

基于FPGA的智能交通灯控制系统设计

赵磊

成都信息工程大学工程实践中心, 四川 成都
Email: zhaolei@cuit.edu.cn

收稿日期: 2021年6月1日; 录用日期: 2021年7月1日; 发布日期: 2021年7月9日

摘要

一直以来, 传统的红绿灯是采用固定时长方式进行路面交通控制, 是成熟有效的系统, 具有通行效率不高的缺点。为优化路口的车辆通行, 本文设计了一种自动感知的交通光控制器系统(ATLC), 对道路交通流进行了研究。该方法利用通行高峰和低谷不同数据, 可以有效地减少非高峰时段的等待时间, 包括紧急情况VIP车辆通行以及行人的安全也都可考虑在内。本文设计的低成本自动交通灯控制系统, 为地面通行提供了稳定的交通流量。该系统利用VHDL语言进行开发, 使用时序仿真进行测试。该系统能够检测不同时刻车辆通过率, 调节相对应的交通灯时间长度随之变化, 实现对红绿灯的智能调节。利用此智能调节控制路面交通, 可使主支干道协调配合, 从而提高路口的通行效率。

关键词

FPGA, 智能交通控制系统, VHDL硬件语言

Design of Intelligent Traffic Light Control System Based on FPGA

Lei Zhao

Electronic Experimental Center, Chengdu University of Information Technology, Chengdu Sichuan
Email: zhaolei@cuit.edu.cn

Received: Jun. 1st, 2021; accepted: Jul. 1st, 2021; published: Jul. 9th, 2021

Abstract

For all the times, the traditional method of fixed length of traffic light for road traffic control has long been a proven and effective system, but it has the disadvantage of low traffic efficiency. In

order to optimize traffic at intersections, a traffic light controller (ATLC) system based on flow sensing is studied, and the traffic flow is discussed. This method takes advantage of peak and slump trough traffic slots to effectively reduce waiting times during off-peak hours, including emergencies and VIP vehicle passes and pedestrian safety. The low-cost automatic traffic light control system designed in this paper provides a stable traffic flow for ground traffic. The system is developed with VHDL and tested by time series simulation. The system can realize the intelligent regulation of the traffic light, detect the arrival rate of vehicles at different times, so that the corresponding length of the traffic light configuration time changes accordingly, so as to intelligently regulate and control the road traffic, so that the main and branch roads coordinate, thus improving the efficiency of the intersection.

Keywords

FPGA, Intelligent Traffic Control System, VHDL Language

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,智慧城市的发展对与城市的可持续发展至关重要[1]。当今城市面临的主要问题是交通拥堵,交通堵塞造成交通成本的上升,并且影响人们的日常生活[2]。交通拥堵是世界上许多城市面临的严重问题。在全球最主要、繁忙的城市中,交通拥堵已经引起了研究人员的重视。交通拥堵直接影响了出行成本、人力成本和市场成本,间接抬高了商品价格[3]。为了解决这些交通问题,我们必须建立新的便利设施和基础设施,但同时设法让这些设施智能化,这将为缓解交通提供新的途径[4]。

在十字路口通常会设置交通灯或交通信号,以便控制交通[5]。这是一种安装在十字路口的电子系统,以预先定义的颜色系统(通常是红、黄、绿)来控制车、人的通行[6]。交通灯控制器 TLC (TLC)可用 ASIC、FPGA 和微控制器系统实现[7]-[12]。FPGA 相对于微控制器,在 I/O 端口的数量、处理速度和性能等方面具有优势。所有这些优势在 TLC 设计中都是非常关键的[13]。在 TLC 的设计中,成本也是一个非常重要的问题。FPGA 的使用可以降低成本[14] [15]。

本文介绍基于的 FPGA 的交通灯控制器设计,采用 VHDL 语言进行编写,并且结合 Quartus II 软件进行系统设计与调试[16]。FPGA 获取当前通过十字路口的车辆的信号,并根据这些信号发送下一步命令。在特定道路上,应设置特别的交通信号灯。另外, FPGA 还需要考虑时间[17] [18] [19] [20]。在本文中,我们可以在每个方向上以一定距离放置三个传感器,我们可以假设第一个传感器流量较低,第二个传感器给出中等流量,第三个给出高流量,根据传感器的状态,我们可以在特定方向上分配不同颜色交通灯的指示时间。在本文中,我们可以假设四种道路交通中的传感器位置,实现各种不同情况,开发这些情况下电路的 VHDL 代码。

2. 设计原理

2.1. 交通灯控制规则

FPGA 的设计流程见如下流程图。图 1 为 FPGA 设计流程的简化版本。

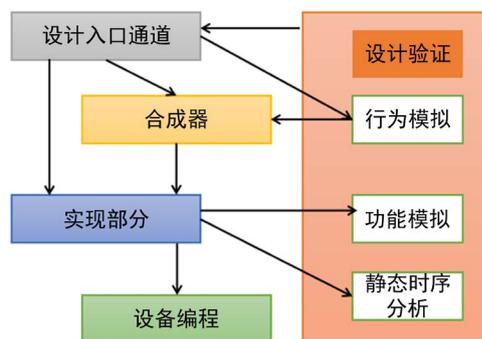


Figure 1. Schematic diagram of FPGA-based design flow
图 1. 基于 FPGA 的设计流程图

交通灯控制器可以从一些任意的假设状态开始设计。比如首先假设向北方向的车辆被允许通行，然后在东、南、西方向的车辆被允许依次移动。编写交通灯控制器程序的优点是，在一个程序中，可以很容易地根据需要进行修改，即假设主干道上的通行时间更多，支路通行时间更少；时钟划分为主要道路时钟时间更多，支路时钟时间更少，这是因为主要道路交通比支路交通繁忙。一般来说，TLC 系统在每个方向都有三个灯(红色、绿色、黄色)，红灯表示停止通行，绿灯表示允许通行，黄色灯表示将在几秒钟内停止通行。图 2 表示交通灯系统控制规则。

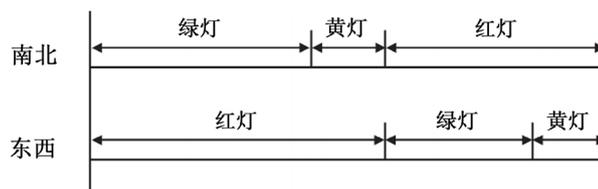


Figure 2. Traffic light system control rules
图 2. 交通灯系统控制规则

2.2. 设计方法及实现

图 3 为交叉口处的交通信号示意图。在该结构中，有四个方向的交通信号，北、南、西、东分别由 R1、R2、R3 和 R4 表示。假设南北方向为主干道，东西方向为支干道。

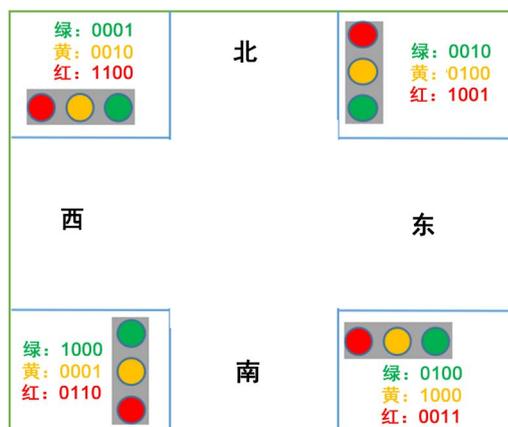


Figure 3. Schematic diagram of traffic signals at intersections
图 3. 交叉路口的交通信号示意

首先,所有信号控制器处于置 0 模式,其中道路信号(R1)为绿色,其他道路 R2、R3 和 R4 均为红色。我们已将这个状态指定为 S0。之后,控制器发送控制信号,使状态变为 S1,其中 R1 为黄色,而其他信号仍然为红色。在此状态下,控制器 X2 检查 R2 处的传感器信号是否为低电平即 R2 处是否有车辆。如果传感器给出该道路上没有车辆的信号,则信号将控制状态使其转移到 S4,表示道路 R3 上的信号变化,而其余信号显示为红色。另一方面,如果道路 R2 上存在车辆,则状态转换为 S2,此时道路 R2 上的信号切换为绿色,其余的信号只有在状态 S2 显示绿色信号后,道路 R2 上的信号从绿色变为黄色,而所有其他信号继续处于红色状态。当控制器处于 S3 状态时,它检查在道路 R3 上的传感器 X3 的响应。如果传感器的输出为低电平,系统将转移到状态 S6,否则转到相应的下一个状态 S4。另一方面,在 S4 时,当道路 R3 的交通信号变为绿色时,道路 R1、R2 和 R4 的信号本身仍保持红色,这时将转移到状态 S5。当控制器的状态为 S5 时,它检查道路 R4 上的传感器 X4 的输出。根据 X4 的输出,进一步的状态也会发生相应的变化。如果传感器输出值为低,则控制被转移到状态 S0。当控制器处于 S5 状态时,道路 R3 上的信号从绿色变为黄色。当控制状态为 S6 时,道路 R4 的信号变成绿色,而所有信号只保持红色信号。然后转移到状态 S0。在状态 S7 中,道路 R4 的信号从绿色变成黄色。同时,检查作为 X1 的第一道路 R1 上的传感器的输出。如果 X1 输出信号为低,则转移到状态 S2,否则转移到初始状态 S0。以上这些控制信号的状态并不是固定的,用户可以指定状态数、交通灯的顺序和交通灯延迟时间,这是设计的优势之一。

2.3. TLC 状态转换

TLC 系统的状态转换主要是在模式控制信号 mode 的作用下工作。控制信号有四种可能值,包括 2 个峰值模式和非峰值模式下的有限状态机(FSM),以及针对紧急模式提出的 2 种状态。峰值模式有 5 个输入和 20 个输出,同时有 12 个状态和 48 个转换方向。同样地,非峰值模式有 11 个输入和 20 个输出,12 个状态和 149 个转换方向。该信息将在合成过程中生成。状态转换从 mode 取不同值开始。状态图的右侧为峰值模式 FSM,在高峰时段将执行该 FSM。这个状态将一直持续到 mode 信号改变。在这个 FSM 中,每个绿灯状态之间有两个中间状态,即黄灯和红灯状态。红灯状态在黄灯之后执行。当紧急情况发生时,将通过更改 mode 值使进入紧急模式。

3. 结果与分析

3.1. RTL 实现及验证

本项目划分为四大模块:1) 控制模块;2) 计时器模块;3) 时钟分频模块;4) 交通灯显示驱动模块。

控制模块:该模块是整个交通灯芯片的 CPU。可控制 UART 通信模块,可读取寄存器,判断工作模式,并相应的改变交通灯状态和通行时间。输出控制信号驱动显示模块进行显示。

计时器模块:计时器模块的基础为计数器,计数器可以对脉冲的个数进行计数,经过时钟分频模块将系统时钟的 50 Mhz 的工作频率分频为计时器模块的工作频率 1 Mhz 后,对每一次的脉冲进行计数,控制模块将发送每个模式所需要计数的时间给计时器模块,当计时器倒数归零则促使控制模块进入下一个模式,而计时器所计数的时间将发送给数码管进行显示。

时钟分频模块:在 FPGA 中一般以晶振作为时钟源,接入到分频器当中。晶振所产生的时钟信号是周期固定的,在需要其他周期时就需对现已有的时钟信号进行分频,分频是将信号的周期变大,频率变小,以满足设计者所需时钟信号周期。为了与智能交通控制系统中的信号周期相匹配,需要将系统频率转换为周期以秒计时的单位。本系统中我们将时钟频率固定为 1 MHz,分频模块能够将 1 MHz 的时钟信号分频为 1 Hz,来驱动后续模块进行工作。

数码管扫描模块：数码管扫描模块将接收由控制模块发送的各个模式需要显示在数码管上的数据，如不同颜色灯的计时时间，并会按照计时器模块上的数值采用动态扫描的方式点亮相应的数码管。

3.2. FPGA 实验结果

根据仿真实验结果，当到达率 $\lambda = 0.03$ 时，对应的平均通行时间即智能交通控制系统中的直行绿灯时长为 18 s，当到达率 $\lambda = 0.02$ 时，对应左转绿灯时长为 10 s，则红灯时长为 34 s；当到达率 $\lambda = 0.037$ 时，直行绿灯时长为 30 s，当到达率 $\lambda = 0.02$ 时，对应左转绿灯时长为 12 s，红灯时长为 50 s。start 为触发信号，当 start 为低电平时，系统为初始清零状态，即所有方向交通灯均为红灯，当 start 为高电平时，系统进入正常信号转换状态，完成主支干道的信号灯指示任务。main_straight_green_light 是主干道由到达率所对应的平均等待时间，即依据交通规则计算出的绿灯时间长度，main_straight_green_light_count 则是由所接收到的数据完成计时工作，在不同的到达率下，绿灯时间数据会根据到达率的改变而智能改变；access_straight_green_light 是支干道直行绿灯时间，同样是根据支干道所获得的平均等待时间即直行绿灯的通行时间长度；其他相位的交通信号灯同样依据此原理。1 表示为左转绿灯，2 表示为直行绿灯，4 表示为黄灯，8 表示为红灯，A 为南北向，B 为东西向。由仿真实验可知：南北向的红灯时长为东西向的直行绿灯、左转绿灯以及黄灯的时长之和，交通灯在完成信号转换功能的同时，同样需要完成相应信号灯的时间倒计时功能，东西方向亦是。在南北向为绿灯状态时，东西向为红灯状态，南北向的直行绿灯倒计时结束后，信号灯转换为黄灯，黄灯倒计时完毕后，左转绿灯的信号开始完成倒计时功能，在左转绿灯倒计时结束后，信号灯转换为黄灯，当作为缓冲的黄灯倒计时结束后，信号灯转换为红灯，直到红灯倒计时结束，南北向一个周期的交通灯指示完成，东西向亦是如此。

3.3. 静态时序分析和综合分析

静态时序分析：静态时序分析(Static Timing Analysis)是一种验证方法。它采用穷举分析方法，提取电路时序路径并计算延迟，检查信号的建立时间和保持时间是否满足设计要求。通过对最大路径和最小路径延迟的分析找出违背时序约束的错误。静态时序分析常用软件是 Synopsys 公司的 PrimeTime。

物理综合：完成了逻辑综合后，相当于完成了前端设计的工作，并生成了具有逻辑功能的门级网表。但目前为止我们所做的都是基于逻辑和代码的设计，并未涉及到物理层面的设计，此时对芯片性能的分析也只是基于算法和计算机模拟的结果，并没有实际的物理意义。因此为了产生真正符合工艺要求，可以进行流片生产的设计文件，我们需要进行后端验证的工作。第一步就是使用物理综合工具，根据门级网标文件和约束，并利用工艺厂商提供的物理库，进行布局、时钟树综合和布线。这种方法是使用比较接近真实的物理信息计算时间参数，综合后的电路接近最优，综合报告能比较真实地反映电路的状态。一般采用的物理综合工具为 Synopsys 公司的 IC Compiler。

3.4. DRC 设计规则检查

满足设计规则的芯片才能保证版图满足厂家流片的要求。设计规则是版图中个图形尺寸的规范。一般设计规则是以器件的特征尺寸为基准，根据制造工艺水平及其他考虑，制定出一套各层掩膜板上图形尺寸的许可范围。设计中采用 ICC 中的 DRC 检查工具 Hercules 进行检查。

4. 结论与讨论

现代的多路交通管理方式在很大程度上改善了交通状况。先进的信号控制器有助于改善城市交通，这与控制器的复杂度成正比。使用状态控制系统可以很好地处理这些复杂的控制器。减少状态控制器有助于减少所需的硬件系统，从而实现低功耗和高效的设计系统。基于 FPGA 的智能交通系统未来目标是

可以通过使用更多智能的电路系统来直接实时应用。这项工作可以进一步扩展到其他 FPGA 系列, 以实现速度升级, 并使用 *arduino* 处理器和 *raspberrypi* 来改善诸如速度之类的参数, 以实现更好的性能。本文依照道路交通情况, 设计出基于 FPGA 的智能交通灯控制系统的结构框架, 并分别完成各个模块的结构设计。将其运用到智能交通控制系统中, 有很高的实用性及高效性。

本设计采用 VHDL 硬件语言描述, 使用分模块设计的方法实现了交通灯交替开启和倒计时时间的显示系统, 以及车辆和行人安全通过的命令实现。程序数据可根据实际情况进行灵活设置。将来我们会完成后续工作, 比如物理综合。综合时可以兼顾电路面积的情况, 可以调整电路结构和布局, 使之在时间和面积之间寻求最大限度的折中, 达到最优点。

参考文献

- [1] Su, K., Li, J. and Fu, H. (2011) Smart City and the Applications. 2011 *International Conference on Electronics, Communications and Control (ICECC)*, Ningbo, 9-11 September 2011, 1028-1031. <https://doi.org/10.1109/ICECC.2011.6066743>
- [2] Nagel, K., Wagner, P. and Woesler, R. (2003) Still Flowing: Approaches to Traffic Flow and Traffic Jam Modeling. *Operations Research*, **51**, 681-710. <https://doi.org/10.1287/opre.51.5.681.16755>
- [3] Sugiyama, Y., Fukui, M., Kikuchi, M., et al. (2008) Traffic Jams without Bottlenecks—Experimental Evidence for the Physical Mechanism of the Formation of a Jam. *New Journal of Physics*, **10**, Article ID: 033001. <https://doi.org/10.1088/1367-2630/10/3/033001>
- [4] de Souza, A.M. and Villas, L.A. (2015) A New Solution Based on Inter-Vehicle Communication to Reduce Traffic Jam in Highway Environment. *IEEE Latin America Transactions*, **13**, 721-726. <https://doi.org/10.1109/TLA.2015.7069097>
- [5] Lindner, F., Kressel, U. and Kaelberer, S. (2004) Robust Recognition of Traffic Signals. *IEEE Intelligent Vehicles Symposium*, Parma, 14-17 June 2004, 49-53.
- [6] Bazzan, A.L.C., De Oliveira, D. and da Silva, B.C. (2010) Learning in Groups of Traffic Signals. *Engineering Applications of Artificial Intelligence*, **23**, 560-568. <https://doi.org/10.1016/j.engappai.2009.11.009>
- [7] Lahade, S.V. and Hirekhan, S.R. (2015) Intelligent and Adaptive Traffic Light Controller (IA-TLC) Using FPGA. 2015 *International Conference on Industrial Instrumentation and Control (ICIC) IEEE*, Pune, 28-30 May 2015, 618-623. <https://doi.org/10.1109/IIC.2015.7150816>
- [8] Nath, S., Pal, C., Sau, S., et al. (2012) Design of an FPGA Based Intelligence Traffic Light Controller with VHDL. 2012 *International Conference on Radar, Communication and Computing (ICRCC) IEEE*, Tiruvannamalai, 21-22 December 2012, 92-97. <https://doi.org/10.1109/ICRCC.2012.6450554>
- [9] Dilip, B., Alekhya, Y. and Bharathi, P.D. (2012) FPGA Implementation of an Advanced Traffic Light Controller Using Verilog HDL. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, **1**, 1.
- [10] El-Medany, W.M. and Hussain, M.R. (2007) FPGA-Based Advanced Real Traffic Light Controller System Design. 2007 *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Dortmund, 6-8 September 2007, 100-105. <https://doi.org/10.1109/IDAACS.2007.4488383>
- [11] Kishore, S.V., Sreeja, V., Gupta, V., et al. (2017) FPGA Based Traffic Light Controller. 2017 *International Conference on Trends in Electronics and Informatics (ICEI)*, Tirunelveli, 11-12 May 2017, 469-475. <https://doi.org/10.1109/ICOEI.2017.8300971>
- [12] Singh, S. and Badwaik, S.C. (2011) Design and Implementation of FPGA-Based Adaptive Dynamic Traffic Light Controller. 2011 *International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Udaipur, 22-24 April 2011, 324-330. <https://doi.org/10.1109/ETNCC.2011.6255914>
- [13] Kaur, S., Varshitha, V., Siddharth, J., et al. (2018) Adaptive Traffic Light Controller Using FPGA. 2018 *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 18-19 May 2018, 1612-1617. <https://doi.org/10.1109/RTEICT42901.2018.9012263>
- [14] Swapna, B., Javeed, M. and Ramesh, G. (2015) FPGA-Based Advanced Traffic Light Controller Simulation.
- [15] Sachdeva, S., Chowdhury, S., Shekhar, S., et al. (2018) Design and Implementation of a Green Traffic Light Controller on FPGA Using VHDL. In: *System and Architecture*, Springer, Singapore, 187-192. https://doi.org/10.1007/978-981-10-8533-8_18
- [16] Palnitkar, S. (2003) Verilog HDL: A Guide to Digital Design and Synthesis. Prentice Hall Professional, Hoboken.

-
- [17] Shokrolah-Shirazi, M. and Miremadi, S.G. (2008) FPGA-Based Fault Injection into Synthesizable Verilog HDL Models. 2008 *Second International Conference on Secure System Integration and Reliability Improvement*, Yokohama, 14-17 July 2008, 143-149. <https://doi.org/10.1109/SSIRI.2008.47>
- [18] Jamieson, P., Kent, K.B., Gharibian, F., *et al.* (2010) Odin II—An Open-Source Verilog HDL Synthesis Tool for Cad Research. 2010 *18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, Charlotte, 2-4 May 2010, 149-156. <https://doi.org/10.1109/FCCM.2010.31>
- [19] Takamaeda-Yamazaki, S. (2015) Pyverilog: A Python-Based Hardware Design Processing Toolkit for Verilog HDL. In: *International Symposium on Applied Reconfigurable Computing*, Springer, Cham, 451-460. https://doi.org/10.1007/978-3-319-16214-0_42
- [20] Anitha, K. and Sujatha, B.K. (2014) FPGA Implementation of High Throughput Digital QPSK Modulator Using Verilog HDL. *International Journal of Advanced Computer Research*, **4**, 217.