

基于 FPGA 的嵌入式三态以太网设计

杨瀚程, 王芳, 张佳杰, 杨光辉

(国家数字交换系统工程技术研究中心, 郑州 450002)

摘要: 提出了一种基于 FPGA 实现嵌入式三态(10MB/100MB/1 000MB)以太网的设计方案, 分别从硬件和软件方面介绍了使用 FPGA 进行嵌入式系统设计的方法, 编写了一个控制系统进行 10MB/100MB/1 000MB 自切换程序, 并在工程中得以实现。

关键词: 大规模汇聚路由器; 嵌入式三态以太网; VxWorks

Design of Embedded Tri-Mode Ethernet Based on FPGA

YANG Han-cheng, WANG Fang, ZHANG Jia-jie, YANG Guang-hui

(National Digital Switching System Engineering & Technological R&D Center, Zhenzhou 450002)

【Abstract】 Based on the development of access convergence router (ACR) of National 863 Plan, this article puts forward a scheme using FPGA to implement the design of embedded Tri-mode Ethernet(10MB/100MB /1 000MB), introduces a way of using FPGA to design embedded hardware and software system, and programs a controlling system for auto change among 10MB/100MB/1 000MB. Such design is realized in project.

【Key words】 large-scale convergence router; embedded Tri-mode Ethernet; VxWorks

随着通信技术的迅速发展, 传统的百兆以太网已经不能满足快速增长的网络业务需求。由 IEEE 于 1998 年推出的 1 000Mb/s 以太网通信标准正日益广泛地应用于各种高速通信设计中。而内嵌处理器且整合了一系列知识产权(IP)核平台 FPGA 由于其自身功能的强大, 使利用 FPGA 结合嵌入式操作系统进行嵌入式千兆级以太网通信的开发成为可能。

1 整体设计

嵌入式三态以太网系统的硬件结构主要围绕平台 FPGA Virtex-4 FX60 进行设计, 系统的通信功能由 FX60 内部的 TEMAC 和 1 个外部三态以太网物理层芯片(PHY)实现。Winder River 推出的 VxWorks6.1 具有优秀的网络处理能力, 为千兆以太网通信提供了全面技术支持, 所以, 选择 VxWorks6.1 作为嵌入式三态以太网的操作系统。由于 FPGA 内部 BlockRAM 的容量无法满足 VxWorks 对存储空间的需求, 因此使用一片 16MB 的 FLASH 来存储 VxWorks 映像和引导程序, 以及一片 256MB 的 DDR 为其提供运行程序的空间。另外, 安装 1 个调试用的 RS232 串口。系统结构如图 1。

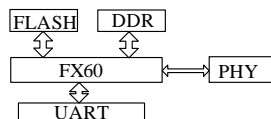


图1 嵌入式三态以太网系统结构

图1以硬核形式内嵌于FX60的微处理器PPC405和三态以太网媒体接入控制器(Tri-mode Ethernet MAC, TEMAC)构成了整个系统的核心。PPC405是一款32bit的Harvard结构处理器, 在450MHz时钟下的处理速度可高达680Mb/s, 而且在FPGA结构中有6Gb/s的峰值通信带宽。TEMAC核的设计符合IEEE 802.3-2002规范, 它提供了可选的网络管理特性, 包括基于每个分组的统计数据矢量和流程控制, 还为不限长度的VLAN帧和JUMBO帧提供了技术支持。

TEMAC支持半双工/全双工操作, 有3种不同的操作模式: 1 000MB, 100 MB和10MB, 可以用以下2种接口任意对其配置^[1]:

(1)在1 000MB模式下, TEMAC支持千兆位级媒体独立接口(GMII), GMII是一个运行速率为125 MHz的字节宽并行SDR接口;

(2)在100MB和10MB模式下, TEMAC均支持MII接口。媒体独立接口(MII)4-bit宽, 在100MB模式下其工作速率为25 MHz; 在10MB模式下, MII的工作速率为2.5MHz。所以, TEMAC完全兼容带有GMII/MII行业标准的三态以太网物理层芯片, 是开发嵌入式三态以太网的理想选择。

2 FPGA设计

针对平台FPGA设计, Xilinx公司推出了一套嵌入式开发专用工具: EDK(embedded development kits)。这是一套专用于FPGA内部位嵌入式处理器的软硬件协同设计工具, 包含了大量与PPC405设计相关的IP核, 使用EDK进行设计可极大地简化设计步骤和缩短开发周期。EDK的开发流程分为硬件设计和软件设计。

2.1 硬件设计

硬件的设计过程实际就是使用EDK对嵌入式三态以太网系统设计需要的知识产权(IP)核进行整合设置, 并最终生成可下载到FPGA中的比特流的过程。

在进行FPGA设计的时候首先应考虑到时钟问题。Xilinx在其FPGA设计中都使用了数字时钟管理(DCM)技术, 主要用于完成时钟的同步、移相、分频、倍频和去抖动等工作。DCM可以产生1MHz~420MHz的精确频率, 因而可以满足

基金项目: 国家“863”计划基金资助项目(2004AA103130)

作者简介: 杨瀚程(1981-), 男, 硕士, 主研方向: 嵌入式处理器及其应用; 王芳、张佳杰、杨光辉, 硕士

收稿日期: 2007-01-10 **E-mail:** yhc@mail.ndsc.com.cn

最先进的总线接口标准，其零延时时钟缓冲可以保证生成正负周期各 50%精确占空比的时钟信号。每个 DCM 都具有相位和频率的综合能力，对相位的精确控制可以达到 1%时钟周期的精度，可以满足时间微调应用的要求，完全能满足具有多个时钟，对时序有严格要求的系统级设计。

全局时钟管脚在大规模的 FPGA 设计中一直属于紧缺资源，就连 Virtex-4 FX60 也只有 16 对全局时钟管脚，针对这种情况，在设计中通过 DCM 使用一个全局时钟管脚来为 PPC405 和 TEMAC 提供 2 个全局工作时钟。

FX60 总共提供 12 个 DCM，可以根据设计需要自由使用。图 2 中，使用 EDK 将由 FPGA 全局时钟管脚输入的 100MHz Dcm_clk_s 时钟送入 DCM0 中，对其进行 3 倍频 1 分频的处理，得到 300MHz 的 PPC405 工作时钟 proc_clk_s；将 DCM0 中 CLK0 送出的 100MHz Sys_clk_s 送入 DCM1 的 CLKIN，经过 5 倍频 4 分频的处理，得到 125MHz 的 TEMAC 工作时钟 GTX_CLK。经测试表明，这 2 个时钟信号完全满足设计要求。

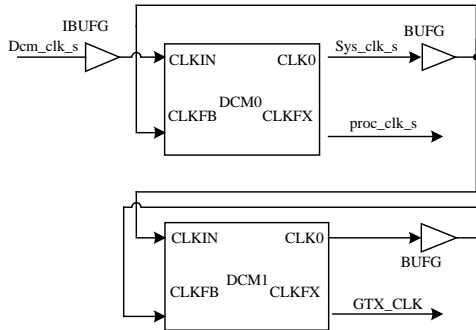


图 2 DCM 设计

时钟设计完成后，就可以着手使用 EDK 对所需的 IP 核进行整合了。IP 核的整合主要基于 IP 资源复用理念。IP 资源复用是指在集成电路设计过程中，通过继承、共享或购买所需的知识产权(IP)内核，然后再利用 EDA 工具进行设计、综合和验证，从而加速设计过程，提高设计质量，降低开发风险^[2]。IP 核整合步骤如下：

(1) 选择片内总线。三态以太网系统的核心处理器 PPC405 完全支持 IBM Core Connect 总线标准。CoreConnect 是由 IBM 开发的片上总线通信链，主要由 PLB(processor local bus)、OPB(on-chip peripheral bus) 总线以及 DCR(device control register)总线组成。设计中只采用 PLB 和 OPB 连接各种控制 IP 核，PPC405 通过它们实现对内部 IP 核及外设的访问控制功能；

(2) 连接 PLB。PLB 是片内的高速数据通道，应该将 DDR、TEMAC 等对速率要求很高的外设通过各自的控制核连接到 PLB 上。软核 PLB_BRAM 用于控制 BRAM，使 CPU 处于一个已知状态而不至于跑飞，在此将其连接到 PLB 上；

(3) 连接 OPB。OPB 慢速总线主要用来访问低速以及低性能系统资源，并不直接与 PPC405 连接，而是通过 PLB2OPB 桥与 PLB 连接，使得 OPB 控制的外设可用通过 PLB2OPB 访问存储器，同时处理器也可用通过该桥访问 OPB 上面的外设。UART 以及 FLASH 对速率的要求并不高所以在设计中将它们控制核连接到 OPB 上。

随着 IP 核的连接完成，IP 核的整合工作就结束了(IP 核结构见图 3)。

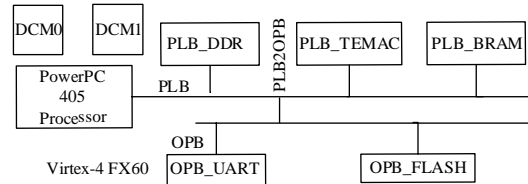


图 3 IP 整合结构

IP 整合完毕后，先在 EDK 中为各 IP 核进行基本参数的设置和地址空间的划分，再填写 EDK 在设计之初生成的 UCF(user constraints file)文件，用于对 FPGA 分配给各控制核与相应外设连接的管脚进行约束。使用 EDK 生成可下载到 FPGA 中的比特流后，硬件设计流程全部结束。

2.2 软件设计

EDK 的软件设计主要包括 BSP (board support package) 的生成和应用程序的编写。首先在 EDK 中选择 VxWorks6.1 作为操作系统，再为各个 IP 核选择相应的驱动，最后使用 EDK 提供的 BSPgen 为系统生成 BSP^[3]。但生成的 BSP 只是一个固定模板文件，不能完全满足当前系统要求。所以除了改动在 BSP 中 Makefile 以及 config.h 的 RAM/ROM 地址定义外还应根据设计需要，对 BSP 部分文件做如下修改：

(1) 将 config.h 中 CONSOLE_BAUD_RATE 配置为与所选 UART 的驱动中波特率的设置一致。

(2) BSP 在生成 TEMAC 相关部分时，并不会自动区分当前使用的 TEMAC ID、中断 ID 等，而使用 _index<temac>、vtag_intrid 以及类似字符来统一代替，所以需要做以下改动：

在 configNet.h 中：

将

```
IMPORT char XTemacLoadStringvtag_index<temac>[];
```

修改为 IMPORT char XTemacLoadString0[];

将

```
{vtag_index<temac>,XTemacEndAdapter_Load,XTemacLoadStringvtag_index<temac>,TRUE, NULL, FALSE};
```

修改为

```
{0,XTemacEndAdapter_Load, XTemacLoadString0, TRUE, NULL, FALSE};
```

在 sysNet.c 中

将

```
char XTemacLoadStringvtag_index<temac>
```

```
[MAX_END_LOAD_STR_SIZE];
```

```
static char XTemacMacAddrvtag_index<temac>[6] = { 0x6, 0x5, 0x4, 0x3, 0x2, 0x1 };
```

修改为

```
char XTemacLoadString0[MAX_END_LOAD_STR_SIZE];
```

```
static char XTemacMacAddr0[6] = { 0x6, 0x5, 0x4, 0x3, 0x2, 0x1 };
```

将

```
sysXTemacEndBldLoadStr(XTemacLoadStringvtag_index<temac>, vtag_deviceid,
```

```
XTemacMacAddrvtag_index<temac>, vtag_intrid);
```

修改为

```
sysXTemacEndBldLoadStr(XTemacLoadString0,XPAR_TRIMODE_MAC_GMII_DEVICE_ID,XTemacMacAddr0, -1)
```

BSP 修改完成后还要根据系统需求编写相应的应用程序。因为在 TEMAC 的软件驱动中，不支持 10MB/100MB/1 000MB 的工作模式的自动切换，而需要通过手动更改驱动

的部分参数来设置 TEMAC 的工作状态, 这样不利于整个系统的维护和操作, 所以应该设计一个程序来实现 10MB/100MB/1 000MB 自切换。

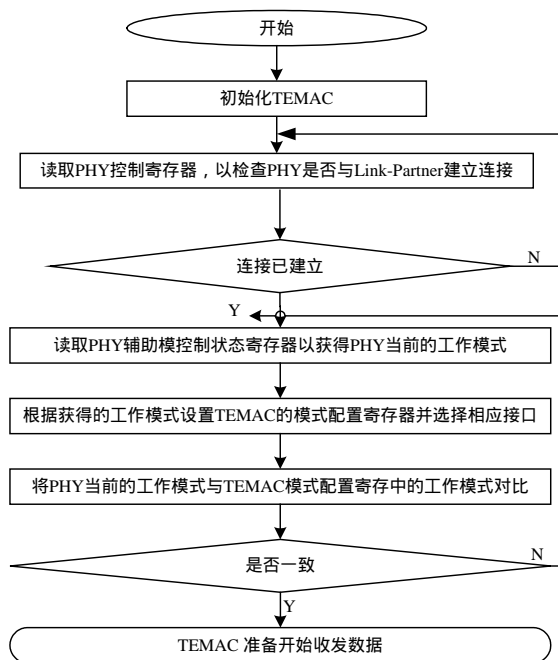


图 4 函数流程

因此, 使用 C 语言编写了函数 TemacAutoNeg()来完成这一功能: 所有 PHY 芯片都包括一个完全符合 IEEE 802.3I 规范的 SMI(serial manage interface)接口, MAC 通过 SMI 接口对 PHY 中的各种寄存器进行读写。程序应该在 TEMAC 初始

化完成后通过 SMI 查询 PHY 的控制寄存器中存放的 PHY 与 Link-Partner 的工作模式(10MB/100MB 或者 1 000MB), 再根据得到的工作模式对 TEMAC 的模式配置寄存器进行相应配置, 选择 TEMAC 采用 10MB/100MB 模式(MII 接口)还是 1 000MB 模式(GMII 接口)。这样, TEMAC 就能根据外部 PHY 的实际工作模式自动完成 10MB/100MB/1 000MB 的切换了。函数流程见图 4。

程序设计完成后, 在 usrAppInit.c 中加入该函数的入口函数, 使用 WorkBench2.3(WindRiver 提供的嵌入式开发环境, 内部集成了 VxWorks6.1, 并采用 C 语言)完成 BSP 编译, 生成可下载 VxWorks 镜像以及 Bootroom 文件。至此, 整个嵌入式三态以太网设计全部完成。

3 小结

本文使用平台 FPGA 进行嵌入式三态以太网系统的设计, 实现了以往需要多个专用芯片才能实现的功能, 大大节约了板面空间和简化了设计。目前该系统在大规模汇聚路由器中运作良好。随着 FPGA 的集成度不断加强, 功能不断更新, 基于 FPGA 的嵌入式开发必将在电子设计领域拥有更为广阔的发展空间和美好的应用前景。

参考文献

- 1 Xilinx. Virtex-4 Embedded Tri-Mode Ethernet MAC User Guide[Z]. 2006.
- 2 任爱锋. 基于 FPGA 的嵌入式系统设计[M]. 西安: 西安电子科技大学出版社, 2004.
- 3 周启平. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 北京: 中国电力出版社, 2004.

(上接第 138 页)

中, $s(t)$ 是 t 时刻源端缓冲器的大小; q_{\max} 是最大被允许的缓冲器大小; T_s 是源端缓冲器在丢弃策略被迫开始时的缓冲器大小初始标志大小; R_a 是编码速率; $R_n(t)$ 是当前网络传输速率; K_{\max} 是媒体位流层的数量, $k=1, 2, \dots, K_{\max}$ 是层索引 (K_{\max} 是最不重要的层); r_k 是 k 层的编码速率; P_k 是 k 层包传输的可能性。算法 1 描述了初始 T_s 过度时一个有选择的丢弃策略就被强迫执行了。包传输可能性在算法 2 中计算^[6]。

源端缓冲初始值 T_s 的选择对整个系统的执行来说很重要。如果初始值过小, 将导致在源端缓冲不必要的包丢失, 如果初始值过大, 将增加延迟并最终引起接收端缓冲下溢。接近理想的初始值为 B-D, 这里 D 是包中图像组的大小, B 是接收端的预设缓冲大小。

通过以上基于缓冲管理的拥塞控制设计, 完成了以下目标: (1) 采用可升级的帧频去适应各种不同的网络环境; (2) 一个简单普通的端到端的缓冲器管理设计作为源端流媒体编码的扩展, 通过变任意的丢包为有控制的丢弃低优先级的包, 能为多变的网络环境提供牢固的适应性; (3) 综合的传输缓冲管理以及接收端的帧频调整来在接收端产生较高质量的媒体播放效果。

4 结束语

本文创新点: 基于 P2P 覆盖网的流媒体传输策略, 充分考虑网络中流媒体传输质量控制的多个方面, 通过基于多跳的 FEC 纠错控制, 有选择地在覆盖网络的中间节点上进行

FEC 纠错, 从而在很大程度上降低了网络上的丢包率和接收端节点上的计算复杂度, 增加了覆盖路径上的有用带宽, 同时综合利用基于缓冲管理的拥塞控制, 变源端任意的丢包为有选择性的丢包, 有效地避免了在流媒体源端重要数据包的丢失, 保证了拥塞环境下依然高质量的流媒体传播。

参考文献

- 1 Tickoo O, Kalyanaraman S, Woods J W. Efficient Path Aggregation and Error Control for Video Streaming[C]//Proc. of IEEE ICIP, Singapore. 2004-10.
- 2 Gringeri S, Egorov R, Shuaib K, et al. Robust Compression and Transmission of MPEG-4 Video[C]//Proc. of ACM Multimedia Conference. 1999.
- 3 Shan Y, Kalyanaraman S. Hybrid Video Downloading/Streaming over Peer-to-Peer Networks[C]//Proc. of International Conference on Multimedia and Expo. 2003: 665-668.
- 4 龚海刚, 刘明, 毛莺池, 等. P2P 流媒体关键技术的研究发展[J]. 计算机研究与发展, 2005, 42(12): 2033-2040.
- 5 Li Jiang. End-to-End Multicast Congestion Control and Avoidance[D]. Troy, New York: Graduate Faculty of Rensse-laer Polytechnic Institute, 2003-07.
- 6 Bajic I V, Tickoo O, Balan A, et al. Integrated End-to-End Buffer Management and Congestion Control for Scalable Video Communications[C]//Proc. of the 11th Packet Video Workshop. 2002-12.

