

东南大学

《数字逻辑与计算机体系结构（含实验）》

实验报告

实验三 时序逻辑与设计

姓 名：赵舞穹

学 号：61520522

同 组：郑瑞琪

学 号：61520523

专 业：工科试验班

实 验 室：计算机硬件技术

实验时间：2021 年 11 月 12 日

报告时间：2021 年 11 月 22 日

评定成绩：

评阅教师：冯熳

目录

1 实验目的与内容	3
2 基本实验原理	3
1 7474 逻辑电路	4
2 8 位数据/地址锁存器（8D 触发器）	4
3 双 8 位数据/地址锁存器（8D 触发器）	5
4 扩展 CD4518 计数器（双 BCD 加计数器）	6
3 实验内容	7
1 简单时序逻辑	7
2 8 位数据/地址锁存器（8D 触发器）特性实验	7
3 双 8 位数据/地址锁存器（8D 触发器）特性实验（8 × 8 点阵）	8
4 扩展 CD4518 计数器（双 BCD 加计数器）应用	9
5 CAD 逻辑设计与仿真	10
1 基本触发器模块	10
2 74LS273 模块	12
3 Cnt10 定时计数器	15
4 选作探索与应用设计	16
1 74LS161 可加载 4 位二进制计数电路	16
2 24 和 60 进制计数显示系统（时钟）	16
5 实验总结	21
参考文献	21
附录 A：实验报告 L^AT_EX 模板	22
附录 B：Vivado 程序真伪判别	22

一. 实验目的与内容

1. 进一步熟悉数字逻辑与硬件接口基本实验仪器的使用及 TPC 实验装置中基本数字逻辑单元的基本原理和组成结构, 学会利用开关输入/LED 发光管/8 段数码管输出电路及示波器、电平逻辑笔等验证测试组合逻辑基本特性;
2. 分析掌握 TPC 实验装置板上 74LS74 (双 D 触发器, 带置位/清零端)、74LS273 (8 位寄存器, 边沿触发带清零端)、扩展 CD4518 (双 BCD 加计数器) 等时序逻辑电路模块的工作原理, 设计实验方案, 配接开关电平转换、LED 发光管驱动、与或非门等组合逻辑电路, 搭建应用实验系统, 完成接线、演示功能及特性验证;
3. 正确认识 HDL 阻塞与非阻塞赋值语句、`always@` 语句 (块) 的正确使用, 学习掌握 FPGA 数字系统基本触发器、锁存器、计数器、移位器的数据流和行为描述的 HDL 模型开发方法, 收集构建基本模块资源;
4. 完成典型时序逻辑实验电路及测试 (`initial`) 模块的 HDL 模型搭建、模块编程、配置、调试、分析环节, 完成仿真运行分析和输入输出演示实验, 掌握基于工具软件的时序逻辑设计概念和设计流程, 为后续复杂可编程数字模块逻辑设计打下基础.

二. 基本实验原理

实验器材 1

- 清华科教仪器厂 TPC-ZK-II_PCI 微机接口实验装置
- EEEC-020A 计算机组装实验箱一台 (自带电压表)
- RIGAL DG1022U 25MHz 双路波形发生器
- RIGOL DS1072E-EDU 70MHz 双踪数字记忆示波器
- Xilinx Vivado HLx 2017.4
- Ubuntu 20 / Windows 11 (x86_64) (主要实验于 Ubuntu 平台完成)
- 没有具体的开发板, 仅限于线上仿真和使用 Verilog 进行电路设计

(一) 7474 逻辑电路

7474 (D Flip Flop) 逻辑电路如图 1 所示，其中 CD 为 clear，SD 为 set，均为低电平有效。其真值表见图 1。

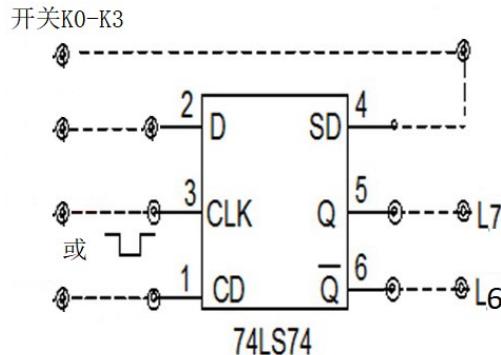


图 1: 74LS74 引脚示意图^[1]

表 1: 74LS74 真值表^[2]

$\overline{S} \ \overline{R}$	Q^{n+1}	\overline{Q}^{n+1}	功 能
1 1	Q^n	\overline{Q}^n	保持
0 1	1	0	置1
1 0	0	1	清0
0 0	1	1	同态， $\overline{S} \ \overline{R}$ 同时由00变为11时，状态不确定

(二) 8 位数据/地址锁存器 (8D 触发器)

74LS273 是 8 位数据/地址锁存器，它是一种带清除功能的 8D 触发器。接线如图 2，利用实验箱简单输入接口实验区的三态总线缓冲器 74LS273。

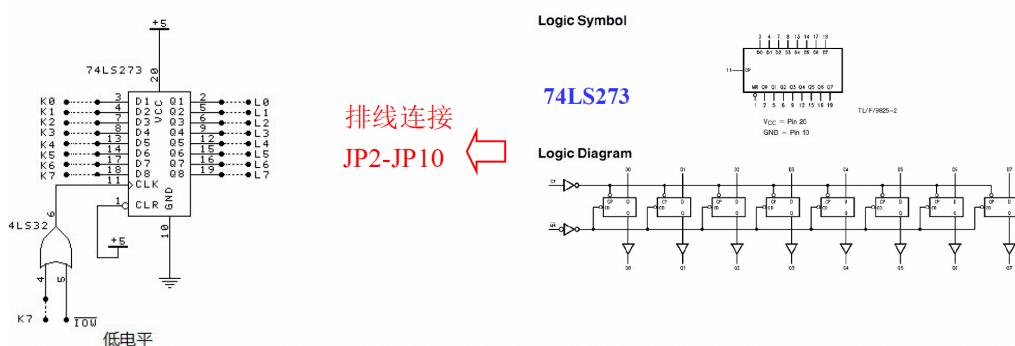


图 2: 74LS273 接线图

(三) 双 8 位数据/地址锁存器 (8D 触发器)

使用实例见图 3.

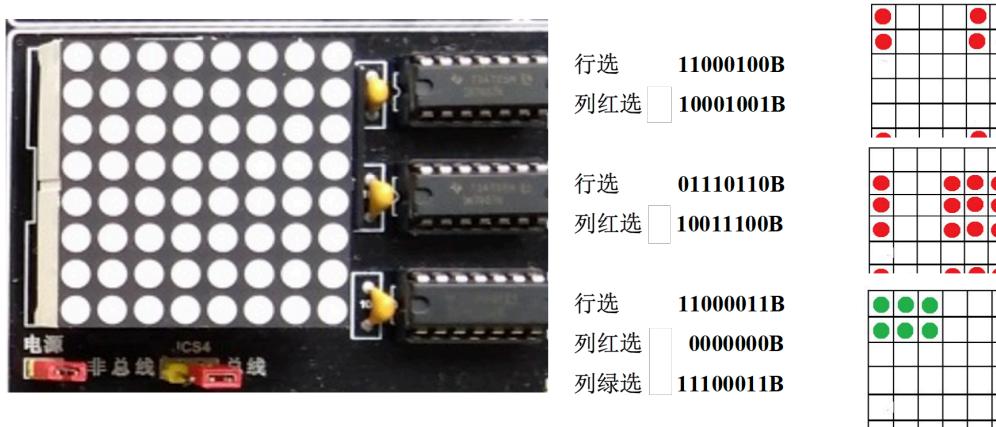


图 3: 双 74LS273 输出 (8×8 点阵显示)

原理如图 4 所示. 双色点阵在列红选为 00000000B 时列绿选才有效. 动态显示可以实现各种图形文字效果, 如“年”字显示 (如图) .

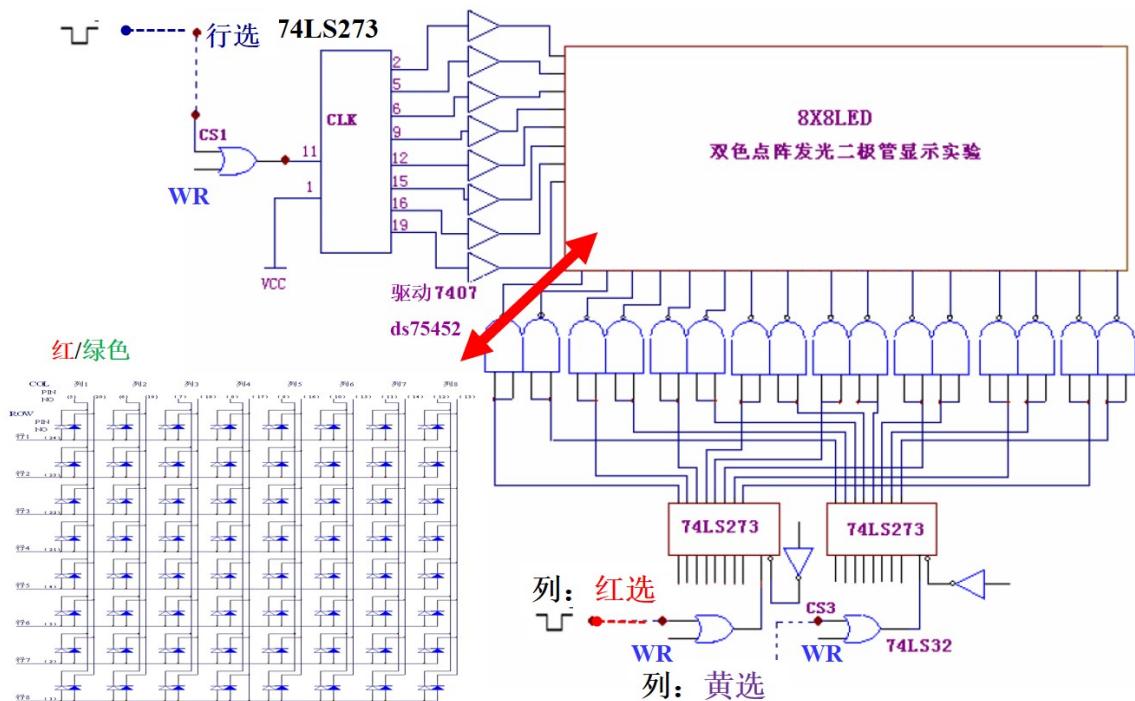


图 4: 双 74LS273 输出 (8×8 点阵显示) 原理

(四) 扩展 CD4518 计数器（双 BCD 加计数器）

CD4518BC 是一双 BCD 加计数器，引脚功能如图 5，真值表如图 6a. 真值表如图 6a.

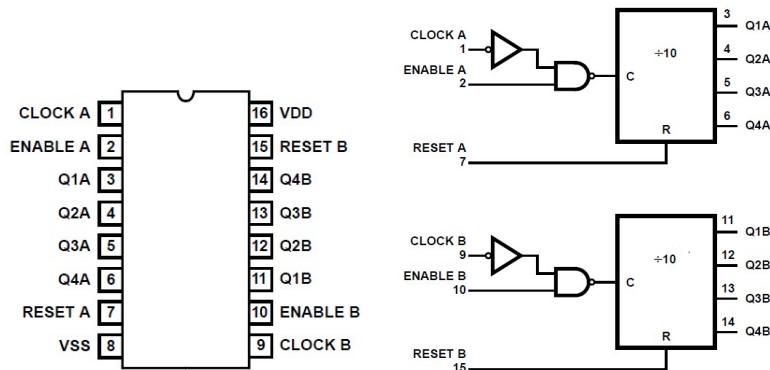
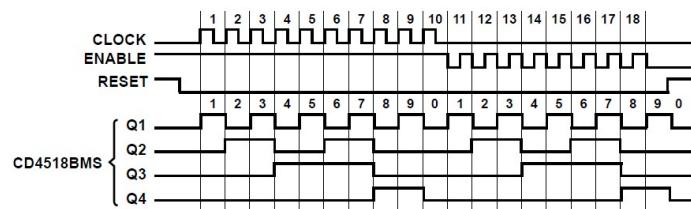


图 5: CD4518 双 BCD 加计数器

TRUTH TABLE			
CLOCK	ENABLE	RESET	ACTION
/	1	0	Increment Counter
0	/	0	Increment Counter
/	X	0	No Change
X	/	0	No Change
/	0	0	No Change
1	/	0	No Change
X	X	1	Q1 thru Q4 = 0

X = Don't Care 1 = High State 0 = Low State

(a) 真值表



(b) 特性波形

图 6: CD4518 双 BCD 加计数器真值表/特性波形

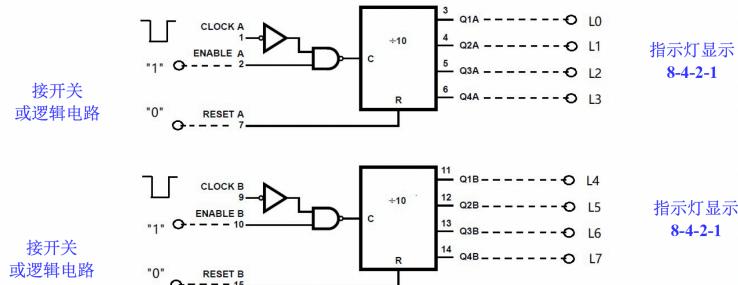


图 7: CD4518 双 BCD 加计数器接线图

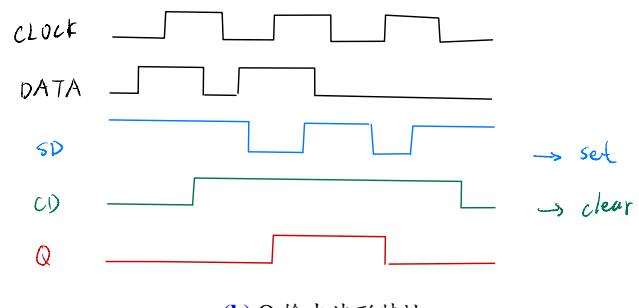
三. 实验内容

(一) 简单时序逻辑

利用实验箱上 7474 逻辑电路，输入接开关，输出接指示灯。连线，记录说明 Q 输出波形特性；真实连线如图 8a 所示，所得输出波形为图 8b.



(a) 真实连线图



(b) Q 输出波形特性

图 8: 7474 简单时序逻辑

结果分析 1: 简单时序逻辑

结果与预期一致，与真值表表 1 相符，验证了 D Flip Flop 的功能。

(二) 8 位数据/地址锁存器（8D 触发器）器特性实验

注意 1: 接线

断开 PC 机总线 D 型插座。排线 1 一端接逻辑开关输入 JP1 (K0-K7)，一端接总线缓冲器输入端 JP9(IN7—IN0)；

排线 2 一端接总线缓冲器输出/JP10 (O7—O0)，一端接 LED 指示灯显示/JP2(L7—L0)。

按动开关 K7 控制使能端，记录示波器观察缓冲器输出 (L7-L0) 及指示灯状态等结果；

结果分析 2: 8 位数据/地址锁存器（8D 触发器）器特性实验

实验结果即完成了一个锁存的效果，只有在下降沿的过程状态会“刷新”。

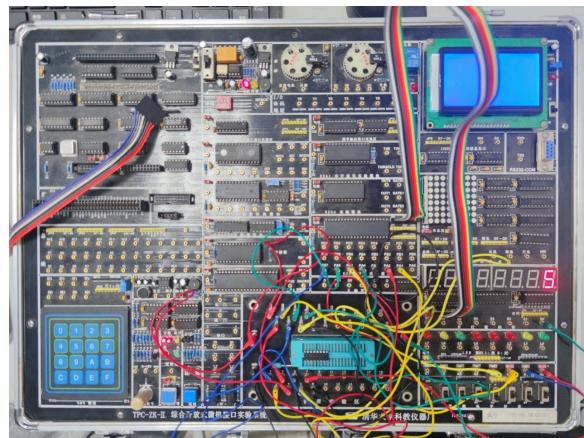


图 9: 8 位数据/地址锁存器 (8D 触发器) 器特性实验真实连线图

(三) 双 8 位数据/地址锁存器 (8D 触发器) 特性实验 (8×8 点阵)

注意 2: 接线

排线 1 一端接逻辑开关输入 JP1 (K7-K0)，一端接总线插座 (D7-D0) 作为触发器输入；WR 接低电平，行选接脉冲 1；红选接脉冲 2。

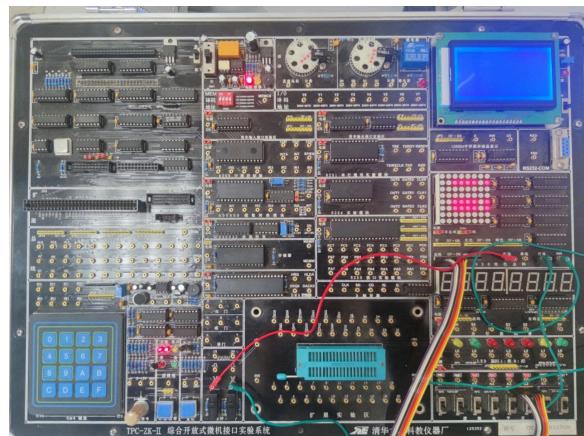


图 10: 双 8 位数据/地址锁存器 (8D 触发器) 器特性实验真实连线图

结果分析 3: 8 位数据/地址锁存器 (8D 触发器) 器特性实验

与上一个实验类似，存在一个锁存的效果，只有在下降沿的过程状态会“刷新”，得到的效果是比較好的，可以做到分別行选列选。

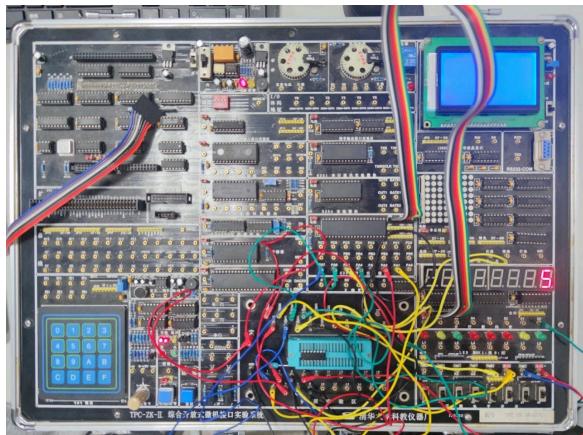
(四) 扩展 CD4518 计数器（双 BCD 加计数器）应用

注意 3: 接线

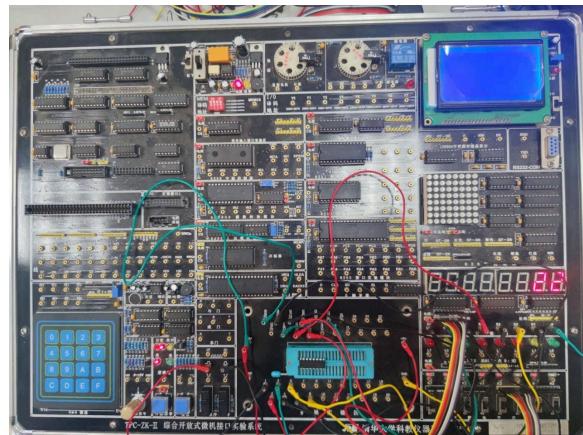
在实验箱扩展实验区顶头插入 CD4518 芯片，引脚 Pin16-Vcc (+5V), Pin8-GND; 引脚 Pin(6-3) Q4A-Q1A 顺序接指示灯 L3-L0; 引脚 Pin(14-11)Q4B-Q1B 顺序接指示灯 L7-L4); 引脚 Pin(2) Enable A 和 Pin(10)ENABLE B 接高电平; Pin (1) Clock A 和 Pin (9) Clock B 分别连接按键脉冲; 如图 7.

通过合理设计组合逻辑 reset 信号即 clock 信号，可以方便地实现不同计数的计数器（8-4-2-1）输出，如双 BCD，24 进制（时计数），60 进制（分、秒计数）。

由真值表可知，CD4518 可以利用 clk 上升沿计数（Enable 接高电平），也可以利用 enable 下降沿计数（clk 接低电平）；进位可以通过各位输出状态用组合逻辑产生，也可利用输出状态 Q4 作为进位接十位 Enable B.



(a) 一位十进制计数器



(b) 两位十进制计数器

图 11: 扩展 CD4518 计数器（双 BCD 加计数器）应用

结果分析 4

一位十进制计数器和双独立的 BCD 显示都成功了，而 0-99 BCD 连接经过助教确认连接没有错误。

思考 1: 24 进制或 60 进制方案

秒信号发生器由 CD4060 和 CD4013 组成，产生频率为 1Hz 的和时间基准信号。CD4060 是 14 级二进制计数器/分频器/振荡器。它与外接电阻、电容、石英晶振共同组成 32768Hz 振荡器，并进行 14 级二分频，再外加一级 D 触发器（CD4013）二分频，输出 1Hz 的时基秒信号。秒、分、时计数器电路均采用双 BCD 同步加计数器 CD4518 计数输出 BCD 码，通过 CD4511 将 BCD 码显示出对应的数字，秒、分计数器是 60 进制计数，时是 24 进制计数。时、分、秒的显示电路完全相同，均使用 CD4511 译码驱动数码管。秒校时采用等待校时法，正常工作时，开关 S3 拨到“走时”，进行对秒信号校对，将 S3 拨到“停”，待标准时间一到，立即拨到“走时”位置，恢复正常走

时. 分和时通过开关直接输入脉冲信号校准，每按一次开关，加一次数。

60 进制电路工作原理：当时间是 60 秒的时候，4518 的十位输出的 BCD 码是 0110，通过一个与门电路（CD4581）接在 4518 的 Q2B 和 Q3B 端，当到 60 秒时，这两端同时输出 1，通过与门电路输出 1 给 CD4518 清零，数字从 00 开始重新计数，同时给分信号或者时信号来一个脉冲加 1。24 进制电路工作原理同 60 进制电路类似，只不过取的信号是 24. 4518 输出的 BCD 码是 0010 0100，取 4518 的 Q2B 和 Q3A 端即可。

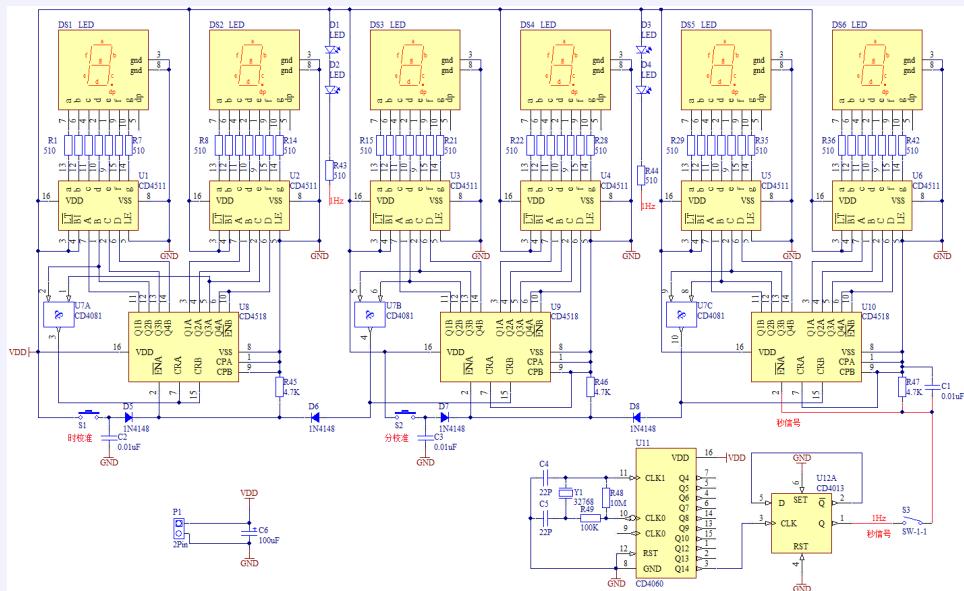


图 12: 24 进制或 60 进制方案电路图

思考 2: 总结

总结一下，主要的区别就是“如何进位”，十进制进位使用 BCD 码的进位信号即可，而在 24 和 60 进制中则需要通过逻辑表达式给出进位的信号，操作过程即是：24 进制取 4518 的 Q2B 和 Q3A 端，60 进制 4518 的 Q2B 和 Q3B 端。

电路没有断电记忆时间功能，断电后需要重新校时。^a

^a参考 <http://www.diyeluan.com/JC/162.html>.

(五) CAD 逻辑设计与仿真

1. 基本触发器模块

编写了基本的触发器模块，分别使用 `always =` 和 `<=` 赋值语句，实现变量的复制（获取），其识别序逻辑 HDL 阻塞（Blocking）与非阻塞（Nonblocking）赋值语句。

其 Verilog 代码在下方，其区别仅在赋值方式的一行。

Blocking	Nonblocking
<pre>module dff #(parameter DELAY = 0)(input d, input clk, output q, output w); reg q, w; always @(posedge clk) begin w = #DELAY d; q = w; // blocking end endmodule</pre>	<pre>module dff #(parameter DELAY = 0)(input d, input clk, output q, output w); reg q, w; always @(posedge clk) begin w <= #DELAY d; q <= w; // nonblocking end endmodule</pre>

结果分析 5: 基本触发器模块

分别运行上述两种代码，可以得到阻塞赋值的 RTL 级电路图 13 和非阻塞赋值的 RTL 级电路图 14。可以观察到，非阻塞赋值需要使用额外的寄存器以实现并行处理，在原理上是比较好的理解的。

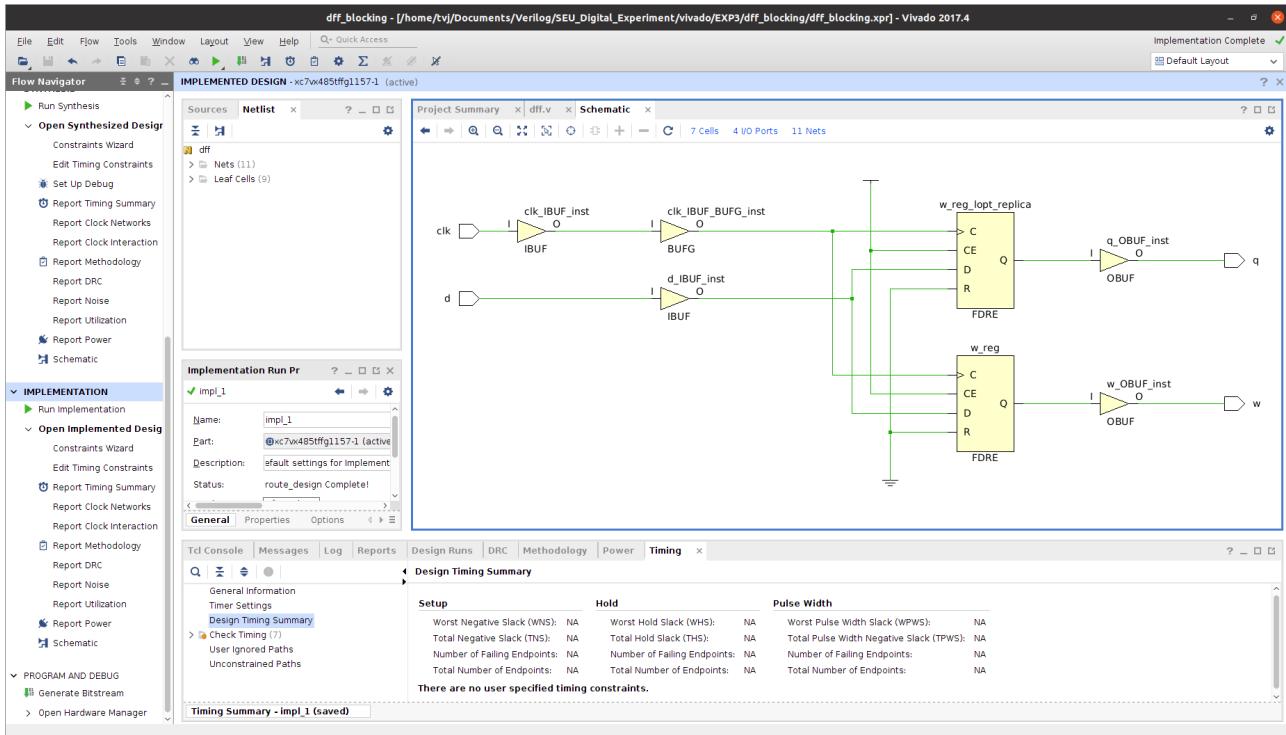


图 13: 阻塞赋值的电路

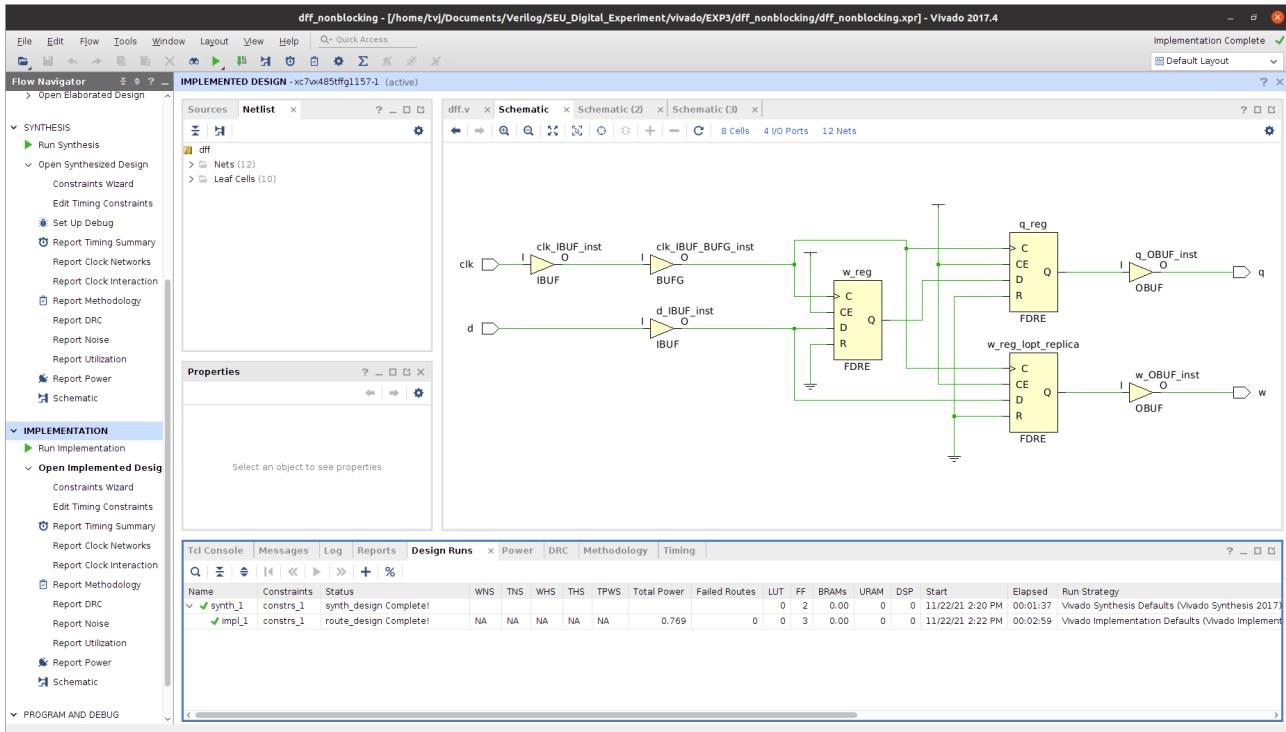


图 14: 非阻塞赋值的电路

2. 74LS273 模块

74LS273 等效功能模块 Verilog HDL 如下：

74LS273

```
module ff273v (din, clk, clr, dout);
    input [7:0] din;
    input clk, clr;
    output [7:0] dout;
    reg [7:0] dout;
    reg [7:0] dinn,dins;
    always @(posedge clk or negedge clr)
    begin
        if (!clr)
            dout <= 8'b00000000;
        else
            begin
                dout<=din;
            end
    end
endmodule
```

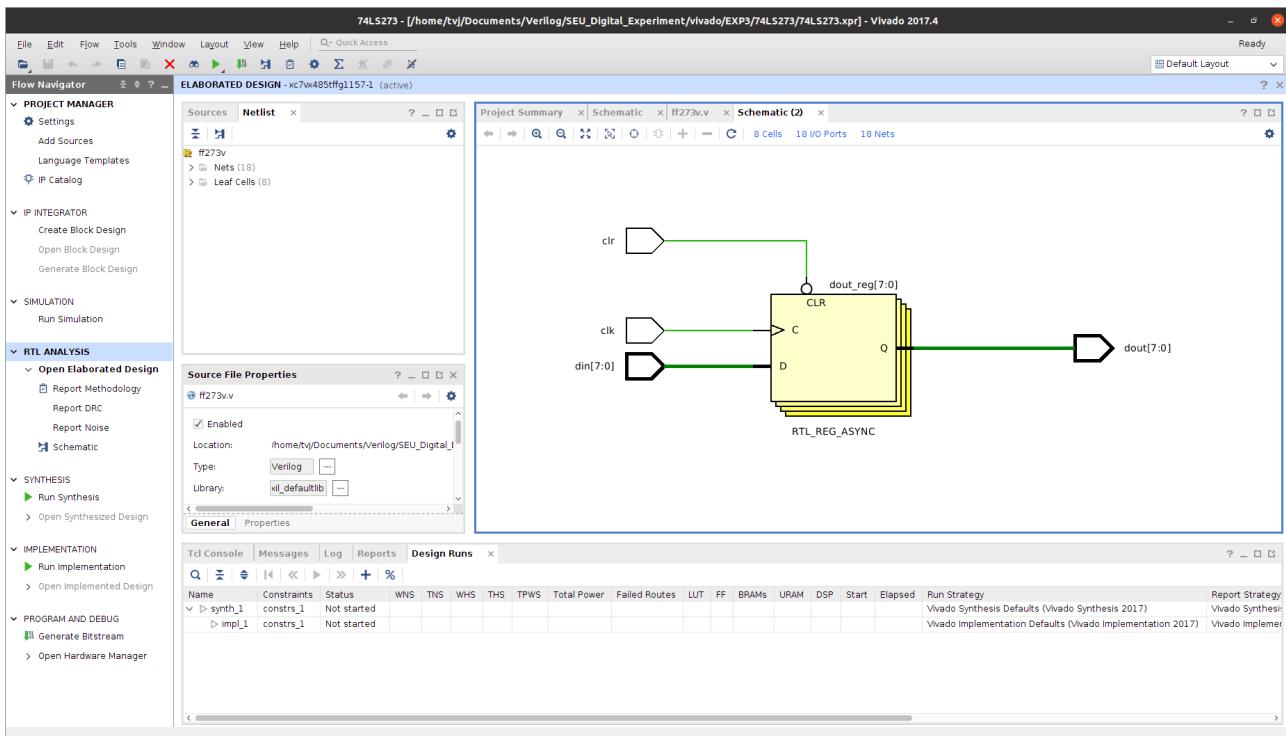


图 15: ff273v RTL 电路

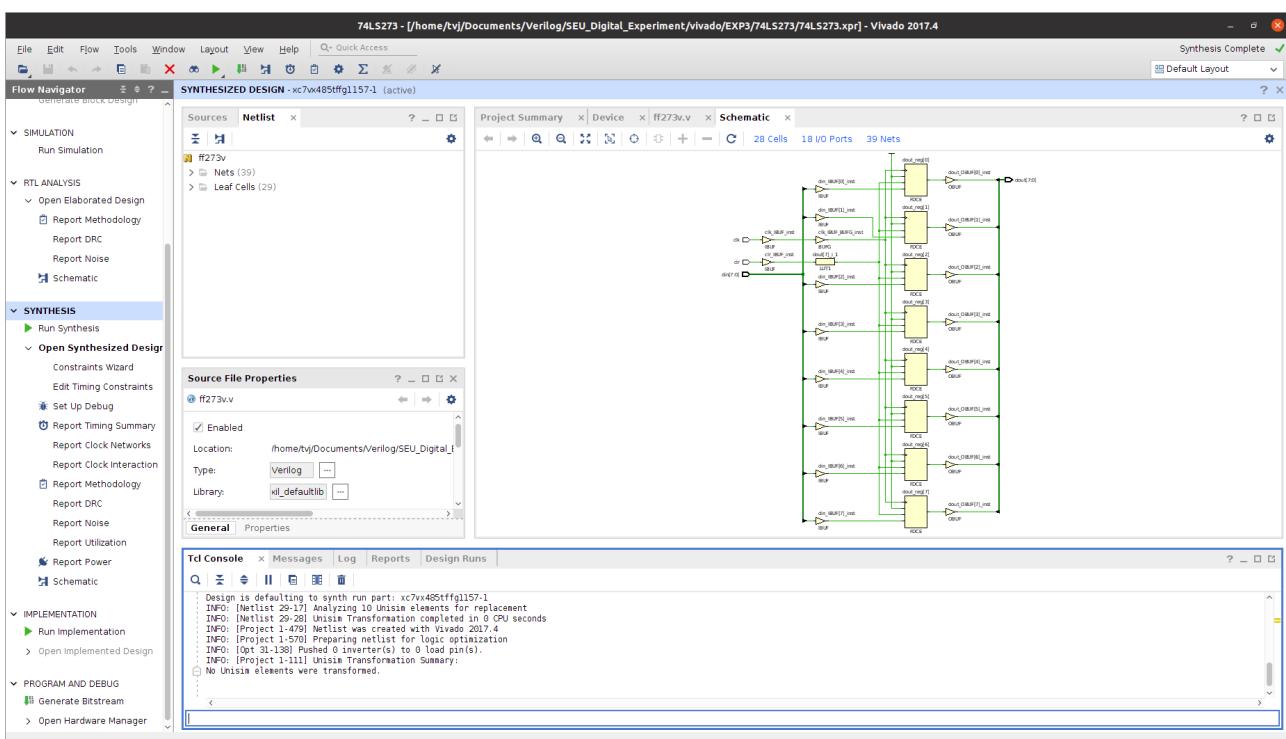


图 16: ff273v Synthesis 电路

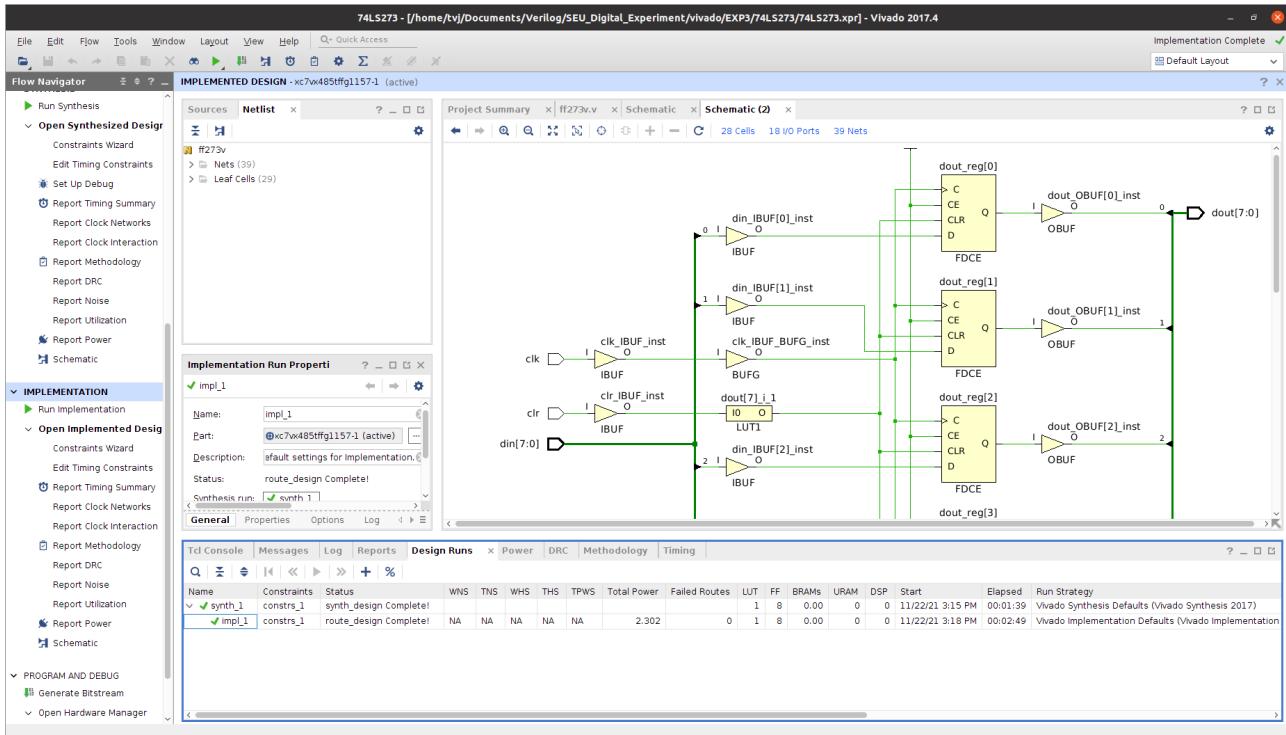


图 17: ff273v Implementation 电路

结果分析 6

Synthesis (图 16) 和 Implementation (图 17) 实际上是一样的, 有可能是和选择的 device 有关. 不过对于其整体的结构已经有了较为深入的了解了. 从图中可以较为清楚地看出 8 位锁存的方式.

3. Cnt10 定时计数器

```
fre_ctrl.v
module fre_ctrl (clk_in2,clk2);
    input clk_in2;
    output clk2;
    reg clk;
    reg [31:0] k;
    reg a, b, c;
    always @(negedge clk_in2)
    begin
        if (k >= 100000000)
            begin
                clk <= ~clk2; // clock signal
                a <= ~clk2;
                b = a ^ clk;
                c = a & clk;
                k <= 0;
            end
        else
            k <= k + 1;
    end
endmodule
```

```
c <= 1;
end
else
begin
    qout <= qout + 1;
    c <= 0;
end
end /* This 'end' is missing in
Ref. [1] */
end
endmodule
```

```
count10.v
module count10 (clk, clr, qout, c);
    input clk, clr;
    output [3:0] qout;
    output c;
    reg [3:0] qout;
    reg c;
    always @ (posedge clk or posedge
        clr)
    if (clr)
        qout <= 0;
    else
        begin
            if (qout >= 9)
                begin
                    qout <= 0;
```

```
disp.v
module disp (qout,clk,led1);
    input [3:0] qout;
    input clk;
    output [6:0] led1;
    reg [6:0] led1;
    always @ (posedge clk)
    begin
        case(qout)
            0: led1 = 7'b0000001;
            1: led1 = 7'b1001111;
            2: led1 = 7'b0010010;
            3: led1 = 7'b0000110;
            4: led1 = 7'b1001100;
            5: led1 = 7'b0100100;
            6: led1 = 7'b0100000;
            7: led1 = 7'b0001111;
            8: led1 = 7'b0000000;
            9: led1 = 7'b0000100;
        default: led1 = 7'b0000001;
        endcase
    end
endmodule /* This is missing in
Ref. [1] */
```

四. 选作探索与应用设计

(一) 74LS161 可加载 4 位二进制计数电路

My74LS161

```

1 module My74LS161(
2     input wire CR, LD, CTP, CTT, CP,
3     input wire [3:0] D,
4     output reg Co,
5     output reg [3:0] Q
6 );
7
8     always @ (posedge CP or negedge CR) begin
9         if ( !CR ) Q = 4'b0; // Asynchronous clear takes the highest
10            priority
11         else if ( !LD ) Q = D;
12         else if ( CTP && CTT ) Q = Q + 4'b1;
13         if ( Q == 4'b1111 ) Co = 1;
14     else Co = 0;
15 end
endmodule

```

学习了网上的代码¹, 大致了解了整体的架构方式.

(二) 24 和 60 进制计数显示系统 (时钟)

TOP.v

```

1 module top_rtc(input rst, clk, output[6:0] S_L, S_M, M_L, M_M, H_L, H_M);
2     wire[3:0] w1, w2, w3, w4, w5, w6;
3     segment_7 s1(.bcd(w1), .y(S_L));
4     segment_7 s2(.bcd(w2), .y(S_M));
5     segment_7 s3(.bcd(w3), .y(M_L));
6     segment_7 s4(.bcd(w4), .y(M_M));
7     segment_7 s5(.bcd(w5), .y(H_L));
8     segment_7 s6(.bcd(w6), .y(H_M));
9     rtc c1(.clk(clk), .rst(rst), .sl(w1), .sm(w2), .ml(w3), .mm(w4),
10        .hl(w5), .hm(w6));
endmodule

```

¹代码: https://github.com/Teddy-van-Jerry/LCDF_Expr/blob/master/Expr13/Report13.md.

BCD_SEVEN_SEGMENT.v

```

1 module segment_7(input [3:0]bcd, output reg [6:0]y);
2   always @(*)
3     case(bcd)
4       4'd0: y = 7'b1111_110;
5       4'd1: y = 7'b0110_000;
6       4'd2: y = 7'b1101_101;
7       4'd3: y = 7'b1111_001;
8       4'd4: y = 7'b0110_011;
9       4'd5: y = 7'b1011_011;
10      4'd6: y = 7'b1011_111;
11      4'd7: y = 7'b1110_000;
12      4'd8: y = 7'b1111_111;
13      4'd9: y = 7'b1111_011;
14    endcase
15 endmodule

```

COUNTER.v

```

1 module rtc(input rst,clk, output [3:0] sl,sm,ml,mm,hl,hm );
2   reg [3:0] sl0,sm0,ml0,mm0,hl0,hm0;
3   wire clk_1sec;
4   assign sl=sl0;
5   assign sm=sm0;
6   assign ml=ml0;
7   assign mm=mm0;
8   assign hl=hl0;
9   assign hm=hm0;
10  assign clk_1sec=clk;//let clk_1sec is driven by 1sec time period clock
11  always@(posedge clk)
12    if(rst) begin
13      sl0<=0;
14      sm0<=0;
15      ml0<=0;
16      mm0<=0;
17      hl0<=0;
18      hm0<=0;
19    end
20    else if(sl0<9) begin
21      if(clk_1sec)
22        sl0<=sl0+1;

```

```

23   end
24   else if(sm0<5) begin
25     sm0<=sm0+1;
26     sl0<=0;
27   end
28   else if(ml0<9) begin
29     sl0<=0;
30     sm0<=0;
31     ml0<=ml0+1;
32   end
33   else if(mm0<5) begin
34     sl0<=0;
35     sm0<=0;
36     ml0<=0;
37     mm0<=mm0+1;
38   end
39   else if(((hm0<2) && (hl0<9)) || ((hm0==2)&&(hl0<3))) begin
40     sl0<=0;
41     sm0<=0;
42     ml0<=0;
43     mm0<=0;
44     hl0<=hl0+1;
45   end
46   else if(hm0<2)
47   begin
48     sl0<=0;
49     sm0<=0;
50     ml0<=0;
51     mm0<=0;
52     hl0<=0;
53     hm0<=hm0+1;
54   end
55   else if((hm0==2)&&(hl0==3)&&(mm0==5)&&(ml0==9)&&(sm0==5)&&(sl0==9))
56   begin
57     sl0<=0;
58     sm0<=0;
59     ml0<=0;
60     mm0<=0;
61     hl0<=0;
62     hm0<=0;
63   end

```

63 endmodule

SIM.v

```

1 module tb(); // testbench
2   wire [6:0] sl,sm,ml,mm,hl,hm;
3   reg clk;
4   reg rst;
5
6   always #1 clk=~clk; // clock signal
7
8   top_rtc r0(rst,clk,sl,sm,ml,mm,hl,hm);
9
10 initial begin
11   clk=0;
12   rst=0;
13   #5;
14   rst=1;
15   #5;
16   rst=0;
17   #4500 $finish();
18 end
19 initial begin
20   $dumpvars();
21   $dumpfile("dump.vcd");
22
23   $monitor($time, "rst= %b, SEC_L= %b, SEC_M= %b, MIN_L= %b, MIN_M= %b,
24   H_L= %b, H_M= %b", rst, sl, sm, sl, mm, hl, hm);
25
26 end
endmodule

```

结果分析 7: 24 和 60 进制计数显示系统（时钟）

学习了网上的代码^a，了解了其整体的结构和具体方法。

其中几个重要的模块就是计数器（COUNTER.v）和 BCD 七段管显示（BCD_SEVEN_SEGMENT.v），这些在之前的研究中已经较为清楚了，此处不加赘述。此处最重要的是较为完整的层次性，这里的 TOP.v 即是做了整合的工作，将计数和现实整合在一个完整的大项目中了，调用了计数器和显示。而 SIM.v 通过已有接口增加了时钟信号。

^a代码地址：https://github.com/T-V-J/24-12_DIGITAL_CLOCK_DESIGN_USING_VERILOG，这是我 fork 后的项目。

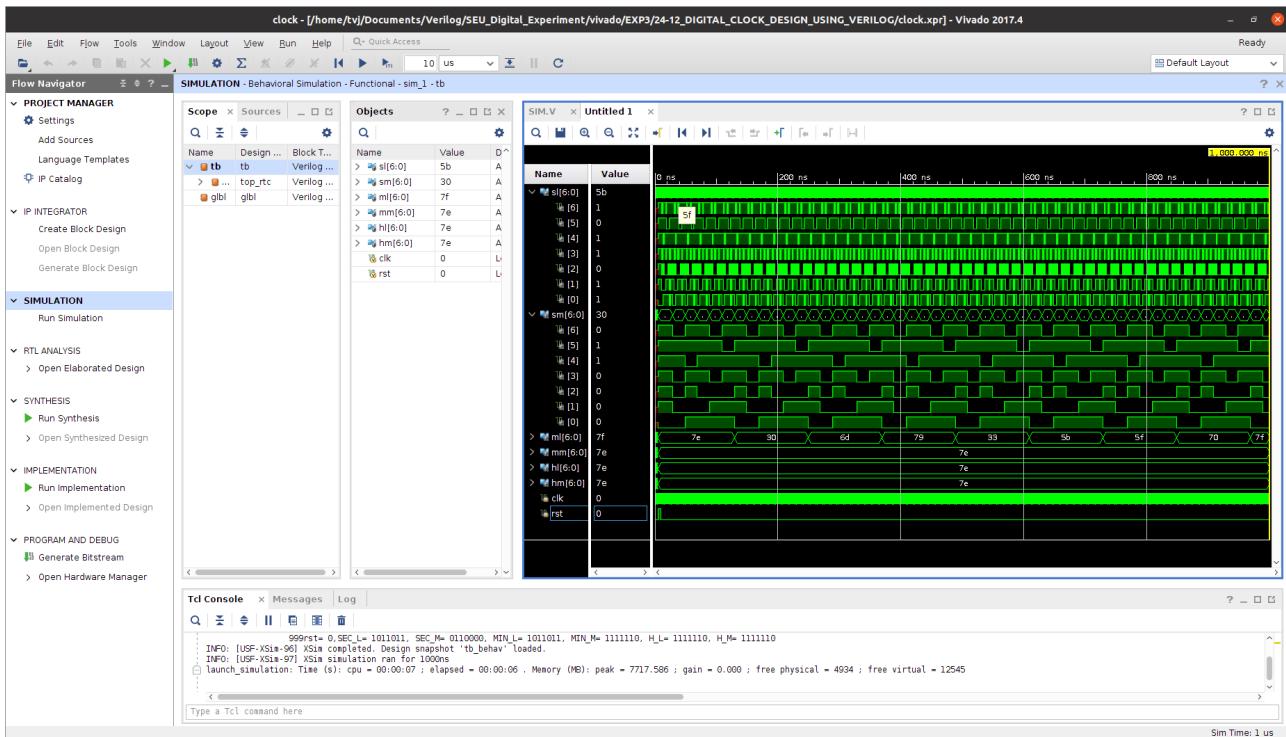


图 18: 24 和 60 进制计数显示系统（时钟）行为仿真结果²

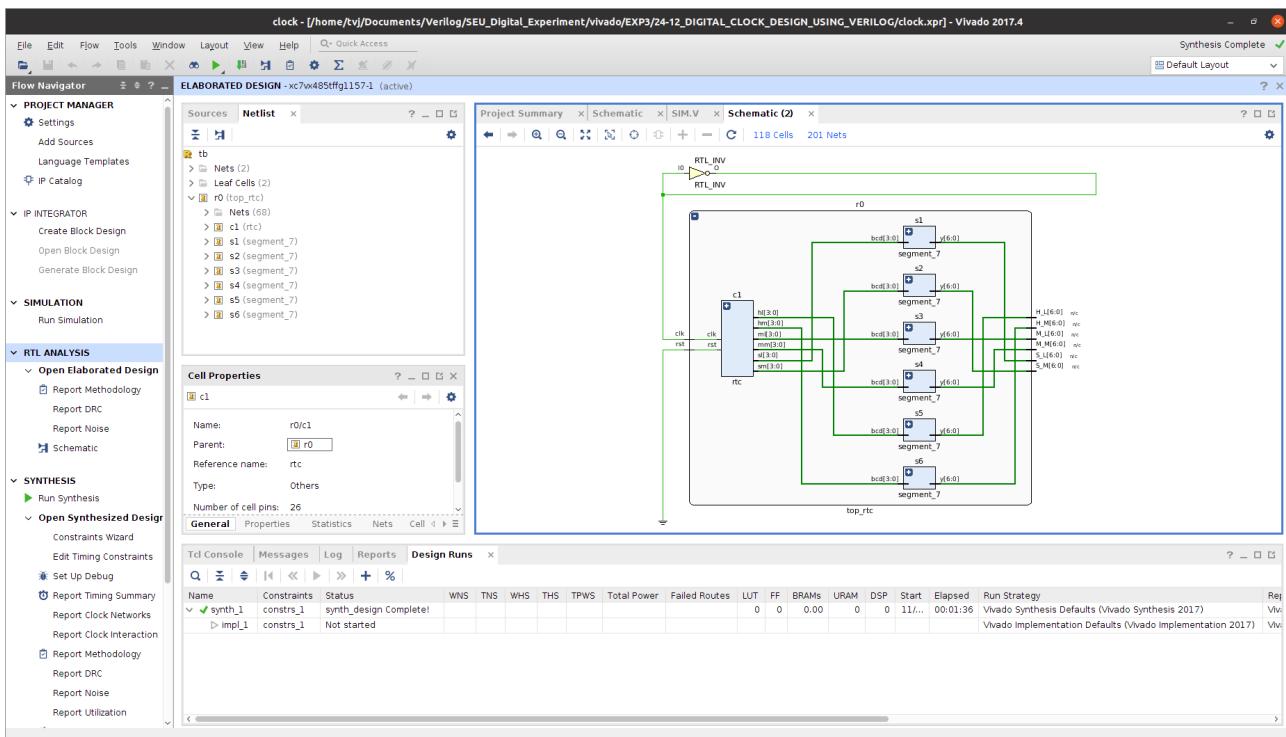


图 19: 24 和 60 进制计数显示系统（时钟）RTL 级电路

²最上面一组是秒的个位，第二组是秒的十位，再往下是分的两位和时的两位。

五. 实验总结

1. 需要用到实验箱的部分和 Vivado 实验关联紧密，在做硬件实验部分的时候就需要对结构和各个元件有深入的认识。一个简单的例子就是我们在理论课上学习的锁存器、寄存器等是需要上升沿信号激活的，而这个信号在实验中即是使用开关/脉冲信号/信号源，并在实际连线操作中会遇到更多的问题。而这个过程是非常锻炼我们的动手操作能力。
2. Vivado 中编写程序的感觉和 C++ 中类似，都是从一个总体思路开始层层向下（至少我是这样的，不过不排除先从模块再整合的思路），有一个总体要求（Verilog 中的 testbench 或者总的 top 结构，C++ 中的 main 函数）向下有其他的模块（module，而在 C++ 中则是类或者函数）。
3. 其他的内容在对应的各个部分已有分析。

参考文献

- [1] 计算机硬件实验室. 《数字逻辑与计算机体系结构》实验指导书[A]. 南京: 东南大学, 2021.
- [2] 冯漫. 3-时序逻辑电路[A]. 南京: 东南大学, 2021.

附录 A：实验报告 L^AT_EX 模板

实验报告使用自己编写的 L^AT_EX 模板 (`SEU-Digital-Report.cls`)，在基本适配 Microsoft Word 版报告的格式要求之外，增加了更多的功能，使得报告看起来更加优雅多彩。

后续升级后，报告模板将于 <https://github.com/Teddy-van-Jerry/TVJ-Digital-Report> 基于 MIT License 开源共享。

编译需要使用 XeLaTeX + Biber，封面页修改如下内容即可。

```

1 %% 使用实验报告模板类（字体大小 11pt 约为五号字）
2 \documentclass[11pt]{SEU-Digital-Report}
3
4 %%%%%%%%%%%%%%%%
5 \expno{三} % 实验序号
6 \expname{时序逻辑与设计} % 实验名称
7 \expauthor{赵舞穹} % 姓名
8 \expID{61520522} % 学号
9 \expmates{郑瑞琪} % 同组
10 \expmatesID{61520523} % 学号（同组）
11 \expmajor{工科试验班} % 专业
12 \explab{计算机硬件技术} % 实验室
13 \expdate{2021年11月12日} % 实验日期
14 \expreportdate{\today} % 实验日期
15 \expgrade{} % 成绩评定
16 \extutor{冯漫} % 评阅教师
17 %%%%%%%%%%%%%%%%

```

附录 B：Vivado 程序真伪判别

1. Vivado 程序我在 Ubuntu 20.4 LTS 平台完成，标题栏为经典的 GNOME 桌面风格，与 Windows 有很大区别。
2. 标题栏现实程序所在文件夹为 `/home/tvj/Documents/Verilog/SEU_Digital_Experiment`，这是我的 GitHub 私有项目。
3. Design Run 中显示的时间也可以辅助判断。