

浙江大学

本科实验报告

课程名称： 计算机逻辑设计基础

姓 名： 刘晨

学 院： 计算机科学与技术学院

专 业： 图灵班

学 号： 3190104666

指导教师： 董亚波

2021 年 1 月 7 日

浙江大学实验报告

课程名称： 计算机逻辑设计基础

实验类型： 综合

实验项目名称： 计数器、定时器设计与应用

学生姓名： 刘晨

专业： 图灵 1901

学号： 3190104666

同组学生姓名：

指导老师： 董亚波

实验地点： 东 4-509

实验日期： 2021 年 1 月 7 日

一、实验目的和要求

掌握同步四位二进制计数器 74LS161 的工作原理和设计方法

掌握时钟/定时器的工作原理与设计方法

二、实验内容和原理

内容：

任务 1：采用行为描述设计同步四位二进制计数器 74LS161

任务 2：基于 74LS161 设计时钟应用

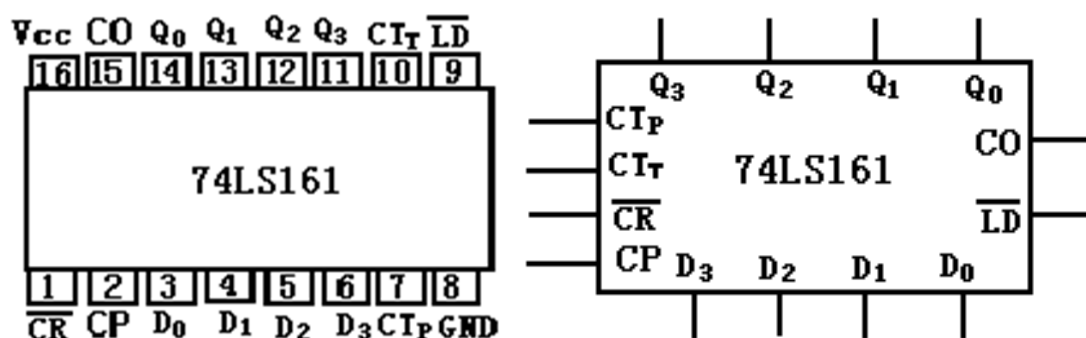
原理：

1. 同步四位二进制计数器 74LS161

74LS161 是常用的四位二进制可预置的同步加法计数器，可灵活运用在各种数字电路，实现分频器等很多重要的功能

2. 74LS161 功能描述

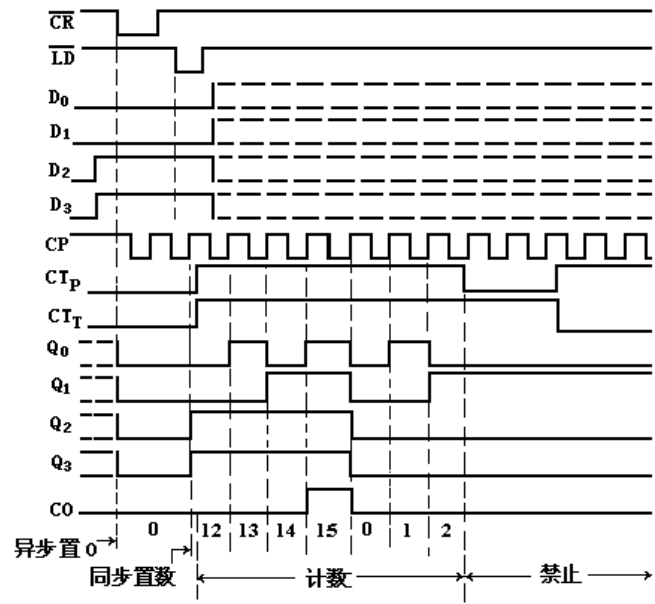
原理图如下：



功能表如下：

输 入					输 出				
\overline{CR}	\overline{LD}	CT_P	CT_T	CP	$D_3D_2D_1D_0$	$Q_3Q_2Q_1Q_0$			
0	×	×	×	×	×	×	×	×	×
1	0	×	×	↑	$d_3d_2d_1d_0$	$d_3d_2d_1d_0$			
1	1	0	1	×	×	×	×	×	×
1	1	×	0	×	×	×	×	×	×
1	1	1	1	↑	×	×	×	×	×

74LS161 时序图如下：

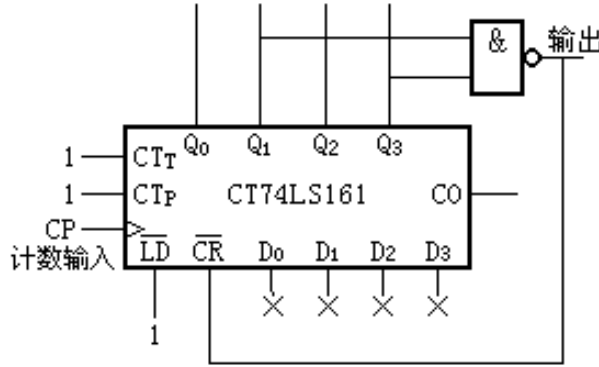


3. 实现十进制计数器

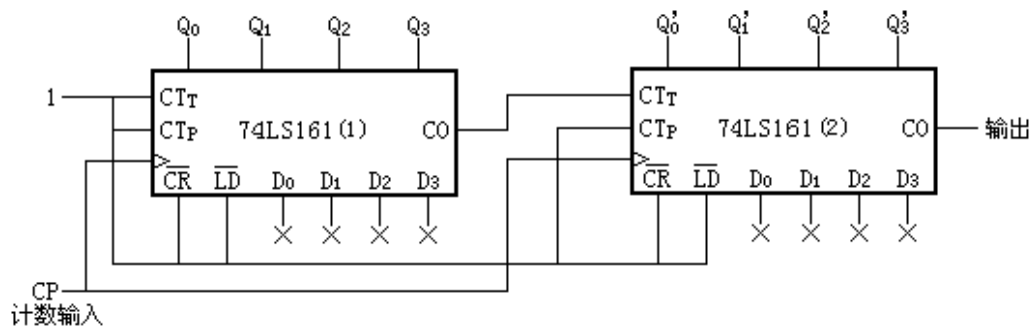
利用与非门判断终止状态 1010 ；

实现十进制计数 （0000 到 1001）

改变与非门的输入信号 ，可以实现其它进制计数。改变与非门输出信号的功能和输入信号，可以实现同步加载。如下为十进制计数器的设计样板图：



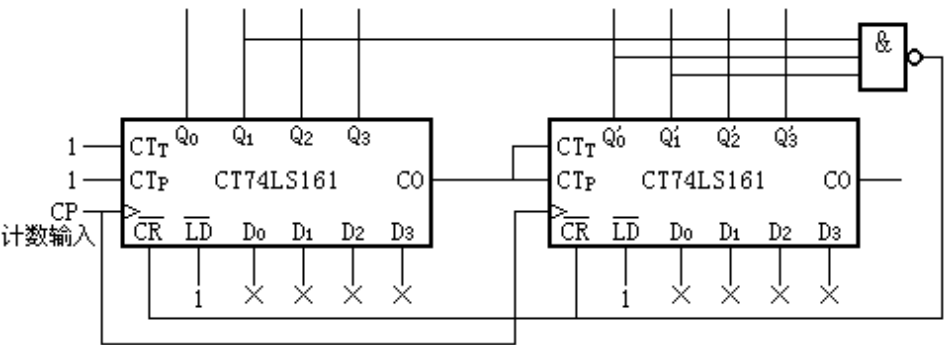
4. 实现 16×16 进制计数器



在计到 1111 以前，CO1=0，高位片保持原状态不变

在计到 1111 时，CO1=1，高位片在下一个 CP 加一

5. 实现 50 进制计数器



十进制数 50 对应的二进制数为 0011 0010

实现从 0000 0000 到 0011 0001 的 50 进制计数器

6. 数字时钟

设计一个数字钟，使用 74LS161 模块，设计 60 进制和 24 进制计数器，实现 24 小时内时间的实时显示。

数字钟的初值通过初始化语句来实现，选择大实验板上的 6 个数码管显示，前两位显示小时的十位和个位，中间两位显示分钟的十位和个位，最后两位显示秒的十位和个位。

三、实验过程和数据记录

任务 1：采用行为描述设计同步四位二进制计数器 74LS161

新建工程，工程名称用 My74LS161。

Top Level Source Type 用 HDL。

用行为描述设计；CR 是异步清零，低电平有效；LD 是同步置位，低电平有效

主要模块代码如下：

Top 模块：

```
module My74LS161new(
    input [3:0]Beginning,//从多少开始计数
    input [3:0]Initial,//初始值
    input CTP,
    input CTT,
    input CR,
    input CP,
    input Ld,//LD
    output CO,
    output reg [3:0]Q
);

always @(posedge CP,negedge CR) begin
    if(~CR) begin
        Q <= Initial;
    end

    else
        if(~Ld) begin
            Q <= Beginning;
        end
    else begin
        if(CTT & CTP)
            begin
                case(Q)
                    4'b0000 : Q <= 4'b0001;//0->1
                    4'b0001 : Q <= 4'b0010;//1->2
                    4'b0010 : Q <= 4'b0011;//2->3
                    4'b0011 : Q <= 4'b0100;//3->4
                    4'b0100 : Q <= 4'b0101;//4->5
                    4'b0101 : Q <= 4'b0110;//5->6
                    4'b0110 : Q <= 4'b0111;//6->7
                    4'b0111 : Q <= 4'b1000;//7->8
                    4'b1000 : Q <= 4'b1001;//8->9
                    4'b1001 : Q <= 4'b1010;//9->10
                    4'b1010 : Q <= 4'b1011;//10->11
                    4'b1011 : Q <= 4'b1100;//11->12
                    4'b1100 : Q <= 4'b1101;//12->13
                    4'b1101 : Q <= 4'b1110;//13->14
                    4'b1110 : Q <= 4'b1111;//14->15
                    4'b1111 : Q <= 4'b0000;//15->0
                endcase
            end
        end
    end
end
```

```
                                endcase
                                end
                                end
                                end
                                assign CO = (&Q) &CTT;
endmodule
```

仿真代码如下：

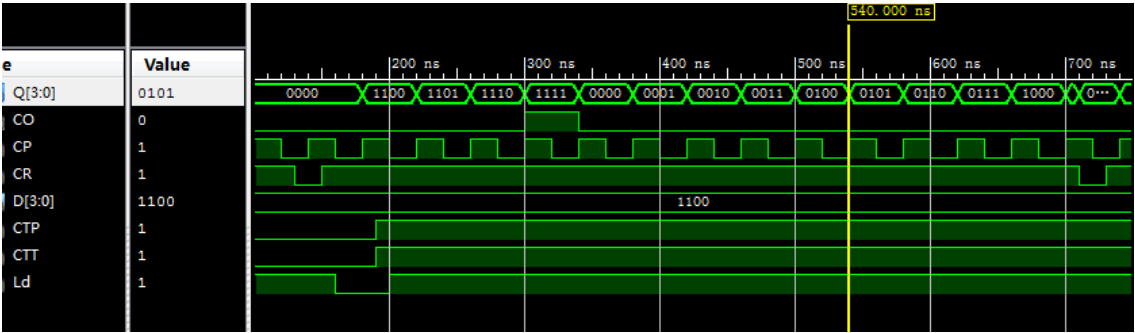
```
initial begin
    CR = 0;
    D = 0;
    CTP = 0;
    CTT = 0;
    Ld = 0;

    #100;
    CR = 1;
    Ld = 1;
    D = 4'b1100;
    CTT = 0;
    CTP = 0;
    #30 CR = 0;
    #20 CR = 1;
    #10 Ld = 0;
    #30 CTT = 1;
    CTP = 1;
    #10 Ld = 1;

    #510;
    CR = 0;
    #20 CR = 1;
    #500;

end
```

仿真结果如下：



任务 2：基于 74LS161 设计时钟应用

新建工程，工程名称用 MyClock。

Top Level Source Type 用 HDL，用结构化描述设计

调用以下模块：My74LS161；分频模块，用 100ms 作为秒的驱动时钟；显示模块
下载验证

主要模块代码如下：

Top 模块：

```
module Top(  
    input wire clk,  
    input wire rst,  
    input wire [2:0]SEG,  
    output wire SEGCLK,//seg clk  
    output wire SEGCLR,//seg clear  
    output wire SEGDT,//seg data  
    output wire SEGEN//seg in  
);  
wire [31:0]clk_div;  
wire clk_100ms;  
wire [23:0]num;  
wire [3:0]sout;  
clkdiv m1(clk, 1'b0, clk_div);  
seg_drv m2(.clkIO(clk_div[3]), .clkScan(clk_div[15:14]), .clkBlink(clk_div[25]),  
.data({8'b0,num}), .point(8'h0), .LES(8'h0),.sout(sout));  
clk_100ms m3(.clk(clk),.clk_100ms(clk_100ms));  
MyClock m4(.clk(clk_100ms),.dispnum(num),.rst(rst));  
assign SEGCLK = sout[3];  
assign SEGDT = sout[2];  
assign SEGEN = sout[1];  
assign SEGCLR = sout[0];  
endmodule
```

MyClock 模块：

```
module MyClock(  
    input rst,  
    input clk,  
    output [23:0]dispnum  
);  
reg [23:0]init;  
wire secondco;
```

```

wire minco;
wire hourco;
initial init <= 24'b0010_0011_0101_1000_0000_0000;
second60 s(.rst(rst),.lnum(init[3:0]),.hnum(init[7:4]),.Q(disnum[7:0]),
.C60(secondco),.clk(clk));
min60 m(.rst(rst),.lnum(init[11:8]),.hnum(init[15:12]),.Q(disnum[15:8])
,.C60(minco),.clk(secondco));
hour24 h(.rst(rst),.lnum(init[19:16]),.hnum(init[23:20]),.Q(disnum[23:1
6]),.C24(hourco),.clk(minco));

endmodule

```

SEG_DRV,clkdiv 等模块在前述实验当中已经出现过。

下载验证

UCF 文件内容如下所示：

```

NET "SEGCLK"LOC = M24 | IOSTANDARD = LVCMOS33;
NET "SEGCLR"LOC = M20 | IOSTANDARD = LVCMOS33;
NET "SEGDT"LOC = L24 | IOSTANDARD = LVCMOS33;
NET "SEGEN"LOC = R18 | IOSTANDARD = LVCMOS33;

NET "SEG[0]"LOC = AF10 | IOSTANDARD = LVCMOS15;
NET "SEG[1]"LOC = AF13 | IOSTANDARD = LVCMOS15;
NET "SEG[2]"LOC = AE13 | IOSTANDARD = LVCMOS15;

NET "rst" LOC = AA10 | IOSTANDARD = LVCMOS15;

NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS15;

```

实验结果：随着时间的推进七段数码管上面的数值在不断的递增。

四、实验结果分析

总体和预期实验结果内容一致。

硬件描述代码和原理图所要实现的功能一致。

仿真结果符合预期。

Verliog 代码内容实现了预期功能。

在开发板上的实验结果验证了应有的功能。

五、讨论与心得

在这次实验中，我们学会了计数器、定时器设计与应用，了解了计数器、定时器

设计与应用的方法。我在实验中深刻体会到了实验当中出现的问题和解决的办法，以及协作的重要性。在这次实验当中，我在实验当中遇到了编写 Verilog 代码和 top 模块这几个问题，所幸通过老师和同学的帮助，这些问题顺利解决了，在开发板上面的实验结果也验证了这个结果。