
    output [2:0] oQ,
    output [6:0] oDisplay
);
    reg rst;
    wire clk;
    wire [3:0] oq;
    assign oq[3]=0;
    assign oq[2:0]=oQ[2:0];
```

```

//defparam div.N=100000000;

initial begin
    rst=1;
    #1 rst=0;
end
Divider div(
.I_CLK(CLK),.Rst(rst),.O_CLK(clk)
);

JK_FF FF0(
.CLK(clk),.J(1),.K(1),.RST_n(rst_n),.Q1(oQ[0]),.Q2()
);
JK_FF FF1(
.CLK(clk),.J(oQ[0]),.K(oQ[0]),.RST_n(rst_n),.Q1(oQ[1]),.Q2()
);
JK_FF FF2(
.CLK(clk),.J(oQ[0]&&oQ[1]),.K(oQ[0]&&oQ[1]),.RST_n(rst_n),.Q1(oQ[2]),.Q2()
);

display7 disp(
.iData(oq),.oData(oDisplay)
);
Endmodule

```

Counter8 为顶层模块，其对分频器、J、K 触发器、七段译码数码管进行实例化，使数字显示在数码管上，分频器同步复位信号 rst 处于上升沿高电平时有效。

子模块 1：J_K 触发器

```

module JK_FF(
    input CLK,
    input J,
    input K,
    input RST_n,
    output reg Q1,
    output reg Q2
);

always @(posedge CLK or negedge RST_n)
begin
    if (!RST_n)
begin
        Q1=0;
        Q2=1;

```

```

        end
    else if(J==1&&K==1)
        begin
            Q1=~Q1;
            Q2=~Q2;
        end
    else if(J==0&&K==0);
    else
        begin
            Q1=J;
            Q2=K;
        end
    end
end
Endmodule

```

子模块 2：七段译码数码管

```

module display7(
    input [3:0] iData,
    output [6:0] oData
);
    assign
oData[6]=iData[3]&iData[2]|iData[3]&iData[1]|(~iData[3])&(~iData[2])&(~iData[1])|(~iData[3])
&iData[2]&iData[1]&iData[0];
    assign
oData[5]=iData[3]&iData[2]|iData[3]&iData[1]|iData[1]&iData[0]|(~iData[3])&(~iData[2])&iDat
a[0]|(~iData[3])&(~iData[2])&iData[1];
    assign
oData[4]=iData[3]&iData[2]|iData[3]&iData[1]|(~iData[3])&iData[0]|(~iData[3])&iData[2]&(~iD
ata[1])|iData[3]&(~iData[2])&(~iData[1])&iData[0];
    assign
oData[3]=iData[3]&iData[2]|iData[3]&iData[1]|(~iData[3])&(~iData[2])&(~iData[1])&iData[0]|i
Data[2]&(~iData[1])&(~iData[0])|iData[2]&iData[1]&iData[0];
    assign
oData[2]=iData[3]&iData[2]|iData[3]&iData[1]|(~iData[3])&(~iData[2])&iData[1]&(~iData[0]);
    assign
oData[1]=iData[3]&iData[2]|iData[3]&iData[1]|iData[2]&(~iData[1])&iData[0]|iData[2]&iData[1
]&(~iData[0]);
    assign
oData[0]=iData[3]&iData[2]|iData[3]&iData[1]|iData[2]&(~iData[1])&(~iData[0])|(~iData[3])&(
~iData[2])&(~iData[1])&iData[0];
Endmodule

```

子模块 3: 分频器

```
module Divider(  
    input I_CLK,  
    input Rst,  
    output O_CLK  
);  
    reg o_clk;  
    reg [26:0] count;  
    parameter N=100000000;  
    always @(posedge I_CLK)  
    begin  
        if (Rst)  
            begin  
                count=0;  
                o_clk=0;  
            end  
        else begin  
            if (count==N/2-1)  
                begin  
                    o_clk=!o_clk;  
                    count=0;  
                end  
            else begin  
                count=count+1;  
            end  
        end  
    end  
    assign O_CLK=o_clk;  
endmodule
```

(2)6.7_2 Divider 分频器

`timescale 1ns / 1ns

```
module Divider(  
    input I_CLK,  
    input Rst,  
    output O_CLK  
);  
    reg o_clk;  
    reg [26:0] count;  
    parameter N=20;  
    always @(posedge I_CLK)  
    begin  
        if (Rst)  
            begin
```

```

        count=0;
        o_clk=0;
    end
    else begin
        if (count==N/2-1)
            begin
                o_clk=!o_clk;
                count=0;
            end
        else begin
            count=count+1;
        end
    end
end
end
assign O_CLK=o_clk;
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) 6.7_1 Counter8 模 8 计数器

```

`timescale 1ns / 1ns
module Counter8_tb();
    reg CLK;
    reg rst_n;
    wire [2:0] oQ;
    wire [6:0] oDisplay;
    Counter8 uut(
        .CLK(CLK),
        .rst_n(rst_n),
        .oQ(oQ),
        .oDisplay(oDisplay)
    );
    initial
    begin
        rst_n=0;
        #10
        rst_n=1;
        CLK=0;
        #1000
        $finish;
    end
    always #10 CLK=~CLK;

```

```
endmodule
```

(2)6.7_2 Divider 分频器

```
`timescale 1ns / 1ns
```

```
module Divider_tb;
```

```
reg I_CLK;
```

```
reg rst;
```

```
wire O_CLK;
```

```
initial begin
```

```
I_CLK=0;
```

```
rst=1;
```

```
#4 rst=0;
```

```
end
```

```
always begin
```

```
#2 I_CLK=~I_CLK;
```

```
end
```

```
Divider div(I_CLK(I_CLK),rst(rst),O_CLK(O_CLK));
```

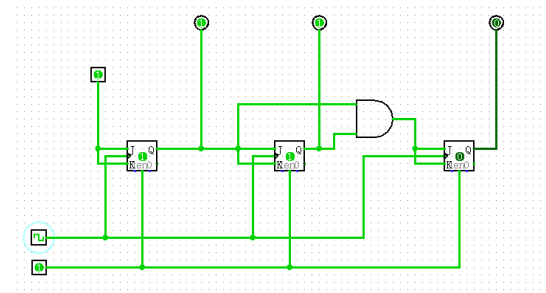
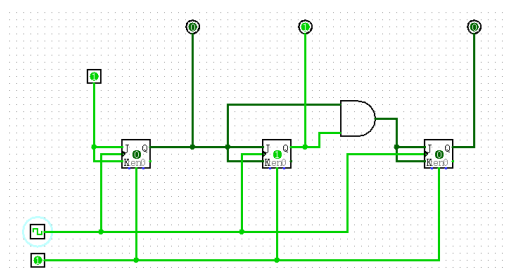
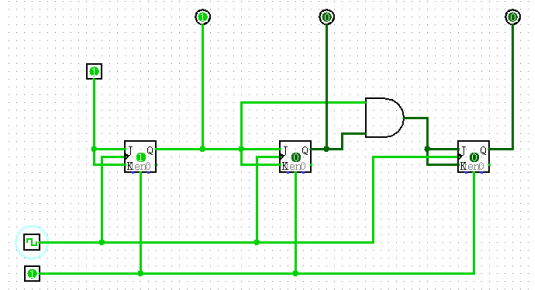
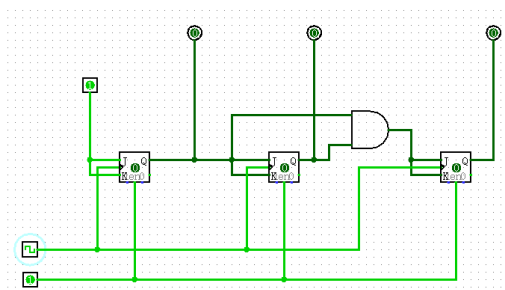
```
endmodule
```

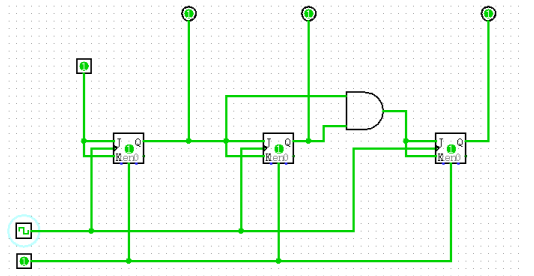
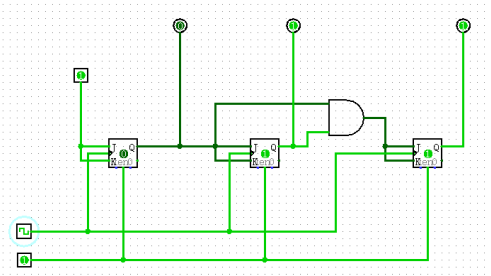
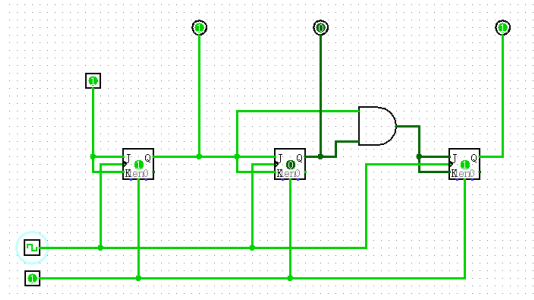
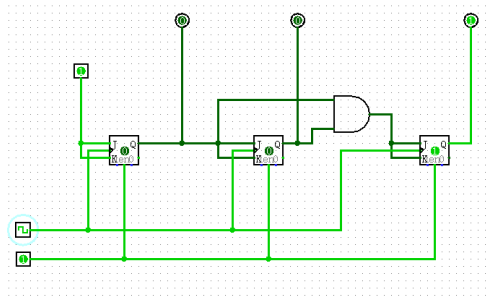
五、实验结果

（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））

logisim 逻辑验证图

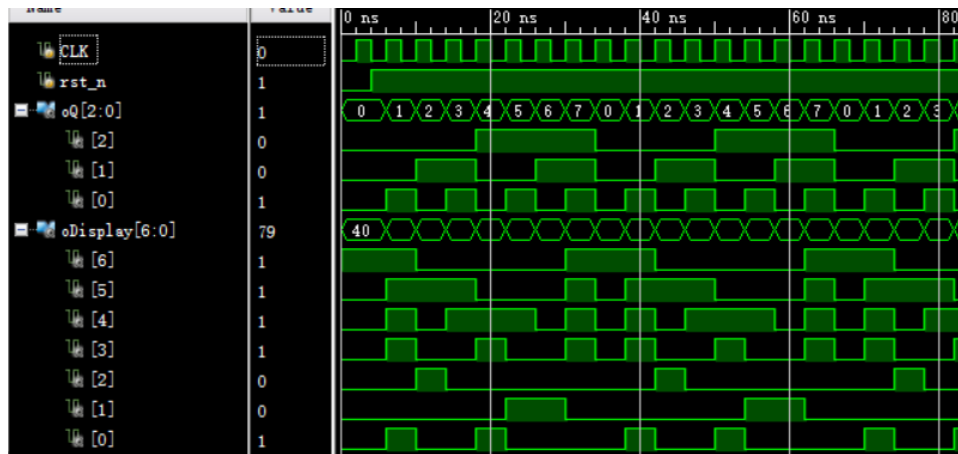
(1)6.7_1 Counter8 模 8 计数器



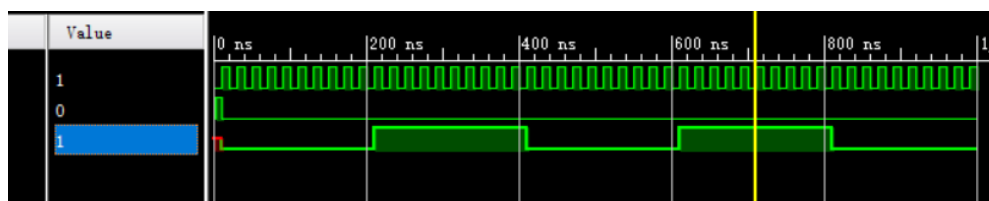


Modelsim 仿真波形图

(1)6.7_1 Counter8 模 8 计数器

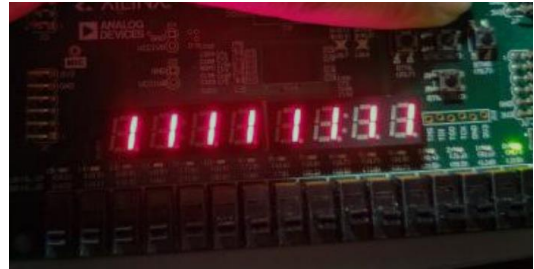
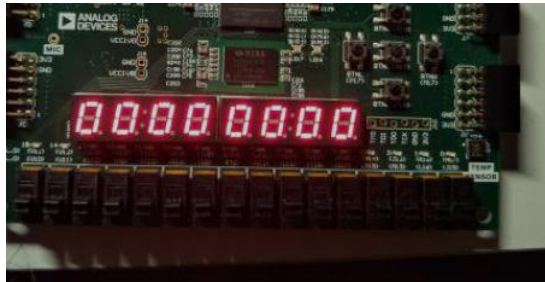


(2)6.7_2 Divider 分频器



下板后贴图

(1)6.7_1 Counter8 模 8 计数器



注：通过分频器实现对 E3 时钟的分频，将 100MHz 的频率分为 1Hz 的频率，即一秒变化一次，使肉眼可见。