

基于 FPGA 的嵌入式 SNMP 代理设计与实现*

张玲 邓巍 何伟 宋炎翼 胡又文

(重庆大学通信工程学院 重庆 400044)

摘要 随着网络设备的广泛应用,网络管理的需求日益增长,同时网络管理代理设计变得越来越重要。笔者基于先进的可编程器件 FPGA,采用 SOPC 技术搭建 SNMP 代理系统结构,使用嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 实现代理的任务调度。将 SNMP 代理划分为 mib、snmpd 算法、报文编解码和 trap 四个关键模块。考虑到工程实际操作系统应用环境,笔者实现了 SNMPv2 的代理,对于 SNMPv3 的实现方式,也提出了参考性建议。结果表明,配合 SNMP 管理站,设计能高效地实现对设备的网络管理。

关键词 简单网络管理协议,代理,现场可编程门阵列,片上可编程系统,管理信息库

1 引言

随着计算机网络的普及和广泛应用,网络管理的需求也在逐步增加。SNMP 作为一种简单有效的网络管理协议,在现实的应用中占据着越来越多的市场份额。现在大多数网络中转设备都支持 SNMP 网络管理协议,而且网络设备供应商都为自己的设备申请了 OID 号^[1,2]。SNMP 管理模型中,管理站是对代理进行管理和监控的中心,代理是对被管设备进行管理和监控的网络前端^[3]。结合嵌入式技术,将 SNMP 网络管理协议应用在网络终端上,达到对网络设备的管理和控制是一种崭新的尝试。对于嵌入式系统,FPGA 理应为首选。笔者提出基于 FPGA 的网络代理模型设计方案,系统地勾画出代理的实现框架和工作流程。结合 SNMP 管理站,一起构成一个 SNMP 管理系统。

2 系统结构

在系统构建上,考虑现有的嵌入式网络实现方案,大多数系统都采用微控制器及其对应的外设接口和相应的软件来实现。笔者介绍的嵌入式 SNMP 代理的设计,采用 SOPC 技术将系统划分为各个功能模块,并把大部分功能模块集中到一块 FPGA 中,对外只保留必要的存储器 and 接口,简化了系统并提高了灵活性,同时也充分说明了 SOPC 片上可编程系统的优势。为了精简系统,并考虑到系统所需的资源和生成代码的大小,设计系统(图 1)主要由以下几部分组成:CYCLONE 1C20 FPGA,包括嵌入 Nios II 软核、系统定时器、TCP/IP 协议栈定时器以及 Avalon 总线等设计,实现系统的主要功能;epcs4 用来在上电时对 FPGA 进行配置;jtag

用来下载可执行文件和应用嵌入式逻辑分析仪进行调试;Flash 用来存放软件代码以及一些需要保存的参数;Sdram 用来存储系统运行时的代码和数据;网络接口芯片采用 Smsc 公司的 Lan91c111 芯片作为网络接口,该器件是一个以太网控制器,实现了网络七层协议栈中的传输层和 MAC 层的功能,另外,它具有 10/100M 自适应、双工/半工自适应等功能和很好的网络兼容性^[4]。

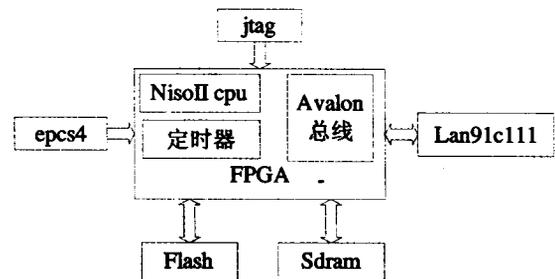


图 1 系统框图

3 硬件设计

硬件设计是指 FPGA 内部构建 CPU 及其外围接口设计。硬件设计使用 QUARTUS II 开发工具,它能提供从设计输入到器件编程的全部功能;采用图形化开发工具 SOPC BUILDER,它可以添加 IP 核和用户自定义模块。在设计中,QUARTUS II 主要编译用户自定义模块和综合设计;SOPC BUILDER 主要添加 CPU、用户自定义模块和外围接口等^[5](图 2)。在添加模块的过程中,一定要把 Flash 和 Sdram 等存储设备的地址锁定,并根据设计的要求分配起始地址和中断优先级。最后,点击 Generate 即可生成系统所需 CPU 和外围器件的 VHDL 文件和针对该设计的 C 语言头文件和库函数^[4]。

张玲 副教授,硕士生导师,主要研究方向:电子系统设计,图像处理;邓巍 硕士研究生,主要研究方向:嵌入式应用,网络管理;何伟 副教授,硕士生导师,主要研究方向:电子系统设计,图像处理。宋炎翼 讲师;胡又文 研究生。

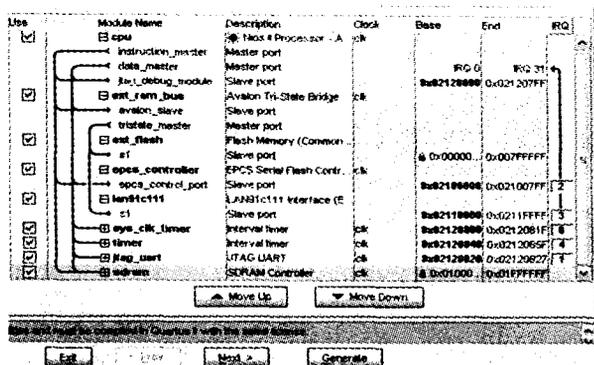


图2 CPU和外围器件设计

4 软件设计

软件设计部分使用 Nios II 开发工具 IDE, 实现 SNMP 代理主体部分。SNMP 代理是一个使用 TCP/IP 协议、多线程的网络系统。因此, 该设计需要 TCP/IP 协议栈和操作系统。在 Nios II 集成开发环境下, 只需在设置工程的属性时, 选中 TCP/IP Lean 协议栈和 RTOS 中的 Micro C OS/II 即可。整个软件部分由操作系统、mib、报文编解码、snmpd 算法和 trap 五部分组成^[6], 如图 3 所示。

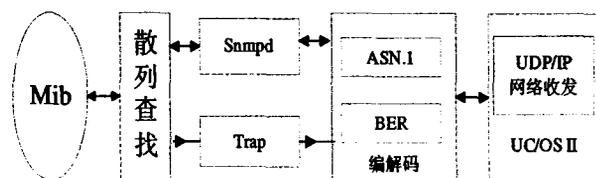


图3 软件设计框图

4.1 操作系统

笔者采用 Nios II 集成开发环境自带的 $\mu\text{C}/\text{OS-II}$ 嵌入式操作系统, 它是一种可固化、可剪裁、占先式的多任务实时操作系统, 并为每个任务分配单独的栈, 提供多种系统服务和中断管理。笔者共建立初始化 TCP/IP 协议栈和网卡芯片、初始化散列表、处理 snmpd 算法和发送 trap 四个任务, 同时对各个任务分配了不同的优先级, 完成任务调度, 并通过邮箱通信。

4.2 mib

mib (Management Information Base) 即管理信息库, 它定义一组由 SNMP 代理维护的变量以及每个变量的含义。mib 变量记录每一个相连网络的状态、发生错误的次数以及内部数据结构的当前内容。因此, 它是 SNMP 操作变量以及生成变量的基础, 包含 SNMP 的所有内容。

实现中, 使用一个数组来保存这些信息, 数组中的每一个元素对应于一个 mib 变量。数组中的每个数据元素都为识别一个变量提供充足的信息, 包含该 ASN.1 名字 (对象标识符)、类型以及是否可写的特征, 并且包含按字典序的下一个变量。当含有 ASN.1 名字的请求分组到达时, 代理必须能够

识别该名字, 并调用一个能够满足该请求的过程。

在实现 mib 以后, 所要考虑的就是如何提高变量名的转换效率, 笔者采用散列表的方式。散列表本身由一个指针数组组成, 其中每个指针给出一个链表地址, 链表节点代表可进行散列寻址的 mib 变量。实际中, 将每一个 mib 变量置入相同大小的数组中, 其结果是链表的平均长度比 1 稍大一些。因此, 大多数查找能立即找到所需值, 并不需要搜索链表。

最后, 只需要将定义好的 mib 变量与 mib 数组中的元素一一对应, 实现 mib 数组的初始化。

4.3 报文编解码

SNMP 有两种数据表示形式: ASN.1 和内部表示形式。其中, ASN.1 数据经 BER 编码后即数据表示成 TLV (Type+Length+Value) 的形式, 用于数据传输; 内部表示形式用于数据处理。编码是把本机内部数据转化为 ASN.1 数据表示形式, 经 BER 模块编码后投递到代理的网络发送接口处, 以供网络发送数据; 解码是把网络接收接口处的数据经 BER 模块解码后, 把 ASN.1 数据表示形式转化为本机能够理解的内部形式, 以供内部处理数据。针对整型、字符以及 OID 等数据类型, 笔者提供相应的编解码函数。

4.4 snmpd 算法

snmpd 算法处理报文的请求与应答。首先, 它打开一个 SNMP 使用的 udp 端口 (默认值 161), 然后进入一个无限循环, 等待下一个传入的报文。当一个 SNMP 报文到达时, snmpd 分析该报文, 包括长度、版本号、pdu 类型以及差错字段, 将其转换为内部格式, 分配一个节点, 并保存到请求描述模块中。除了提取首部中的字段之外, snmpd 从报文中提取对象表示符序列 OID, 并把对象表示符序列 OID 转换到一个链表中。在转换时, snmpd 调用解码模块把每个 ASN.1 对象标识符转换成内部表示形式。

当报文和变量名序列都已成功转换为内部格式后, snmpd 解决报文中提出的请求, 包括为链表中的每个对象标识符执行指定的 set、get、getnext 或 getbulk 操作。在完成这一工作后, snmpd 生成应答报文, 并用 write 操作向管理站发送应答报文。代理发送了一个应答报文后, snmpd 释放变量名链表, 并返回到主循环的头部, 等待下一个输入的报文。

如果因出错而无法顺利完成上述工作, 则代理生成一个差错应答报文, 该报文含有差错类型代码 (error type code) 和差错索引 (error index)。差错类型码给出差错发生的原因, 而差错索引指出在变量绑定列表中导致差错的变量名。

4.5 trap

trap 通知管理站被管设备出现异常情况, 报告异常出处, 它是一种主动向管理站发送信息的格式, 不同于上述 get 等操作^[7]。

根据指定的接收管理站, 打开一个 udp 端口 (默认值 162), 然后进入一个无限循环, 等待接收 trap 信息, 系统出现差错或异常时, 接收邮箱发送的 trap 信息, 包括系统时间、 trap 类型以及产生 trap 的 OID 和值。然后将 trap 信息绑定到 trap pdu 变量绑定列表中, 生成报文, 并用将报文发送到管理站。最后释放资源, 返回循环头部, 等待接收下一条 trap 信息。

5 结果分析

笔者使用 AdventNet 公司的 MibBrowser, 模拟 SNMP 管理站。通过测试, 该设计能够及时响应管理站发出的请求, 并在系统异常时, 向管理站发送告警信息。结果如图 4 所示。

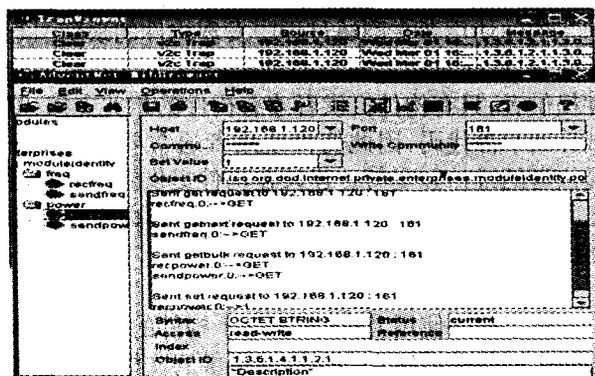


图 4 测试结果

结束语 该设计不但在计算机上得到仿真验证, 而且在具体工程实践中得到应用。μC/OS-II 只

是一个实时操作系统的内核, 没有网络、文件系统等额外模块。对网络功能要求比较高的 SNMP 代理系统而言, μC/OS-II 确实不如嵌入式 Linux、Vx-Works 等具有强大网络功能的操作系统方便, 虽然可以移植 TCP/IP 协议栈, 但高层开发依然十分繁琐, 特别是 SNMPv3 代理。笔者实现了 SNMPv2 代理, 若要实现 SNMPv3 代理, 以下方法可供参考:

(1) 使用 FPGA&μCLinux 的方式。μCLinux 适用于不含 MMU 微处理器的运行环境, 如 Nios II 软核。它具有稳定、强大的网络功能, 较低的资源占用率。用它实现 SNMPv3 代理将会事半功倍。

(2) 使用 ARM&Linux 的方式。ARM9 以上的处理器都有 MMU, 支持完整的 Linux。在 Linux 的环境下使用日益成熟的 ARM 处理器实现 SNMPv3 代理也会十分的方便。

参考文献

- 1 王兆峰, 彭晓燕. 三层以太网交换机 SNMP 代理软件的设计与实现[J]. 微计算机信息, 2004, 20(5): 60~61
- 2 Stalling W. SNMP 网络管理[M]. 胡成松, 江凯译. 北京: 中国电力出版社, 2003
- 3 罗雪松, 罗蕾, 许子辛. 嵌入式 SNMP Agent 的设计与实现[J]. 计算机应用研究, 2004, 21(10): 220~221
- 4 白璘, 孙肖子, 白玉霞. 基于 NIOS II 的嵌入式 Web 服务器[J]. 现代电子技术, 2005, 28(12): 30~31
- 5 周博, 邱卫东. 挑战 SOC-基于 Nios 的 SOPC 设计与实践[M]. 北京: 清华大学出版社, 2004
- 6 刘红, 白栋, 孔令山, 等. 嵌入式 SNMP 软件的设计与实现[J]. 计算机工程与应用, 2001, 37(21): 52~54
- 7 顾华玺, 邱智亮, 刘增基. 嵌入式代理在以太网交换机中的实现[J]. 电信科学, 2002, 18(6): 24~28

(上接第 138 页)

- 2 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proceedings of 1993 ACM-SIGMOD International Conference on Management of Data, Washington, D. C., May 1993. 207~216
- 3 Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C, eds. Proc. of the 20th Int'l Conf. on Very Large Data Bases. Santiago; Morgan Kaufmann, 1994. 478~499
- 4 Han J, Pei J, Yin Y. Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data, Dallas, TX: ACM Press, 2000. 1~12
- 5 Pei Jian, Han Jiawei, Mao Runyng. Closet: An efficient algorithm for mining frequent closed itemsets. In: SIGMOD International Workshop on Data Mining and Knowledge Discovery, May 2000
- 6 Pei Jian, Han Jiawei, Wang Jianyong. Closet+: Searching for

- the best strategies for mining frequent closed itemsets. In: SIGKDD '03, August 2003
- 7 Lucchese C, Orlando S, Perego R. Fast and Memory Efficient Mining of Frequent Closed Itemsets; [Technical Report CS-2004-9]. Nov 2004
- 8 杨红菊, 梁吉业. 一种挖掘频繁项集和频繁闭包项集的算法. 计算机工程与应用, 2004, 40(13): 176~178
- 9 Mohammad El-Hajj, Zaiane O R. COFI-tree Mining: A New Approach to Pattern Growth with Reduced Candidacy Generation. In: Workshop on Frequent Itemset Mining Implementations (FIMI'03) in conjunction with IEEE-ICDM 2003, Melbourne, Florida, USA, November 2003
- 10 范明, 李川. 在 FP-树中挖掘频繁模式而不生成条件 FP-树. 计算机研究与发展, 2003, 40(8): 1216~1222
- 11 Gopalan R, Suchay YG. High Performance Frequent Patterns Extraction using Compressed FP-tree. In: Proceedings of the SIAM International Workshop on High Performance and Distributed Mining, Orlando, USA, April 2004