

基于 PLB4 总线的 DDR3 控制器的设计与优化

李 哲,田 泽,王世中,郑 斐

(西安航空计算技术研究所 集成电路与微系统设计航空科技重点实验室,陕西 西安 710068)

摘 要:内存是计算机系统的信息存储部件,主设备与内存间信息交换的速度是影响系统性能的关键因素。PLB 总线是 IBM 提出的嵌入式总线标准,用于主设备与片内存储以及 PCIE、DMA、SRIO 等高速设备的互联,在 SoC 设计中使用广泛。该项目中 DDR3 作为从设备挂接到 PLB4 总线上,而选用的 DDR3 控制器 IP 核基于 HIF 接口,使用该 IP 核需要设计一套简单高效的总线桥逻辑,以满足系统访存性能要求。文中提出一种基于 PLB4 总线接口的 DDR3 控制器的设计方案,通过对数据流、控制流进行深入分析,采用请求合并、多级流水、数据预测、地址与控制信息复用、读数据乱序处理等方式,对访存效率影响较大的总线桥进行了速度和面积优化。仿真证明,优化后访存性能得到显著提升。

关键词:内存;性能;速度;面积;优化

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2016)03-0181-04

doi:10.3969/j.issn.1673-629X.2016.03.042

Design and Optimization of DDR3 Controller Based on PLB4 Bus

LI Zhe, TIAN Ze, WANG Shi-zhong, ZHENG Fei

(Aeronautics Science and Technology Key Laboratory of Integrate Circuit and Micro-system Design,
AVIC Computing Technique Research Institute, Xi'an 710068, China)

Abstract: Memory is the information storage component in computer systems. The message transmission speed between the master and the memory is the key factor to affect the system performance. PLB bus put forward by IBM is a embedded bus standard, which is used for the interconnection among masters, memory and other high-speed devices like PCIE, DMA, SRIO. It is wildly used in SoC design. This project takes DDR3 as a slave connected to the PLB4 bus, which has a host interface named HIF. So an high effective cross bus bridge logic is designed to interconnect each other and improve the memory access efficiency. A DDR3 controller solution based on the PLB4 bus interface is proposed, through analysis on the data and control flow, adopting the methods of request combination, multi-pipeline, data forecast, address and control information multiplexing, data reading out of order processing to optimize speed and size of the bridge logic which will influence the memory access delay. Simulation proves that after optimization the performance has been improved remarkably.

Key words: memory; performance; speed; size; optimization

0 引言

在高性能 SoC 设计中,高速的片上总线和高效的片上内存管理是不可或缺的组成部分。DDR3 是由 JEDEC 制定的新一代 DDR 内存技术标准。在现行的各种总线标准中,IBM 公司提出的 CoreConnect 总线结构具有突出的性能和效率优势,能够满足日益复杂化和高速化的 SoC 设计要求,成为业内标准总线之一。CoreConnect 总线采用分级结构,分别通过 PLB (Processor Local Bus) 总线和 OPB (On-chip Peripheral Bus) 总线将高速设备与低速设备分离,同时把读写控

制寄存器的总线操作独立开来,形成单独的 DCR (Device Control Register) 总线,大大减轻了数据总线的负载^[1]。它支持各种传输操作,总线/事务分离以及地址流水化等功能,最大限度提升总线带宽,从而提高系统性能^[2]。

1 总体结构设计

该项目需要设计满足 PLB4.6 协议标准的总线接口,满足 DDR3 协议规范的内存控制器。DDR3 控制器用于用户访问片外 DDR3 SDRAM 存储器芯片,它

收稿日期:2015-06-16

修回日期:2015-09-22

网络出版时间:2016-03-04

基金项目:中国航空科学基金(2015ZC51036)

作者简介:李 哲(1985-),男,工程师,研究方向为集成电路设计;田 泽,博士,研究员,中航首席技术专家,研究方向为 SoC 设计、VLSI 设计、嵌入式系统开发和应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160304.1508.006.html>

提供了用户访问外部存储器芯片的通道,支持片外 SDRAM 数据位宽度为 64 位。提供软件可访问的 DCR 接口寄存器,用于根据不同外存芯片进行大小、延迟等参数的配置,并能读取存储控制器的当前状态^[3]。

为提高 SoC 设计效率,采用 IP 复用技术,从经过硅验证的 PLB4 从设备 IP 核中剥离出 PLB4 从接口,选用成熟的 DDR3 标准 IP 核。需要做的主要工作就是从 PLB4 从接口到 DDR3 IP 核之间的转接逻辑,如图 1 所示,包括 PLB4 从接口模块,DDR3 控制器模块,DDR PHY 模块。其中 PLB4 从接口模块后端为 MCIF(Memory Control InterFace)总线接口,MCIF 总线是 IBM 公司定义的一种存储器控制接口,DDR3 模块分为控制器和 PHY 两大部分,DDR 控制器的主机接口为 HIF(Host InterFace)接口,控制器与 PHY 之间是标准的 DFI(DDR PHY Interface)接口。其中 DDRC 和 PHY 内部有可配置的寄存器,通过 DCR 接口访问。PLB4 从接口模块是 DDR3 控制器与 PLB4 总线的接口处理模块,满足 PLB4 的总线规范要求,可以接收、传输外部 PLB4 总线上的指令和数据,实现 PLB 主设备对 DDR 的访问。该从接口支持 64/128 位 PLB 主设备,支持的操作类型包括 1~16 字节单拍、4 字、8 字 line 操作和双字、4 字突发的 PLB 访问^[4-5]。其中 MCIF2HIF 模块,完成从 PLB4 接口到 DDRC 控制器核之间的命令和数据转换。

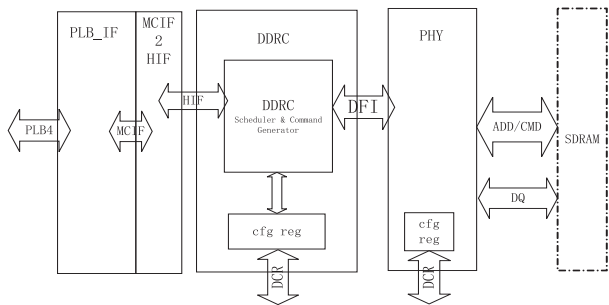


图 1 总体结构

2 MCIF2HIF 模块设计

由于 PLB 从接口后端为标准的 MCIF 接口,而 DDRC 控制器主机接口为 HIF 接口,需要该模块实现 MCIF 到 HIF 的数据率匹配、时序映射和转换功能。MCIF 与 HIF 接口采用同步时钟,该模块的转换效率对访存效率影响很大,应尽量减少转换延迟。

2.1 接口时序

MCIF 接口读、写操作时序如图 2 所示。

HIF 接口的读、写时序如图 3 所示。

HIF 在进行写操作时 co_ih_rxcmd_valid 有效同时写地址与写操作类型有效。ih_co_wdata_ptr_valid 有

效后表示读操作请求得到响应,然后可以发出写数据。写数据位宽为 256 位,一次写操作可以发出 2 个 256 位的写数据。co_wu_rxcmd_valid 为高表示写入数据有效,同写数据一同发出的还有写掩码信号 co_wu_rxdta_mask,用来表示传输的对应字节。ih_co_stall 为高时表示系统忙,直到 ih_co_stall 变低之前系统无法响应新的请求^[3]。

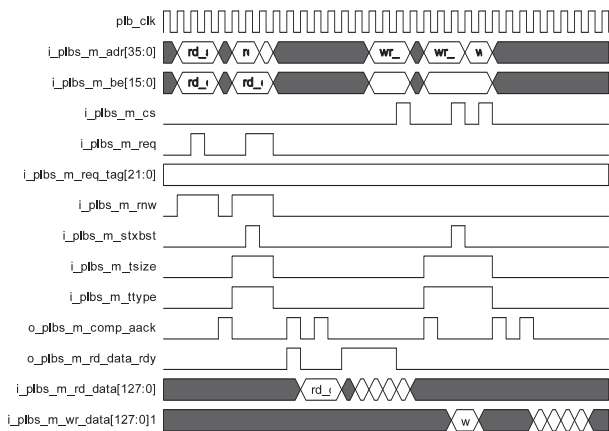


图 2 MCIF 接口时序

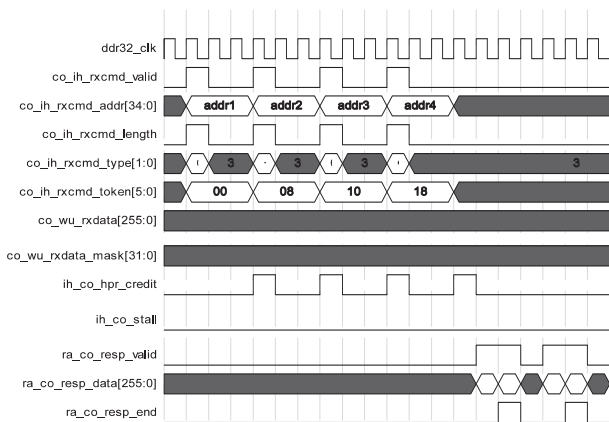


图 3 HIF 接口时序

HIF 在进行读操作时 co_ih_rxcmd_valid 有效同时读操作地址与读操作类型有效。读 token 信号 co_ih_rxcmd_token 有效,ih_co_hpr_credit 有效后表示读操作高优先级信用有效,然后可以发出写数据。读数据位宽为 256 位,一次读操作可以读出 2 个 256 位的读数据。ra_co_resp_valid 为高表示读出数据有效,同读数据一同读出的还有读 token 信号 ra_co_resp_token,用来表示读出数据的 token 信息^[3]。

2.2 数据及控制通路设计

由于 HIF 接口数据线为 256 bit,而 MCIF 接口数据线为 128 bit,因此在写操作时需要让 MCIF 接口尽量满负荷给 PLB 总线发送地址响应和请求写数据,并将写数据进行缓冲,对于读操作也需要对 HIF 返回的读数据进行缓冲。加快回地址响应的速度,可以让更多的 PLB 请求进来,并通过增加 compaack_fifo 与 wr_addr_fifo,将进来的 PLB 请求和配套信息缓存,保证不

要丢失请求及配置信息。由于 PLB 总线是地址和数据独立的,因此可以对写数据单独设计一个 FIFO 进行缓冲,与 addr_fifo 形成流水。

通过分析得出该模块需要使用 4 个 FIFO: compaack_fifo、wr_addr_fifo、wr_data_fifo、rddata_fifo。compaack_fifo 用于各个命令的调度、缓冲写请求并返回地址响应和写完成信号,可以将请求缓存并迅速接收下一条指令;wr_addr_fifo 用于缓冲读写地址信息与操作指令信息,需要设计一个命令控制字调度状态机控制来自 MCIF 接口的命令控制字信息调度;wr_data_fifo 用于缓冲写数据;rddata_fifo 用于缓冲读数据。

通过读 wr_addr_fifo,将其中的控制信息取出,同时,根据读出的控制信息适时地将 wr_data_fifo 中的写数据读出,通过判断 HIF 接口当前 wr_addr_fifo 的状态将写数据进行拼接并锁存。由于 HIF 写操作可能是 256 bit 或 512 bit,当为 256 bit 时,HIF 接口写操作为 1 拍,否则为 2 拍。HIF 读操作均为 2 拍,即一次返回 512 bit 数据,这时由于 DDRC 控制器读操作仅支持 BURST8 操作,存储总线宽度为 64 bit,所以每次读操作从 HIF 接口返回 512 bit 数据,转换逻辑需要根据控制信息从 512 bit 数据中适当的位置提取出正确的数据并按照预定的顺序返回给 MCIF 接口。控制信息的存储需要单独设计一个双口存储器,当发送 HIF 读数据请求时,将控制信息存储在该双口中,当 HIF 读数据返回后,需要读出双口中的控制信息,并根据控制信息取出相应位的数据返回给 MCIF 接口。

核心指令状态机转移图如图 4 所示。

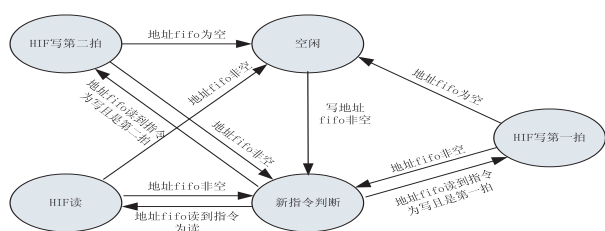


图4 核心指令状态机转移图

2.3 乱序处理设计

由于 PLB4 总线读操作不支持乱序^[2],而 DDRC 是支持乱序的,即对 HIF 接口来说,由于每次 HIF 读请求需要对应一个 token,而不同 token 的读请求返回给 HIF 接口的读数据可能是乱序返回的。因此,不能直接将 HIF 读数据返回给 MCIF 接口。

设计一个 32 位的 flag 寄存器和深度为 32 的双口存储器(先暂定为 32 深度,通过仿真评估后可能调整),根据 HIF 接口时序要求,当发送 HIF 读请求时,需要同时发送 co_ih_rxcmd_token 信号,简称为 token 信号。当控制指令状态机处于新指令判断状态时,将当前空闲的 flag 位的值赋给 token,并将 flag 寄存器中

对应位置 1,flag 采用从低位到高位依次顺序分配,但只有分配到 31 位,且 flag[0] 为 0 且已经被读走时,才能重新分配 flag。否则等待 flag 空闲后再发送读请求。当 DDRC 返回读数据时,由于返回的读数据的 token 与之前发送的读请求的 token 是一致的,所以将返回数据的 token 作为双口存储器的地址,将对应数据写入双口中,并将 flag 清 0。同时设计一个 32 位的 dpram_flag 寄存器,用来存储返回的 token 值。由于前面发送 HIF 读请求的 token 的顺序是确定的,因此期望返回的 token 的顺序也是确定的。当返回一个 token 时,将其对应数据写入双口的 token 值对应的地址中,同时将该 token 对应的 dpram_flag 置 1,并将该 token 与期望返回的值比较,相等则下一拍将该笔数据从双口中取出返回给 MCIF 接口,并清除 dpram_flag 寄存器的对应位,否则,将期望值加 1(当加到 31 时重新返回 0),等待下一次返回读数据。直到期望值与后面返回的数据相等时,将该笔数据从双口中读出,从而保证返回给 PLB 总线的读数据的顺序。

另外,由于两笔 HIF 读数据返回间隔的不确定性,数据在通过双口读出后需要经过一个 rddata_fifo 缓存后输出到 MCIF 接口。设计功能结构图如图 5 所示。

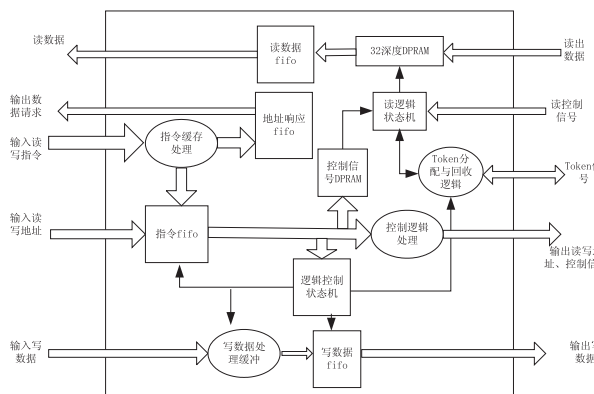


图5 功能结构图

3 设计优化

3.1 对 BURST 操作的优化

由于 BURST 操作通常是带有连续性的,因此 MCIF 接口上相邻的两次读或写请求具有强相关性。可将其地址和控制信息进行合并,两次仅有一个入 FIFO 队列,从而提高效率和 FIFO 利用率。对于前面预判可合并,后面实际判断不能合并的,最终会分成两次请求入队^[6-7]。

3.2 获取写数据的加速

如果上一步进来的 PLB 请求是写请求,则需要在回 compaack 之后向 PLB 从接口要写数据。首先对 compaack_fifo 和 addr_fifo 进行分工调整,由于 compaack_fifo 的最终目的是要写数据,因此只将写请求及

配置信息入队 compaack_fifo,而读请求不入队^[8-9]。另外,通过对状态机进行优化,在回 compaack 同时向 PLB 接口发起写数据请求,并在将写数据缓冲到 wrdata_fifo 的同时,请求下一次的写数据,从而加速写数据的获取。这里写入 wrdata_fifo 的数据已经根据从 compaack_fifo 读出的信息,按照 HIF 接口的要求重组好,根据后面流程直接从 wrdata_fifo 读出后输出到 HIF 接口,实现零换乘,省去拼接和产生规定数据的时间^[10-11]。

3.3 对读、写请求的优化

由于 wr_addr_fifo 作为该模块控制信息来源,存储了所有来自 PLB4 的总线请求及配套信息。控制调度核心命令状态机通过对该 FIFO 的读出信息来控制向 HIF 发送读、写请求及写数据。缩短两次读取 wr_addr_fifo 的间隔,对读操作,由于不需要数据,因此发完 HIF 读请求后,可以快速切换到下一次对 wr_addr_fifo 的读取状态。对于写操作,由于在发完 HIF 写请求之后,既要等待 HIF 接口的授权,又要等待写数据准备好。仿真发现此处延迟较大,若增加一个 hif_wr_fifo,缓冲 wr_addr_fifo 的写控制信息,不用等待授权和写数据准备好,核心命令状态机就可转入下次读 wr_addr_fifo 状态。使 HIF 写命令和写数据流水,从而降低

延迟^[12-13]。

3.4 利用 token 进行速度和面积优化

co_ih_rxcmd_token 信号是 DDRC 控制器中 CAM 存储器的地址,根据 token 发送和接收不会改变其值的特点,可以将控制信息通过 token 进行传递和回收。仿真发现,HIF 接口最坏情况下,堵在双口中的读数据最多为 4 个,考虑设计余量,因此将双口深度调整为 8,同时将 flag 和 dpam_flag 寄存器宽度也调整为 8。由于 HIF 接口的 co_ih_rxcmd_token 信号宽度固定为 6 bit,因此可以考虑将 token 低三位负责传递控制信息,高三位作为 CAM 和双口的地址,由于 token 分配方式一轮操作内不会重复(前一个不释放,后一个不会重用),因此 CAM 地址不会出现冲突。采用 token 为读请求来传递控制信息,可省去一个存储控制信息的双口存储器,同时将读数据双口及 rddata_fifo 深度均从 32 减为 8,节省了面积,简化了逻辑设计,降低了双口读写延迟,避免双口两端同时写操作的冲突风险^[14]。

4 仿真结果

仿真环境是在 PLB 总线及 DDRC 时钟均为 200 MHz 的情况下。将优化前与优化后的设计进行仿真,统计多种操作的访存延迟,结果如表 1 所示。

表 1 优化前后对比表

PLB 操作类型		PLB 操作名称	优化前/ μ s	优化后/ μ s	提升百分比/%
4 字 burst 操作	连续发出 32 次 4 字 burst 长度为 8 的写操作(4 kB)		2.064	1.714	16.957
	连续发出 32 次 4 字 burst 长度为 8 的读操作(4 kB)		3.863	2.903	24.851
	交替发出 64 次 4 字 burst 长度为 8 的写操作与读操作(4 kB/4 kB)		4.284	3.573	16.597
8 字 line	连续发出 32 次 8 字 line 的写操作(1 kB)		1.084	0.749	30.904
	连续发出 32 次 8 字 line 的读操作(1 kB)		1.828	1.348	26.258
	交替发出 64 次 8 字 line 的写操作与读操作(1 kB/1 kB)		2.323	2.188	5.811
8 bit 单拍操作	连续发出 128 次 8 bit 单拍写操作(128 Byte)		3.959	2.349	40.667
	连续发出 128 次 8 bit 单拍读操作(128 Byte)		6.933	5.013	27.694
	交替发出 256 次 8 bit 单拍写操作与读操作(128 Byte/128 Byte)		10.388	9.793	5.728

5 结束语

文中论述了 PLB4DDR3 控制器的总体结构设计,重点研究了 MCIF2HIF 转换接口设计优化。仿真结果表明,优化后访存效率得到明显提升。文中对片上内存设计及优化具有一定参考价值。

参考文献:

[1] 王宏亮,毛永毅,张宏君. 基于 CoreConnect 的 OPB SPI 接口设计与实现[J]. 测控技术,2013,32(5):72-76.

[2] IBM. 128-bitprocessor local bus architecture specifications version 4.6[M/OL]. [s. l.]:IBM Inc. Press. 2004. http://

www.chips.ibm.com.

[3] Design ware cores enhanced universal DDR memory controller databook[M]. [s. l.]:Synopsys Press,2014:189-193.

[4] 陈超文,彭国杰,王忆文,等. 基于 PLB 总线的 NOR FLASH 控制器设计[J]. 微电子学与计算机,2014,31(5):83-86.

[5] 卢俊,颜哲,田泽. 基于 PLB 双总线高速存储接口的设计与实现[J]. 计算机技术与发展,2015,25(4):233-237.

[6] 陈卓,杨爱良,王骥. 基于 PLB 总线的多通道 SGDMA 设计[J]. 航空电子技术,2009,40(1):12-15.

[7] 吴从中,项磊,蒋建国. 基于 PLB 总线的 H. 264 整数变换量化软核的设计[J]. 电子技术应用,2008,34(10):35-38.

(下转第 189 页)

IPv6。与 IPv4 相比,IPv6 具有地址空间更大、路由表更小、安全性更高等多项优势。因此随着当前及未来 IPv6 网络的普及,IPv6 地址势必将逐渐取代 IPv4 地址。但是 IPv6 地址由 128 位二进制数码表示,相对于 IPv4 地址却更加难以记忆。为了能够使用户更方便地访问互联网,而不用去记住那些能够被机器直接读取的 IPv6 地址数字串,解决 IPv6 地址信息不便于记忆的问题,网络中离不开 IPv6 的域名解析技术(即 DNSv6 技术)。通过主机名,最终得到该主机名对应的 IPv6 地址^[15]。

DNSv6 代理技术生成映射转发表,建立客户端发起的 DNSv6 请求和服务器响应的 DNSv6 应答之间的映射,通过正确转发 DNSv6 请求和应答,使客户端最终获取 DNS 查询结果,完成域名解析,而且实现过程简单。

DNSv6 代理技术具有如下优点:

(1)简化网络管理。使用 DNSv6 代理,如果 DNS 服务器地址等信息发生变化,只需改变 DNSv6 代理上的配置,无需改变局域网内每个 DNS 客户端的配置,从而简化了网络的管理,提高了网络的稳定性。

(2)减少网络传输流量,提高解析效率。利用 DNSv6 代理中的缓存可以大大地节约域名解析的时间,减少 DNSv6 代理上行网络中的传输流量,提升网络传输速度,同时利用缓存快速完成域名解析,也可提高网站的访问速度。

(3)避免真实服务器遭受攻击。客户往往会使用域名来访问网站资源,通过 DNSv6 代理技术,真实的 DNS 服务器便对 DNS 客户端不可见,这样可以避免真实的 DNS 服务器遭受客户端的恶意攻击,达到保护真实 DNS 服务器安全的目的,提高网络的安全性。

参考文献:

[1] 百度百科. IPv6 [DB/OL]. 2015-02-27. <http://baike.baidu.com/link?url=HCQ7hhqOmptyjQojI6UZYf0bB8VoF3>

(上接第 184 页)

[8] 苏鹏,卞春江,张磊. PowerPC 处理器 MPC8548E 的 DDR2 接口实现[J]. 微计算机信息,2010,26(12-2):174-176.

[9] 刘宁宁,田泽,裴希杰. 基于 CoreConnect 总线的 Nand Flash 控制器设计[J]. 计算机应用,2014,34(S1):327-329.

[10] IBM. Device control register bus 3.5 architecture specifications[M]. Armonk:IBM,2006.

[11] 马秦生,魏翠,孙力军,等. 嵌入式 SoC 总线分析与研究

Rg2np3yEaacrc20JGCyC9Avo43wkaSCQ4J2HQn7UNU9ECY0iNmymswJa.

[2] 王大阜. 基于 Linux 平台下的 IPv6 DNS 系统的实现[J]. 计算机与信息技术,2009(4):72-73.

[3] Thomson S,Huitema C. DNS extensions to support IP version 6[S]. RFC 1886,1995.

[4] Crawford M,Huitema C. DNS extensions to support IPv6 address aggregation and renumbering[S]. RFC 2874,2000.

[5] 华为技术有限公司. AR150&AR160&AR200&AR1200&AR200&AR3200 产品文档[EB/OL]. 2014. <http://support.huawei.com/ehedex/hdx.do?docid=DOC1000043775&lang=zh>.

[6] 武浦军,余镇危,张英,等. IPv6 DNS 代理软件的设计与实现[J]. 计算机工程,2005,31(11):211-213.

[7] 冯中华. VFP 系统动态数据报表程序设计[J]. 计算机工程与应用,2003,39(2):194-195.

[8] 盖一凡,吴国升. IPv6 地址以及 IPv6 DNS[J]. 通信管理技术,2005(1):50-50.

[9] 张惠姝. DNS 安全防护关键技术研究[J]. 电信技术,2015(9):99-104.

[10] 孔政,姜秀柱. DNS 欺骗原理及其防御方案[J]. 计算机工程,2010,36(3):125-127.

[11] 杨驩宇. 高并发环境中路由器平台上 DNS 代理的设计与实现[D]. 武汉:华中科技大学,2012.

[12] 孙文芳,冯坚. 智能 DNS 系统建设[J]. 有线电视技术,2015(7):35-37.

[13] Cisco Systems, Inc. IP addressing: DNS configuration guide, Cisco IOS release 12.4T[EB/OL]. 2011. http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dns/configuration/12-4t/dns-12-4t-book.pdf.

[14] 百度百科. DNS(域名系统)[DB/OL]. 2014-12-25. http://baike.baidu.com/link?url=E0XaYtVkbmleQjC3pK-sPr9j5t6JqsEj45YS7srRVpnbQ00YUVVJhmiIYX9PzBbb68KCQqhFlx_TZfb1qYZVFs-mLoEySii0OK5tJYtFGA4i.

[15] 百度文库. 代理服务器的作用[DB/OL]. 2012-08-25. <http://wenku.baidu.com/view/a97bda345a8102d276a22f21.html>.

[J]. 中国集成电路,2007,16(3):45-49.

[12] 潘波,杨根庆,孙宁,等. 基于多级片上总线的并行图像处理系统设计[J]. 计算机应用研究,2008,25(7):2208-2209.

[13] IBM Inc. Processor local bus functional model toolkit user's manual version 4.7[M]. [s.l.]:IBM Inc. Press,2002:214-215.

[14] Xilinx Inc. BFM simulation tutorial[EB/OL]. 2006-07-18. <http://www.xilinx.com>.