

# 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: 仇国智

学 院: 竺可桢学院

专 业: 计算机科学与技术

学 号: 3220102181

指导教师: 董亚波

2023 年 12 月 16 日

## 一、实验目的和要求

### 1 实验目的

- 1.1 掌握同步四位二进制计数器 74LS161 的工作原理和设计方法
- 1.2 掌握时钟/定时器的工作原理与设计方法

### 2 实验任务

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

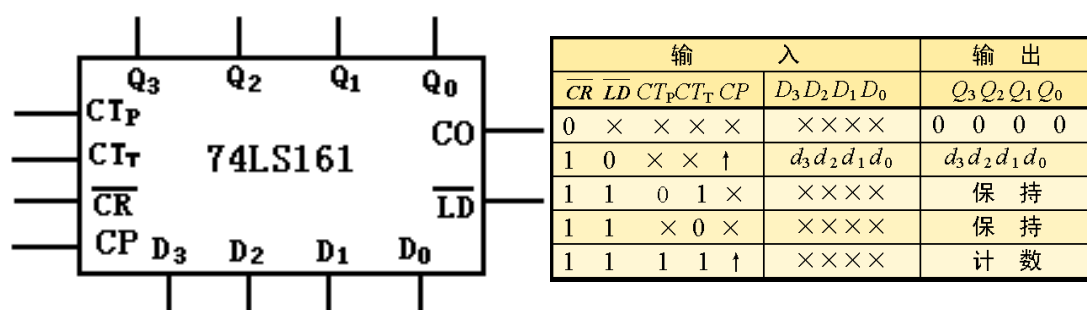
## 二、实验内容和原理

### 1 实验内容

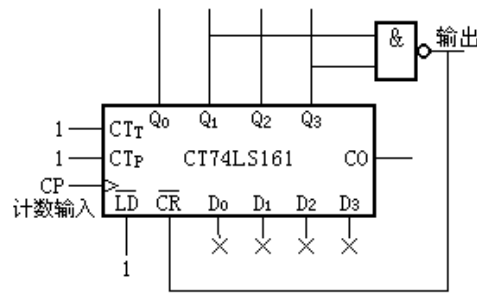
- 1.1 任务 1: 采用行为描述设计同步四位二进制计数器 74LS161
- 1.2 任务 2: 基于 74LS161 设计时钟应用

### 2 实验原理

- 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。接口: 异步清零端  $\overline{CR}$ , 同步加载端  $\overline{LD}$ , 使能端  $CT_P$ 、 $CT_T$ , 进位输出端  $CO$ , 状态输出  $Q$ , 并行输入  $D$ 。



- 2.2 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。
- 2.3 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,可以实现同步加载控制结构不变,每路输入向量化

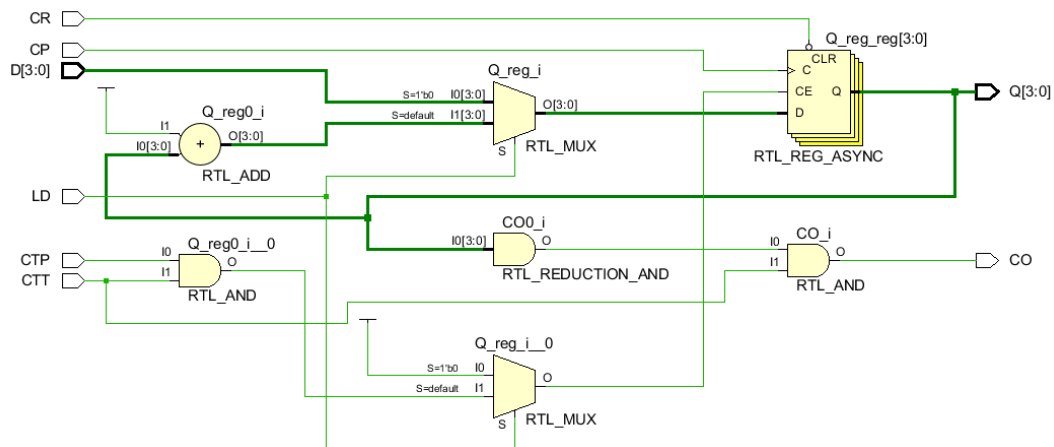


### 三、实验过程和数据记录

#### 1 四位二进制计数器 74LS161

##### 1.1 使用 verilog 语言进行设计

```
`timescale 1ns / 1ps
module My74LS161 (
    input wire [3:0] D,
    input wire CP,
    CR,
    LD,
    CTT,
    CTP,
    output wire CO,
    output wire [3:0] Q
);
assign CO =(Q)&CTT;
reg[3:0] Q_reg;
assign Q=Q_reg;
initial begin
    Q_reg = 4'b0000;
end
always @(posedge CP or negedge CR) begin
    if(!CR) begin
        Q_reg <= 4'b0000;
    end
    else if(!LD) begin
        Q_reg <= D;
    end
    else if(CTP&CTT) begin
        Q_reg <= Q_reg + 1;
    end
end
end
endmodule
```



## 1.2 编写激励代码，进行波形仿真

激励代码：

```
`timescale 1ns / 1ps
module My74LS161_test1 ();
    reg [3:0] D;
    reg CP, CR, LD, CTT, CTP;
    wire CO;
    wire [3:0] Q;
    My74LS161 test1 (
        .D (D),
        .CP (CP),
        .CR (CR),
        .LD (LD),
        .CTT (CTT),
        .CTP (CTP),
        .CO (CO),
        .Q (Q)
    );
    always begin
        CP = 0;
        #20;
        CP = 1;
        #20;
    end
    initial begin
        CR = 0;
        D = 0;
        CTP = 0;
        CTT = 0;
        LD = 0;
    end
endmodule
```

```

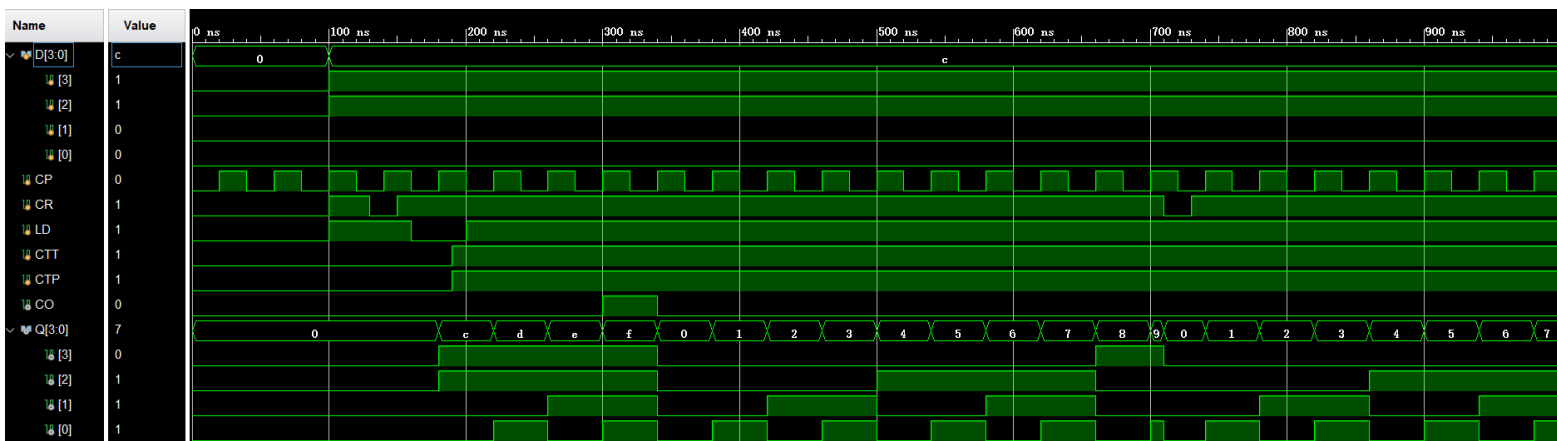
#100;
CR = 1;
LD = 1;
D = 4'b1100;
CTT = 0;
CTP = 0;
#30 CR = 0;
#20 CR = 1;
#10 LD = 0;
#30 CTT = 1;
CTP = 1;
#10 LD = 1;

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

endmodule

```

仿真波形：



## 2 基于 74LS161 设计时钟应用

### 2.1 用 verilog 语言进行设计

#### MyClock 模块

```

`timescale 1ns / 1ps
module MyClock (
    input wire clk,
    output wire SEGCLK,

```

```

        output wire SEGCLR,
        output wire SEGDT,
        output wire SEGEN
    );
    wire [31:0] init = 32'h00235830;
    wire [31:0] num;
    assign num[31:24] = init[31:24];
    SEG8 seg8 (
        .num(num),
        .clk(clk),
        .SEGCLK(SEGCLK),
        .SEGCLR(SEGCLR),
        .SEGDT(SEGDT),
        .SEGEN(SEGEN)
    );
    wire [31:0] clk_100ms;
    // 24
    clkdiv d_0 (
        .clk(clk),
        .rst(0),
        .clkdiv(clk_100ms)
    );
    reg [5:0] LD;
    wire [5:0] CTT, CR;
    wire [5:0] CO;
    initial begin
        LD = 6'b000000;
    end
    always @(posedge clk_100ms[24]) begin
        LD <= 6'b111111;
    end
    My74LS161 d_1 (
        .D (init[3:0]),
        .CP (clk_100ms[24]),
        .CR (CR[0]),
        .LD (LD[0]),
        .CTT(CTT[0]),
        .CTP(1),
        .CO (CO[0]),
        .Q (num[3:0])
    );
    My74LS161 d_2 (
        .D (init[7:4]),
        .CP (clk_100ms[24]),

```

```

        .CR (CR[1]),
        .LD (LD[1]),
        .CTT(CTT[1]),
        .CTP(1),
        .CO (CO[1]),
        .Q (num[7:4])
    );
    My74LS161 d_3 (
        .D (init[11:8]),
        .CP (clk_100ms[24]),
        .CR (CR[2]),
        .LD (LD[2]),
        .CTT(CTT[2]),
        .CTP(1),
        .CO (CO[2]),
        .Q (num[11:8])
    );
    My74LS161 d_4 (
        .D (init[15:12]),
        .CP (clk_100ms[24]),
        .CR (CR[3]),
        .LD (LD[3]),
        .CTT(CTT[3]),
        .CTP(1),
        .CO (CO[3]),
        .Q (num[15:12])
    );
    My74LS161 d_5 (
        .D (init[19:16]),
        .CP (clk_100ms[24]),
        .CR (CR[4]),
        .LD (LD[4]),
        .CTT(CTT[4]),
        .CTP(1),
        .CO (CO[4]),
        .Q (num[19:16])
    );
    My74LS161 d_6 (
        .D (init[23:20]),
        .CP (clk_100ms[24]),
        .CR (CR[5]),
        .LD (LD[5]),
        .CTT(CTT[5]),
        .CTP(1),

```

```

        .CO (C0[5]),
        .Q (num[23:20])
    );
    assign CR[0] = ~(num[3] & num[1]);
    assign CR[1] = ~(num[6] & num[5]);
    assign CR[2] = ~(num[11] & num[9]);
    assign CR[3] = ~(num[14] & num[13]);
    assign CR[4] = ~(num[19] & num[17] | (num[21] & num[18]));
    assign CR[5] = ~(num[21] & num[18]);
    assign CTT[0] = 1'b1;
    assign CTT[1] = num[3] & num[0];
    assign CTT[2] = num[6] & num[4] & num[3] & num[0];
    assign CTT[3] = num[11] & num[8] & num[6] & num[4] & num[3] & num[0];
    assign CTT[4] = num[14] & num[12] & num[11] & num[8] & num[6] & num[4]
& num[3] & num[0];
    assign CTT[5] = num[19] & num[16] & num[14] & num[12] & num[11] &
num[8] & num[6] & num[4] & num[3] & num[0];
    // assign CTT[0]=1'b1;
    // assign CTT[1]=num[3]&num[0];
    // assign CTT[2]=num[6]&num[4]&CTT[1];
    // assign CTT[3]=num[11]&num[8]&CTT[2];
    // assign CTT[4]=num[14]&num[12]&CTT[3];
    // assign CTT[5]=num[19]&num[16]&CTT[4];
endmodule

```

2.2 修改 MyClock 模块，并编写激励文件，进行波形仿真：  
修改后的 MyClock 模块：

```

`timescale 1ns / 1ps
module MyClock_test (
    input  wire clk,
    output wire SEGCLK,
    output wire SEGCLR,
    output wire SEGDT,
    output wire SEGEN,
    output wire [23:0] O,
    output wire clk_128
);
    wire [31:0] init = 32'h00235958;
    wire [31:0] num;
    assign num[31:24] = init[31:24];
    assign O=num[23:0];

    SEG8 seg8 (

```



```

        .num(num),
        .clk(clk),
        .SEGCLK(SEGCLK),
        .SEGCLR(SEGCLR),
        .SEGDT(SEGDT),
        .SEGEN(SEGEN)
    );
    wire [31:0] clk_100ms;
    assign clk_128=clk_100ms[7];
    // 24
    clkdiv d_0 (
        .clk(clk),
        .rst(1'b0),
        .clkdiv(clk_100ms)
    );
    /*reg[7:0] clk_100ms;
    always begin
        clk_100ms=8'b0;
        #1;
        clk_100ms=8'b11111111;
        #1;
    end*/
    reg [5:0] LD;
    wire [5:0] CTT, CR;
    wire [5:0] CO;
    initial begin
        LD = 6'b000000;
    end
    always @(posedge clk_100ms[7]) begin
        LD <= 6'b111111;
    end
    end
    My74LS161 d_1 (
        .D (init[3:0]),
        .CP (clk_100ms[7]),
        .CR (CR[0]),
        .LD (LD[0]),
        .CTT(CTT[0]),
        .CTP(1'b1),
        .CO (CO[0]),
        .Q (num[3:0])
    );
    My74LS161 d_2 (
        .D (init[7:4]),
        .CP (clk_100ms[7]),

```

```

        .CR (CR[1]),
        .LD (LD[1]),
        .CTT(CTT[1]),
        .CTP(1),
        .CO (CO[1]),
        .Q (num[7:4])
    );
    My74LS161 d_3 (
        .D (init[11:8]),
        .CP (clk_100ms[7]),
        .CR (CR[2]),
        .LD (LD[2]),
        .CTT(CTT[2]),
        .CTP(1),
        .CO (CO[2]),
        .Q (num[11:8])
    );
    My74LS161 d_4 (
        .D (init[15:12]),
        .CP (clk_100ms[7]),
        .CR (CR[3]),
        .LD (LD[3]),
        .CTT(CTT[3]),
        .CTP(1),
        .CO (CO[3]),
        .Q (num[15:12])
    );
    My74LS161 d_5 (
        .D (init[19:16]),
        .CP (clk_100ms[7]),
        .CR (CR[4]),
        .LD (LD[4]),
        .CTT(CTT[4]),
        .CTP(1),
        .CO (CO[4]),
        .Q (num[19:16])
    );
    My74LS161 d_6 (
        .D (init[23:20]),
        .CP (clk_100ms[7]),
        .CR (CR[5]),
        .LD (LD[5]),
        .CTT(CTT[5]),
        .CTP(1),

```

```

        .CO (CO[5]),
        .Q (num[23:20])
    );
    assign CR[0] = ~(num[3] & num[1]);
    assign CR[1] = ~(num[6] & num[5]);
    assign CR[2] = ~(num[11] & num[9]);
    assign CR[3] = ~(num[14] & num[13]);
    assign CR[4] = ~(num[19] & num[17] | (num[21] & num[18]));
    assign CR[5] = ~(num[21] & num[18]);
    assign CTT[0] = 1'b1;
    assign CTT[1] = num[3] & num[0];
    assign CTT[2] = num[6] & num[4] & num[3] & num[0];
    assign CTT[3] = num[11] & num[8] & num[6] & num[4] & num[3] & num[0];
    assign CTT[4] = num[14] & num[12] & num[11] & num[8] & num[6] & num[4]
& num[3] & num[0];
    assign CTT[5] = num[19] & num[16] & num[14] & num[12] & num[11] &
num[8] & num[6] & num[4] & num[3] & num[0];
    // assign CTT[0]=1'b1;
    // assign CTT[1]=num[3]&num[0];
    // assign CTT[2]=num[6]&num[4]&CTT[1];
    // assign CTT[3]=num[11]&num[8]&CTT[2];
    // assign CTT[4]=num[14]&num[12]&CTT[3];
    // assign CTT[5]=num[19]&num[16]&CTT[4];
endmodule

```

## 2.3 设计实现 4 位 7 段数码管动态扫描显示 DispNum 模块

```

`timescale 1ns / 1ps
module DispNum(
    clk,
    HEXS,
    LES,
    points,
    RST,
    AN,
    Segment
);
    input wire clk,RST;
    input wire[3:0]LES,points;
    output wire[3:0]AN;
    input wire[15:0] HEXS;
    output wire[7:0] Segment;
    wire[31:0]clkd;
    wire [3:0]HEX;
    wire point,LE;

```

```

    clkdiv d0(clk,RST,clkd);
    dispsync d1(HEXS,clkd[18:17],points,LES,HEX,point,LE,AN);
    MyMC14495
d2(HEX[0],HEX[1],HEX[2],HEX[3],point,LE,Segment[0],Segment[1],Segment[2]
],Segment[3],Segment[4],Segment[5],Segment[6],Segment[7]);
endmodule

```

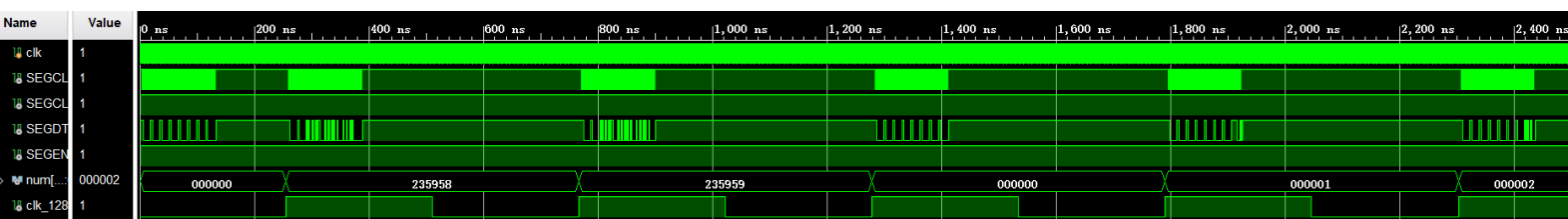
激励代码:

```

`timescale 1ns / 1ps
module MyClock_test2 ();
    reg clk;
    wire SEGCLK;
    wire SEGCLR;
    wire SEGDT;
    wire SEGEN;
    wire [23:0] num;
    wire clk_128;
    MyClock_test test2 (
        .clk(clk),
        .SEGCLK(SEGCLK),
        .SEGCLR(SEGCLR),
        .SEGDT(SEGDT),
        .SEGEN(SEGEN),
        .O(num),
        .clk_128(clk_128)
    );
    always begin
        clk = 0;
        #1;
        clk = 1;
        #1;
    end
    initial begin
        #2500;
        $finish;
    end
endmodule

```

仿真波形



## 2.4 编写引脚约束文件：

```
set_property IOSTANDARD LVCMOS18 [get_ports clk]
set_property PACKAGE_PIN AC18 [get_ports clk]
set_property PACKAGE_PIN M24 [get_ports SEGCLK]
set_property PACKAGE_PIN M20 [get_ports SEGCLR]
set_property PACKAGE_PIN L24 [get_ports SEGDT]
set_property PACKAGE_PIN R18 [get_ports SEGEN]
set_property IOSTANDARD LVCMOS33 [get_ports SEGCLK]
set_property IOSTANDARD LVCMOS33 [get_ports SEGCLR]
set_property IOSTANDARD LVCMOS33 [get_ports SEGDT]
set_property IOSTANDARD LVCMOS33 [get_ports SEGEN]
```

Scalar ports (5)											
clk	IN	AC18	<input checked="" type="checkbox"/>	32	LVCMOS18	1.800			NONE	NONE	
SEGCLK	OUT	M24	<input checked="" type="checkbox"/>	13	LVCMOS33*	3.300	12	SLOW	NONE	FP_VTT_50	
SEGCLR	OUT	M20	<input checked="" type="checkbox"/>	13	LVCMOS33*	3.300	12	SLOW	NONE	FP_VTT_50	
SEGDT	OUT	L24	<input checked="" type="checkbox"/>	13	LVCMOS33*	3.300	12	SLOW	NONE	FP_VTT_50	
SEGEN	OUT	R18	<input checked="" type="checkbox"/>	13	LVCMOS33*	3.300	12	SLOW	NONE	FP_VTT_50	

## 2.5 生成并下载 bit 文件，进行实际测试



## 2.6

### 四、实验结果分析

波形图同 ppt 相同，输出随 O 随选择信号线 S 在 I0,I1,I2,I3 中依次选择输出；下载验证符合预期，小数点随开关拨动依次亮起，数字随开关拨动依次消隐，每按一次键数字会发生变化。但是唯一的不足是数字每次不是只变化 1。

### 五、讨论与心得

按钮按下过程中电压不稳定，会导致数字递增多次，可以使用开关来稳定，或者采用其它方式避免电压波动造成的影响。