基于以太网的 DSP 网络加载技术研究

沈发江
（江苏自动化研究所，江苏 连云港 222006）

摘 要：提出了一种基于以太网的 DSP 程序加载技术，介绍了基于以太网的加载方法，包括网络接口控制、HPI 接口控制及 Flash 程序控制等关键技术，并给出了该方法在工程中的实际应用。与传统的加载技术相比，该技术灵活方便，可脱离仿真器实现远程、大容量的程序代码加载，快速完成 DSP 系统的软件更新。

关键词：以太网；TMS320C6713；HPI；引导加载
中图分类号：TP391 **文献标识码**：A **文章编号**：1674-7720(2013)07-0058-03

Technology research for DSP bootloader based on Ethernet

Shen Fajiang
(Jiangsu Automation Research Institute, Lianyungang 222006, China)

Abstract: A technology for DSP bootloader based on Ethernet is presented. Bootloader methods based on Ethernet are studied, including network interface, HPI interface operations and Flash storage model design. A practical application in engineering systems is given. Compared with conventional bootloader technologies, this technology is more flexible and convenient for a long-range capacity program code load without an emulator so as to update the software rapidly.

Key words: Ethernet; TMS320C6713; HPI; bootloader

随着以太网技术和 DSP 技术的迅猛发展，基于网络的 DSP 设备已成为仪器仪表、工业控制和远程测控的重要发展方向。在以 DSP 为核心的应用系统中，程序代码的引导加载具有至关重要的作用。传统的 DSP 程序加载是通过硬件仿真器来完成的，但在工程应用中，一旦系统组装为成品后，再对系统进行软件更新和维护时，传统的加载方式就显得十分不方便，而且该方法不能解决程序代码的远程加载问题。因此，需要一种更加灵活、高效的程序加载方式，而基于网络的 DSP 软件更新就成为一个新的热点。

本文介绍了 DSP 程序加载的基本原理，设计了 DSP 加 MicroBlaze 软处理器的系统，并以 TI 公司的 DSP 为例，提出了一种基于以太网的 DSP 程序加载技术，利用网络通信实现 DSP 的动态加载及 DSP 资源的远程访问，提高了软件更新的效率及远程访问的便利性。

1 DSP 程序网络加载原理概述

要实现 DSP 程序的网络加载，需要解决的关键技术有数据的网络发送和接收，DSP 的 HPI 主机接口控制，Flash 存储器的读写以及程序文件的网络格式转换。具体的实现过程为：首先上位机工具将编译好的程序输出文件进行网络分包转换，按照网络包的格式将代码传送给板卡的主控制器，主控制器 MicroBlaze 接收到数据后通过 DSP 的 HPI 接口把程序写到内存中，完成程序代码的引导加载。同时，也可以把接收到的网络数据通过主控制器 MicroBlaze 写入到外部 Flash 存储器，再次上电后完成程序的加载。

要实现网络数据的发送和接收，必须要使系统能够实现网络功能，按照一定的协议进行网络数据包的组织。目前，应用最广泛的网络协议是 TCP/IP 协议，但由于控制器 MicroBlaze 自身资源有限，无法实现标准的 TCP/IP 协议，再加上该系统网络环境较简单，因此本设计采用效率较高且协议简单的 UDP 协议来发送网络数据。将 UDP 协议嵌入到主控制器 MicroBlaze 中，即可实现系统的网络接口功能。

HPI 接口是 TI 公司 DSP 留给外部主控制器访问其资源的接口，通过 HPI 接口，外部主控制器能够实现

DSP 程序的引导。具体过程为：当系统上电后，DSP 的 CPU 处在复位状态，外设部分已工作正常，外部主控制器通过 HPI 接口访问 DSP 的所有资源，当主控制器完成所有加载工作后，把 CPU 从复位状态唤醒，开始从地址 0 处执行程序。

通常 DSP 编译器编译后的程序文件一般都是 COFF 格式，它由多个数据段组成。其中.text 通常包含可执行代码，.data 通常包含已初始化的数据，.bss 通常为未初始化的数据保留空间。通过对 COFF 文件结构的分析，读取 DSP 编译产生的.out 文件，根据文件本身携带的信息，直接提取生成可供下载的二进制文件，再把二进制文件按照 UDP 协议分包后通过网络发送到系统的主控制器，即可完成 DSP 的网络加载。

2 系统设计与实现

2.1 硬件设计

系统采用基于 DSP 的主从式双 CPU，如图 1 所示。选用 Xilinx 公司的 V5 系列 FPGA 作为系统的控制单元，内部运行嵌入式 MicroBlaze 处理器用来作为主机，负责网络数据的接收、解析、HPI 接口的控制、Flash 存储器的读写等工作。TMS320C6713 主要完成和外部信号的接口，完成信号处理算法的实现，可以通过网络对其算法程序进行不断的修改，以达到最佳的信号处理效果。从图 1 可以看出，DSP 与 FPGA 之间通过 HPI 接口连接，DSP 还负责模拟信号的采集和输出。

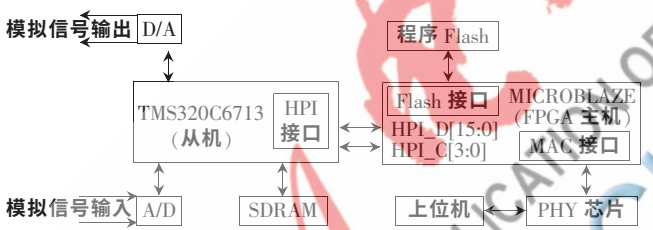

图 1 基于 DSP 的主从式系统框图

2.1.1 网络接口设计

实现系统程序的网络加载需要解决的关键技术之一是模块的网络接口设计，使该模块能够接入网络。在本系统中，为了提高灵活性并减少成本，网络接口的设计直接通过 FPGA 对物理层器件的控制实现，然后由 MicroBlaze 软核处理器完成上层协议的封装和解析，这样提供了 FPGA 对网络的灵活控制，通过修改软核处理器的程序即可完成不同协议的封装。

FPGA 通过内部总线接收网络数据的逻辑框图如图 2 所示。FPGA 通过内部总线接收网络数据时，数据从物理器件 MII 接口的 RX_D[0:3]进入 FPGA，根据 MII 的接收时序，检测到网络数据帧中的帧前序和帧起始符后，触发地址产生逻辑，将接收到的 RX_D[0:3]和 RX_EN 信号转换为 4 bit 的数据写入到 2 KB 的双端口 RAM 中。双端口 RAM 是在 FPGA 内部生成的，其 A 端口为 4 bit 的数据线，B 端口为 32 bit 的数据线，用于 FPGA 内 MII 接收与 HPI 接口间进行数据转换。从双端口 RAM 端口 A 写入数据，由 B 端口读出，经过转换后与 DSP 的 HPI 接口相连。在对 DSP 的 HPI 接口进行配置后，网络接收到的数据在 FPGA 软核处理器的控制下可以写入到 DSP 的内存中，从而实现网络数据的接收。

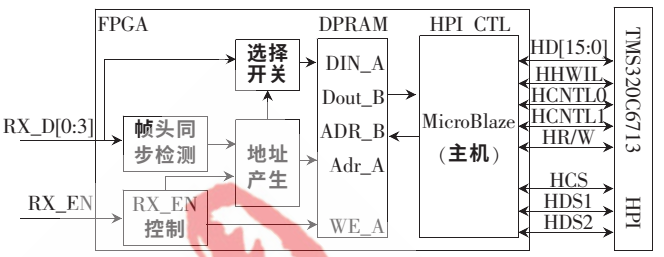

图 2 网络接口和 HPI 接口框图

2.1.2 HPI 接口设计

TMS320C6713 的 HPI 端口为一个 16 bit 宽的并行端口，此接口为主机与 DSP 的数据交换提供了最方便有效的方案。主机掌管着该端口的主控权，可通过 HPI 接口直接访问 DSP 的存储空间和外围设备。主机对 HPI 内存的访问通过 HPIC（HPI 控制寄存器）、HPIA（HPI 地址寄存器）和 HPID（HPI 数据寄存器）3 个专用的 DSP 寄存器来实现。主机可以对这 3 个寄存器进行读写，而 DSP 只能对 HPIC 进行访问。HPID 中存放的是主机从存储器中读取的数据，或者是主机向 DSP 存储空间写入的数据。HPIA 中存放的是主机访问的地址，HPIC 中存放的是控制信息。

在 FPGA 中做一个 HPI 的控制核，可以产生对 HPI 接口的控制时序，包括地址线和读写信号线等控制逻辑，在软核处理器 MicroBlaze 上编写 HPI 接口的相应驱动程序，包括 HPI 接口的读、写操作及复位操作等函数。用户通过调用某个 HPI 驱动函数，驱动函数转化为底层的时序控制逻辑，就可以完成对 HPI 接口的访问。

2.2 软件设计

硬件是基础，软件是核心，硬件必须搭配稳定、高效的软件才能组成一个完整的系统。根据硬件的层次结构，加载系统的软件主要完成以下功能部分的设计：上位机控制程序、上位机网口程序、FPGA 软核处理器网络驱动程序以及软核处理器服务程序。

2.2.1 上位机软件设计

上位机软件是用户实现 DSP 网络加载的平台，采用 C++ Builder 6.0 设计完成。该开发工具内部集成了 UDP 协议的通信控件，只要再经过简单的封装就可以形成 UDP 协议的网络驱动程序，不需要底层进行开发。上位机控制程序对外采用 MDI 界面，主要实现程序的加载、烧录以及内存的查看等功能。

2.2.2 底层软件设计

底层驱动软件的核心是网络接口的驱动程序。考虑到 FPGA 实现网络协议的简单性和实时性的要求，选用 UDP 协议作为传输协议，网络数据封装必然要满足 UDP

协议的帧格式。数据帧中包含了 7 B 的帧前序和 1 B 的帧起始符。完整的 UDP 协议的帧结构如表 1 所示。由表可知，UDP 协议是在 IP 协议和 MAC 帧格式的基础上封装的。

表 1 以太网 UDP 帧结构表

帧前序(7 B)	帧起始符(1 B)	
目的 MAC 地址(6 B)	源 MAC 地址(6 B)	长度/类型(2 B)
	UDP 检验和	
源端口		目的端口
UDP 长度		UDP 检验和
UDP 数据区		帧校验(4 B)

网络接口的驱动程序就是按照表 1 的数据包组成结构将接收到的数据进行解析，分解出数据。驱动程序包括 UDP 数据的发送、接收和 ARP 数据的回复等。

2.3 网络加载流程

在实现了网络数据的收、发以及 HPI 接口的控制之后，就可以实现 DSP 程序的网络加载。网络加载的流程图如图 3 所示，主要包括加载网络硬件的驱动、启动 UDP 服务、接收网络数据、向 DSP 内存地址写入数据、判断数据是否接收完成、数据接收完成后唤醒 DSP 以及开始执行下载的程序代码等过程。

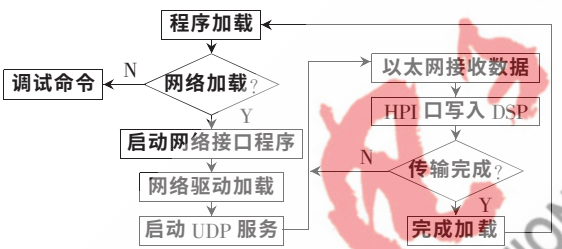

图 3 网络加载数据流程框图

3 系统设计中的关键技术

在系统设计中，主要涉及：网络接口和 HPI 接口两个接口，对这两个接口的访问控制非常重要。此外，HPI 程序加载代码的生成也非常关键。

3.1 网络接口的实现

网络接口采用 FPGA 编写 IP 核的方式来实现，主要是 MAC 层控制器的设计，其既可以集成于网络终端设备中实现网络接入，同时又是开发网桥、交换机等网络互联的设备。整个 MAC 分为 10 个模块，其构成图如图 4 所示。每个模块都完成相对独立的一系列功能，各模块的功能如下。

(1)PHY 接口模块。根据 PHY 的工作模式，将 MII 接口不同的数据位宽进行转换，从而提供给上层发送模块和接收模块统一的位宽。

(2)发送模块。其主要功能是按照 CSAM 机制完成信道接入控制，以及将上层的待发送数据封装成以太网帧的格式，为其添加前导码、帧起始定界符、PAD 和 CRC 校验字段并发出。

(3)接收模块。进行单播/组播/广播帧的过滤，进行 CRC 校验，滤除帧碎片，把合法的帧传输至上层，并在接收结束后向上层报告帧接收的状态。

(4)流量控制模块。完成全双工下流量控制的功能。

(5)发送缓存/接收缓存。实现对发送/接收帧缓存的管理。

(6)AHB 总线接口。它为外部总线接口，完成与 MicroBlaze 软核及其他 AHB 接口单元的通信。

(7)MII 管理模块。完成对 PHY 工作模式的监控和设置。

(8)时钟管理模块。其产生不同工作模式下各个模块的工作时钟和时钟使能信号。

(9)寄存器和中断模块。负责系统模式配置及中断管理。

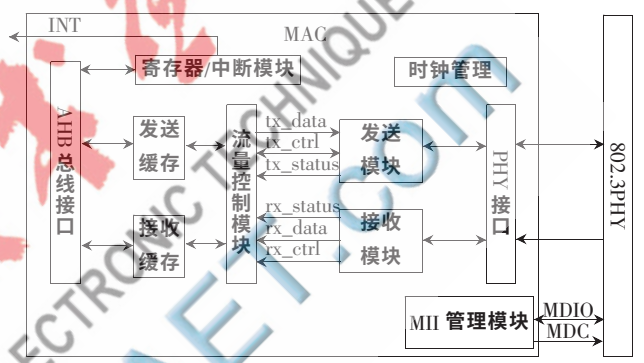

图 4 网络接口控制器实现框图

3.2 HPI 接口的访问控制

DSP 引导配置引脚 HD[4:3]决定了 TMS320C6713 的引导加载模式，将 DSP 的 HPI 接口引脚 HD[4:3]通过下拉电阻接地，即选择 HPI 加载。HPI 接口主要信号有 HD[15:0]，并行双向数据地址共用总线，主要传输控制寄存器的值、初始化访问地址及数据；HCNTL[1:0]主要控制当前访问的是哪一个寄存器，当控制线不同译码时可以访问不同的寄存器；HHWIL 信号用于区分数据线上传输的是数据的高 16 bit 还是低 16 bit；HR/$\overline{W}$ 信号用于读/写选择，高电平表示主机读，低电平表示主机写；HR$\overline{D}$Y 信号用于表明 HPI 是否已经准备好传输数据，其作用是在接口时序上插入等待周期；HI$\overline{N}$T 信号表示向主机发出的中断，当 DSP 需要和主机通信时，将该位置 1，主机就会响应中断。根据上述的信号功能，MicroBlaze 处理器的局部总线和 HPI 的数据总线相连，MicroBlaze 处理器的 3 bit 地址线和 HPI 接口的控制线相连，MicroBlaze 处理器映射的控制寄存器的控制位连接到片选信号。通过上述信号线的物理连接，在软件访问具体的地址空间时就可以译码出 HPI 接口读写的一系列片选、控制信号，完成对 HPI 接口的访问。

3.3 HPI 程序加载代码生成

在 CCS 开发环境下生成的目标文件是“*.out”，即通

用目标文件格式(COFF)。该文件可以在 CCS 开发环境下通过仿真器下载到 DSP 目标板中运行调试,但代码文件并不是 DSP 中实际运行的程序代码,需要 Hex 工具进行转换,然后自编程序将可执行的程序代码提取处理,可执行的程序代码作为数据段封装成帧,通过以太网加载到 TMS320C6713 中。由于 HPI 引导完成后,DSP 是从地址 0 开始执行的,而 DSP 程序的入口是 c_int00,因此要在地址 0 处添加一条跳转指令到程序入口。在编程时要特别注意需将中断代码段放在存储空间的 0000H~01FFH。

本文提出了一种基于以太网的 DSP 程序加载技术,该技术在多 DSP 系统的程序加载及远程控制系统的软件更新等工程应用领域具有独特的优越性和较好的应用前景。该方法已经在某 DSP 信号处理系统中得到应用。实际测试表明,该加载技术灵活可靠,完全满足现场调试的需要。

参考文献

[1] Texas Instruments. TMS320C6000 DSP host port interface reference guide[Z]. 2006.

[2] 左幔睿,刘永清,张傲华,等.基于以太网的 DSP 远程加载技术研究 [J]. 单片机与嵌入式系统应用,2012(5):24-26.

[3] 夏军营,乔纯捷,王刚,等.基于以太网接口的多 DSP 监控技术研究[J].计算机测量与控制,2007,15(7):913-915.

[4] 张晓亮.基于 SOPC 以太网技术的研究与实现[D].大连:大连理工大学,2007.

[5] 江华. 基于 TigerSHARC 的可配置 DSP 系统动态加载技术[D].西安:西安电子科技大学,2006.

(收稿日期:2012-12-12)

作者简介:

沈发江,男,1981 年生,硕士,工程师,主要研究方向:多媒体显示及信号处理。