浙江大學

《EDA技术应用》实验报告

项目名称：实验一 – My First FPGA

姓　　名：

学　　号：

学　　院：生物医学工程与仪器科学学院

专　　业：生物医学工程

指导老师：黄添添、裘利坚

2026 年 3 月 22 日

浙江大学 实验报告

课程名称：EDA 技术应用　实验类型：EDA 实验　指导老师：黄添添、裘利坚

实验名称：实验一：My First FPGA

姓　　名：　　学　　号：　　专　　业：生物医学工程

实验地点：玉泉教六 204　日　　期：2026 年 3 月 22 日

目　录

一、实验目的 2
二、实验内容 2
三、主要仪器设备 2
四、操作方法和实验步骤 2
4.1 分配设备与建立工程 2
4.2 逻辑设计环节 2
4.3 编译与验证 5
五、实验结果与分析 5
六、实验心得 6

一、实验目的

通过本实验，了解如何创建 FPGA 设计并在 DE2-115 开发板上运行设计。

二、实验内容

实验要求实现一个基于按键 KEY0 控制频率的 LED 闪烁系统。该系统能够通过直观的视觉反馈来检验设计逻辑的正确性。具体任务包括：采用 Verilog HDL 编写一个基础的 32 位计数器；引入锁相环（PLL）宏模块作为系统的时钟信号源；并调用一个 2 输入的数据选择器（多路复用器）宏模块。在实际的下板验证阶段，操作者可通过拨动物理开关，对驱动 LED 显示的计数器输出位进行动态切换与复用。

三、主要仪器设备

Quartus ii 13.1, DE2-115 开发板。

四、操作方法和实验步骤

典型的 FPGA 开发周期涵盖了设计输入、工程编译、逻辑仿真、程序烧录以及板级验证等核心环节。若在硬件测试中发现异常，需返回修改原始设计并重新走通上述开发流程。其中，设计输入阶段通常支持原理图绘制与硬件描述语言（HDL）编码两种主流方式。

4.1 分配设备与建立工程

在 Quartus 中建立全新工程时，需配置项目路径及名称。必须严格保证项目存放目录、工程文件名中不含任何空格字符，且顶层实体名（Name of the top-level design entity）必须与项目名称（Name of the project）完全保持一致。随后，需根据实际使用的硬件指定 FPGA 芯片型号（本实验目标器件为：EP4CE115F29C7），并完成相应的管脚绑定工作。

4.2 逻辑设计环节

在具体的设计实现中，通常以模块化框图文件（.bdf）作为工程的顶层架构。在此基础上，开发者可以灵活地调用参数化宏模块库（LPM），或混合使用纯 Verilog HDL 脚本编写的自定义逻辑单元，这两种开发方式在实际工程中可自由组合。

4.2.1 添加锁相环宏功能模块

锁相环（Phase-Locked Loop, PLL）是一种电子电路，常可用于时钟生成、频率合成、信号恢复、频率调制解调和频率多路复用等情况。在 FPGA 设计中我们常会调用 IP 核来使用 PLL，目的是为了得到想要的时钟频率。

创建顶层新原理图文件(.bdf)的方式为 File>New>Block Diagram/Schmatic File。新建 Verilog HDL 文件（.v）的方式为 File>New>Verilog HDL。

完成对之前的 simple_counter.v 文件的模块设计，用于输出计数值，进而控制 LED 灯的显示。再将 Verilog 代码通过 File>Create/Update>Create Symbol Files for Current File，为这个文件生成对应的符号文件（.sym）。（生成 符号文件是将 HDL 文件转换为原理图符号的过程，这个可以更加方便地使用和调用这些模块。）将转化好的符号文件插入到对应顶层文件的原理图中。之后就可以通过双击该项目 Project 库中对应 Verilog 文件生成相应原理图模块，并且添加到原理图中。

simple_counter.v 的代码如下。

```
//It has a single clock input and a 32-bit output port
module simple_counter (
CLOCK_50,
counter_out
);
input CLOCK_50;
output [31:0] counter_out;
reg [31:0] counter_out; // 32位二进制计数器

always @(posedge CLOCK_50)
begin
counter_out <= #1 counter_out + 1;// increment counter
// 赋值操作延时1个时间单位进行，溢出之后重新从0开始
end
endmodule
```

4.2.2 添加 PLL 宏功能模块。

本实验使用的是 DE2-115 板上的晶振来产生 PLL，该振荡器的频率为 50MHz。在 Edit>Insert Symbol 使用 Megawizard Plug-in Manager 来新建一个 custom megafunction variation。在使用过程中，选择正确的 IO、器件系列和文件输出类型以及文件名、设别系列、设备速度，设置 inclock0 输入频率。通过 Megawiard Plug-in Manager 生成 PLL megafunction，也可以通过 PLL 符号文件（和 Verilog 文件一样都转化成符号文件，便于添加到原理图设计中）。

常用器件插入方式是在原理图右键单击 >Insert Symbol 在 primitive/pin 下有常见的输入输出和 bidir。

原理图总线宽度定义方法为 name[X..Y] 中，X 是最高位，Y 是最低位。

在此基础上，进行初步的器件连接，包括时钟电路与简单计数电路，电路的输入输出之间的连接，如下图所示。

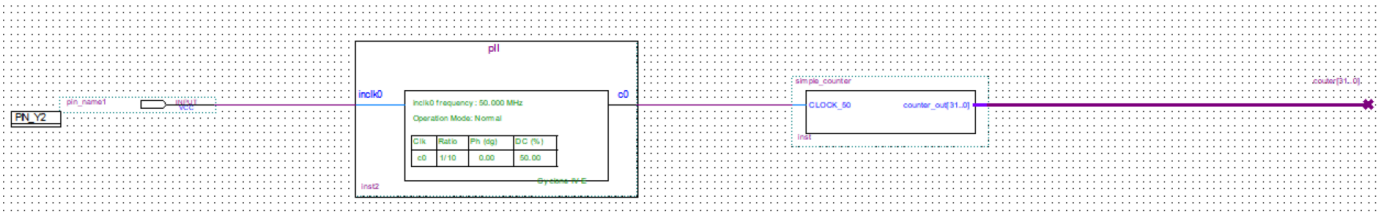

图 4.1 – 时钟电路与计数电路之间的连线设计

4.2.3 添加多路复用器

使用 MegaWizard Plug-in Manager 来添加多路复用器 lpm_mux。将两种不同的计数器总线多路复用到 DE2-115 开发板上的 4 个 LED。（多路复用器即数据选择器，用来将 N 个输入通道的数据复用到一个输出通道上。）

开发方式：在 MegaWizard Plug-in Manger 中选择 Install Plug-Ins > Gates > LPM_MUX，选择对应的开发板设备和文件（counter_bus_mux.v）输出类型。并且根据自己的设计需求选择输入信号路数（2）和输出信号的宽度（4bits）。使用该方法可以生成多路复用器的符号，然后将设计好的多路复用器添加到原理图中，用于 LED 输出控制。

4.2.4 多路复用器的使用

通过编辑不同数据线的名称表示器件引脚之间的相连关系，在多路复用器中，其中一路使用 counter[26..23]，另一路使用 counter[24:21] 的数据作为输出结果，这两种输出之间的频率相差 4 倍。且多路复用器的选择通过 selection 信号决定，在本实验中应当对应 KEY[0] 按键的事件。

多路复用器的具体连线规划以及整个电路的电路设计图如下所示。

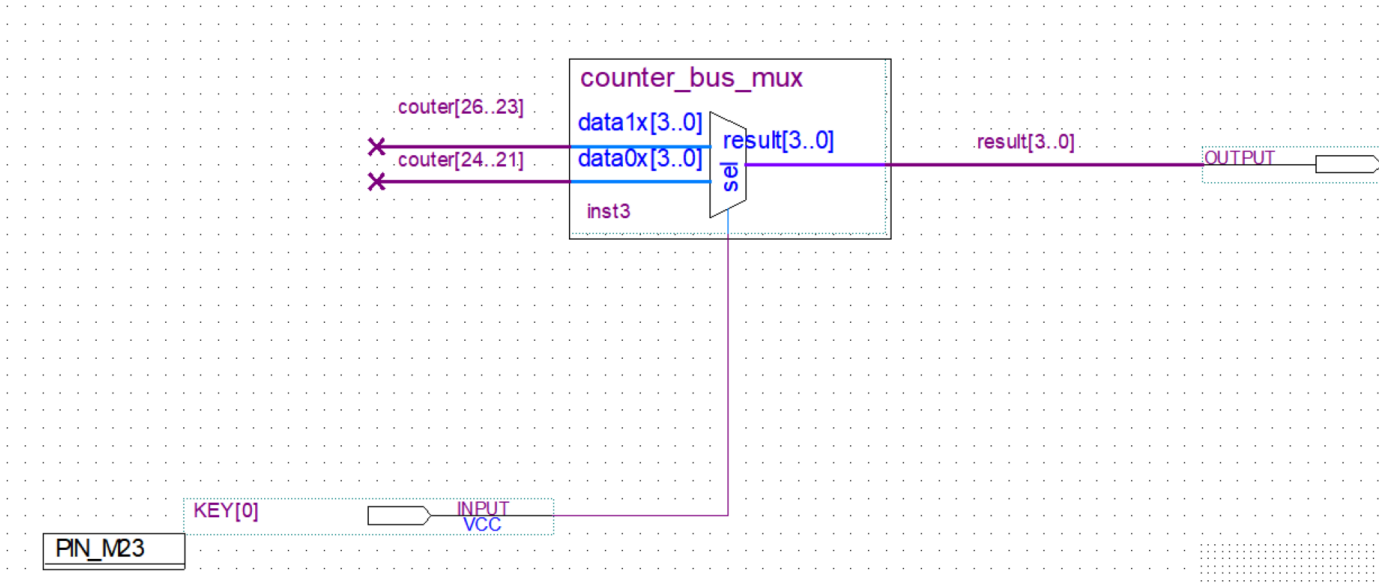

图 4.2 – 多路复用器连线设计

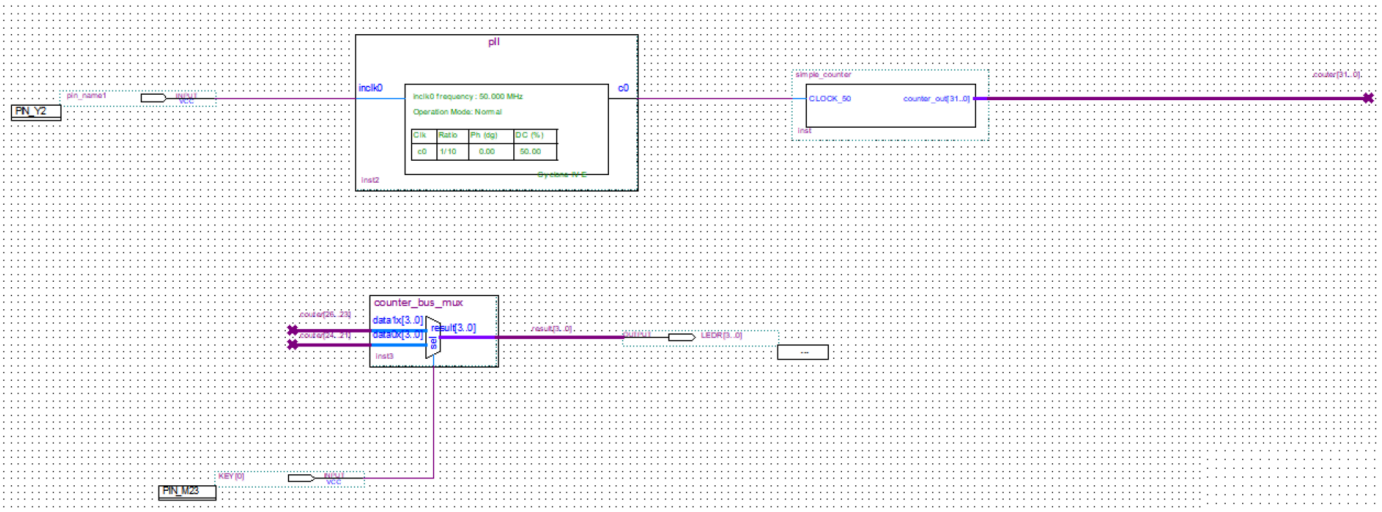

图 4.3 – 原理图

4.2.5 引脚分配

完成原理图设计之后为输入输出信号分配硬件引脚，分别是输入信号 KEY[0] 对应 M23，CLOCK_50 对应 Y2；输出信号 LEDR[3] 对应 F21，LEDR[2] 对应 E19，LEDR[1] 对应 F19，LEDR[0] 对应 G19。

4.2.6 编写时序约束文件

时序约束文件 Default TimeQuest SDC File 该文件是 Synopsys 设计约束文件。

创建文件的过程为：使用 Tool 打开 TimeQuest Timing Analyzer，选择创建新的 SDC 文件，并且命名应当与项目名称一致。

SDC 文件内容如下所示

```
create_clock -period 20.000 -name CLOCK_50
derive_pll_clocks
derive_clock_uncertainty
```

4.3 编译与验证

使用 compile 完成编译与 Debug，将.sof 文件烧到开发板上进行验证。在硬件验证中，需要将 J9 与 USB-Blaster 连接，并且将 USB-Blaster 与 USB 连接到主机上。

五、实验结果与分析

实验结果需要在时间连续的范围上可以直接观察到以下实验结果。当按键 KEY[0] 未被按下时，LEDR0、LEDR1、LEDR2、LEDR3 这四个 LED 灯作为输出指示灯，指示 0000b 到 1111b 十六个二进制数的周期变化，即从 0000b 递增到 1111b，1111b 的下一个变化又回到 0000b，以此循环。当 KEY[0] 被按下的时候，LEDR 们的变化顺序与未被按下时一致，但是变化的频率更快，根据多路复用器的设计，理论上按键被按下时的 LED 变化频率为未被按下的变化频率的 4 倍。从实验实际操作中可以明显地观察到两种情况下 LED 灯变化速度的差异。

此外，还根据实验指导手册的说明，调整未使用到的 LED 灯的输入类型为三态输入引脚（As input tri-stated），保持输出结果的正确性，让理论上的闲置 LED 灯不会发出微弱的亮光。总结来说，实验结果与实验开发设计的目的一致。

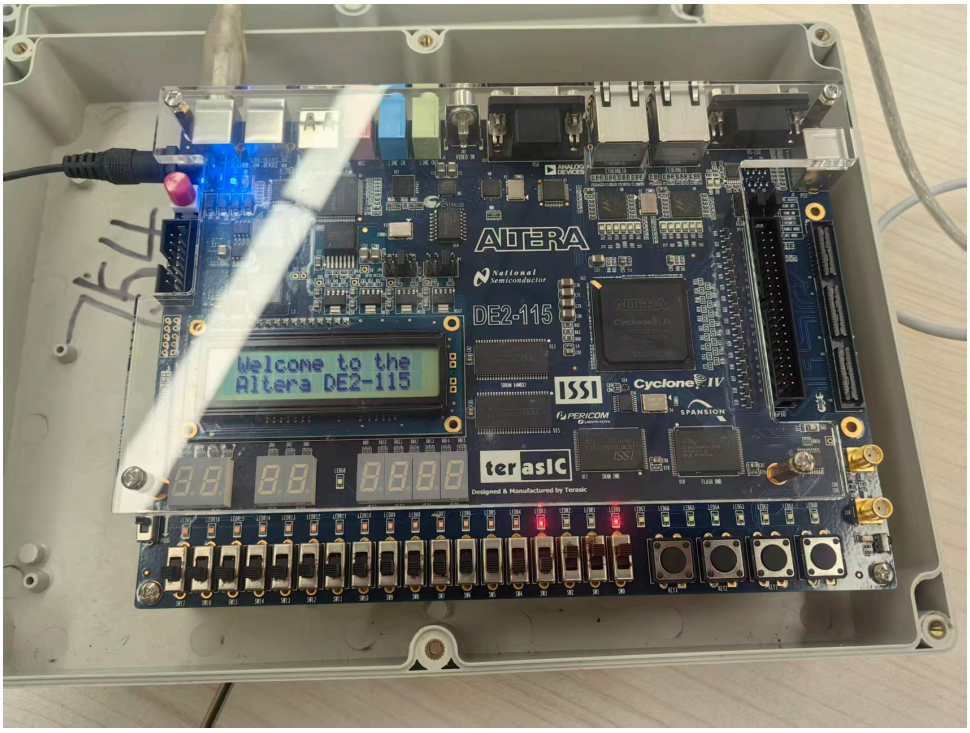

图 5.1 – 未进行限制管脚三态输入时的输出结果

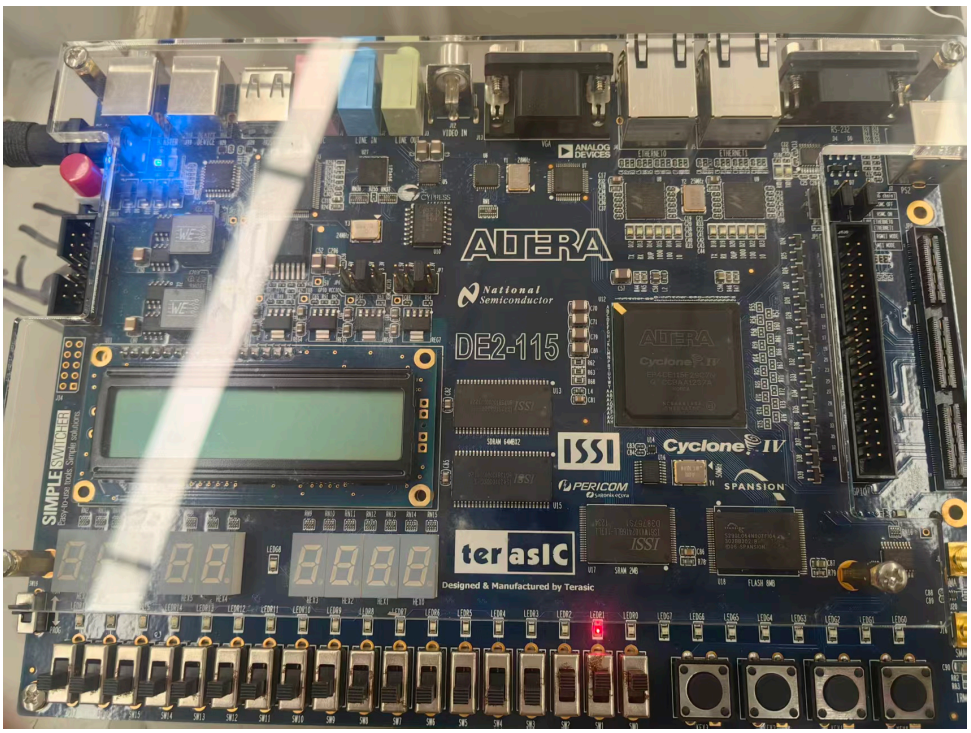

图 5.2 – 进行限制管脚三态输入时的输出结果

六、实验心得

本次实验不仅是我首次真正意义上接触 FPGA 开发，也是我从其他专业课更注重的软件顺序执行思维向硬件并行运行转变的起点，有点类似于《微机原理及应用》课程中田翔老师强调的软硬结合的思想。尽管本次实验的实验手册提供了详尽的操作步骤，但在一步步执行的过程中，我结合理论知识进行了深入的思考，对整个 EDA 设计流程有了更加立体的认知。

首先，我深刻体会到了自顶向下的模块化设计思想在 FPGA 开发中的重要性。在本次设计中，整个系统被清晰地划分为三个子模块：负责提供稳定时钟节拍的 PLL 宏模块、负责核心逻辑与分频的 32 位计数器模块（simple_counter.v），以及负责信号路由的多路复用器模块（LPM_MUX）。Quartus 平台极大地体现了设计的灵活性：对于简单的计数逻辑，我们可以使用 Verilog HDL 进行底层代码编写；而对于 PLL 和多路复用器这种标准化的高级模块，通过 MegaWizard Plug-in Manager 调用 IP 核则显得更加高效且不易出错。将代码与 IP 核统一转换为符号文件（.sym）并在顶层原理图（.bdf）中连线，这种图文结合的方式直观地展现了数据流的走向。

其次，通过对 Verilog 代码细节的深挖，我初步理解了硬件描述语言的本质特征。在编写 32 位计数器时，Verilog 语句让我直观感受到了时序电路受时钟沿触发的特性。同时，我在代码中注意到了非阻塞赋值（<=）以及人为加入的 1 延时操作（counter_out <= 1 counter_out + 1;）。结合资料我了解到，虽然在真实硬件综合中这种微小的 1 延时可能被忽略，但它在逻辑仿真阶段对于模拟真实的门延迟、防止时序竞争冒险具有重要意义，这与 C 语言中简单的变量赋值有着本质的区别。另外，巧妙地截取计数器的高位（[26..23] 和 [24..21]）来作为多路复用器的输入，利用二进制逐级分频的特性将 50MHz 的高频降至人眼可分辨的闪烁频率，这种软硬结合的设计思路让我感到非常巧妙。

最后，在下板验证阶段，我也体会到了硬件工程师的严谨。除了顺利完成按键控制 LED 快慢闪烁（4 倍频差）的核心要求外，我还根据指导对未使用的引脚设置了三态输入（As input tri-stated）。

这个细节处理让我认识到，FPGA 中的闲置管脚如果处于悬空或未定义状态，极易受到外部电磁干扰而产生微弱漏电或不可预知的逻辑错误 。

总而言之，通过本次实验，我走通了从“建立工程 -> 逻辑设计 -> 引脚分配 -> 编译综合 -> 烧录验证”的完整闭环 。我深知目前对 Quartus II 以及底层时序约束（如 TimeQuest SDC 的使用）的理解还比较浅显 ，在未来的学习中，我需要进一步强化 Verilog 语法基础，并多在实际问题中锻炼排错（Debug）能力，真正将 EDA 技术内化为自己的工程开发利器。