

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:	李瀚轩
学院:	竺可桢学院
系:	所在系
专业:	计算机科学与技术
学号:	3220106039
指导教师:	董亚波

2024 年 1 月 3 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: 寄存器及寄存器传输设计

学生姓名: 李瀚轩 专业: 计算机科学与技术 学号: 3220106039

同组学生姓名: 指导老师: 董亚波

实验地点: 东四 509 实验日期: 2023 年 12 月 21 日

一、实验目的和要求

1. 掌握同步四位二进制计数器 74LS161 的工作原理和设计方法
2. 掌握时钟/定时器的工作原理和设计方法

二、实验内容和原理

内容:

1. 任务 1: 采用行为描述设计同步四位二进制计数器 74LS161
2. 任务 2: 基于 74LS161 设计时钟应用

原理

2.1 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

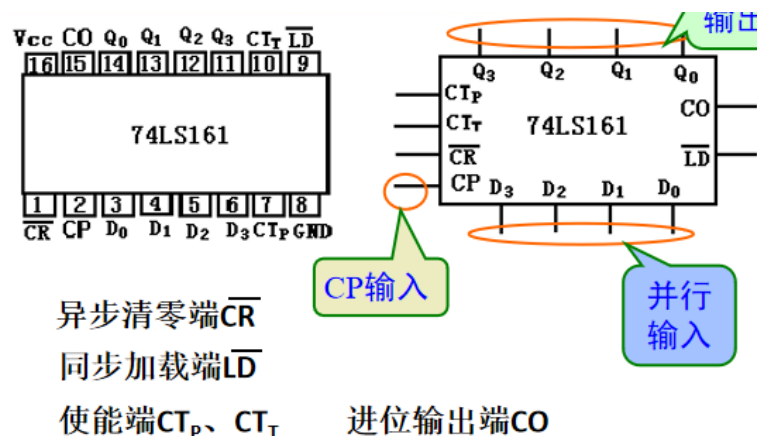


图 1: 74LS161 功能描述

其中 CR 为异步清零端， LD 为同步加载端。异步清零功能最优先。

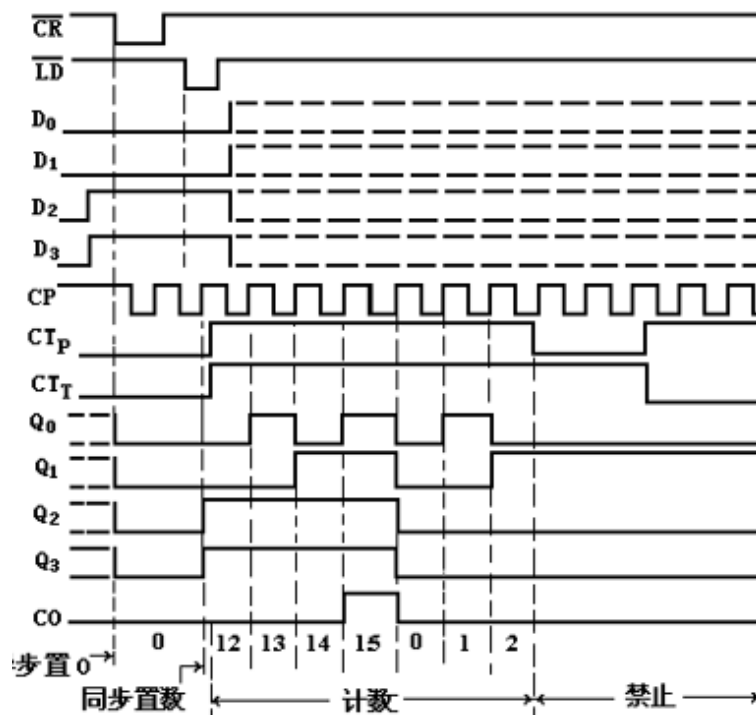


图 2: 74LS161 时序图

当利用 74LS161 实现十进制计数器时，只需要用非门判断终止状态 1010，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，如下图所示：

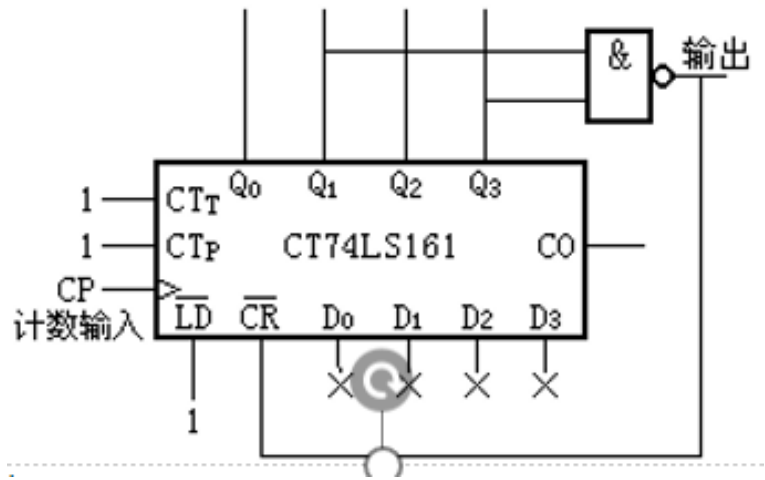


图 3: 十进制计数器

2.2 时钟应用设计

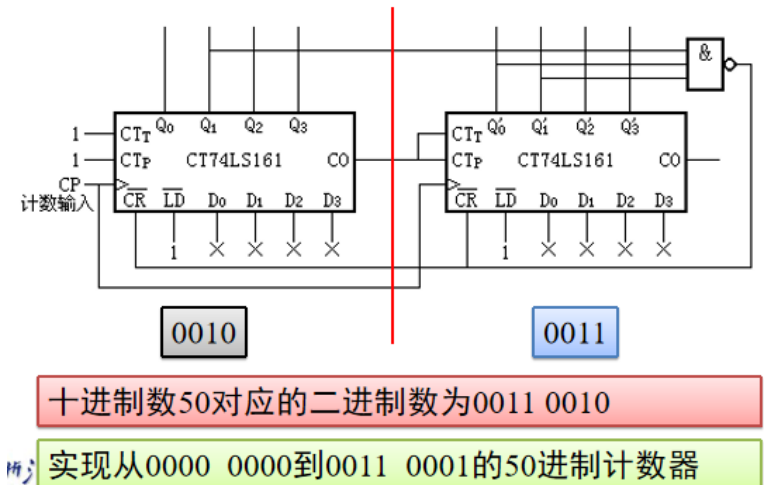


图 4: 50 进制计数器

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。数字钟的初值通过计数器同步置位的方式实现，默认加载 23:58:30。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

3.1 采用行为描述设计同步四位二进制计数器 74LS161

1. 新建工程 My74LS161
2. 用行为描述设计，CR 异步清零，低电平有效，LD 同步置位，低电平有效，代码如下：

```
module My74LS161(  
    input CP,CR,  
    input Ld,  
    input CTP,CTT,  
    input [3:0] D,  
    output reg [3:0] Q,  
    output CO  
);  
    initial Q = 4'b0;  
    always @(posedge CP,negedge CR) begin  
        if(~CR) begin  
            Q <= 0;  
        end  
        else  
            if(~Ld) begin  
                Q <= D;  
            end  
            else begin  
                if (CTT & CTP) begin  
                    Q <= Q + 1;  
                end  
            end  
        end  
    end  
end
```

```

        assign CO=(&Q)&CTT;
    end module

```

3. 仿真，代码如下：

```

module My74LS161_sim;

    // Inputs
    reg CP;
    reg CR;
    reg Ld;
    reg CTP;
    reg CTT;
    reg [3:0] D;

    // Outputs
    wire [3:0] Q;
    wire CO;

    // Instantiate the Unit Under Test (UUT)
    My74LS161 uut (
        .CP(CP),
        .CR(CR),
        .Ld(Ld),
        .CTP(CTP),
        .CTT(CTT),
        .D(D),
        .Q(Q),
        .CO(CO)
    );

    initial begin
        // Initialize Inputs
        CP = 0;
        CR = 0;
        Ld = 0;

```

```

        CTP = 0;
        CTT = 0;
        D = 0;

        // Wait 100 ns for global reset to finish
        #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;

        // Add stimulus here

    end
    always begin
        CP=0;#20;
        CP=1;#20;
    end

endmodule

```

得到如下波形图：

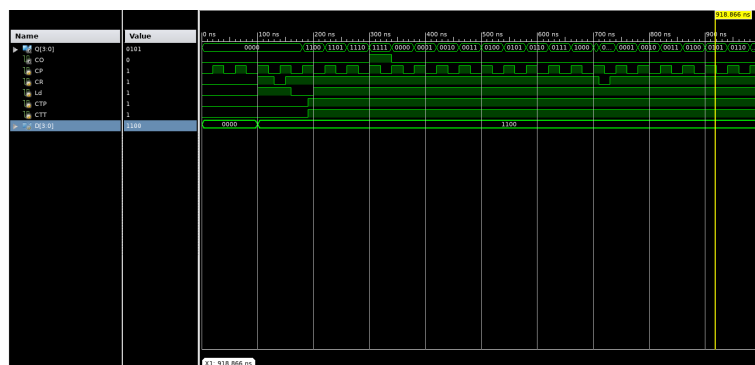


图 5: 仿真波形图

3.2 基于 74LS161 设计时钟应用

1. 新建工程 MyClock
2. 用行为描述设计，调用 My74LS161。调用分频模块，用 100ms 作为秒的驱动时钟。调用实验 13 的 8 位数码管显示模块显示时分秒。用 SW[15] 初始化时钟为 23:58:30 Verilog 代码如下:

```

module top(
    input clk ,
    input SW,
    output SEG_CLK,
    output SEG_DT,
    output SEG_EN,
    output SEG_CLR
//    output [31:0] num
);

wire [31:0] num;
wire [7:0] sec ;
wire [7:0] min;
wire [7:0] hour ;

wire clk_100ms , clk_1ms , start ;

assign num[7:0] = sec ;

```



```

assign num[11:8]=4'b0000;
assign num[19:12]=min;
assign num[23:20]=4'b0000;
assign num[31:24]=hour;

clk_100ms m0(.clk(clk),.clk_100ms(clk_100ms));

My74LS161 m1(clk_100ms,~(sec[3]&sec[1]),~SW,1'b1,1'b1,4'b0000);
My74LS161 m2(clk_100ms,~(sec[6]&sec[5]),~SW,sec[3]&sec[0],s);
My74LS161 m3(clk_100ms,~(min[3]&min[1]),~SW,sec[6]&sec[4]&s);
My74LS161 m4(clk_100ms,~(min[6]&min[5]),~SW,(sec[6]&sec[4]&s));
My74LS161 m5(clk_100ms,~(hour[2]&hour[5]|hour[3]&hour[1]),~SW,1'b1);
My74LS161 m6(clk_100ms,~(hour[2]&hour[5]),~SW,hour[3]&hour[1]);

wire [31:0] clk_div;
clkdiv m9(clk, 1'b0, clk_div);
Load_Gen m10(.clk(clk),.btn_in(clk_100ms),.Load_out(start));

SEGP2S m7(.clk(clk),.reg_num(num),.start(start),.SEG_CLK(SEGCLK),.SEG_DT(SEGDT));

endmodule

```

其中 SEGP2S 为实验 13 的 8 位数码管显示模块。

3. 仿真。分频模块改为 128 个 clk 周期驱动秒计数。将时钟初始值改为 23:59:58。控制 clk 周期宽度和仿真时长，仿真到时钟输出 00:00:02。在 Top 模块中将 6 个 74LS161 模块的输出合并为 num[23:0] 输出，在仿真时用 16 进制显示。同时输出 SEGCLK 和 SEGDT，观察是否有正确输出。仿真代码如下：

```

`timescale 1ns / 1ps

module top_sim;

// Inputs
reg clk;
reg SW;

```

```

// Outputs
wire SEG_CLK;
wire SEG_DT;
wire SEG_EN;
wire SEG_CLR;
wire [31:0] num;

// Instantiate the Unit Under Test (UUT)
top uut (
    .clk(clk),
    .SW(SW),
    .SEG_CLK(SEG_CLK),
    .SEG_DT(SEG_DT),
    .SEG_EN(SEG_EN),
    .SEG_CLR(SEG_CLR),
    .num(num)
);

initial begin
// Initialize Inputs
clk = 0;
SW = 0;

// Wait 100 ns for global reset to finish
#100;

// Add stimulus here
SW=1;
#20;
SW=0;
#3000;
SW=1;
end

always begin

```

```

clk=0;#10;
clk=1;#10;
end

endmodule

```

得到波形图如下：

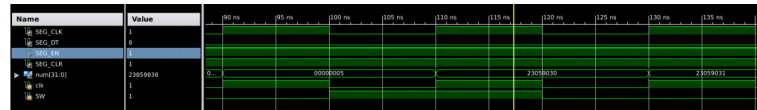


图 6: 仿真波形图

4. 下载验证，引脚约束文件如下：

```

NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
NET "SW" LOC = AA10 | IOSTANDARD = LVCMOS15;
NET "SEG_CLK" LOC = M24 | IOSTANDARD = LVCMOS33 ;
NET "SEG_CLR" LOC = M20 | IOSTANDARD = LVCMOS33 ;
NET "SEG_DT" LOC = L24 | IOSTANDARD = LVCMOS33 ;
NET "SEG_EN" LOC = R18 | IOSTANDARD = LVCMOS33 ;

```

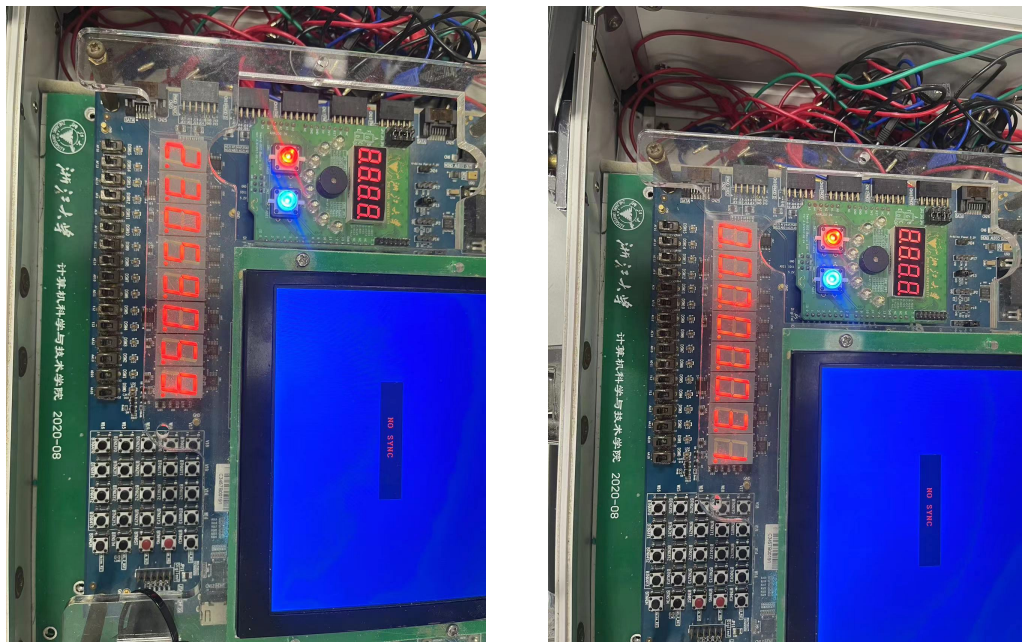


图 7: 验证时钟

由上图可知，时钟正常工作。这里只截取了两张图，设计的正确性已在验收时得到验证。

四、实验结果分析

相关实验结果都在前文给出，实验结果符合要求。

五、讨论与心得

这次实验相比上次来说是简单了挺多，任务 1 的逻辑功能比较简单，而任务二在清楚功能后便也水到渠成。但是计逻实验终于结束了！我不知道是应该开心还是怀念，或者是二者皆有吧。固然有调不通代码，上板失败的痛苦经历，但不得不说，在看到写的代码，画的电路图后在实验板上成功工作的结果后，我都会异常兴奋与开心，感觉努力得到了回报。

总而言之，在这一学期的计逻实验中，我学到了很多，也将理论课上的知识应用到了实践中，收获颇丰。感谢老师和助教一学期的辛勤付出与孜孜不倦的答疑解惑！