

# 一种基于以太网加载 FPGA 和 DSP 的实现方法

吴湘宁, 江 瑾, 冯俊涛

(广州海格通信集团股份有限公司, 广东 广州 510663)

**摘要:** 介绍了脱离仿真器直接使用外部计算机通过网口进行程序代码加载的基本原理, 讨论分析了网络接口、FPGA 接口和 HPI 接口的访问控制等关键技术。详述了在包含 CPU、FPGA 和 DSP 的复杂系统设计方案中基于以太网加载 FPGA 和 DSP 的实现。该技术在系统工程化的应用中具有很好的前景。

**关键词:** CPU; 以太网; FPGA; DSP; HPI; 引导加载

中图分类号: TN911.72

文献标识码: B

文章编号: 1674-7720(2011)18-0055-04

## A realization for FPGA and DSP loading based on Ethernet

Wu Xiangning, Jiang Jin, Feng Juntao

(Guangzhou Haige Communications Group Co., Ltd., Guangzhou 510663, China)

**Abstract:** This paper describes the basic principles of the method that using an external computer through the network interface directly to load the program code without an emulator, discusses and analyzes the access control for the network interfaces, FPGA interface and HPI interface and other key technologies. This paper also describes the detail of the realization for FPGA and DSP loading based on Ethernet which is a complex systems that contains CPU, FPGA and DSP. and it has a good prospect for the application of systems engineering.

**Key words:** CPU; Ethernet; FPGA; DSP; HPI; boot loader

随着嵌入式技术的发展, 嵌入式系统逐渐演变为由微处理器、大规模集成电路、软件技术和各种具体的行业应用技术相结合的结果。嵌入式设备已经在航空航天、通信、电子等行业得到广泛应用。在以微处理器为核心的嵌入式应用系统中, 对其中的 FPGA 和 DSP 的传统加载都是通过硬件仿真器来完成的。这在产品的初研阶段非常适用, 然而在实际工程应用中各种产品组装成一个庞杂的系统时(如需对系统进行软件更新和维护时), 传统的加载方式就显得极其不方便, 程序代码的引导加载就需要一种更加灵活、高效的方法来实现, 而基于以太网加载 FPGA 和 DSP 的实现方法能很好地解决这个难题。

本文介绍了基于以太网加载 FPGA 和 DSP 的基本原理, 并通过该技术在某嵌入式主控设计中的应用详述了在包含 CPU、FPGA 和 DSP 的复杂系统设计方案中基于以太网加载 FPGA 和 DSP 的实现, 讨论分析了网络接口、FPGA 接口和 HPI 接口的访问控制等关键技术。

### 1 基于以太网加载 FPGA 和 DSP 的基本原理

基于以太网加载 FPGA 和 DSP 的实现过程为: 首先, 外部计算机通过网络接口将代码传送给微处理器, 并存储到微处理器的 Flash 存储器模块; 然后, 由微处理器分别通过 FPGA 的从串加载模式和 DSP 的主机引导加载(HPI 接口)实现对 FPGA 和 DSP 程序代码的引导加载。

基于 TCP/IP 协议的以太网是目前使用最广泛的网络。网络协议通常分不同层次进行开发, 每一层负责不同的通信功能。嵌入式设备通常根据微处理器的结构特点和系统需要实现的功能, 将 TCP/IP 协议嵌入到微处理器中, 从而实现嵌入式系统的网络接口功能<sup>[1-2]</sup>。

#### 1.1 FPGA 加载

FPGA 常见的配置文件的加载模式有 5 种: 主串加载模式、从串加载模式、主并加载模式、从并加载模式及 JTAG 加载模式。其中, JTAG 加载模式在开发调试阶段使用。其余 4 种加载模式可分为串行加载方式和并行加载方式, 而串行加载方式和并行加载方式都有主、从两

《微型机与应用》2011 年 第 30 卷 第 18 期

## 网络与通信 Network and Communication

种模式。由于主模式的加载同步时钟(CCLK)由 FPGA 提供,从模式的加载同步时钟(CCLK)由外部时钟源或者外部控制信号提供,为了方便微处理器控制加载过程,一般选择使用从串模式。

### 1.2 DSP 加载

DSP 常见的三种引导加载方式为仿真器引导加载、ROM 引导加载、主机引导加载。

(1)仿真器引导主要用于调试阶段,是一种特殊的主机引导加载方式。

(2)ROM 引导加载:在 DSP 外部复位信号被释放后,芯片处于内部复位保持状态,DSP 芯片自动地通过 EDMA/QDMA 将外部 ROM 中的 1 KB 程序搬入地址 0 处。内部复位完成后,DSP 开始从地址 0 处执行指令。

(3)主机引导加载:DSP 上电复位后,I/O 部分将处于激活状态,可以进行正常的通信,但 DSP 的内核仍处于复位状态,一直等待主机向 HPI 控制寄存器中的 DSPINT 位置 1。这样,主机就可以通过 HPI 接口访问 DSP 的整个存储空间,对它们进行初始化。完成相关设置后,向 DSPINT 位写入 1,DSP 内核结束复位状态,执行地址 0 处的指令。

## 2 嵌入式数字处理系统设计与实现

### 2.1 硬件设计

嵌入式数字处理系统采用微处理器 PowerPC 作为主控 CPU,主要执行网络控制等各种控制任务,其程序代码相对固定,存储在 Flash 中。如图 1 所示,选用 Xilinx 公司的 XC3S1000 和 XC4VSX35 这两种 FPGA 分别做逻辑和变频<sup>[3-4]</sup>,选用两片 TI 公司的 DSP TMS320C6416 芯片完成数据处理与算法实现,这些 FPGA 和 DSP 的程序代码需实时更新<sup>[5]</sup>。

#### (1)网络接口设计

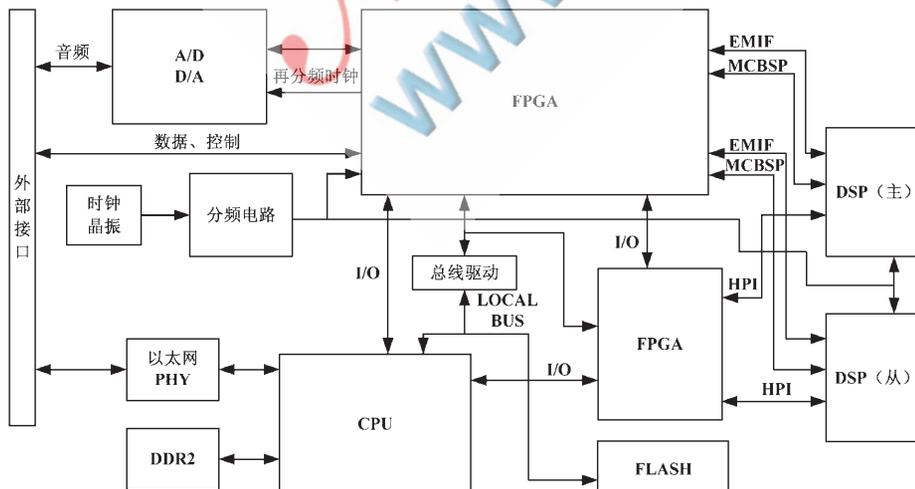


图 1 嵌入式数字处理系统框图

以太网 PHY 采用 Marvell 公司 PHY 协议层千兆网卡 88E1111<sup>[6]</sup>。88E1111 是高性能千兆以太网收发芯片,可以完成所有 PHY(物理层)功能,支持全双工或半双工的 10/100/1 000 Mb/s 三速以太网,支持 8 bit 数据总线。CPU 和以太网 PHY、FPGA 及 DSP 的接口基本电路如图 2 所示。CPU 的网络接口引脚和以太网 PHY 的网络接口相连接,通过硬件配置模式 HWCFG MODE[3:0]来确定 88E1111 的操作模式,从而确定需要连接的引脚,如图 2 所示,该设计选择了 RGMII 模式,以太网 PHY 通过网络隔离滤波器 HX5008 与外部以太网接口相连。

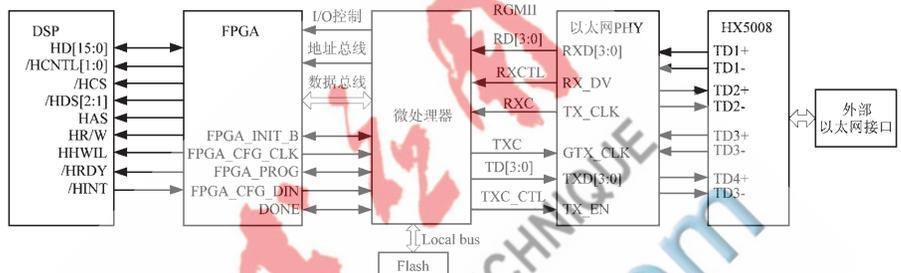


图 2 网络接口、FPGA 接口和 HPI 接口电路

#### (2)FPGA 接口设计

在从串模式下进行 FPGA 程序加载仅需要使用 CCLK、PROG\_B、INIT\_B、DONE、DIN 这 5 个信号引脚,微处理器使用通用 IO 管脚来配置 FPGA。此外还需要设置 M[2:0]信号以选择配置模式,根据芯片 XC3S1000 和 XC4VSX35 的模式设置其从串模式的配置均为 M[2:0]=111。

#### (3)HPI 接口设计

引导配置引脚 BEA[19:18]决定了 TMS320C6416 的引导加载模式,将 DSP 的 EMIFB 接口引脚 BEA[19:18]通过上下拉电阻配置成 BEA[19:18]=01,即选择 DSP 为 HPI 加载方式。HPI 接口电路如图 2 所示,主控 CPU 的数据总线、地址总线及控制 I/O 连接到 FPGA 上,DSP 的 HPI 接口数据总线和控制线也连接到 FPGA 上,通过 FPGA

的接口逻辑设计将 CPU 和 DSP 连接起来。

### 2.2 软件设计

#### 2.2.1 网络接口程序设计

##### (1)网络通信协议设计

网络通信协议设计主要是实现 TCP/IP 协议。根据嵌入式系统结构特点和系统设计实现的功能需求,在 TCP/IP 协议的基础上实现系统所需功能。

##### (2)网络接口的访问控制

驱动程序设计包括以太网 PHY 芯片的初始化、数据包的发送和接收。Freescale 的 MPC8377 自带了三速以太网控制器 ETSEC,作为以太

## 网络与通信 Network and Communication

网接口与外部的 PHY 芯片相连接<sup>[7]</sup>。本设计中主控 CPU 与 PHY 之间是 RGMII 模式接口。88E1111 可以实现多种数据速率和访问控制模式,通过 HWCFG MODE [3:0] 确定其操作模式,设定 DIS\_FC 为 0,可以自适应地选择 FIBER、COPPER 接口。

Linux 网络驱动程序的实现主要是在 Linux 内核中提供的网络设备接口及以上层次的代码基础上,移植或编写特定网络硬件的驱动程序工作,也就是完成设备驱动功能层,主要包括数据的接收、发送等控制。

### 2.2.2 FPGA 接口程序设计

FPGA 从串模式的配置过程将配置比特流载入到 FPGA,有 4 个主要阶段:(1)清除配置内存;(2)初始化;(3)载入配置数据;(4)设备启动。其配置时序流程图如图 3 所示。

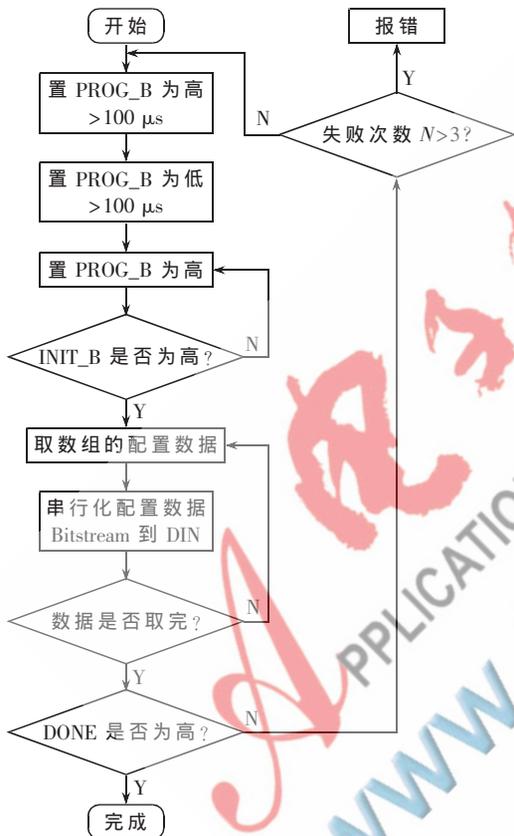


图3 FPGA的从串配置时序流程图

系统上电后,将 PROG\_B 拉低以复位 FPGA 内部逻辑重新配置 FPGA,充分复位内部逻辑后,将 PROG\_B 置高。

INIT\_B 为低电平,在 PROG\_B 拉高保持一定时间后,FPGA 将 INIT\_B 置高。在 INIT\_B 由低向高跳变的瞬间,采样配置模式 M[2:0]。此系统采用从串配置模式。在 FPGA 采样配置模式后,微处理器就可以向 FPGA 配置时钟 CCLK 和数据,在 CCLK 的上升沿,传输数据至 DIN,数据字节先发送低位,再发送高位。配置过程中若发生错误,则 INIT\_B 为低电平。

当 DONE 为高电平时,FPGA 释放全局三态(GTS),激

活 I/O 引脚,释放全部置位复位(GSR)和全局写使能(GWE)有效,开始执行配置区的逻辑。因此在所有的配置数据传送完成,CRC 校验无误时,DONE 为高电平,否则为低电平<sup>[6]</sup>。

微处理器从串配置 FPGA 的时序图如图 4 所示。

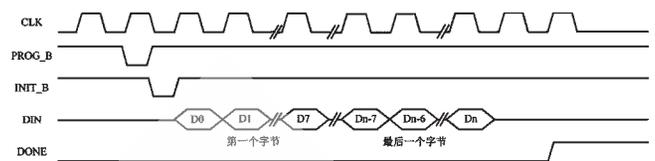


图4 FPGA的从串配置时序图

### 2.2.3 HPI 接口程序设计

本文的嵌入式数字处理系统设计中,主控 CPU 通过 FPGA 访问 HPI 接口的 HPIC、HPIA、HPIDA 和 HPIDF 这 4 个寄存器,实现与 TMS320C6416 的通信,从而完成引导加载 DSP 程序代码。图 5 为 DSP 的 HPI 加载流程图。



图5 DSP的HPI加载流程图

本设计中采用 HPI16 模式,16 根数据线通过 FPGA 逻辑与主控 CPU 数据总线的低 16 位相连,将 HPI 的控制信号通过 FPGA 连接到主控 CPU 上。

对 DSP 进行 HPI 接口加载首先要对 DSP 进行复位,主控 CPU 对 DSP 的全部存储空间进行访问,先对 HPIC 寄存器初始化,主要针对 HPI16 模式最低位 HHWIL 位设置,决定数据传输格式是按高半字在前(设置为 0),还是低半字在前(设置为 1),该位对于 HPI32 模式无效,可不设置;然后,对 HPIA 寄存器初始化,设置访问单元的地址;最后通过读写数据寄存器(HPIDA、HPIDF)实现数据读写操作,其中读写 HPIDA 寄存器是完成连续地址单元读写操作,读写 HPIDF 寄存器是完成固定地址单元读写操作。数据线、地址线定义如图 6 所示。

LD、LA 分别表示主控 CPU 的数据和地址,addr 表示

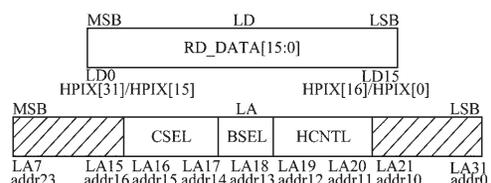


图6 数据线、地址线定义

## 网络与通信 Network and Communication

FPGA 内部地址, HPIX 表示 DSP 的 HPI 控制器数据。

CSEL:0 表示选择 FPGA, 1 表示选择主 DSP, 2 表示选择从 DSP, 3 作为备用选择。

BSEL:0 表示 HPIX 的低 16 位, 1 表示 HPIX 的高 16 位。

HCNTL:0 表示 HPIC, 1 表示 HPIA, 2 表示 HPID (HPIA 自动递增), 3 表示 HPID (HPIA 保持不变)。

在软件编程时, HPI 接口寄存器的定义如下所示:

```
/* HPI FPGA registers */
#define HPI_HCNTL_HPIC      (0x0<<11)
#define HPI_HCNTL_HPIA     (0x1<<11)
#define HPI_HCNTL_HPIDINC  (0x2<<11)
#define HPI_HCNTL_HPIDFIX  (0x3<<11)
#define HPI_BSEL_LOW      (0x0<<13)
#define HPI_BSEL_HIGH     (0x1<<13)
#define HPI_CSEL_FPGA     (0x0<<14)
#define HPI_CSEL_HPI1     (0x1<<14)
#define HPI_CSEL_HPI2     (0x2<<14)
#define HPI_FPGA_MEM_SIZE (2<<16) /* bytes */
```

### 2.3 HPI 程序加载代码的生成

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)。该代码文件不是 DSP 中实际运行的程序代码, 需用 Hex 工具进行转换, 然后使用自编程序将可执行的程序代码提取出来, 封装成帧, 通过以太网加载到 DSP 中。HPI 引导完成后, DSP 从地址 0 开始执行, 而 DSP 程序的入口是 \_c\_int00, 所以要在地址 0 处添加一条跳转指令到程序入口<sup>[7]</sup>。

### 3 工程应用

以前研制的数字处理系统采用 CPU+DSP 或 CPU+FPGA, 在研制阶段和应用过程中需采用仿真器进行程序加载, 无法实现远程加载。然而在产品组装成品进行系统联调时需要多次对 FPGA、DSP 程序进行优化和完善, 在对 FPGA、DSP 程序进行更新时极其不方便, 每次更新程序都需要先解除系统连接, 然后拆卸产品再用仿真器对该主控单元进行 FPGA、DSP 程序进行升级更新,

整个过程繁琐并且费时费力, 严重制约研制进程, 而且后续工程应用的维护工作难度也较大。

本文所述嵌入式数字处理系统基于以太网加载 FPGA 和 DSP, 只要将需要进行更新程序的产品通过网络接口和外部计算机连接起来进行数据交换, FPGA 和 DSP 程序的加载与启动直接由外部计算机控制, 无需拆卸系统和产品, 摆脱了仿真器。这种方式使得整个加载简单方便, 易于操作, 而且可以实现对复杂系统的多个 FPGA 和多个 DSP 进行基于以太网接口的加载。

随着现在数字技术的高速发展, 数字系统设计的功能越来越强大, 设计中综合微处理器、FPGA、DSP 也越来越常见, 本文提出了一种基于以太网加载 FPGA 和 DSP 的实现方法, 文中嵌入式数字处理系统实现了多个 FPGA、DSP 的程序加载进行软件更新, 证实了该技术简单方便、可实现远距离加载, 且具有良好的可靠性。该技术在工程应用领域具有独特的优越性和良好的应用前景。

#### 参考文献

- [1] 朱伟, 王广君. 基于 ARM 的嵌入式系统中从串配置 FPGA 的实现[J]. 仪表技术, 2007(5): 13-15.
- [2] 李文禄, 蒋宇中. 一种基于以太网的 TMS320C6713B 程序加载技术[J]. 海军工程大学学报, 2008, 20(6): 35-39.
- [3] Spartan-3 FPGA Family Data Sheet[S]. Xilinx. 2009.
- [4] Virtex-4 User Guide v2.3[S]. Xilinx. 2008.
- [5] TMS320C6416 fixed-point digital signal processor[S]. Texas Instruments. 2005.
- [6] 88E1111 Datasheet INTEGRATED 10/100/1000 ULTRA[S]. Marvell. 2009.
- [7] MPC8379E PowerQUICC™ II Pro integrated host processor family Reference manual[S]. Freescale Semiconductor. 2009.

(收稿日期: 2011-04-16)

#### 作者简介:

吴湘宁, 男, 1981 年生, 本科, 工程师, 主要研究方向: 通信系统, 控制平台。

江瑾, 男, 1981 年生, 博士, 主要研究方向: 短波通信。

冯俊涛, 男, 1982 年生, 硕士, 工程师, 主要研究方向: 嵌入式软件开发。