

# 东南大学

## 《微机系统与接口实验》

### 实验报告

#### 实验一 数据传送

姓 名：薛宇飞

学 号：04020235

同 组：

学 号：

专 业：信息工程

实 验 室：金智楼硬件实验室

实验时间：2022 年 3 月 27 日

报告时间：2022 年 3 月 27 日

评定成绩：

评阅教师：裴文江

## 目录

<b>1 实验目的与内容</b>	<b>3</b>
<b>2 基本实验原理</b>	<b>3</b>
1 信号源和示波器 . . . . .	3
2 TPC 实验装置 . . . . .	3
3 D 触发器 . . . . .	3
4 环形振荡器 . . . . .	4
<b>3 方案实现与测试</b>	<b>4</b>
1 信号源和示波器 . . . . .	4
2 D 触发器控制 LED . . . . .	5
3 8 段数码管的显示 . . . . .	5
4 环形振荡器 . . . . .	6
5 Vivado 开发初探 . . . . .	6
1 行为仿真 . . . . .	6
2 RTL 级电路 . . . . .	7
3 硬件实现 . . . . .	8
4 其他工作 . . . . .	8
<b>4 提高与创新研究</b>	<b>9</b>
1 双色点阵发光二极管显示控制 . . . . .	9
2 Vivado 开发设计 . . . . .	10
<b>5 分析与总结</b>	<b>11</b>
<b>参考文献</b>	<b>11</b>

## 一. 实验目的与内容

1. 熟悉数字逻辑与硬件接口基本实验仪器（双踪数字示波器、双路波形发生器等）的使用；
2. 了解掌握 TPC 实验装置中基本数字逻辑单元的基本原理和组成结构, 学会测试检查 TTL 电平、开关输入/LED 发光管/8 段数码管输出控制、与或非门和 D 触发器的基本特性；
3. 分析了解时钟发生和分频器电路原理，学会使用双踪数字记忆示波器检测电平和脉冲特性的方法；
4. 正确掌握基本组合逻辑与或非和时序逻辑（D 触发器）芯片特性；
5. 了解可编程逻辑器件设计仿真工具软件的基本工作原理和 HDL 编辑编程、仿真、分析操作过程，形成分层结构概念，为 HDL 模块验证及后续实验搭建基本工程框架。

## 二. 基本实验原理

### (一) 信号源和示波器

信号源与示波器在电路实验中已有较为深入的了解，此处不再赘述。

### (二) TPC 实验装置

TPC 实验装置分为许多模块，具体详见实验指导书<sup>[1]</sup>。

### (三) D 触发器

D 触发器的结构如图 1 所示，其真值表为表 1。

表 1: D 触发器的真值表

输入				输出	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	×	×	H	L
H	L	×	×	L	H
L	L	×	×	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	×	$Q_0$	$\overline{Q_0}$

图 2 左侧模块是按键单脉冲电路，和 D 触发器联合的原理为：CLK 端为高电平时，当 CD 端为低电平时，L7 熄灭，当 CD 端为高电平时，L7 点亮；CLK 端为低电平时，L7 保持原状态。

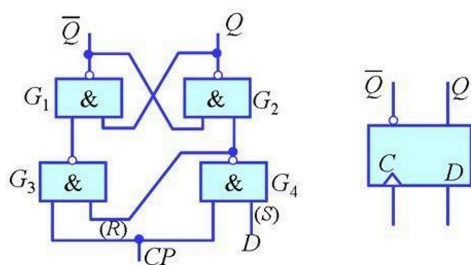


图 1: D 触发器结构图

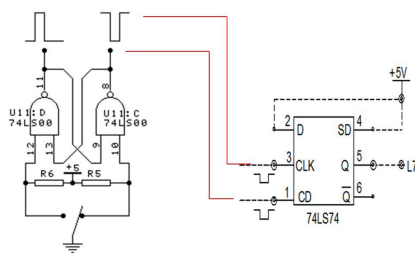
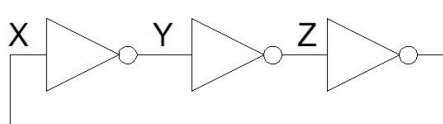


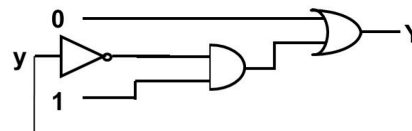
图 2: D 触发器基本特性测量电路

#### (四) 环形振荡器

环形振荡器的原理图如图 3a 所示,由于实验箱与门、或门、非门各仅有一个,所以需要使用图 3b 所示电路来等效。此处震荡是靠门延时来实现,每经过一个门输出  $F = \bar{y}$ 。



(a) 原理图



(b) 等效图

图 3: 环形振荡器电路图

### 三. 方案实现与测试

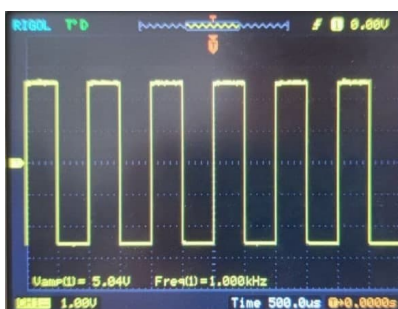
#### (一) 信号源和示波器

RIGOL DG1022U25MHz 双路波形发生器和 RIGOL DS1072E-EDU 70MHz 双踪数字记忆示波器的操作使用。

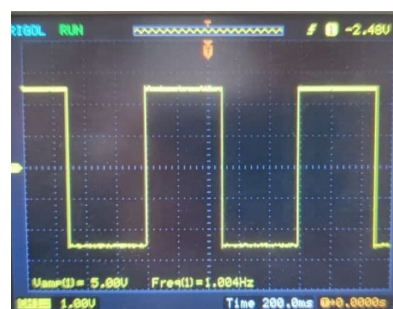
图 4a, 4b, 4c 分别是 32.768kHz、1kHz 和 1Hz 时示波器观察到的波形。



(a) 32.768kHz



(b) 1kHz



(c) 1Hz

图 4: 高低电平方波的示波器显示

#### 注意 1: 示波器的使用

示波器使用时需要和输入信号共地, 否则可能会有很大的区别 (例如, 不接入任何有效电压的时候可以观测到 50Hz 的信号波动), 虽然由于仪器可能共用拖线板达到了共地的效果。

## (二) D 触发器控制 LED

通过与门、或门、非门，控制 LED 灯的亮和灭，具体电路如图 5 所示。实验结果与理论的亮灭情况一致。

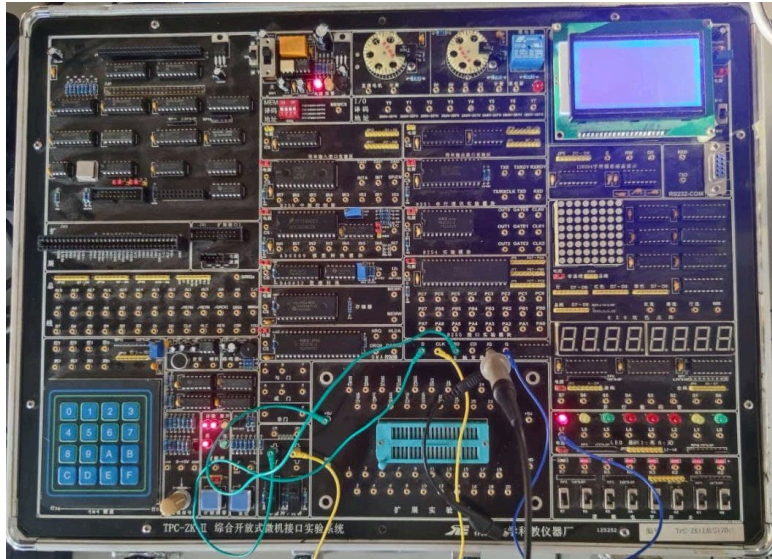


图 5: D 触发器控制 LED 的实际电路

## (三) 8 段数码管的显示

通过电路连接可以做到 8 段数码管的显示，如图 6 所示。我们通过排线让所有的位有同样的数字显示。

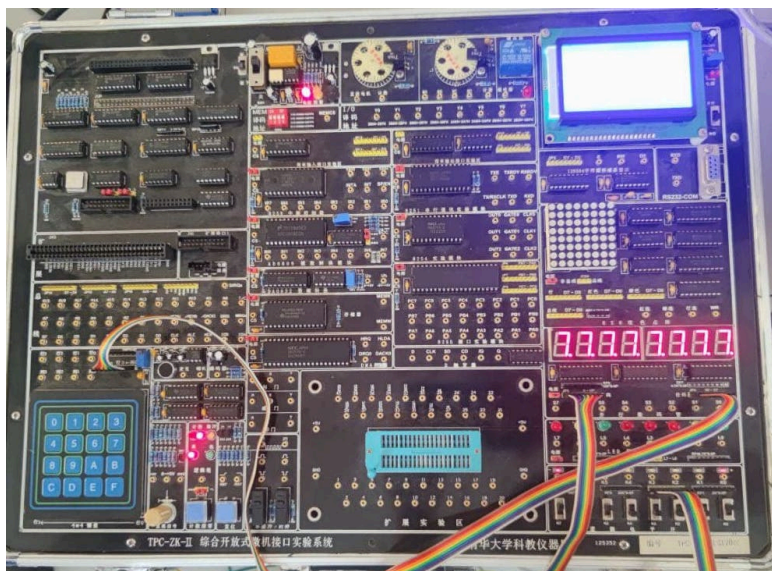


图 6: 8 段数码管显示的实际电路

#### (四) 环形振荡器

环形振荡器电路按照图 3b 搭建，通过示波器可以看到震荡的效果（如图 7 所示），两个通道之间的相位差可以清晰的看到。

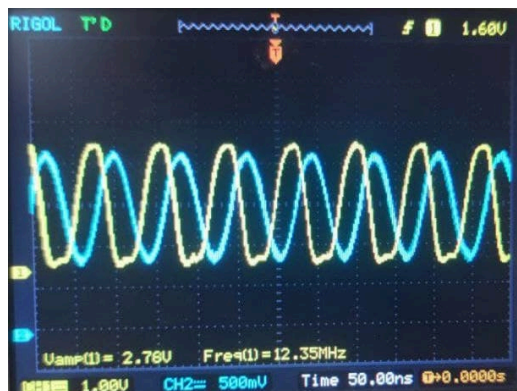


图 7: 环形振荡器示波器波形显示

#### 注意 2: 环形振荡器

实验前需要检验设备，这很重要！我们的实验过程中电路经过检查没有问题，最终发现与门存在故障，不管输入电平如何输出永远都是高电平。

#### (五) Vivado 开发初探

##### 实验器材 1: Vivado 实验器材

- 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

##### 1. 行为仿真

通过行为仿真 (behavioral simulation)，我们可以观测到图 8 中的结果。switches[7:0] 的结果基本可以说明其计数器的功能。调整如下代码中第 5 行的延时，即修改计数一次维持的时间，而第 6 行的延时则是改变上升沿的时间。

tutorial\_tb.v

```

1 initial
2 begin
3     for (i = 0; i < 255; i = i+2)
4         begin
5             #50 switches=i; // time for each step
6             #10 e_led = expected_led(switches); // transition time

```



```

7      if (leds == e_led)
8          $display ("LED output matched at", $time);
9      else
10         $display ("LED output mis-matched at ", $time, ": expected:
           %b, actual: %b", e_led, leds);
11     end
12 end

```

### 分析 1: 时延的影响

上述代码中的实验控制可以理解为梯形的几个几何参数：上底长度和其腰的投影长度。

### 注意 3: 行为仿真的查看

注意图 8 中观测行为仿真需点开数组（例如 `switches` 和 `leds`），另外需要将窗口展示的范围调整到合适的值。

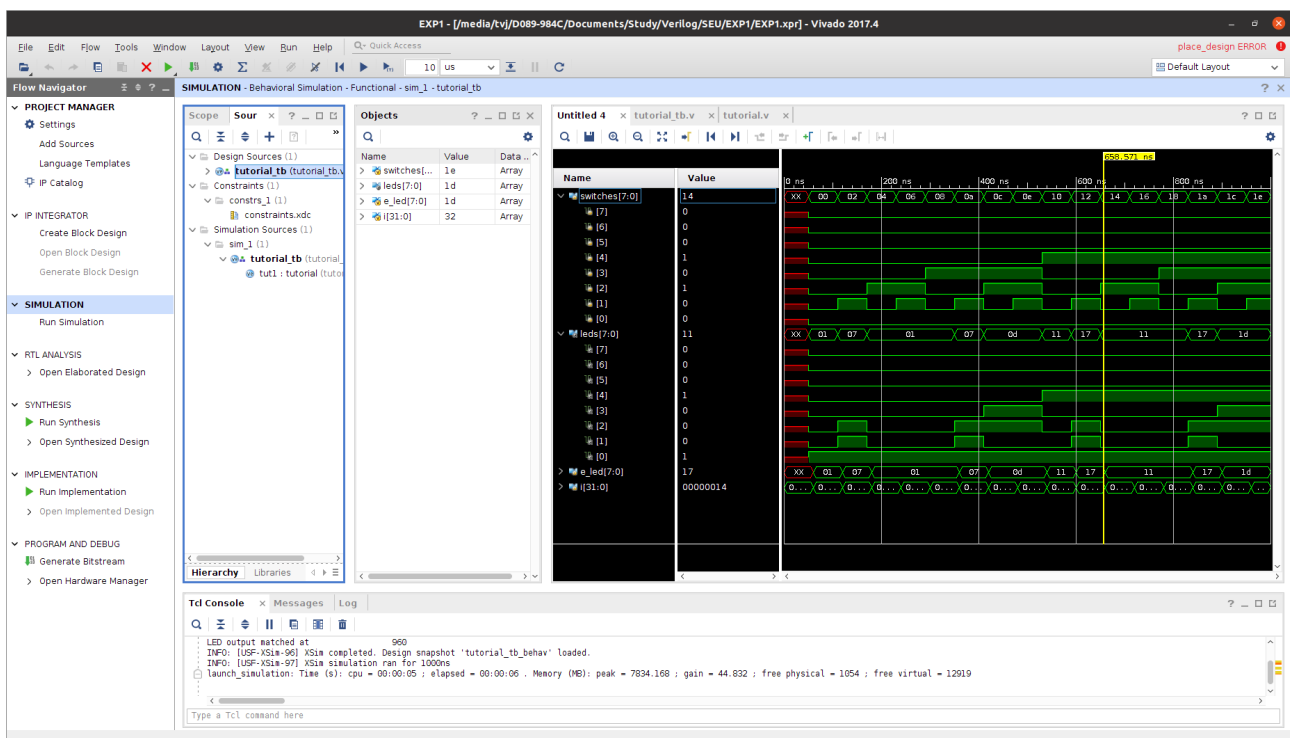
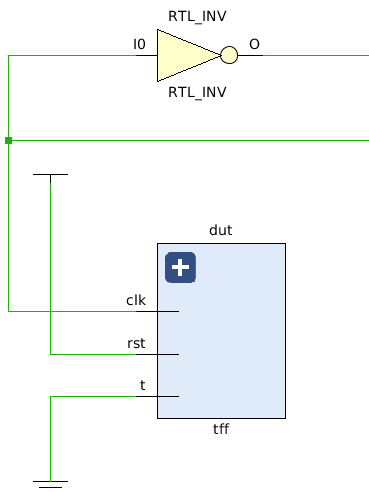


图 8: 行为仿真图 (tutorial)

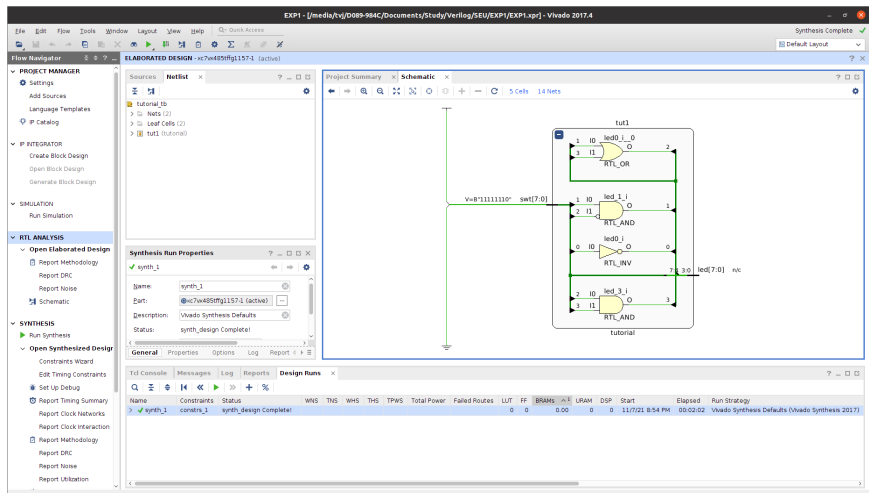
## 2. RTL 级电路

通过 `run synthesis` 可以获得电路的 RTL 级电路. 图 9a 保留了完整的 `tff1` 模块, 不过也可以像图 9b 一样展开模块的内容.

<sup>1</sup>代码来自<sup>[2]</sup>.



(a) tff



(b) tutorial

图 9: RTL 级电路

### 3. 硬件实现

通过添加约束文件(constraint),对硬件的结构(例如引脚)做出约束,然后 run implementation,得到例如图 10 的复杂电路. 不过由于没有具体的开发板,后面的步骤无法继续进行,目前只做了了解.

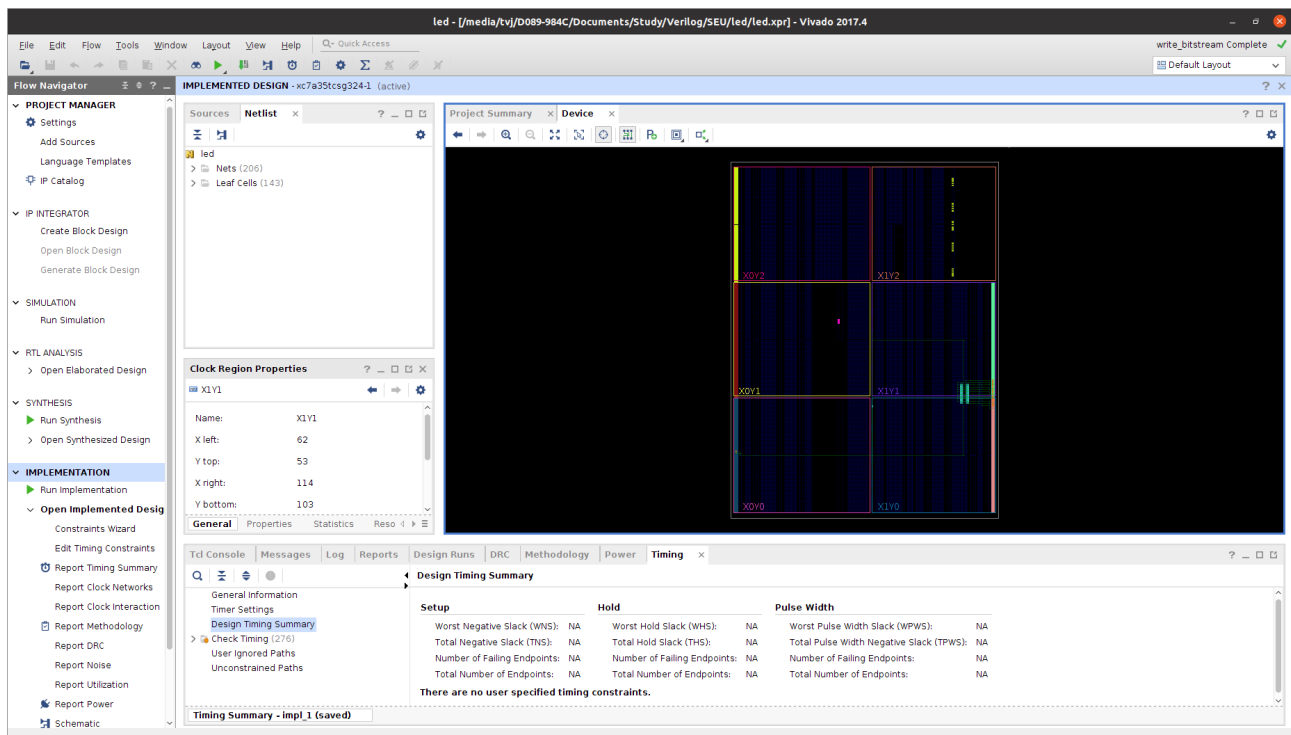


图 10: 硬件实现 (led)

### 4. 其他工作

led.v 的 RTL 级电路图为图 11.



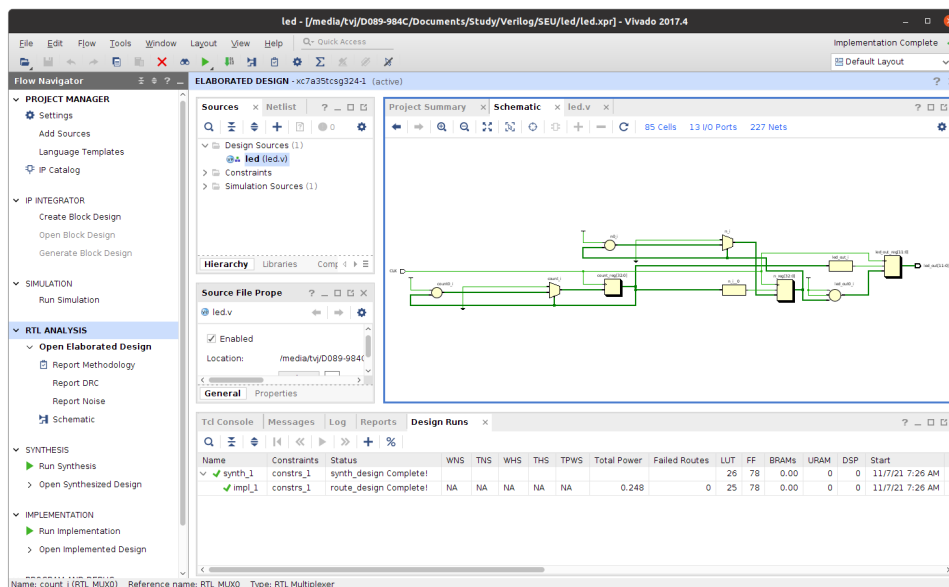


图 11: RTL 级电路 (led)

另外一部分工作详见第 4 章第 2 节.

## 四. 提高与创新研究

### (一) 双色点阵发光二极管显示控制

通过连接电路, 我们控制了双色点阵发光二极管的显示, 效果如图 12 所示, 每一次变化除了需要调节开关, 还需要控制单脉冲电路以激活状态.

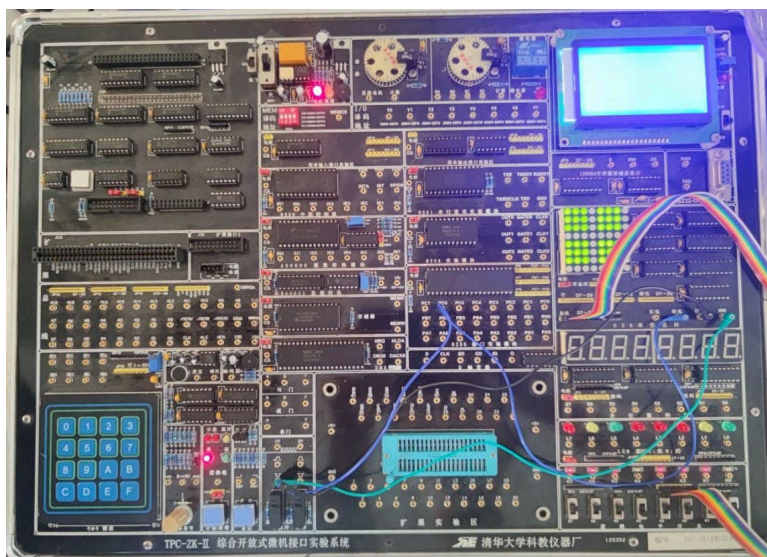


图 12: 双色点阵发光二极管的显示控制电路

## (二) Vivado 开发设计

我主要通过 tff<sup>[2]</sup> 的代码对 Vivado 做了更深的了解.

### 思考 1: Vivado 层次调用

此处有 tff\_tb.v (test bench) 和核心模块 tff.v, test bench 通过创建一个 tff 类变量即可实现调用, 体现在 schematic 图 (图 9) 上即一个可以展开或收起对框.

下面是 test bench 的代码.

tff\_tb.v

```
module tff_tb;
    reg clk, reset, t;
    wire q;
    tff dut (.clk(clk), .t(t), .rst (reset), .q(q));

    initial clk = 1'b0;
    always #5 clk = ~clk; // clock generation

    initial
    begin
        reset = 1'b1;
        t = 1'b1;
        #20 reset = 1'b0;
        #50 t = 1'b0;
        #30 reset = 1'b1;
        #30 $stop;
    end
endmodule
```

下面是 tff 的核心代码.

tff.v

```
module tff(
    input clk, rst, t,
    output reg q,
    output qb
);
    always @ (posedge clk, posedge rst) // asynchronous reset
    begin
        if (rst)
            q <= 1'b0;
    end
endmodule
```

```
        else if (t)
            q <= ~q;
        else
            q <= q;
    end
    assign qb = ~q;
endmodule
```

## 思考 2: Vivado 整体设计思路

从思考 1 中总结,可以得到 Vivado 设计的总体思路. Vivado 在结构上类似 C++, 能够定义类来实现层次的堆叠. 因此 Vivado 项目是基于众多 components 的, 而这些 components 既可以来源于 Verilog 标准 (相当于 C 语言), 又可以来自例如 xup 库 (相当于 C++ 标准库) 等地方, 当然我们也可以自己定义我们的类.

我的导师在研究生阶段研究极化码 (polar codes), 其中极化码 BP 译码 (belief propagation) 架构从 butterfly 结构层层向上加, 结构清晰且易于证明架构的正确性.

## 五. 分析与总结

具体的问题与思考在对应的位置已有分析, 此处主要阐述对实验总体的体悟.

我目前对于算法的研究较多 (例如在课题组的工作方向是“智能反射表面的信道估计”), 而在硬件实现的能力上有所欠缺. 一方面, 需要较强的动手操作能力, 另一方面, 硬件设计是一个和算法相差很多的工作领域, 虽然关联非常密切, 但需要考虑的问题会和算法有一定的出入.

不过目前我还能较好适应这样的工作内容, 经过锻炼, 一方面是学习到了“如何独立自主学习”的技能, 另一方面在利用代码能力优势的同时加强代码与硬件联系的能力, 最后在成果总结 (例如文字总结和 L<sup>A</sup>T<sub>E</sub>X 技巧) 上也有不小的提升.

在下一阶段, 我将对 Vivado (Verilog) 做更深入的了解.

## 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] Yasho2249. Fpga-verilog — GitHub[EB/OL]. 2020. <https://github.com/yasho2249/fpga-verilog>.