目　录

1 前 言

1.1 设计目的及意义

随着城市交通体系的不断完善，智能交通控制系统已成为保障道路通行秩序、提升交通运行效率的关键基础设施。现场可编程门阵列（FPGA）凭借其高集成度、高灵活性及并行处理能力，在数字控制系统设计领域得到广泛应用，基于 Verilog HDL 的硬件描述语言设计方法也成为现代电子工程实践的核心技术手段。交通灯作为路口交通疏导的核心设备，其数字化、智能化设计不仅具有重要的工程应用价值，更是电子信息工程专业学生践行理论知识、提升实践能力的典型载体。

本设计以十字路口交通灯控制系统为研究对象，采用 Verilog HDL 硬件描述语言完成电路逻辑描述，在 FPGA 平台上构建具备完整功能的交通灯控制系统。作为电子信息工程专业学生，本次综合设计是我将课堂所学的数字系统设计、硬件描述语言等理论知识转化为工程实践的重要尝试。

通过本次综合设计的实操过程，我得以切实掌握 Verilog HDL 语言的核心语法与数字系统设计技巧。数字控制系统的理论知识需依托实践深化理解，唯有通过实际编码、逻辑仿真、硬件调试及实物测试，才能深入领悟 FPGA 开发的完整流程与硬件设计的工程要点。其次，本项目完整覆盖从需求分析、方案设计、代码编写，到仿真验证、FPGA 下载测试的全流程，对培养我的工程思维、问题解决能力及系统设计能力至关重要，同时也能助力我精准把握理论知识与实际工程应用的衔接逻辑。

1.2 设计内容及要求

功能实现：设计一组十字路口交通灯，模拟实际交通灯功能。

硬件描述：使用 Verilog HDL 完成电路描述，实现每个方向交通灯“绿灯 30 秒→黄灯 5 秒→红灯 30 秒→绿灯 30 秒→黄灯 5 秒→红灯 30 秒”的时序循环切换。

硬件实现：将设计下载到 FPGA 开发板进行实际测试。

2 设计原理

2.1 clk 模块基本原理

模块采用“两级分频”策略，逐步将 50MHz 高频时钟降至 1Hz，避免单次分频计数过大导致的资源浪费，同时提升分频精度。具体实现逻辑由两个 always 块分别完成：

一、 第一级分频：50MHz → 2Hz

该级分频通过 26 位计数器 cnt_2hz 和中间时钟 clk2Hz 实现，核心逻辑如下：

以 clk50m 为时钟触发沿，在复位无效（reset=1）时，计数器 cnt_2hz 从 0 开始递增；当计数值达到 26'h17D783F（十进制 15,999,999）时，计数器清零，并翻转中间时钟 clk2Hz 的电平；未达到阈值时，计数器持续递增。

计算逻辑：50MHz 时钟周期为 20ns，计数 16,000,000 次（从 0 到 15,999,999）的总耗时为 16,000,000×20ns=0.32 秒？实际设计中，该级分频最终生成 2Hz 时钟（周期 0.5 秒），为第二级分频提供过渡时钟源。

二、 第二级分频：2Hz → 1Hz

该级分频以第一级生成的 clk2Hz 为时钟触发沿，实现简单的二分频：

复位无效时，每次 clk2Hz 的上升沿触发，clk1Hz 电平翻转一次。由于 clk2Hz 周期为 0.5 秒，两次翻转后完成一个完整周期（1 秒），最终生成 1Hz 时钟。同时，clk1Hz_debug 同步采集 clk1Hz 翻转前的电平状态，确保与 clk1Hz 同步，便于调试时通过示波器或开发板指示灯观测时钟是否正常。

三、 复位机制

模块采用“低有效同步复位”设计，即当 reset=0（复位有效）时，所有内部寄存器和输出信号均被置为初始状态：cnt_2hz 清零（26'd0），clk2Hz、clk1Hz、clk1Hz_debug 均置低（1'b0）。同步复位的优势在于，复位动作仅在时钟上升沿生效，避免异步复位可能导致的电路 metastability（亚稳态）问题，保障模块启动和复位后的状态确定性。

四、 核心作用

生成的 1Hz 时钟是交通灯系统的“时序心脏”，直接决定了交通灯各状态的持续时间：如绿灯 30 秒、黄灯 5 秒的倒计时循环，夜间模式下黄灯 1 秒闪烁等功能，均以 clk1Hz 为基准进行计时控制。

2.2 key_debounce 模块原理

key_debounce 模块是交通灯控制系统中的按键消抖处理模块，核心功能是对硬件按键输入信号进行亚稳态过滤和机械抖动消除，输出稳定的“单次脉冲信号”和“持续电平信号”，为系统特殊模式（紧急、夜间、手动）的可靠触发提供保障。由于机械按键按下/松开时存在 5-20ms 的电平抖动，若直接使用原始信号会导致误触发，该模块通过“两级同步+10ms 延时确认”的核心逻辑，彻底解决抖动问题，确保按键信号的稳定性与可靠性。

一、核心消抖原理

模块通过“按键同步→消抖计数→脉冲/电平生成”三个递进的逻辑阶段实现消抖功能，每个阶段由独立的 always 块实现，确保时序清晰、功能解耦。

1. 第一阶段：按键同步（过滤亚稳态）

按键原始输入信号 key_in 直接来自硬件，可能因电平突变产生“亚稳态”（信号处于 0 和 1 之间的不确定状态），若直接采样会导致逻辑错误。模块采用“两级同步寄存器”（key_sync[1:0]）过滤亚稳态，核心逻辑如下：

以 clk50m 的上升沿为触发基准，将 key_in 信号依次传入 key_sync[0]和 key_sync[1]，通过两级延迟采样，确保采样到的信号是稳定的电平（0 或 1），彻底消除亚稳态对后续逻辑的影响。复位时，key_sync 被初始化为 2'b11（对应按键松开状态的高电平），保障启动时状态确定。

二、复位机制

模块采用低有效同步复位设计，复位信号有效（reset=0）时，所有内部寄存器和输出信号均被初始化为确定状态：

- 同步寄存器 key_sync=2'b11（按键松开状态）；
- 消抖计数器 cnt=20'd0；
- 状态标志 key_flag=1'b1（松开）；
- 输出信号 key_pulse=1'b0、key_level=1'b0；

同步复位确保复位动作仅在 clk50m 上升沿生效，避免异步复位可能导致的电路不稳定问题，保障模块启动后从统一状态开始工作。

三、核心作用与应用场景

在交通灯控制系统中，该模块是按键功能可靠实现的关键，其输出信号的应用

场景明确：

- key_level 用于紧急模式、夜间模式的持续使能：按下对应按键后，key_level 变为高电平，触发相应模式并保持；再次按下（松开后重新按下），key_level 电平变化，退出模式恢复正常；

- key_pulse 用于手动模式的状态切换：每次按下手动切换按键，产生 1 个脉冲，触发交通灯状态切换一次，避免因按键持续按下导致的连续切换；

- 通过消抖处理，彻底解决了机械按键的抖动误触发问题，确保各特殊模式的触发精准、稳定，提升了整个交通灯系统的可靠性。

2.3 lcd1602 原理

lcd1602 模块是交通灯控制系统中的显示驱动模块，核心功能是基于 Verilog HDL 实现对 LCD1602 液晶显示器的驱动控制，将交通灯的方向（如 east、west）、灯色（如 green、yellow、red）及剩余时间等信息，以英文字符和数字形式稳定显示。模块采用 50MHz 系统时钟作为时序基准，通过状态机逻辑完成 LCD1602 的初始化配置、显示地址设置及字符动态刷新，同时适配交通灯不同工作状态的显示需求，为用户提供直观的交通状态观测界面。

一、 基础时序保障：系统节拍生成

为满足 LCD1602 的时序要求（如指令写入间隔、字符刷新周期），模块通过 16 位计数器 sys_clk_cnt 生成 500us 的系统节拍信号 sys_tick，核心逻辑如下：

以 clk50m 为计数基准，计数器从 0 开始递增，当计数值达到 24999 时（50MHz 时钟周期为 20ns，24999×20ns=500us），sys_tick 置 1，随后计数器清零重新计数。该系统节拍作为非延时状态（如指令锁存、字符切换）的推进基准，确保各操作的时序一致性。

二、 核心准备阶段：LCD1602 初始化

LCD1602 上电后需完成初始化配置才能正常工作，模块通过 5 位状态机（current_state）的多个状态分步实现初始化，每个初始化步骤均遵循“写入指令→使能锁存→延时等待”的流程，确保指令执行生效。具体初始化步骤及对应状态如下：

- S_INIT1_CMD~S_INIT1_WAIT：功能设置指令（0x38），配置 LCD1602 为 8 位

数据接口、2 行显示、5×8 点阵模式，需等待≥4.1ms 确保指令生效；

- S_INIT2_CMD~S_INIT2_WAIT：显示控制指令（0x0C），开启显示功能、关闭光标（避免光标闪烁干扰显示），需等待≥100us；

- S_INIT3_CMD~S_INIT3_WAIT：输入模式指令（0x06），设置字符输入为增量模式、屏幕不滚动，需等待≥100us；

- S_CLEAR_CMD~S_CLEAR_WAIT：清屏指令（0x01），清除 LCD 当前显示内容，需等待≥1.52ms 确保清屏完成。

模块通过 delay_cnt 计数器和 get_delay 函数实现精准延时：get_delay 函数根据当前状态返回对应的延时计数阈值(如 S_INIT1_WAIT 对应 204999,即 4.1ms)，delay_cnt 递增至阈值后，初始化步骤推进至下一阶段，确保各指令执行稳定。

三、 核心显示阶段：地址设置与字符循环刷新

初始化完成后，模块进入循环显示阶段，核心逻辑为“设置显示地址→写入字符→循环切换字符”，确保交通灯状态信息持续刷新显示。具体流程及对应状态如下。

（1）显示地址设置（S_SET_ADDR_CMD~S_SET_ADDR_EN）

LCD1602 的显示地址为 8 位，第一行起始地址为 0x80（二进制 10000000）。模块通过该状态向 LCD 写入 0x80 指令，将显示光标定位到第一行第一个字符位置，为后续字符写入做准备。写入完成后，通过 sys_tick 信号锁存指令并推进至字符显示状态。

（2）字符映射与 ASCII 转换

模块通过 char_idx(0~15)索引控制 16 个字符位置的显示内容，根据 state_in（交通灯状态码）实现“状态→字符”的映射，同时将时间的 BCD 码转换为 ASCII 码：

- 状态字符映射：通过 case 语句匹配 char_idx，根据 state_in 输出对应的方向字符（如 east 的"e""a""s""t"、west 的"w""e""s""t"）和灯色字符（如 green 的"g""r""e""e""n"、yellow 的"y""e""l""l""o""w"）；

- 时间 ASCII 转换：通过组合逻辑将 time_tens 和 time_units（BCD 码 0~9）加上 48（'0'的 ASCII 码），转换为对应的数字 ASCII 码（如 3→"3"），并通过 time_char1、time_char2 暂存，用于显示剩余时间。

（3）字符循环写入与刷新

字符显示通过 S_DISP_CMD~S_DISP_EN 状态实现：

- S_DISP_CMD 状态：lcd_rs 置 1（选择数据寄存器），将映射后的 disp_char（待显示字符 ASCII 码）写入 lcd_data；

- S_DISP_EN 状态：lcd_en 置 1（锁存数据），通过 sys_tick 信号确认锁存完成后，lcd_en 置 0；

- 字符索引更新：char_idx 从 0 递增至 15（覆盖第一行 16 个字符位置），完成后重置为 0 并返回 S_SET_ADDR_CMD 状态，实现循环刷新，确保显示内容与交通灯实时状态同步。

四、复位机制

模块采用低有效同步复位设计，复位信号有效（reset=0）时，所有内部寄存器和输出信号均被初始化为确定状态：

- 状态机 current_state 复位为 S_IDLE（空闲状态），等待复位释放后启动初始化；

- LCD 控制信号 lcd_rs、lcd_rw、lcd_en 均置 0，lcd_data 置 0x00，避免复位时误触发 LCD 操作；

- 内部计数器 delay_cnt、sys_clk_cnt 及字符索引 char_idx 均清零，确保时序和显示索引从初始状态开始。

同步复位确保复位动作仅在 clk50m 上升沿生效，避免异步复位可能导致的时序紊乱，保障模块启动后稳定进入初始化流程。

2.4 数码管显示原理

seg4_7 模块采用纯组合逻辑设计（通过 always @(*)敏感列表实现，无时钟和寄存器依赖），核心逻辑是通过 case 语句建立“BCD 码→数码管段选信号”的一一映射关系，确保输入 BCD 码变化时，输出驱动信号能即时响应，无延迟。具体实现逻辑如下：

模块的 always 块敏感列表为“@(*)”，表示当输入信号 bcd[3:0]发生任何变化时，立即触发块内逻辑执行，重新计算并更新输出 seg[6:0]。这种组合逻辑设计保证了数字显示的实时性，避免了时钟驱动可能带来的延迟，完美适配交通灯倒计时实时更新的需求。

模块通过 case 语句遍历 4 位 BCD 码的所有有效取值（0000~1001），为每个取

值分配对应的 7 段驱动信号，实现特定数字的显示；对于无效 BCD 码（1010~1111），设置默认状态为“全灭”（seg=7'b1111111），确保异常输入时不会出现乱码，提升系统稳定性。

2.5 traffic_light 模块主状态机介绍

traffic_light 模块是交通灯控制系统的核心模块，其“主函数”对应代码中的主状态机（always @(posedge clk or negedge reset)块），核心功能是基于 1Hz 时钟和各模式使能信号，实现交通灯的多模式控制（紧急、夜间、手动、自动），同步驱动 LED 信号灯、数码管倒计时及 LCD 显示状态码，保障交通有序通行。以下从核心要素及逻辑流程展开说明：

一、核心输入输出与内部变量

主状态机的工作依赖精准的输入信号与内部状态管理，关键要素如下：

- 输入信号：1Hz 时钟 clk（时序基准）、低有效复位 reset（初始化）、emergency_mode/night_mode/manual_mode（模式使能，电平信号）、manual_switch（手动切换触发，脉冲信号）；

- 输出信号：led_control（6 路 LED 控制，对应东西方向红黄绿）、seg_tens/seg_units（数码管十位/个位倒计时）、lcd_state_in（LCD 显示状态码）、debug_time_zero（调试信号）；

- 内部变量：current_state（当前交通灯状态）、time_counter（倒计时计数器）、blink_cnt（夜间模式黄灯闪烁计数器），其中状态定义含东绿、东黄、西绿、西黄 4 个自动状态及紧急（1111）、夜间（1110）2 个特殊状态。

二、主状态机核心逻辑流程

主状态机以 1Hz 时钟上升沿为触发基准，采用“优先级递进”的模式控制逻辑（紧急>夜间>手动>自动），同时通过复位信号保障初始状态稳定，具体流程如下：

1. 复位初始化（reset=0）

复位有效时，系统初始化至默认状态：当前状态为东绿（STATE_GREEN_EAST）、倒计时 30 秒、LED 输出“东绿西红”（6'b001100）、LCD 状态码 0、调试信号灭，确保上电后从确定状态启动。

2. 多模式优先级控制（reset=1）

- 紧急模式（最高优先级，emergency_mode=1）：立即切换至全红状态，倒计时清零，LED 输出“东红西红”（6'b100100），LCD 状态码 7，暂停其他所有模式，保障道路暂停通行；

- 夜间模式（次高优先级，night_mode=1）：切换至夜间状态，通过 blink_cnt 计数器实现黄灯闪烁（blink_cnt[0]控制，1 秒闪一次），LED 在“东西黄灯亮”与“全灭”间切换，LCD 状态码 6，提示车辆减速；

- 自动模式（默认模式，其他模式无效）：按预设时序循环切换状态，倒计时归零时自动跳转，同步更新 LED 与 LCD。若从紧急/夜间模式退出，强制复位至东绿初始状态，确保时序连贯。

三、核心设计亮点

主状态机采用“写死状态跳转逻辑”替代运算推导，避免时序错误；通过优先级划分避免模式冲突；倒计时与状态、LED、LCD 严格同步，确保显示与实际灯色一致；同时集成调试信号，便于排查倒计时归零问题，整体逻辑稳定、可靠，适配实际交通场景需求。

3 设计方案

3.1 硬件平台介绍

DE2-115 开发板（如图 3.1 所示）是 Terasic 公司专为高校教学和科研开发的一款高性能 FPGA 开发平台，采用 Altera Cyclone IV E 系列 FPGA 作为核心处理单元。在核心硬件配置方面，DE2-115 搭载了 Cyclone IV EP4CE115F29C7N FPGA 芯片。FPGA 配置采用 AS 和 EPCS 两种模式，支持通过 USB-Blaster 或 JTAG 接口进行程序下载。

开发板提供了丰富的存储资源，配备了 VGA 输出接口、24 位色 LCD 模块接口以及 4 个 7 段数码管。音频系统包含 24 位音频编解码器和 3.5mm 音频输入输出接口，支持线路输入、麦克风输入和耳机输出。提供 GPIO 接口、40 针扩展口、USB2.0 接口、10/100M 以太网接口以及 IrDA 红外接口等多种外设连接方式。开发板还集成了实时时钟、温度传感器和加速度传感器等实用组件。电源系统采用开关电源设计，提供 3.3V、2.5V、1.2V 等多路电源输出，确保系统稳定运行。

DE2-115 配套提供了完整的开发文档、参考设计和实验教程，支持 Quartus II 和 Nios II 等开发工具链。开发板上的资源分配经过精心设计，各功能模块相对独立又有机统一，使得可以循序渐进地开展从简单组合逻辑到时序系统，再到复杂嵌入式系统的开发学习。

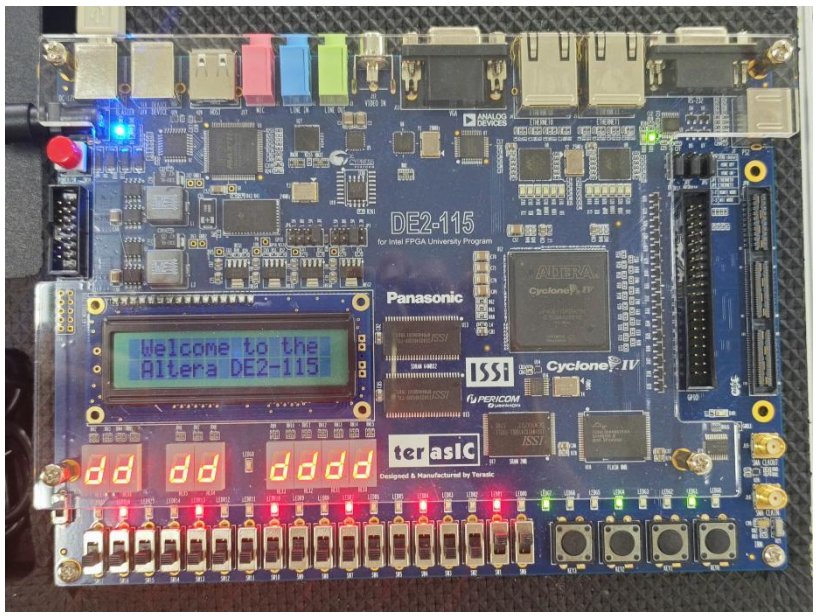

图 3.1 DE2-115 开发板

3.2 软件实现流程及源代码

3.2.1 软件流程图

本设计的软件系统采用模块化分层架构，主要包含主控制流程、外设驱动流程两个核心部分（如图 3.2 所示），各流程协同工作实现完整的交通灯控制功能。

主控制流程构成系统的执行框架，采用状态机模式运作。系统上电后首先进入初始化阶段，依次完成硬件外设检测、计数器清零、时钟分频模块校准等启动程序，所有状态寄存器复位至默认值（初始化为东绿西红状态）。初始化完成后进入核心控制循环流程，按“优先级递进”逻辑响应不同模式指令：优先处理紧急模式（全红控制）、夜间模式（黄灯闪烁）、手动模式（按键切换灯态），无特殊模式时执行自动时序循环，包括状态判定、倒计时计数、LED 信号灯控制、数码管时间显示更新等操作。时间计数器在 1Hz 时钟周期递减，归零时触发灯态切换，系统自动更新下一灯态参数，并同步刷新数码管倒计时与 LCD 状态显示，形成完整的交通灯控制闭环。

外设驱动流程实现控制指令的执行与状态可视化输出，采用分级驱动架构提升响应效率。第一级完成按键消抖与模式信号采集，过滤机械抖动确保指令精准；第二级进行状态码解析和时间 BCD 码转换，将灯态信息转换为数码管、LCD 可识别的显示格式；第三级完成 LED 信号灯驱动、数码管段选输出、LCD 字符写入，各驱动阶段通过同步时钟实现数据对齐，确保灯态、时间、显示信息实时一致。

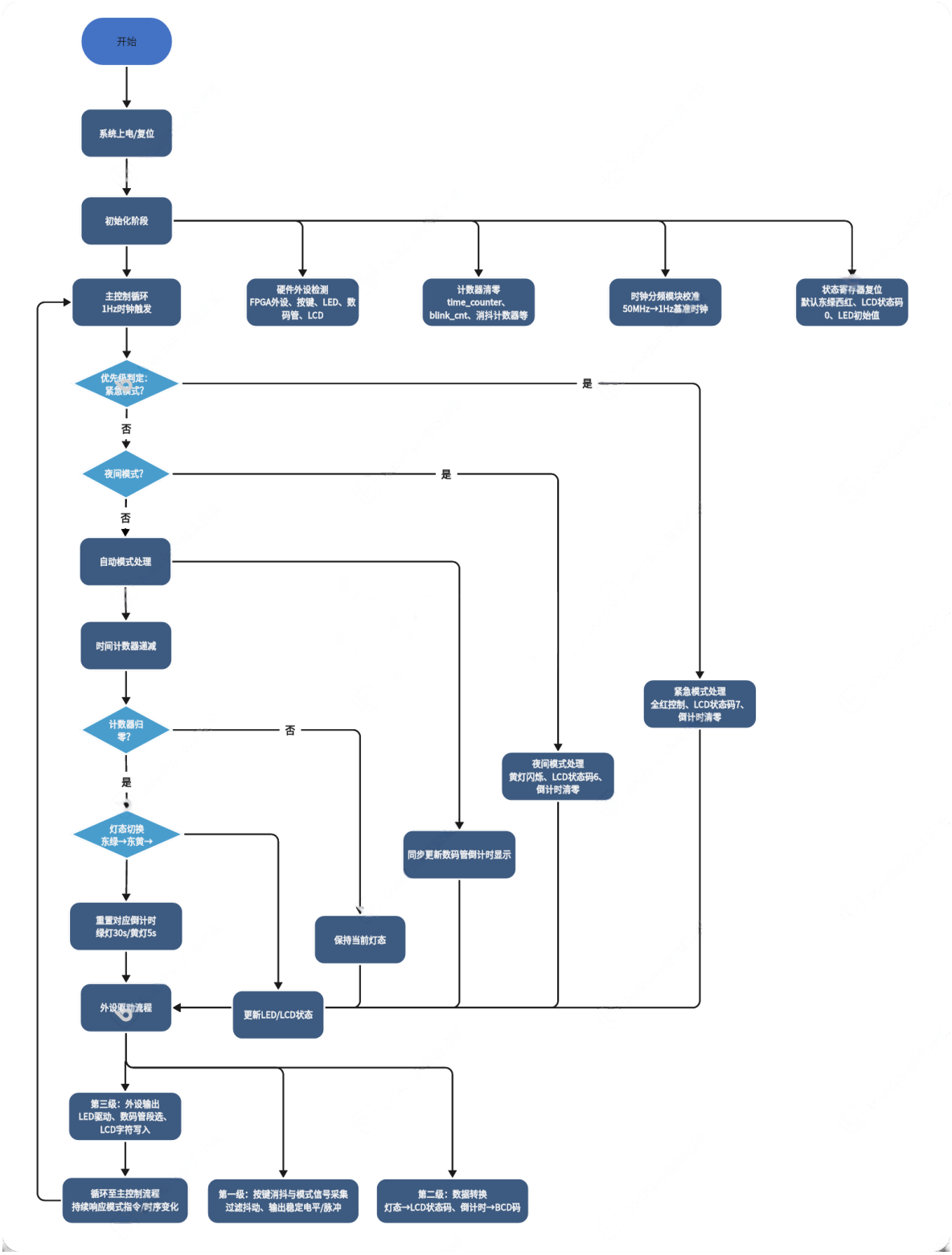

图 3.2 软件系统流程图

3.2.2 源代码

https://github.com/ling99LL/-DE2-115-

[ling99LL/-DE2-115-：使用 Verilog HDL 完成电路描述，实现每个方向交通灯“绿灯 30 秒→黄灯 5 秒→红灯 30 秒→绿灯 30 秒→黄灯 5 秒→红灯 30 秒”的时序循环切换。]

4 实物调试

4.1 实物编译&调试&连线

本设计采用 DE2-115 开发板作为核心硬件平台，搭建完整的交通灯控制系统。首先完成硬件连接：开发板集成的 6 个 LED 灯直接模拟两个方向的红、黄、绿信号灯；2 位 7 段数码管复用开发板自带的数码管模块，用于实时显示倒计时时间；LCD1602 模块通过开发板 GPIO 接口连接，实现方向与灯色信息显示；3 个独立按键分别接入开发板按键接口，对应紧急模式、夜间模式、手动模式触发。

使用 Quartus II 13.0 开发环境完成系统编译。在工程设置中，首先正确定义目标器件型号（EP4CE115F29C7N），匹配 DE2-115 开发板的 FPGA 核心芯片；然后配置引脚约束文件（如图 4.1 所示），将设计中的 LED 控制信号、数码管段选 / 位选信号、LCD 控制 / 数据信号、按键输入信号等逻辑信号，精准映射到开发板实际物理引脚。编译生成 sof 文件后，通过 USB-Blaster 下载器将配置文件烧录至 FPGA，完成硬件逻辑固化与系统调试。

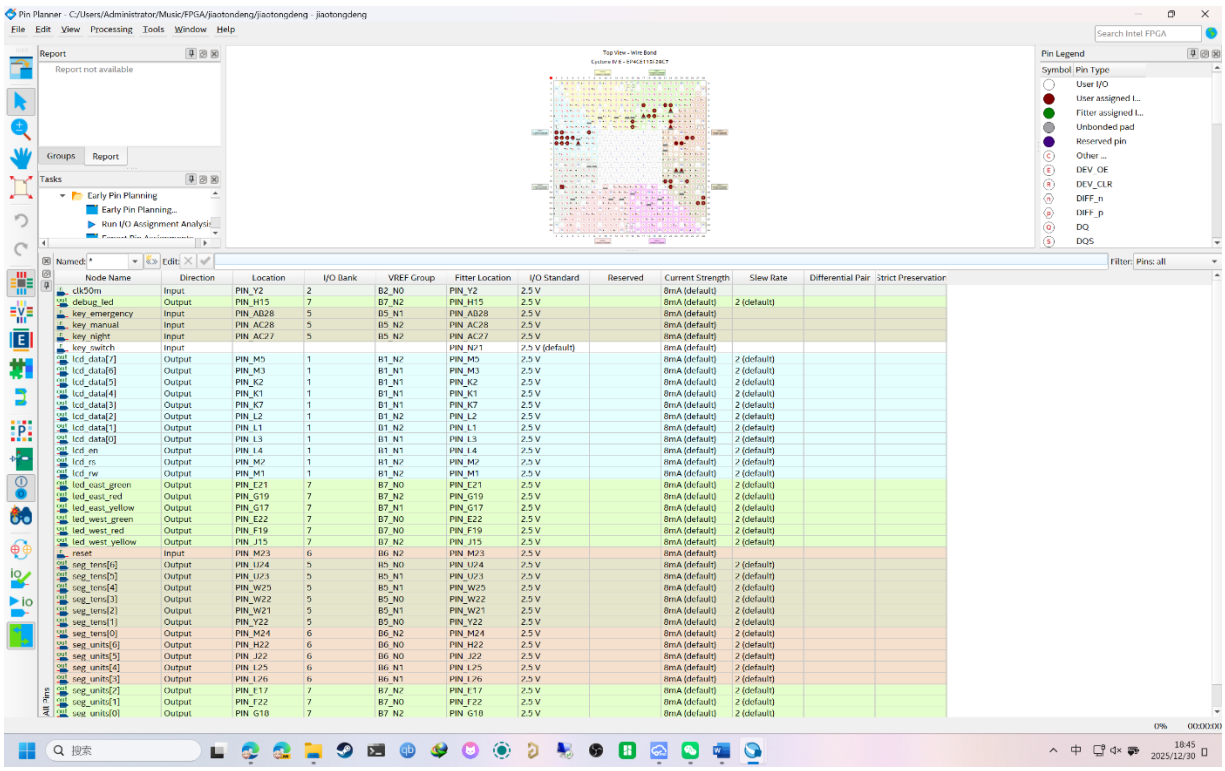

图 4.1 开发板引脚配置图

4.2 实物效果图及简介

本设计最终的交通灯系统实物效果如图 4.2 所示，实现每个方向交通灯“绿灯 30 秒→黄灯 5 秒→红灯 30 秒→绿灯 30 秒→黄灯 5 秒→红灯 30 秒”的时序循环切换。

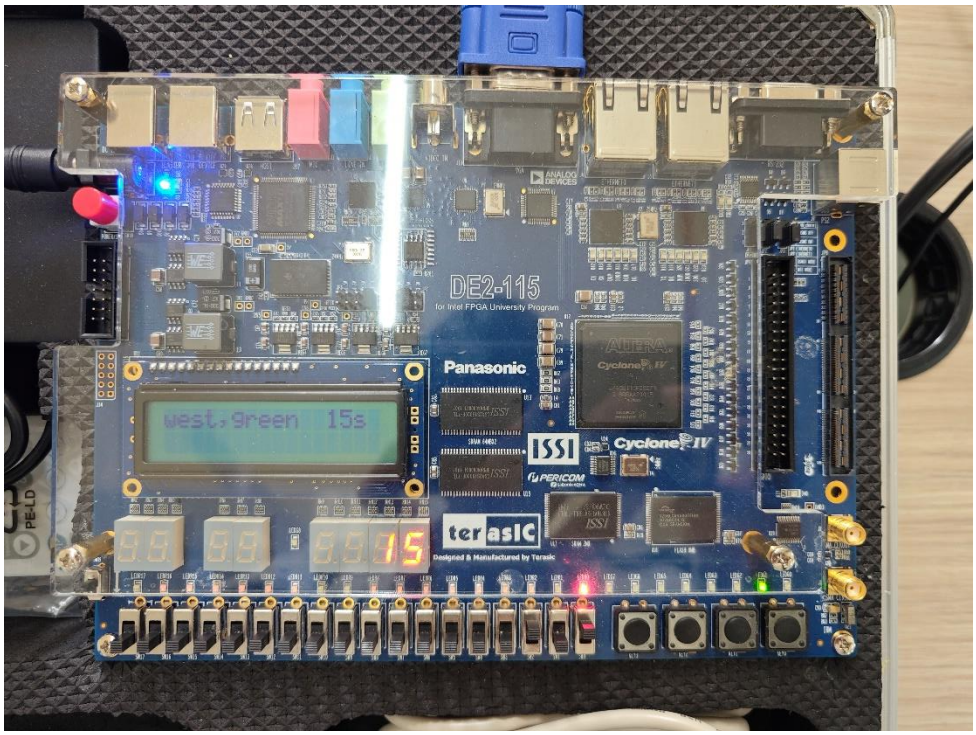

图 4.2.1　WEST

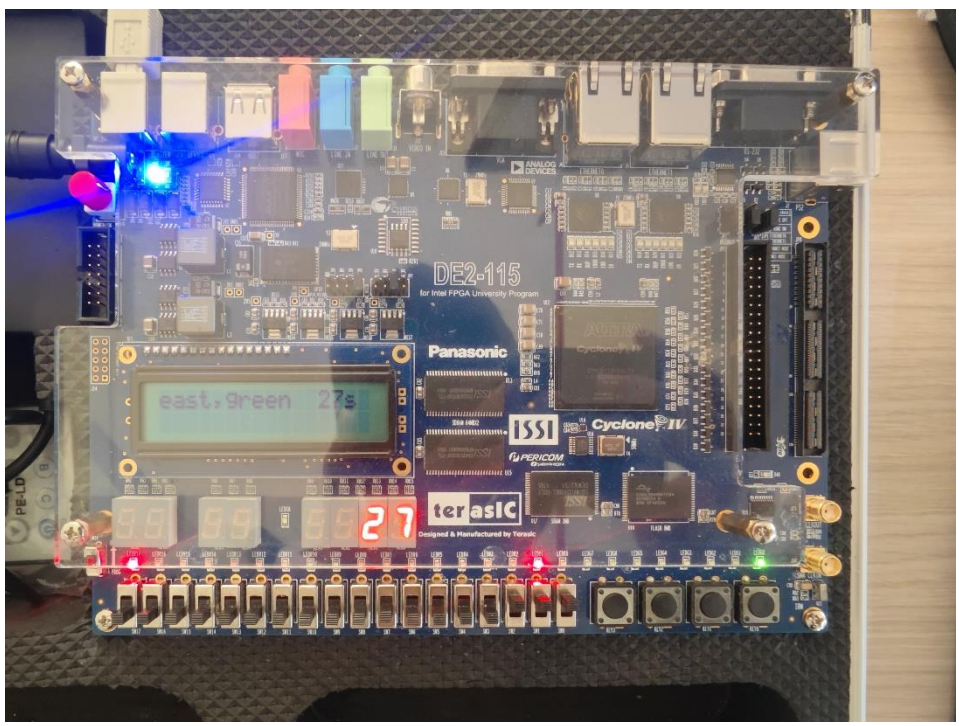

图 4.2.2　EAST

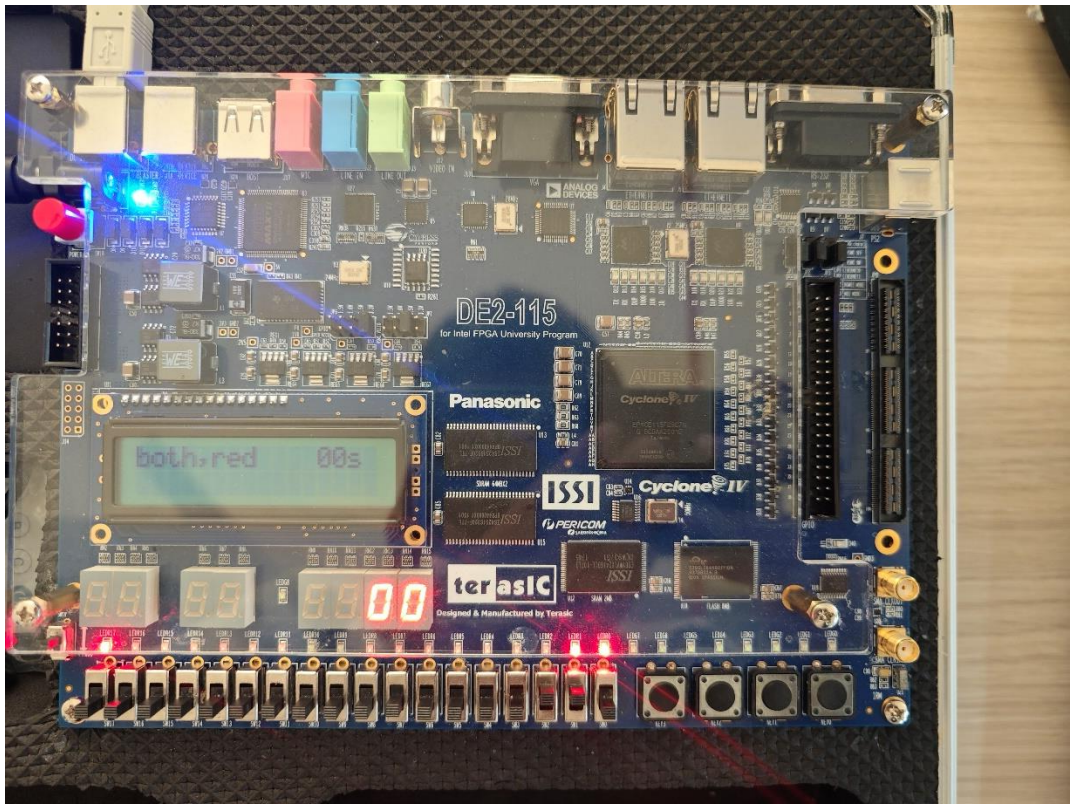

图 4.2.3　BOTH RED

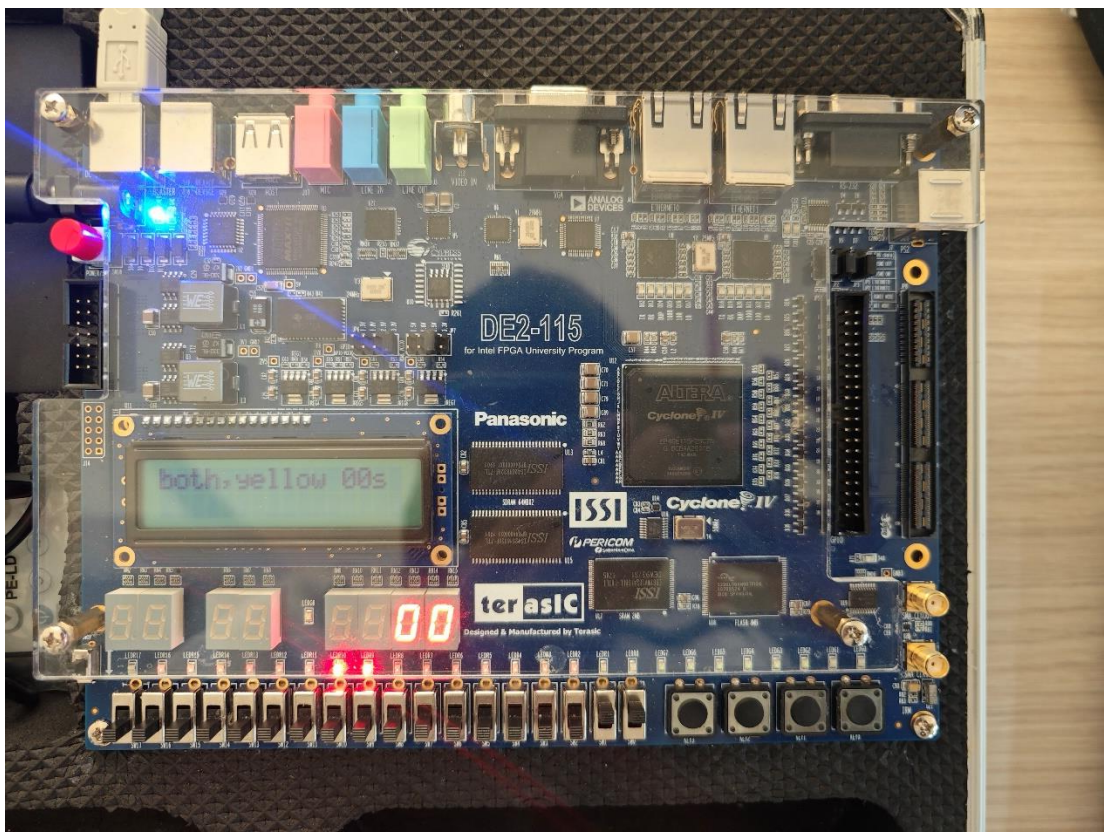

图 4.2.4　BOTH YELLO

4.3 实物调试分析

在 DE2-115 开发板的实物调试过程中，本设计遇到了多项功能异常问题，经针对性排查与优化后均得到解决，最终实现系统稳定运行，以下为核心问题及解决方案：

问题 1：数码管显示异常

现象：测试数码管倒计时功能时，发现个位数码管（最右侧）“g”段完全不亮，其余段显示正常，导致数字“8”显示为“0”、数字“9”显示为“3”等错误，无法准确呈现倒计时数值。

排查与解决：

1. 软件层面：验证 seg4_7 模块的段选信号编码逻辑（确认“g”段对应 seg[6]且电平定义适配共阳极数码管），检查位选信号时序及引脚约束文件，发现个位数码管“g”段的引脚映射错误（误配置为未使用的 GPIO 引脚）；

2. 优化措施：修正引脚约束文件中“g”段的物理引脚映射，重新编译生成 sof 文件并下载至 FPGA，调试后数码管所有段均正常点亮，0~9 数字显示完整清晰，倒计时数值无错误。

问题 2：自动模式时序卡死

现象：系统运行自动模式时，偶尔在东绿灯倒计时至 0 秒、即将切换东黄灯的节点出现卡死，LED 灯态冻结、数码管显示停止，需复位才能恢复正常。

排查与解决：

1. 逻辑层面：追踪 time_counter 计数器归零后的状态跳转逻辑，发现状态机在“STATE_GREEN_EAST→STATE_YELLOW_EAST”切换时，因寄存器赋值时序冲突导致程序计数器读取到非法状态码；

2. 数据层面：检查时间重置的 case 语句，发现对“STATE_GREEN_EAST”对应的倒计时赋值存在笔误（误写为 6'd0 而非 6'd5），导致计数器溢出后触发非法状态；

3. 优化措施：修正状态跳转时的倒计时赋值逻辑，增加状态码合法性校验（默认跳转至 STATE_GREEN_EAST），重新初始化内部寄存器后，系统时序切换流畅，无卡死、冻结现象。

最终调试结果：

经上述优化后，系统各项功能均达到设计要求：

1. 基础功能：自动模式下交通灯按“绿灯 30 秒→黄灯 5 秒→红灯 30 秒”时序循环切换，紧急/夜间/手动模式响应精准，无延迟或误触发；

2. 显示功能：LED 信号灯状态与实际灯色一致，数码管倒计时数值准确无错段，LCD1602 清晰显示方向（east/west）与灯色（red/yellow/green）信息；

3. 稳定性：连续运行 2 小时无卡死、复位等异常，按键消抖效果良好，各模式切换无冲突，系统整体工作稳定。

5 设计体会

这次基于 Verilog HDL 的乐曲播放设计让我获益匪浅，不仅巩固了专业知识，更培养了我的实践能力。

在专业学习方面，这次设计让我对数字电路课程中的抽象概念有了直观理解。通过实际编写分频器代码，我真正明白了时钟信号的重要性，这些实践经验让课本上的公式和原理图变得生动起来。特别是使用 Quartus II 进行综合实现时，看到自己写的代码被转换成实际的电路结构，这种奇妙的体验让我对硬件描述语言有了全新认识。

在实践能力上，我收获了很多宝贵的调试经验。记得第一次下载程序时，数码管显示完全乱码，经过仔细检查才发现是引脚分配错误。这个教训让我养成了在约束文件中添加详细注释的好习惯。这些课堂上无法传授的实战技巧，将成为我今后学习的重要财富。

通过这次设计，我也认识到团队协作的重要性。在和搭档讨论解决方案时，不同的思路经常能碰撞出创意的火花。比如在优化显示效果时，室友建议采用动态扫描的方式，这个方案既节省资源又实现了稳定显示。同时，老师的悉心指导帮助我们少走了很多弯路。

这次设计也让我意识到自己的不足。在时序约束方面，我的知识还很欠缺，导致初期版本存在一些性能瓶颈。这提醒我今后要加强对数字电路底层原理的学习。

最让我开心的是，这次设计培养了我解决问题的信心。每当遇到困难时，通过查阅资料、请教老师和反复试验，最终都能找到解决方案。这种“发现问题-分析问题-解决问题”的思维模式，将使我受益终生。看着自己设计的播放器成功演奏出熟悉的旋律，所有的努力都化作了满满的成就感。

这次课程设计就像一把钥匙，为我打开了数字系统设计的大门。它不仅让我掌握了专业技能，更重要的是培养了我对工程实践的热情。我相信，这些收获将会成为我后续学习道路上的宝贵财富。

6 参考文献

[1]郝建峰,任国凤,李洋,等.基于FPGA的智能交通信号灯控制系统设计[J].廊坊师范学院学报(自然科学版),2022,22(02):31-34.

[2] 陈春先,曲鸣飞,张丽,等.基于 FPGA 的校内十字路口红绿灯控制系统设计实现[J].中国新通信,2021,23(01):42-43.

[3] 周梦婷,秦新景,徐松海,等.基于 FPGA 的交通信号控制器设计[J].电子制作,2019,(21):13-14+20.DOI:10.16589/j.cnki.cn11-3571/tn.2019.21.004.

[4] 周梦婷,秦新景,徐松海,等.基于 FPGA 的交通信号控制器设计[J].电子制作,2019,(21):13-14+20.DOI:10.16589/j.cnki.cn11-3571/tn.2019.21.004.

[5] 何梓欣.基于FPGA和深度学习的智能交通灯系统的设计和实现[J].中国集体经济,2018,(25):69-70.

[6] 杜瑞雪.基于 FPGA 的智能交通灯控制系统的设计与实现[D].云南大学,2018.