文章编号:1671-4598(2017)08-0139-04 DOI:10.16526/j.cnki.11—4762/tp.2017.08.036 中图分类号:TP368 文献标识码:A

# 基于三级队列缓存的 FlexRay 网络监控平台

刘 彪,白卫伟,钟 韦
(北京交通大学 电气工程学院,北京 100044)

**摘要**:FlexRay 以其实时性、灵活性、高带宽的优势,被称为“下一代车载网络”;然而其协议相对复杂,网络组网调试过程较困难;目前的辅助调试工具如 CANoe 不能进一步分析网络性能,在应用上具有局限性;针对上述问题,提出了一种同时对多个数据采集与存储的解决方案,并以 FPGA 为核心设计了 FlexRay 网络监控平台,利用 SPI 和 USB 总线设计了通信协议,在数据缓存方面设计了三级队列缓存方案;该平台实现了通信数据监测功能,可以获取结点 CPU 中寄存器状态、记录数据帧到达时间,进而分析网络运行及故障状态,同时解决了大量数据缓存溢出的问题;经过测试表明:系统能够稳定运行,适用于快速、优化地设计 FlexRay 网络。

**关键词**:FlexRay;测试平台;FPGA;三级缓存

# FlexRay Network Monitoring Platform Based on Tri—stage Queue Buffer

Liu Biao, Bai Weiwei, Zhong Wei
(College of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract**: FlexRay is known as " the next generation of vehicle network" with the advantages of real—time, flexibility and high bandwidth. However, the complex protocol makes it difficult to build and debug the node network. Currently debugging tools can not be used to further analyze the network performance, which is the limitation in application, such as CANoe. In order to solve the problems above, a solution for multiple datas acquisition and storage at the same time is presented, and the FlexRay network monitoring platform is designed with FPGA as the core, where the protocol is based on the SPI and USB bus and tri—stage queue buffer is used for data cache. It realizes the monitoring function of the communication data, obtains the register state of the node CPU, records the arrival time of the frame, and then analyzes the network operation and fault state, and solves the problem of massive data buffer overflow. The result shows that the system can run stably and is suitable for designing FlexRay network quickly and optimally.

**Keywords**: FlexRay; test platform; FPGA; tri—stage buffer

## 0 引言

FlexRay 具有 10 MHz 带宽、双通道冗余、消息发送时间的确定性等优点[1-4],更适用于日益复杂的车载主干网络以及对实时性要求严苛的线控系统[5]。然而 FlexRay 协议较复杂,需要完成四类参数——全局参数、节点参数、物理层参数与辅助参数共计超过 70 个参数的配置[6],同时由于启动和同步过程的特殊性,在开发初期的结点组网调试过程中很不方便。

目前国外的一些商业化的辅助调试工具,比如 Vector 公司的 CANoe.FlexRay 环境及配套采集结点 VN7600,可以协助完成组网,但作为集成产品只能获取通信数据,不能进行进一步的网络性能分析,无法获取网络运行状态参数,在通信故障情况下无法获得运行参数,也无法解决 FlexRay 网络数据大量缓存的问题,在应用上还有一定的局限性。

本文提出了一种同时采集多个数据采集与存储的解决方案,并基于此方案设计了 FlexRay 网络监控平台,此平台除了可以监测通信数据以外,还可以获取网络运行及故障状态并且解决了大量数据缓存溢出的问题,能够帮助工程师更快速、更优化地设计 FlexRay 网络,同时成本相对于 CANoe 大大降低。

**收稿日期**:2017-02-09; **修回日期**:2017-02-28。

**基金项目**:国家自然科学基金(61401017);中央高校基本科研业务费(2014JBM112)。

**作者简介**:刘 彪(1982-),男,河北保定人,副教授,主要从事智能检测技术和现场总线与控制网络方向的研究。

## 1 Flexray 总线技术

FlexRay 是由 BWM 和 Daimler Chrysler 等公司形成的 FlexRay 联盟(FlexRay Consortium)为了满足未来汽车的通信需求而共同开发的车载网络协议。FlexRay 是一个高速、容错的车辆总线系统,核心概念是时间触发的网络通信协议,更加关注汽车行业的一些核心需求,比如数据传输率,数据通信的灵活性,更全面的拓扑选择等。FlexRay 电气物理层为一个发送与一个或多个接收通信模块提供了一种差分电压连接(即总线),差分电压通过测量两个表示为 BP(总线正)和 BM(总线负)的信号线得到[7]。相较传统的车用总线,FlexRay 在很多方面都有突出的表现,其优点如下:

1)传输速率高:FlexRay 有两条分离的通信信道,相比于 CAN 总线 1 Mbps 的传输速率,FlexRay 信道的数据最高传输速率为 10 Mbps,双通道运行时,数据的传输速率可以达到 20 Mbps。

2)预知消息的到达时间:根据 FlexRay 总线的通信机制,消息是在不断循环的周期中进行发送的。各类消息在通信周期中所占的位置已事先设定,所以各类消息的接收节点可以提前知道消息到达的时间。

3)可靠性高:FlexRay 有通信冗余能力,采用分布的时钟同步机制,实时纠正相位合速率偏差,提供了多个等级的容错功能;并且可以采用双通道冗余通信。

4)灵活性强:FlexRay 可以实现同步和异步的数据传输,也可进行网络进度的检测;FlexRay 可以实现总线、星型和混合拓扑网络结构的灵活配置,通过采用一种或结合两种或两种

以上的拓扑结构来进行分布式系统的配置；FlexRay 媒体访问机制灵活，静态段采用基于时间触发的媒体访问机制，保证了数据通信的可预测性和时钟同步，动态段采用基于事件触发的媒体访问机制，更有利于不同带宽消息的传输，可以有效提高带宽利用率。

## 2 总体方案设计

依据 FlexRay 总线的特点，该监控平台需要实现如下功能：能够接收多路 FlexRay 结点中 CPU 的数据帧；能够查看 CPU 的寄存器；能够接收错误类消息；能够获取数据帧到达时间，数据收集器能够将数据发送至上位机进行存储、分析。

本监控平台的总体方案包括由 3 个 FlexRay 结点组成的通信网络；基于 FPGA 进行搭建的用于对通信网络进行数据采集的数据收集器以及上位机。3 个 FlexRay 节点之间通过 FlexRay 总线进行数据通信，FlexRay 节点和 FPGA 通过 SPI 总线连接，FPGA 和上位机之间通过 USB 总线进行数据交互[8]。该监控平台采用的总体方案如图 1 所示。

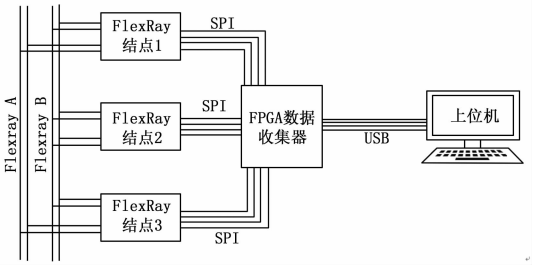


图 1 FlexRay 网络监控平台总体设计方案

该监控平台采用的 FPGA 是基于 Altera 公司的 EP4CE10F17C8 芯片。USB 接口是基于 Cypress 的 CY7C68031A 芯片，该芯片支持高速 USB2.0 协议。该监控平台需要为 FlexRay 结点中的 CPU 嵌入一段数据打包及一段用于 SPI 数据通信的程序，为了提高兼容性，采用通用 IO 模拟 SPI 接口。

## 3 平台设计

### 3.1 多级总线协议设计

如图 2 所示，该平台中涉及到 FlexRay 结点 CPU、FPGA (Field－Programmable Gate Arrary 现场可编程器件)[9] 和上位机的数据交互以及 SPI 总线和 USB 总线通信。数据通信协议设计为 4 层，如图 2 所示。SPI 字节流用于完成 CPU 和 FPGA 之间一个字节的发送，SPI 帧用于 CPU 和 FPGA 之间一帧数据的传输，USB 帧用于 FPGA 和上位机之间的数据传输。其中单片机到电脑的数据传输过程为，单片机将应用层数据打包成 SPI 帧，SPI 帧之后被拆解成单个的字节流发送至 FPGA，FPGA 将字节流组成 SPI 帧，之后 FPGA 将 SPI 帧打包成为 USB 帧发送至 PC，FPGA 和 PC 之间的底层通信由 Cypress 的 USB 控制器完成。最后 PC 应用层拆解 USB 帧，再拆解 SPI 帧即可得到 FlexRay 结点 CPU 的应用数据。PC 和 FPGA 之间的通信方式为 PC 上位机将数据打包成 USB 帧发送至 FPGA，FPGA 拆解 USB 帧即可得到数据。

SPI 帧分为 3 种：寄存器数据观测格式、不带时间戳和带时间戳的 FlexRay 通信数据观测格式，如表 1、表 2、表 3 所示。

表 1 寄存器数据观测格式

| 0x01 类别 (1 字节)： | 节点表示 (1 字节)： | 寄存器编码 (2 字节)： | 寄存器数据 (2 字节)： |
|---|---|---|---|

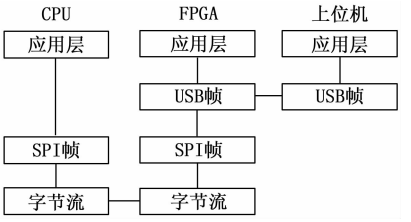


图 2 通信协议架构图

表 2 不带时间戳的 FlexRay 通信数据观测格式

| 0x02 类别 (1 字节)： | 节点表示 (1 字节)： | 通信周期 (2 字节)： | 数据长度 (2 字节)： |
|---|---|---|---|
| 时槽编号 (1 字节)： | 时槽长度 (1 字节)： | 时槽数据 (不固定)： | 重复长度与数据 |

表 3 带时间戳的 FlexRay 通信数据观测格式

| 0x03 类别 (1 字节)： | 节点表示 (1 字节)： | 通信周期 (2 字节)： | 数据长度 (2 字节)： |
|---|---|---|---|
| 时槽编号 (1 字节)： | 时槽长度 (1 字节)： | 时槽数据 (不固定)： | 时间戳 (8 字节)： |

带时间戳的 SPI 帧记录数据到达 FPGA 的时间，可以一次发送多帧 FlexRay 帧数据，不带时间戳的 SPI 帧不记录帧到达 FPGA 的时间，只能一次发送一条 FlexRay 帧数据。

在 FlexRay 结点 CPU 中嵌入 SPI 帧发送程序，首先按照上述表中数据格式生成 SPI 帧，采用了队列形式实现多帧的存储，包含队列创建、1 帧数据压入队列、3 种帧格式的数据结构创建和将 1 个队列中存储的 1 帧数据发送。

USB 帧数据用于 FPGA 和上位机之间的通信，本文设计 3 中 USB 帧格式，如表 4～6 所示。

表 4 数据内容格式

| 0x01 类别 (1 字节)： | 节点表示 (1 字节)： | 数据长度 (2 字节)： | SPI 帧(长度不固定)： |
|---|---|---|---|

表 5 控制命令格式

| 0x02 类别 (1 字节)： | 节点表示 (1 字节)： | 命令表示 (2 字节)： |
|---|---|---|

表 6 错误命令格式

| 0x03 类别 (1 字节)： | 节点表示 (1 字节)： | 错误数据 32 字节()： |
|---|---|---|

### 3.2 基于三级缓存的 FPGA 程序设计

本监控平台采用了 Altera 的 FPGA，其内部集成的 SPI IP 核模块和定时器模块一般是作为主机设计的，本平台需要将 FPGA 作为从机设计，所以 SPI 模块和定时器模块的程序都是通过 Verilog HDL 语言进行描述；另外 Altera 的 FPGA 内部嵌入了 Nios II 处理器，该处理器是用 HDL 语言进行描述之后映射到 FPGA 的通用逻辑单元中。

利用 FPGA 中的 Nios II 处理器完成控制功能。由于需要与多个 FlexRay 结点连接，FPGA 在存储 SPI 帧数据时采用队列形式，本文共设计了三级队列缓存，以 4 个 FlexRay 结点为例的队列缓存如图 3 所示，其中 Queue2 _ 1 表示第 2 号结点的一级缓存。一级缓存用于收集离散的字节，二级缓存用于存储一个 SPI 模块的帧数据，三级缓存用于向 USB 输出数据。

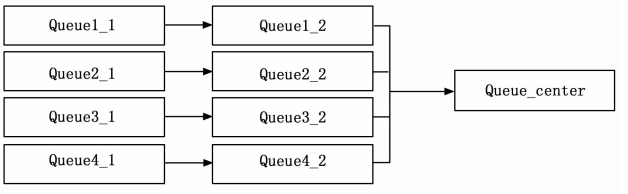


图 3　SPI 帧数据三级缓存

Nios II CPU 使用 Nios II 11.1 Software Build Tools 软件开发，使用 c 语言进行编程。Nios II 程序开发软件开发环境和 JAVA 编程语言的开发环境类似，都是基于 Eclipse IDE 制作的。其重要部分分为初始化程序，SPI 握手程序，CRC 校验程序，SPI 组帧程序，数据管理程序，USB 帧封装程序，队列模块程序。二级缓存数据管理和将三级缓存中的 SPI 帧封装成 USB 帧的程序实现均在 Nios II 软核中实现。

二级缓存数据管理的流程如图 4 所示。

二级缓存向第三级缓存输出时采用依次从每个二级缓存中取一帧数据到第三级缓存中，如果数据帧有错误则将对应的 SPI 模块错误标志位置 1，并且将 SPI 帧转换 USB 帧。错误处理函数会将对应的 SPI 模块的二级缓存清零，直到第三级的缓存等于 400 字节或者第三级缓存加上一帧的数据大于 400 字节。

图 5 是将三级缓存中的 SPI 帧数据封装成 USB 数据帧的流程图。

其中 $q1$ 和 $q2$ 是两个队列，$q1$ 中存储了 *SPI* 帧数据，$q2$ 中存储了 *USB* 帧数据流，$q1$ 是二级缓存，$q2$ 是三级缓存。

当上位机发起 USB 请求时，Nios II 处理器会进入 USB 中断程序，将 USB 帧数据从三级缓存的数据块转成无符号整型数据存入到无符号整型的数据块中。USB 的中断程序的流程如图 6 所示，起始第三级缓存存入数据块即为将字节流转换为 128 个无符号整型的数据块。

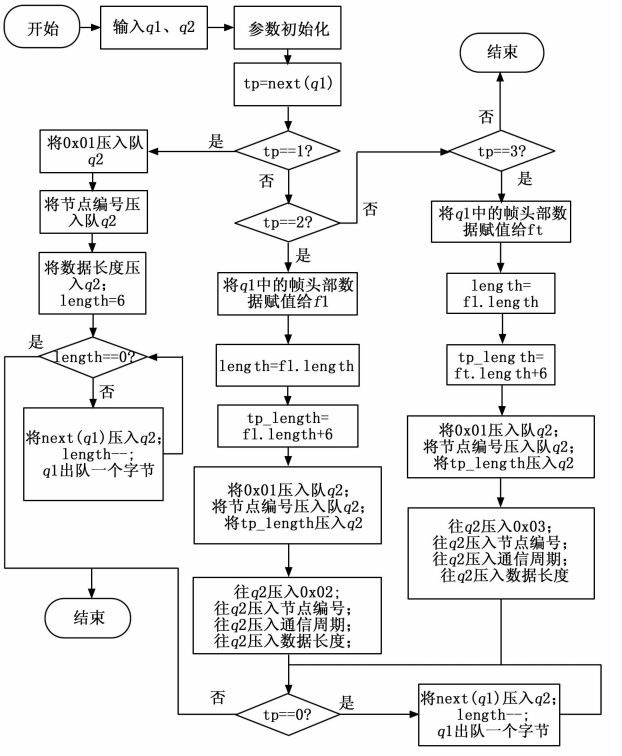


图 5　SPI 帧封装成 USB 帧的流程图

### 3.3　上位机程序设计

该测试平台 PC 端软件的功能有采集 USB 数据、USB 界面交互、对数据进行解码、将数据存入数据库、静态段时槽规划、数据显示以及软件测试等。限于篇幅，此处不再介绍。

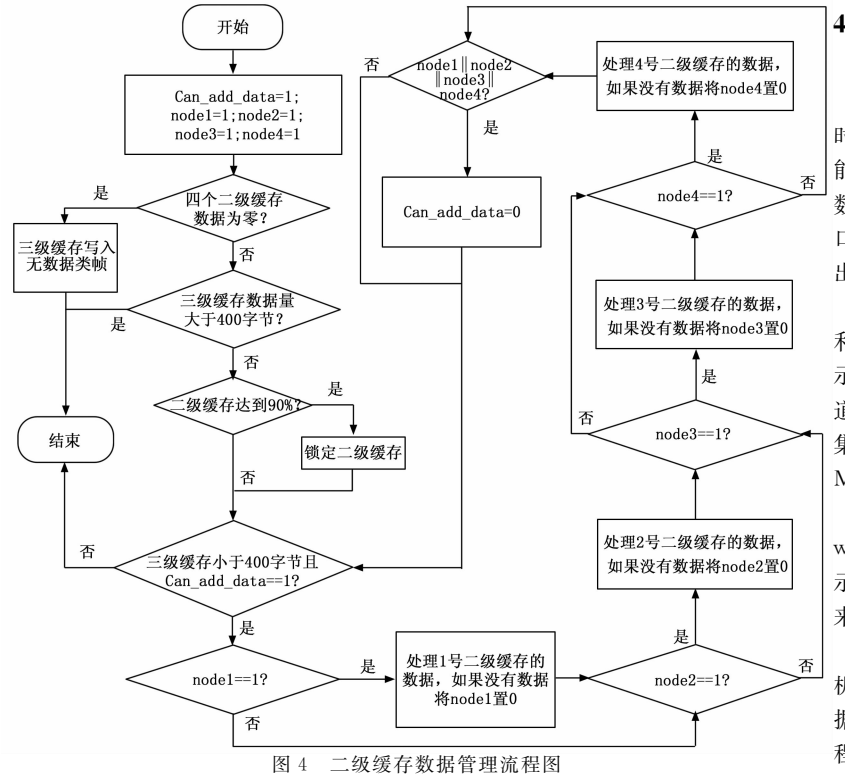


图 4　二级缓存数据管理流程图

## 4　实验与分析

针对一个 3 结点的 FlexRay 网络进行了测试。

首先对 FPGA 中 SPI 利用 Modelsim[10] 进行时序仿真，如图 7 所示。Cs 端口为外部使能功能，Clk 为时钟输入，SI 为数据输入端口，SO 为数据输出端口。Indata 是 16 位内部数据输入端口，InterCs 是内部使能端口，OutEven 是数据输出通知端口，DataOut 是 16bits 数据输出端口。

实际测试中，SPI 时钟频率设定为 500 kHz，利用逻辑分析仪采集到的 SPI 时序信号如图 8 所示。通道 0 中采集的信号为主机 MISO 管脚，通道 1 中采集的信号为主机的 SCK 管脚，通道 2 采集的为主机的片选信号，通道 3 采集的是主机的 MOSI 管脚。

在 FPGA 中的数据需要使用，NIOS II Software Build Tools 软件查看其中的数据，如图 9 所示，队列 FlexRay _ node2 _ queue2 接入接收到了来自节点 1 的数据，该数据为寄存器数据。

将 FPGA 数据采集器通过 USB 接口与上位机进行连接，通信网络中的数据就能通过查询数据界面进行显示，并转存到数据库中，USB 通讯程序查询数据界面如图 10 所示。

从上位机数据库中可以查询 FlexRay 通信数

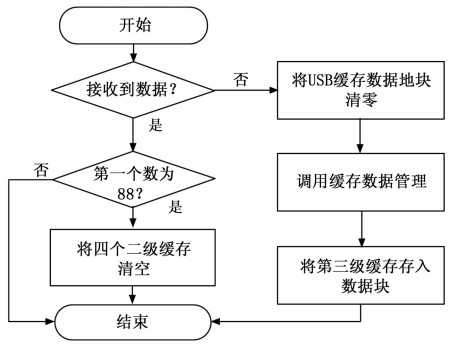


图 6 USB 中断流程图

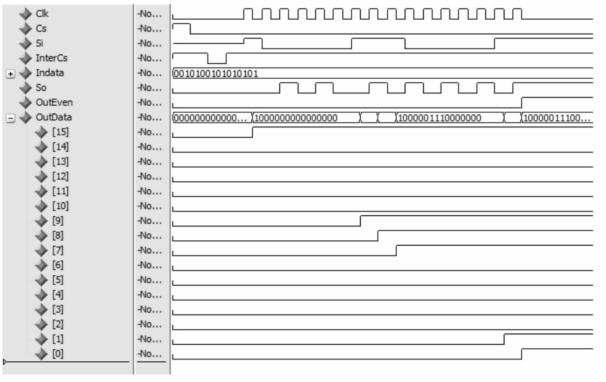


图 7 SPI 模块仿真时序图

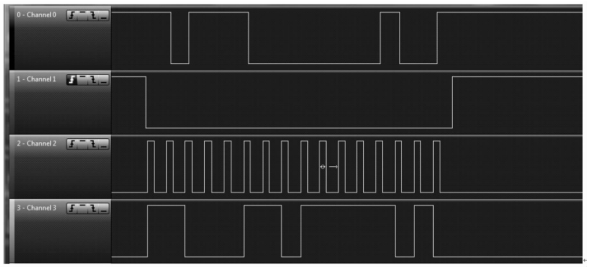


图 8 SPI 时序信号

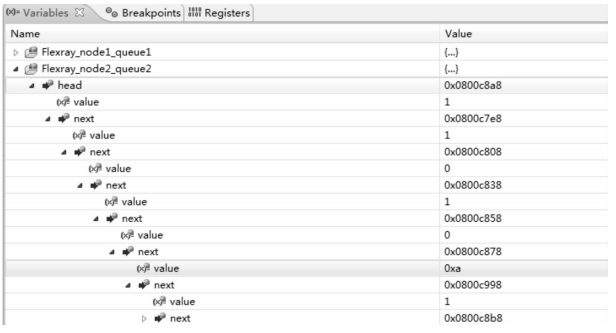


图 9 FPGA 中队列中的数据

据，查询结果如图 11 所示。

图 10 和图 11 表明，测试平台能够正常收集 FlexRay 节点的通信数据，通过对比 FlexRay 网络实际发送数据，没有出现缓存溢出及数据丢失情况。

## 5 结论

本文利用 FPGA 及 NIOS II 开发技术，设计了针对 FlexRay 网络的监控平台，结合 SPI 和 USB 总线设计了四层网

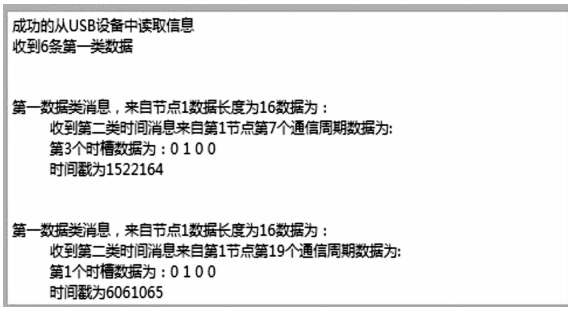


图 10 USB 通讯程序查询数据


图 11 FlexRay 通信数据展示界面

络通信协议，设计了三级缓存来解决大量数据溢出的问题。经过测试，FPGA 中 SPI 模块时序运行正常，可以正确接收到 FlexRay 网络结点 CPU 的寄存器状态以及数据到达的时间，在大量数据发送的情况下也没有出现缓存溢出及数据丢失的情况。本文所设计的监控平台针对 FlexRay 开发初期中组网调试能够提供较大的便利，从而缩短开发周期。

**参考文献：**

[1] 陈 龙. 车载 FlexRay 主干网的构建与性能分析 [D]. 哈尔滨：哈尔滨工业大学，2008.

[2] 包 玥. FlexRay 车载网络的调度与控制协同设计 [D]. 武汉：武汉理工大学，2014.

[3] 孙 雨. FlexRay 网络通信协议在车载系统中的应用与研究 [D]. 长沙：中南大学，2014.

[4] FlexRay Communications System — Protocol Specification，Version 3.0.1 [S]. FlexRay Consortium，Oct. 2010.

[5] Paulitsch，M. Hall，B.. FlexRay in aerospace and safety — sensitive systems [J]. IEEE Aerospace and Electronic Systems Magazine，2008，23 (9)：4 - 13.

[6] 庄江丽. FlexRay 总线的组网及其在混合动力客车上的应用 [D]. 北京：北京交通大学，2012.

[7] 苑利维，李治国，朱晓荣. 一种 FlexRay 总线通讯测试系统设计 [J]. 计算机测量与控制，2016，24 (11)：20 - 23.

[8] 杨建军. CAN 总线技术在汽车中的应用 [J]. 上海汽车，2007 (6)：32 - 34.

[9] 曹 晶. FPGA 在智能交通产品中的应用研究 [J]. 电子技术与软件工程，2016 (6)：139.

[10] 张桂兴，张英敏，张 鹏. 基于 IP 核与 ModelSim 的正弦波发生器仿真平台建立 [J]. 测控技术，2011，30 (1)：28 - 31.