

基于 FPGA 的 Web 服务定时系统设计与实现

邵 磊, 倪 明, 王 强

(中国电子科技集团公司第三十二研究所, 上海 200233)

摘 要: 根据现代高速网络及高 Web 服务性能的应用需求, 提出一种基于现场可编程门阵列的 Web 服务器定时系统, 实现定时服务模块及子模块的设计与验证。给出其与嵌入式 RAM 等其他硬件实现的对比和分析。该系统的时钟频率达 125 MHz, 可实现 10 万个客户端连接数的毫秒级定时服务设计要求, 并移植到专用 Web 服务器的 ASIC 中。

关键词: 现场可编程门阵列; Web 服务器; 定时服务; TCP/IP 协议

Design and Implementation of Web Service Timing System Based on FPGA

SHAO Lei, NI Ming, WANG Qiang

(The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 200233, China)

【Abstract】 With respect to the requirement of high speed Web and high performance Web service, a scheme of Web service timing system based on FPGA is introduced. The contrast and analysis is given with other methods to realize the Web service by debugging and verification. This design allows one hundred thousand clients to connect the Web server and can reach more than 125 MHz clock frequency by the final debug and test. This system is of advantageous transplant ability and appropriate expansibility for special Web server ASIC.

【Key words】 Field Programmable Gate Array(FPGA); Web server; timing service; TCP/IP protocol

DOI: 10.3969/j.issn.1000-3428.2011.21.076

1 概述

Web 服务器的主要功能是提供网上信息浏览服务, 其定时服务能确保用户在一定的时间内获取 Web 服务。由于用户在通过 Web 浏览器访问信息资源的过程中, 关心的是在其忍耐允许的情况下能不能享受到快速的网络浏览, 因而 Web 服务的定时系统显得尤为重要^[1]。

传统的 Web 服务器是基于高性能的多处理机和大容量的冗余外部存储空间的大型机架构, Web 服务定时系统也是用软件实现的定时器提供定时服务。国内外的研究还没有把定时服务系统和整个 Web 服务系统一起用一个 ASIC 实现的, 目前的产品也只有单个芯片实现 Web 服务部分的服务(如韩国生产的实现 TCP/IP 协议的芯片)。

本文介绍一种基于完全由 FPGA 硬件实现的 Web 服务中的定时和超时管理服务系统。该系统采用了 Xilinx 公司 Virtex-5 系列的, 型号为 XC5VSX50T 的 FPGA 芯片和 BEECube 开发平台。系统内全局时钟实现、消息的建链、拆链、空闲及数据包确认超时管理都基于 FPGA 用硬件实现; 实现了时钟频率达 125 MHz 以上, 客户端连接数达 10 万个连接管理的设计要求。目的是为整个 Web 服务器系统设计实现 ASIC 的一部分。这样设计出来的 Web 服务器在思博伦网络测试仪下, 速度和并发连接数都提高 90.4%, 用软件实现同等功能相比降低了 40.6%。它也将是未来云计算的云端 Web 服务器中不可缺少的一部分。

2 Web 服务器的总体方案及架构

2.1 Web 服务器系统总体结构

Web 服务系统主要包括消息分解、消息处理、TCP 封装、URL 解析、连接管理表和定时服务 6 个模块。其中, 消

息分解把消息分解为: Get 消息处理消息, 建链请求消息, 拆链请求消息, 拆链确认消息和数据包确认消息 5 类消息。整个消息传递过程中都由定时服务系统确定消息文件的传送, 确保和满足客户端获取消息的时间及速度。每个连接都由连接管理表管理, 每个模块从连接管理表读取任务, 并通过状态机给出每个模块之间的处理结果, 实现状态处理转换。

2.2 基于 FPGA 的 Web 服务器定时服务系统

基于 FPGA 的 Web 服务器定时服务系统划分为顶层模块和外部接口模块, 图 1 为基于 FPGA 的 Web 服务器定时服务结构图。

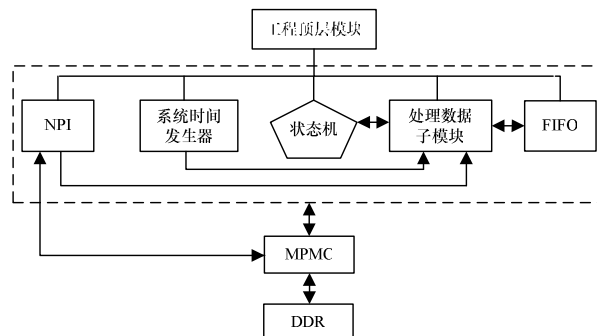


图 1 基于 FPGA 的 Web 服务器定时服务结构

顶层模块包括 NPI(Native Port Interface)、系统时间发生

基金项目: 国家“863”计划基金资助重点项目“新概念高效能计算机体系结构及系统研究开发”(2009AA012201)

作者简介: 邵 磊(1984—), 男, 硕士研究生, 主研方向: 计算机系统结构; 倪 明, 研究员; 王 强, 硕士

收稿日期: 2011-04-20 **E-mail:** shaolei5167@163.com

器、状态机、处理数据模块和先进先出队列(FIFO)接口队列模块。系统时间发生器提供精确到毫秒级的时钟时间,数据处理模块所要用的时钟和处理都是从系统时间发生器获得。处理数据子模块与状态机和NPI互通有无;处理数据子模块处理结果送到FIFO队列。NPI接口的提供是为了更好地利用MPMC(Multi Port Memory Controller)特性;NPI提供一个简单的内存接口以适应不同的协议;允许同时通过FIFO进行数据的读写;提供地址线、数据线和控制信号进行内存操作;以及可以配置成32位或64位的数据宽度以适应不同的系统需求。

顶层模块外的MPMC模块,最多可以有8个端口,这些端口是相互独立的^[2]。也就是说可以通过任意一个端口访问DDR内存。如果有2个端口同时访问内存,MPMC的总裁器会按一定的规则总裁并决定其先后。同时MPMC的每一个端口均带有缓存,所以在向一个端口写入数据的时候,数据是先写入缓存,等到这个端口取得内存写入权后再从缓存写入DDR内存。

3 定时服务的设计与实现

3.1 具体功能描述

基于FPGA的Web服务器定时服务提供精确到毫秒级的系统计时和负责各种定时器的超时判断,并进行相应的处理。定时服务所用到的定时器有:建链握手定时器,拆链握手定时器,空闲定时器和数据包确认定时器。连接建立定时服务从连接管理表接口开始,轮询读取连接状态和查找已经定义的处理相应事务的时间片长,转向相应的定时服务进行计时并做出相应的超时判断,如果超时则转向超时处理,如图2所示。

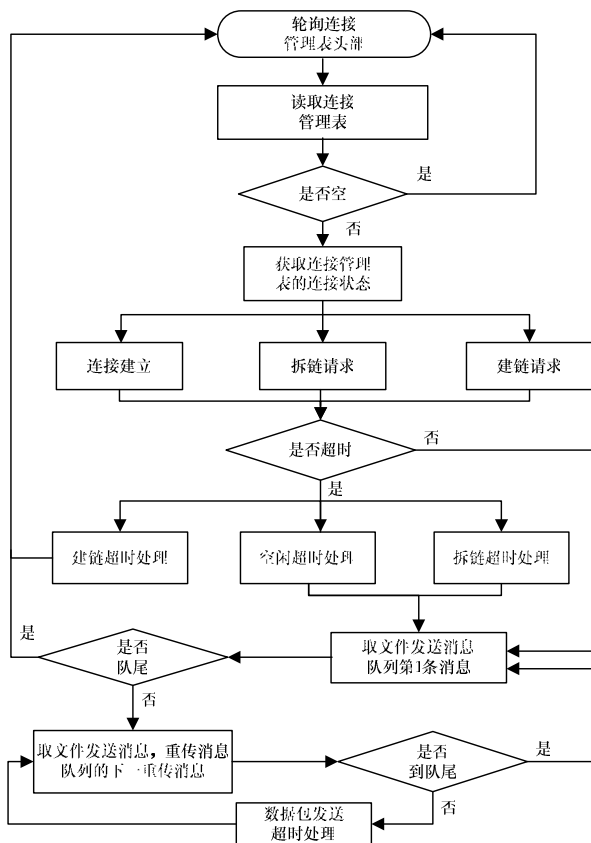


图2 基于FPGA的Web服务器定时服务系统处理流程

定时服务的处理流程如下:

(1)访问连接管理表链表,读取下一个连接管理表,若为

空,则回到连接管理表链表的头部,重新执行。

(2)访问连接管理表的连接状态:

1)若连接的状态为建链请求,超时即触发建链超时处理,然后转到第(1)步。

2)若连接的状态为连接建立,超时触发空闲超时处理,然后转到第(3)步。

3)若连接的状态为拆链请求,超时即触发拆链超时处理,则跳到第(3)步。

(3)取文件发送消息队列的第1条消息,若到队列尾,则跳转到第(1)步。

(4)取文件按发送消息,重传消息队列的下一重传消息,若到队尾,则跳到第(3)步。

(5)判断时钟是否超时,若超时则触发数据,发送超时处理,跳到第(4)步。

3.2 系统时间发生器模块FPGA的设计与实现

本设计是在XilinxISE12.2平台上用Verilog开发实现的^[3]。功能仿真平台是ModelSim6.5f,调试平台是BEECube FPGA平台。因为定时服务由系统时间发生器提供系统时间。本文的系统时间发生器产生毫秒级的计时,所用的时钟是FPGA Virtex5-SX50T提供的10 MHz的时钟,用此时钟做计时器,因为此时钟周期是10 ns,所以每10万个时钟周期为1 ms,生成的输出时钟为定义的32位时钟格式。板级调试结果如图3所示。

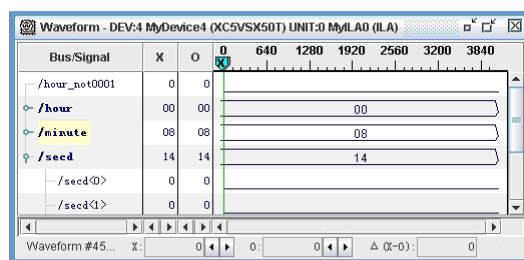


图3 毫秒级系统时间发生器的ChipScope调试结果

3.3 Web服务器定时服务超时管理模块设计

本文的设计与实验基础是每次重传设置的超时定时器的时长是固定不变的。而在TCP协议中,重传的时间和2个参数有关,一个是RTT时间,另外一个重传的次数。整个基于FPGA实现的Web服务器中所要用到的超时管理有:TCP建立连接过程中是否超时判断及处理;轮询连接管理表所查询的连接状态为空闲,或是拆链过程中所对应产生的空闲/拆链超时判断及相应处理;数据包确认超时判断与重传超时判断处理。

(1)定时服务建链超时处理

Web服务开始轮询是首先读取连接管理表的下一个地址,如果连接管理表的状态非空,获取连接管理表的当前状态,判断超时处理类型,若为空则返回链首。如果是建链请求超时,即做相应的建链请求超时处理:修改连接管理表状态为关闭;将连接管理表地址压入空闲连接管理堆栈,并释放连接管理表地址空间;从连接管理表队列删除该连接,删除源地址,即管理表的索引记录(删除客户端IP、客户端端口号对应的连接管理表指针)。

(2)定时服务空闲连接或拆除链接超时处理

查询连接管理表的状态为空闲或拆除链接状态时,释放文件发送消息队列中的消息,及其拥有的HTTP头部资源和重传消息队列中的消息。然后修改连接管理表状态为关闭;从连接管理表队列删除该连接,将连接管理表地址压入空闲

连接管理堆栈, 释放其空间; 删除源地址即管理表的索引表记录。

(3)数据包确认超时处理

连接管理表的状态为数据包确认时, 判断已经发送数据包的次数是否已经大于 4 次, 如大于 4 次则执行空闲超时处理过程; 如果已经发送数据包的次数小于 4 次, 就将 TCP 包封装消息体中的重发次数执行加 1 操作, 重新设置计时器, 将消息地址压入 TCP 包封装请求队列的头部。

3.4 MPMC 接口与内存通信的实现

Web 服务器定时服务系统设计的目的是定时服务时间精准, 数据处理速度快, 这样就要求数据处理子模块的处理能力以及和内存通信的能力要强。MPMC 控制和管理内存的数据存取。通过 NPI 提供的内存接口和内存进行通信。当系统时钟到来或系统初始化后, 使 NPI 的地址读/写信号有效, 使 MPMC 地址引脚和内存相应的 NPI 地址相连。

定时服务时钟模块 PIM 引出地址请求、地址应答、32 位地址线等信号线与 MPMC 对应的信号线相连接。初始化后定时服务时钟模块发出地址请求信号, 占用地址后向 MPMC 发出地址应答信号; 同时把数据放到地址线上, 根据读写模式的选择读或写数据; 然后把处理数据写入 FIFO 队列。读内存控制时序就是模式为读模式, 有几种不同情况的延迟, 其他内存操作时序与写类似。

4 调试验证及与其他实现方法的对比分析

验证的重点是定时服务系统通过 MPMC 对内存数据读写的实现。调试分析思想是, 只要用超级终端对内存进行写数据, 用 ChipScope 通过 JTAG 口先对写入的数据读出, 然后对比写入和读出的数据。如果前后数据一致就能说明对内存的数据存取设计正确。也可以把定时服务模块加入到 Web

服务器的总工程里, 通过总工程的综合、布局布线等调试, 由客户端提供请求, 分析定时服务模块设计的正确性。对比写入数据和读出的 32 位数据, 结果数据一致, 通过反复地在开发平台上进行调试和分析, 超时处理机制和 TCP 包重传机制的计时器调用也均达到设计要求。

通过发包工具和 5 台客户机完成发包实验。结果显示单客户端连接数能达到 2 万个, 在定时系统控制下丢包率在 0.01% 以下。客户等待时间控制在 0.5 s, 超时重发 2 s, 相比其他嵌入式 ARM: Web 服务器下的浏览器等待时间 30 s, 超时重发 15 s~35 s, 该系统性能大幅提高。

5 结束语

本文介绍了一种基于 FPGA 的 Web 服务器定时服务系统, 对系统级计时器进行设计、超时管理及 TCP 重传, 对内存数据的存取等关键问题给予解决方案。对设计的模块进行验证, 调试结果稳定可靠, 并与其他硬件实现方法进行了简单的对比分析。随着物联网、云计算的发展和不断深入, Web 服务技术将得到更广泛的应用和发展, 设计的全硬件定时服务系统将发挥更加积极的功能和作用^[4]。

参考文献

- [1] Reed K D. TCP/IP 基础[M]. 张 文, 邢淑琴, 杨彦昌, 等, 译. 北京: 电子工业出版社, 2002.
- [2] 叶肇晋, 张稀楠, 马 磊. 基于 XILINX FPGA 片上嵌入式系统的用户 IP 开发[M]. 西安: 西安电子科技大学出版社, 2008.
- [3] 田 耘, 徐文波, 胡 彬, 等. Xilinx ISE Design Suite 10.X FPGA 开发指南[M]. 北京: 人民邮电出版社, 2008.
- [4] 邵华钢, 程 劲, 王 辉, 等. 面向物联网的系统及其中间件设计[J]. 计算机工程, 2010, 36(17): 84-86.

编辑 顾逸斐

(上接第 221 页)

测试内容有: 驱动程序是否正确地安装到内核模块中; 驱动程序是否能准确无误地响应上层软件对其的请求; 驱动程序是否能正常卸载。

加载/卸载驱动程序的测试可用 Linux 系统提供的命令来检测, 在此不多做解释, 详情请参考相关工具书。驱动响应上层软件的成功与否可用通过“dmesg”命令来查看打印信息来判定驱动程序执行是否正确。发送和接收数据功能的测试是通过双绞线与另一台装有网络测试卡(Condor 卡)和相关测试软件的计算机相连, 通过本机与测试计算机发送和接收数据, 对比双方的数据来判断发送接收数据的正确性。以发送为例, 上层软件向 2 个不同 VL 各发送 10 个数据, Condor 卡配备的抓包软件就接收到了 20 个包, 如图 5 所示。

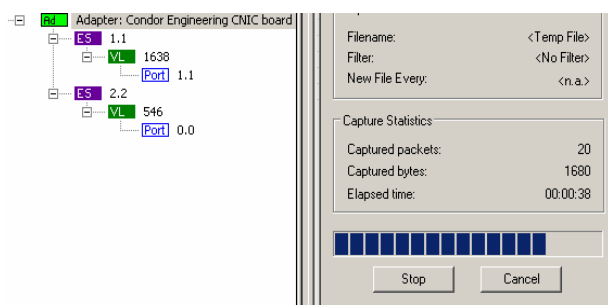


图 5 发送数据测试结果

6 结束语

本文介绍了 Linux 下 AFDX 端系统驱动的设计与实现。该驱动使 AFDX 网络能够适应于 Linux 这样的流行操作系统环境, 扩展了 AFDX 网络的适用范围。经测试验证本驱动程序能够很好地将 AFDX 硬件板卡连接到 Linux 操作系统中, 并实现相应的功能。

参考文献

- [1] ARINC Inc.. Arinc Project Paper 664: Aircraft Data Network, Part 7—Avionics Full Duplex Switched Ethernet(Afdx) Network[EB/OL]. (2009-09-01). https://www.arinc.com/cf/store/catalog_detail.cfm?item_id=1270.
- [2] 武 华, 马捷中, 翟正军. AFDX 端系统通信端口的设计与实现[J]. 测控技术, 2009, 28(3): 56-58.
- [3] ARINC Inc.. Arinc Specification 653-2: Avionics Application Software Standard Interface Part 1—Required Services[EB/OL]. (2008-12-01). https://www.arinc.com/cf/store/catalog_detail.cfm?item_id=1072.
- [4] 毛德操, 胡希明. Linux 内核源代码情景分析[M]. 杭州: 浙江大学出版社, 2001.
- [5] 藏 浩, 刘嘉勇, 方 勇. Linux 文件加/解密过滤驱动的设计与实现[J]. 计算机工程, 2009, 35(9): 144-146.

编辑 任吉慧

