

基于 802.11p 协议的实时图像通信系统的设计与 FPGA 实现^{*}

付广钰, 罗 涛, 覃建策
(北京邮电大学 北京 100876)

摘 要

基于 Altera 公司 CycloneIII EP3C120 硬件开发套件, 设计实现基于 802.11p 及 NiosII 软核的实时图像传输系统。根据项目需求自定义 SOPC 硬件架构, 并在 NiosII 软核中运行的 MicroC/OSII 嵌入式操作系统中设计多任务, 完成图像数据的接收与发送。本设计用 VerilogHDL 语言实现 802.11p 协议, 并给出了 NiosII 软核与 VerilogHDL 硬件模块数据交互的一种实现方案。

关键词 NiosII; SOPC; FPGA; 多任务; FIFO

1 引言

CycloneIII EP3C120 是 Altera 公司推出的一款低功耗、低成本、高性能的 FPGA 开发套件^[1]。SOPC(system on a programmable chip)为可编程片上系统, 是一种可定制的硬件平台, 用户可根据需要添加 CPU、存储器驱动、网卡驱动、输入输出接口等。其中 NiosII 是一款通用的 RISC(reduced instruction set computer)结构的 CPU, 它定位于广泛的嵌入式应用。NiosII 处理器分为 3 个系列: 快速型(NiosII/f)、经济型(NiosII/e)和标准(NiosII/s)内核。考虑到实时图像传输需要较大的处理速度, 本文选择快速型软核来构建系统。

IEEE 802.11p 协议^[2]主要是针对车辆通信网络, 特别是电子道路收费系统、车辆安全服务与车上的商业交易系统等应用设计的一种中长距离通信的空中接口标准, 它能够提供给高速的车到车和车到中心台的数据传输。在 802.11p 协议中, 信道带宽为 10 MHz, 信息传输速率为 3~27 Mbit/s, 工作频率为 5.85~5.925 GHz。目前, 对基于

802.11p 的车辆通信系统的研究才刚起步, 演示系统就更少了。基于此, 本文基于 CycloneIII EP3C120 硬件平台, 主要研究基于 802.11p 协议的车辆实时通信系统的软硬件设计及实现。

2 系统概述

基于 CycloneIII EP3C120 和 802.11p 协议的车辆实时通信系统整体结构如图 1 所示。其中, 服务器完成实时图像采集功能, 并将图像数据经 socket 连接传输给发射机^[3]。在发射机中, 从服务器接收到的图像数据按照 802.11p 协议处理后被转换为模拟信号。

接收机的主要功能是接收发射端的模拟信号, 根据 802.11p 协议解调出图像数据, 并将校验正确的数据经 socket 连接传输给客户端。客户端则完成图像的连续播放, 实现实时视频的效果。

项目分为硬件平台搭建和软件设计两大部分。

3 硬件平台搭建

本项目的硬件平台主要包括 SOPC 平台定制、NiosII 软核中多任务设计、802.11p 硬件编程模块设计三大部分, 其中 NiosII 软核多任务是建立在 SOPC 硬件平台基础上的。

^{*} 国家自然科学基金项目 (No.60971082, 60972073, 60871042, 60872049), 国家“863”计划基金项目 (No.2011AA100706) 和国家科技重大专项基金项目 (No.2010ZX03001-003)

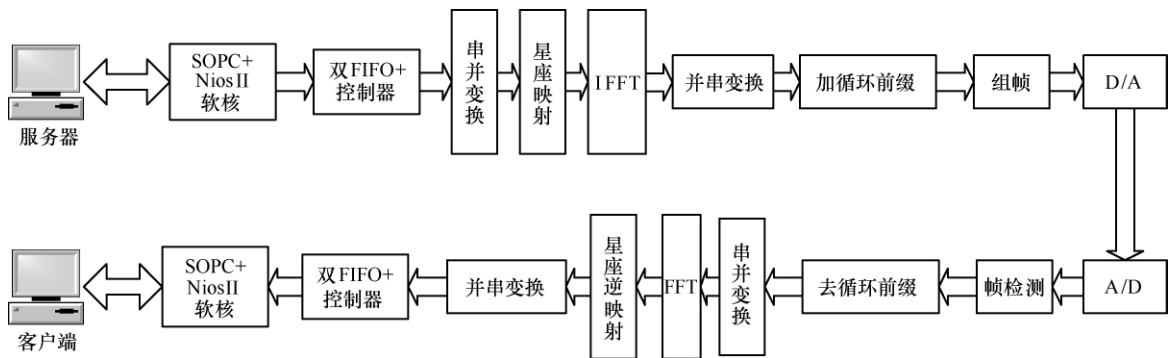


图1 系统整体结构

3.1 SOPC 硬件平台定制

根据项目对硬件的需求,我们自定制了一种硬件系统架构,发送端和接收端均基于此 SOPC 平台架构来建设。SOPC 硬件平台总体架构如图 2 所示。

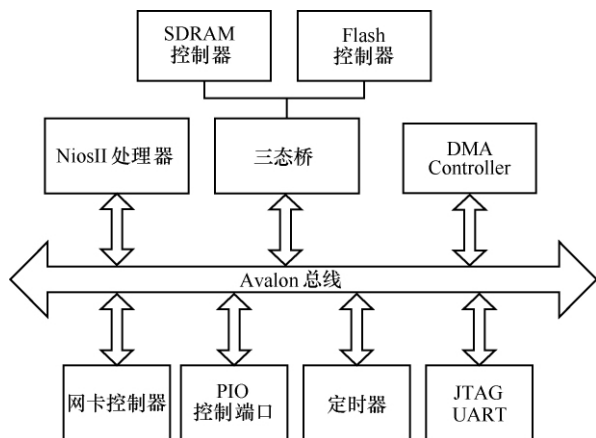


图2 SOPC 硬件架构

在此架构中,NiosII 作为中央处理器,控制所有其他部件的动作,实时多任务嵌入式操作系统 MicroC/OSII 也运行于这个处理器中^[4]。Altera 自定义了一种总线格式——Avalon,它是系统的内部总线,用于 NiosII 和其他部件之间进行数据交互^[5]。

3.2 NiosII 中的多任务设计

基于 SOPC 硬件平台可以在 Altera 公司提供的 NiosII 集成开发环境中创建 c 工程,该工程会提供给用户所有 SOPC 硬件对应的驱动程序,即硬件抽象层(hardware-abstraction-layer,HAL)。NiosII 处理器中可运行 MicroC/OSII 嵌入式实时操作系统^[6]。在此系统之上运行 NicheStackTcp/Ip 协议栈和一些系统初始化任务,进而运行我们设计的多任务。操作系统通过任务调度机制完成复杂的任务切换。

在本项目中,NiosII 软核中的多任务按照所在位置的不同可分为发射机多任务和接收机多任务。这两种多任务调度分别运行在发射机 SOPC 和接收机 SOPC 硬件平台上,且所有任务切换均采用消息邮箱的机制实现。

3.2.1 发射机 NiosII 中的多任务设计

发送机中设计了两个任务。任务一通过 socket 从服务器接收图像数据,校验正确后存入开发板的 RAM 中。任务二读取 RAM 中的图像数据,通过输入输出端口输出到连接后续模块的 FIFO 中。其中,任务二需要依次循环检测两个 FIFO,若第一个 FIFO 为空,则写入图像数据直到满,然后切换任务,等到下次到达任务二时,则检测第二个 FIFO,并将其写满。这样设计可提高数据的传输效率。

3.2.2 接收机 NiosII 中的多任务设计

与发射机类似,接收机 NiosII 中同样开启了两个任务,只是这两个任务的功能稍有不同。接收机任务一检测双 FIFO,并依次从两个 FIFO 中读取图像数据,根据发送端定义的校验格式判断接收数据是否正确,若校验通过则将数据写入 RAM 中,否则对本次 FIFO 数据做丢弃处理。任务二则与目的主机建立 socket 连接,将正确的图像数据发送给客户端。

3.3 IEEE 802.11p 硬件编程模块设计

IEEE 802.11p 硬件编程模块可分为发射机图像数据处理模块和接收机图像数据恢复模块。

3.3.1 IEEE 802.11p 发射机图像数据处理模块设计

使用 VerilogHDL 语言编写实现 FIFO 和 802.11p 协议的物理层基带部分。如图 1 中发射端部分所示,该模块主要分为 FIFO 控制器部分和 802.11p 协议实现两部分。首先,从 FIFO 中按照一定的规则读取数据,即每 64 个数据空 16 个时隙,每 4K 个数据空 500 个时隙,这样便于后面进行 64 点 IFFT、加 CP 以及组帧。双 FIFO 控制器的程序

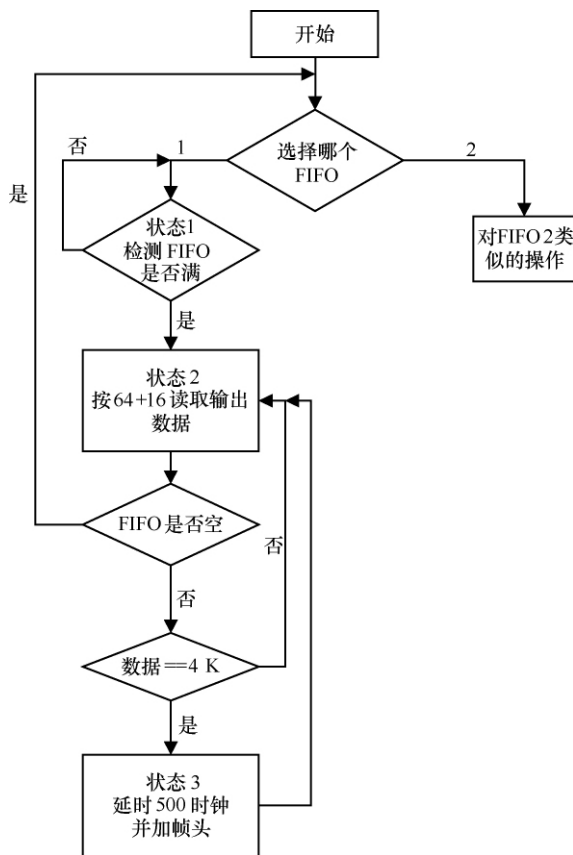


图 3 FIFO 控制程序流程

流程如图 3 所示。802.11p 协议包括串并变换 BPSK 星座映射、64 点 IFFT、加 CP、加帧头等功能模块^[7],具体的实现细节此处不再赘述。

3.3.2 IEEE 802.11p 接收机图像数据恢复模块设计

由于项目主要进行物理层基带部分的设计,发送机和接收机共用同一个采样时钟,故暂未考虑采样定时同步。直接根据训练序列的特性对模数转换后的数字信号进行同步,然后进行去 CP、64 点 FFT 以及星座逆映射,将得到的解调数据依次循环写入两个 FIFO 中,可参见图 1 中接收端部分。

其中,同步包括帧同步、载波同步和符号同步三个阶段。具体的同步仿真结果如图 4 所示。同步算法使用的是短训练序列的延时相关算法。在此算法中,首先对短

训练序列进行延时相关运算。作为判决变量的延时相关值连续超过阈值一定长度后,判定帧头到来;当判决变量低于阈值一定长度时,判定此帧结束。同步算法中,开始的第一个相关值直接影响同步性能。在实际仿真中,经常会出现前面若干相关值不稳定,从而导致出现帧定位错误,丢失整个帧,使得误码率上升。基于此,算法选取中间的 5 个短训练序列的相关值作为判决变量来进行帧同步。载波同步阶段,对载波频率和相位偏移进行估计,提取中间的 5 个短训练序列,计算延时相关累加值,进而求取补偿因子的固定相移角,最后求得每个样点的补偿因子对数据进行补偿。在符号同步阶段,根据 10 个短训练序列自相关算法检测出 10 个数据峰值,来确定数据帧的精确位置^[8]。

4 软件设计

基于§3 的硬件平台,软件设计主要集中在视频采集服务器和终端接收客户端。

4.1 服务器图像采集程序设计

服务器端采用 Microsoft 公司的 VFW(video for windows) 数字视频开发包。在 Visual Studio2010 环境下进行 MFC 编程,启动多线程实现实时采集图像并传输的功能^[9]。软件流程如图 5 所示。

4.2 客户端图像接收显示程序设计

目的主机基于 VFW 类库利用 Visual Studio 2010 进行 MFC 编程,开启两个线程。线程一用 socket 接收图像数据,通过 File 类保存图片到本地磁盘上。线程二则用位图的方式显示保存好的图片。值得注意的是,两个线程访问同一个图片文件时会发生资源共享冲突,所以在程序设计时要对访问共同文件的两个线程加锁,保证同一时间只能有一个线程访问该资源。实现的伪代码如下。

```
task1_access_flag = true;
```

```
task2_access_flag = true;
```

任务一:

```
if(task1_access_flag){
```

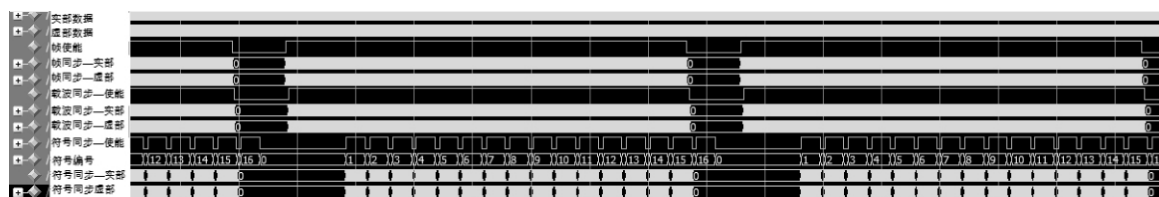


图 4 同步仿真结果

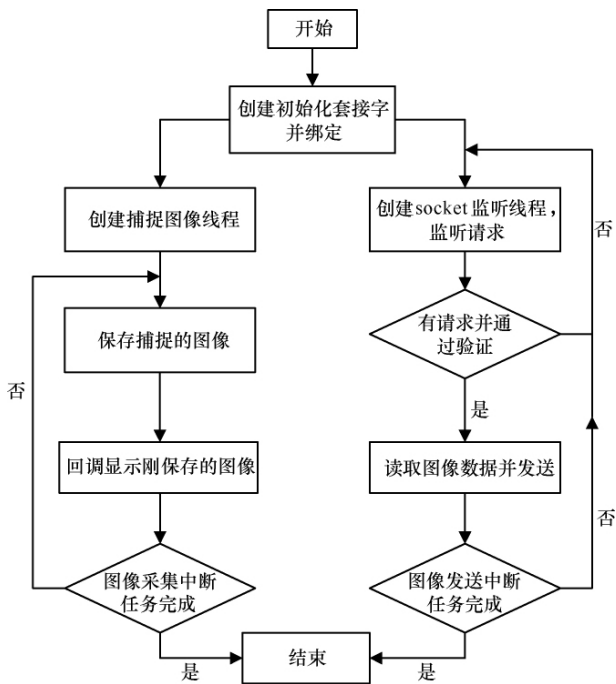


图5 服务器图像采集程序流程

```

task2_access_flag=false;
//接收图像数据,修改保存图像文件
task2_access_flag=true;
}
任务二:
if(task2_access_flag){
    task1_access_flag=false;
    //访问图像文件,用位图方式显示
    task1_access_flag=true;
}
  
```

5 结束语

项目中各设备具体连接方式如图6所示。服务器通过双绞线与发射机 CycloneIII 开发板连接,开发板通过 HSMC(high speed mezzanine cards)接口与 D/A 子板连接。D/A 子板的输出端口与接收端 A/D 子板输入端口通过 SMA(sub-miniature-A)接口线连接,然后 A/D 子板与接收机 CycloneIII 开发板通过 HSMC 接口连接,最后 CycloneIII 开发板与客户端通过双绞线连接。其中发射机通过共用时钟线为接收机提供共用时钟源。

该设计最终的实现效果如图7所示,这样,服务器采

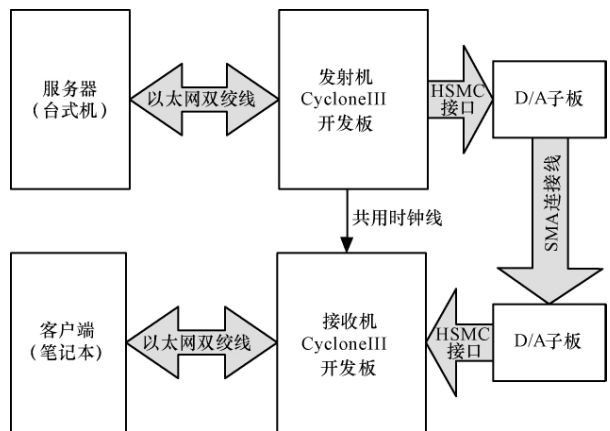


图6 车辆实时通信系统的硬件连接



图7 图像传输演示

集图像,图像数据经发射机和接收机实时传输最终在客户端上显示。由图7可以看出,该设计完成了预期效果,实现了基于 NiosII 和 802.11p 的点对点图像数据传输。

参考文献

- 1 Cyclone III development board reference manual. Altera Corporation, 2008
- 2 IEEE 802.11p-2010. IEEE standard for information technology-telecommunication and information exchange between systems-local and metropolitan area networks-specific requirements part 11: wireless LAN medium access control (MAC) and Physical layer (PHY) specifications amendment 6: wireless access in vehicular environments, Jul, 2010
- 3 Using the nicheStack TCP/IP stack-Nios II edition tutorial. Altera Corporation. 2010
- 4 史运锋. 基于 NiosII 软核的嵌入式以太网设计. 南京理工大学硕士学位论文, 2009
- 5 SOPC Builder User Guide. Altera Corporation, 2010
- 6 Jean J Labrosse 著, 邵贝贝译. 嵌入式实时操作系统林 UC/OS II(第二版). 北京: 北京航空航天大学出版社, 2003
- 7 徐小传. OFDM 系统及其 FPGA 实现. 北京邮电大学硕士学位论文, 2009

8 覃建策. OFDM 通信系统同步算法研究及其实现. 天津大学本科毕业论文, 2011

9 宋坤, 刘锐宁, 马文强. 视频技术方案宝典. 北京: 人民邮电出版社, 2008

IEEE 802.11p Based Real-Time Image Transmission System Design and FPGA Implement

Fu Guangyu, Luo Tao, Qin Jiance

(Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract A real-time image transmission system is designed based on CycloneIII EP3C120 development kit of Altera. According to needs in the project, a new System-On-a-Programmable-Chip (SOPC) hardware architecture is customized and multitask is run upon the MicroC/OSII embedded operating system which runs in the NiosII soft-core. So that images can be transmitted from one PC to another one. This design is based on the 802.11p protocol and implemented with VerilogHDL programming language and also gives a method to the data communication between NiosII soft-core and other hardware modules.

Key words NiosII, SOPC, FPGA, multitask, FIFO