

# 基于网络处理器 IXP1200 的以太网上联卡设计

**摘要:** 以太网上联卡是以 ATM 技术为内核的 DSLAM 设备中的一块板卡, DSLAM 设备通过它可直接与 IP 网络相连, 由于在进行 ATM 与 IP 转换时要消耗大量资源, 因而很容易使上联卡成为整个系统的瓶颈。文章提出了一种基于网络处理器 IXP1200 的以太网上联卡设计方案, 该方案利用 IXP1200 网络处理器强大的数据处理能力和高度的灵活性来实现对数据的线速处理, 同时也可根据需要增加新的功能。

**关键词:** 以太网上联卡; 网络处理器; ATM; 以太网; 微码; IXP1200 变压器隔离电路 YL18-2050S

随着网络通讯技术的高速发展, 宽带接入技术成了当前电信接入技术的热点。由于早期的宽带技术以 ATM 为核心, 各大厂家提供的核心芯片和线路接口芯片都是基于 ATM 技术的。而数据网络主要以 TCP / IP 为核心, 因此, 为了解决 ATM 和 TCP / IP 的融合问题, 就需要在 DSLAM 设备上提供 ATM 到以太网的转换。但转换过程中需要进行大量的数据处理, 因此容易产生系统瓶颈, 而上联卡的设计就是为了解决在 DSLAM 设备中的 ATM 信元和以太网帧之间的高速转发问题。本文提出了一种基于网络处理器 IXP1200 的上联卡设计方案, 并对该方案的实现过程进行了详细分析。

## 1 网络处理器 IXP1200 主要特性

网络处理器是一种硬件可编程器件, 通常是一种芯片, 它是专门为处理网络数据包而设计的。通过对硬件架构和指令集的优化, 该网络处理器不但可提供线速处理数据包的高质量硬件功能, 同时还具备极大的系统灵活性。

IXP1200 是英特尔公司生产的一款高档网络处理器, 也是 IXA (Internet Exchange Architecture) 架构的核心产品。IXP1200 的内部结构如图 1 所示, 它内含 1 个主频最高可达 232 MHz 的处理核心 Strong ARM、6 个 RISC 结构的可编程微引擎 (每个微引擎包含 4 个硬件线程)、64 位和最高 104 MHz 的 I X B u s 、32 位的 S R A M 接口单元 (工作频率为核心频率的一半)、64 位的 S D R A M 接口单元 (工作频率为核心频率的一半)、32 位和最高 66 MHz 的 P C I 总线接口单元等。IXP1200 通过 F B I 接口单元和 I X B u s 相连接。另外还有一套集成开发环境, 可用于对微引擎进行应用开发, 它支持汇编和 C 编程语言。

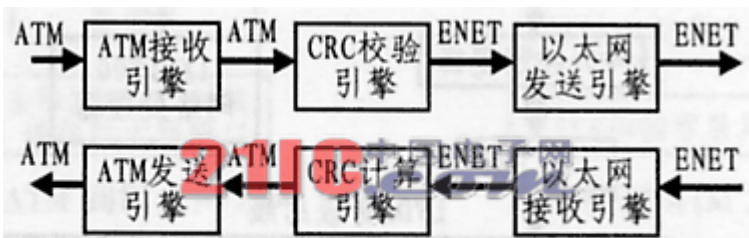


图 2 微引擎的任务分配

(1) Strong ARM Core

通过 Strong ARM Core 可实现 CPU 的主要功能, 同时可启动系统、管理和控制对网络处理器的其它单元、处理微引擎无法处理的数据包和一些异常状况。

### (2) 微引擎

微引擎是可编程的 32-bit RISC 处理器, 它的指令集是专门针对网络和通信应用而设计的。通过对各个线程进行编程, 可单独执行数据包的转发和处理, 而无需 Strong ARM Core 干预, 因而可减轻 Strong ARM Core 的负担, 特别适合高速数据的处理和转发。

### (3) SDRAM 单元

SDRAM单元可提供IXP1200与SDRAM的接口，最大可支持256M字节的SDRAM。虽然SDRAM的访问速度较慢，但存储空间大，因而可用来存储大容量的数据结构（如数据包和路由表等），并可在系统运行时存储操作系统的代码。

#### (4) SRAM单元

SRAM单元可为三种类型设备提供通用总线接口。这些设备包括最大可达8M字节的SSRAM、复位后StrongARM Core执行代码所在的FLASH或E-PROM等、BOOTROM设备和其它慢速端口设备（如CAM）、加密设备和MAC或PHY设备的控制状态接口。SRAM访问速度较快，但存储空间小，主要用来存储查找表和缓存描述符等需要快速访问的数据结构。

#### (5) PCI单元

PCI单元用于提供与PCI设备相连的接口，可用于下载操作系统和配置程序。

#### (6) FBI单元

图1中的哈希单元、IX总线接口和Scrachpad内存统称为FBI单元。IXP1200通过FBI单元和IX Bus相连，来实现外设与IXP1200之间数据包的收发，以便使微引擎可以访问这些数据包，并利用线程对其进行转发。实际上，StrongARM Core也可以访问这些数据包，并对其进行异常处理或上层协议处理。

## 2 以太网上联卡的设计方案

以太网上联卡的基本功能是实现ATM信元和以太网帧之间的转发，即从LVDS接口收到来自核心卡的ATM信元流后，根据封装协议（如RFC1483桥接协议）转换成以太网帧，然后建立相应的MAC地址与ATM PVC的对应关系，并通过以太网上联口送往IP网络；也可以从以太网上联口接收来自IP网络的以太网帧，然后根据建立的MAC地址与ATM PVC的对应关系，将其转换成ATM信元流，再通过LVDS接口送往核心卡。

在上联卡中，ATM信元和以太网帧之间的转发是由网络处理器中的微引擎完成的。要使以太网上联卡不成为网络的瓶颈，微引擎必须能以线速来处理数据包（以太网帧或ATM信元），即在下一个数据包到来以前，完成对当前数据包的处理。因此，每个数据包的最大允许处理时间应小于数据包之间的间隔时间。在进行设计时，应根据以太网上联卡具体功能的实现，并结合网络处理器IXP1200所拥有的硬件资源来进行合理的分配使用。这样可以最大限度地发挥系统性能，本设计中，以太网上联卡需要实现以太网接收处理、CRC计算产生、ATM发送处理、ATM接收处理、CRC校验、以太网发送等六个主要任务，而由于IXP1200刚好拥有六个微引擎，因此，将这六个单独的任务分配在每个微引擎上，并在处理上将其搭建成多流水线结构的程序架构，可以取得很好的处理效果。图2给出了网络处理器IXP1200的六个微引擎的任务分配方案，该分配方案的整个处理流程可以分为两个方向，一是上行方向，即ATM到以太网的数据映射，二是下行方向，即以以太网到ATM的数据转换。

在上行方向，ATM接收引擎把收到的ATM信元组装成AAL5 PDUs，并根据封装协议转换成以太网帧，同时建立相应的MAC地址与ATM PVC的对应关系，然后送到CRC-32校验队列。接下来由CRC-32校验引擎对队列中的PDUs执行CRC校验并把PDUs送到以太网的发送队列。而以以太网发送引擎的任务则主要是把发送队列中的以太网帧从以太网上联口发送出去。

在下行方向，以太网接收引擎接收来自以太网上联口的以太网帧，并将其封装成AAL5 PDUs后送到CRC-32产生队列，同时根据建立的MAC地址与ATM PVC的对应关系进行查找以得到ATM信元头部。接着由CRC-32产生引擎为队列中的PDUs生成CRC校验值，并把PDUs送到UBR队列。

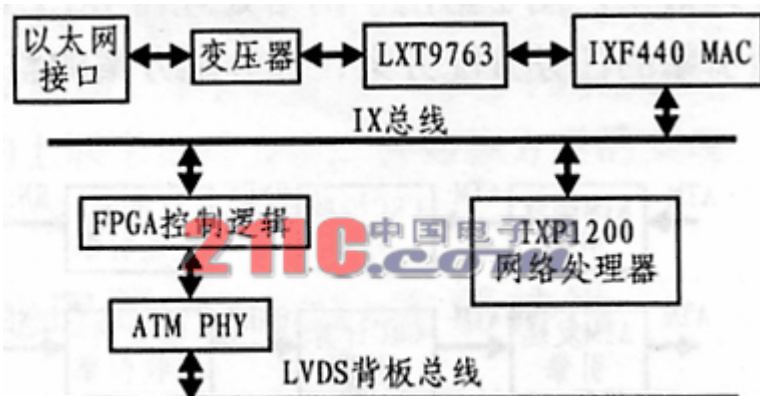


图3 以太网上联卡硬件结构图

最后由ATM发送引擎把PDUs分割(segment)成ATM信元后,从ATM端口发送出去。

### 3 以太网上联卡的硬件设计

图3所示是以太网上联卡的硬件电路,该硬件电路主要包括四个部分:以太网处理单元、IXP1200网络处理单元、FPGA控制逻辑单元、ATM及LVDS背板总线处理单元。

#### 3.1 以太网处理单元

以太网处理单元是上联卡的上联处理部分,用于连接路由器或者三层交换机等数据网络设备。该单元主要包括RJ45接口、变压器隔离电路YL18-2050S、LXT9763以太网物理层芯片和IXF440MAC层芯片。其中RJ45接口以及变压器隔离电路YL18-2050S是以太网处理接口的标准单元电路,LXT9763主要完成802.3协议中描述的物理层功能,它主要通过MII总线和IXF440芯片相连接。IXF440芯片主要完成802.3协议中描述的MAC层功能,同时提供与网络处理器的IX总线接口,实际上,该芯片是网络处理器中IX总线的SLAVE设备。

#### 3.2 IXP1200网络处理单元

IXP1200网络处理单元是整个以太网上联卡的核心,它主要通过IX总线与外部芯片进行相连,是IX总线的MASTER设备,所有的处理软件均运行在网络处理器中。

IXP1200网络处理单元由网络处理器IXP1200及外部芯片(如SDRAM、SRAM、Flash等)构成。SDRAM和SRAM单元是可共享的智能单元。其中SDRAM单元可以被IXP1200的StrongARM内核以及微引擎和PCI总线上的设备直接访问,这样可以支持SDRAM与微引擎或IX总线以及PCI总线之间的快速移动数据,而SRAM单元则具有比SDRAM单元更快的访问时间,通常可以用来存储需要快速查找的表格,以提高性能。

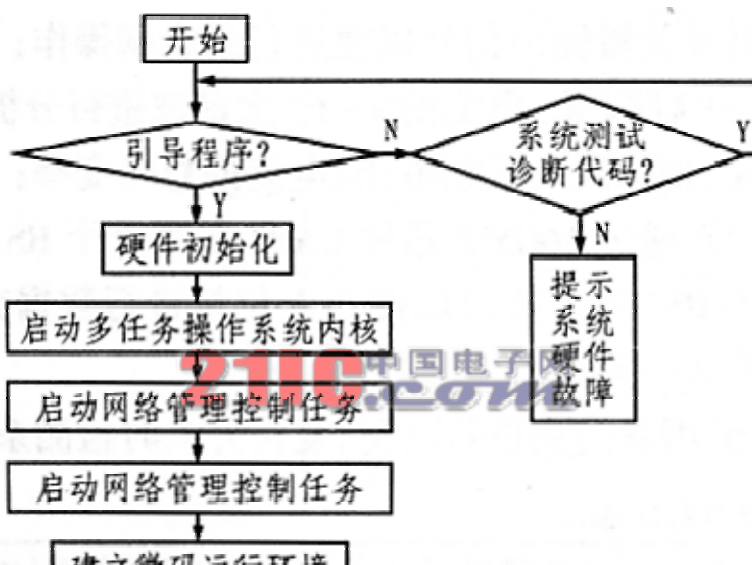
#### 3.3 FPGA控制逻辑单元

由于在英特尔公司所提供的网络处理器解决方案中,外部的数据接口是IX总线,该总线是英特尔提供的专有数据总线,而以太网上联卡中所采用的ATM芯片的外部接口为标准的UTOPIA总线。所以,为了实现芯片间的互联,应采用FPGA来完成IX总线和UTOPIA总线间的变换,即在IX总线端实现IX总线的SLAVE接口,在ATM端实现UTOPIA总线的SLAVE接口。通过该FPGA逻辑控制单元可为ATM到以太帧的转换提供物理层的控制功能。FPGA逻辑控制单元的实现对于完成以太网上联卡的设计非常关键。

#### 3.4 ATM与LVDS背板总线单元

该处理单元主要完成以太网上联卡中的网络处理器单元与背板ATM的无缝连接。由于DSLAM设备的设计核心是基于ATM技术,为了将网络处理器单元应用在基于ATM的DSLAM设备中,必须采用该处理单元来实现系统互连。

DSLAM设备系统中的其它板卡主要用于完成ATM交换以及ADSL设备的线路接口。而背板是基于LVDS总线的高速差分总线,它具有抗干扰能力。这对于高密度的DSLAM设备来说是非常重要的。



实际上，上联卡就是通过ATM的物理层芯片与高速LVDS总线进行互联，从而使该板卡无缝插接在系统之中。

#### 4 以太网上联卡的软件设计

以太网上联卡的软件主要运行在网络处理器IXP1200中。为了方便基于网络处理器IXP1200的开发，英特尔公司特别推出了高度集成且具有强大开发能力的开发工具SDK2.0。这个开发工具包中包含有IXP1200 Developer WorkBench，是一个集成的开发工具，专门用来写符号微码，并且具有汇编器以及优化设备，还提供了一个不需要硬件的IXP1200模拟器，可支持软件模式下的仿真和调试，因而具备友好的用户接口和调试环境。

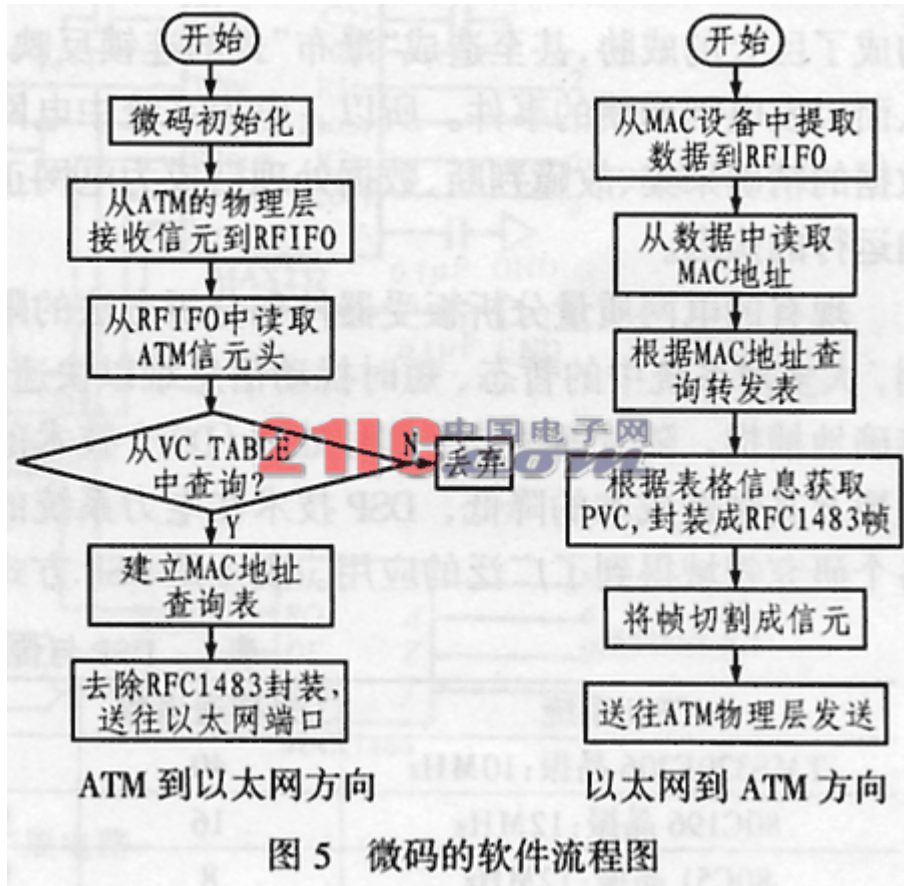


图5 微码的软件流程图

网络处理器IXP1200的软件开发主要基于两个层面，一个是高层软件，通常指运行在网络处理器IXP1200 StrongArm内核上的管理软件、路由协议软件以及所有的系统所需任务，这部分软件通常需要一个嵌入式操作系统，目前的开发主要基于Linux操作系统。另一个层面是底层软件，这部分软件主要运行于六个微引擎之上，可用于完成包的快速处理，包括包的快速转发和基本的二层协议处理等，这部分软件采用微码形式来完成，但应特别注意软件部分的代码优化，即用尽可能少的指令来完成处理。在网络处理器IXP1200中，每个微引擎提供有2k字大小的代码存储空间。此外，每个微引擎中也包含四个线程，这四个线程可构成硬件多线程。由于微引擎内部包含大量的GPR以及SRAM、SDRAM传输寄存器，因此，在采用微线程进行相对寻址模式时，每个线程都具有自己特定的寄存器组，从而极大地加快了线程切换的速度。在IXP1200中进行微码设计有一个重要原则：即当一个线程在等待资源时，应将该线程切换出去，以让其它线程占用微引擎的处理，这样可进行快速切换，以保证各个线程都能够充分利用微引擎的处理机，而不会因为一个在等待资源线程，造成处理器的浪费。微码的组织也是按照这一原则来进行的。图4所示是高层软件的程序主流程图。高层软件的目的是完成整个硬件和软件的初始化，同时将微码程序加载到网络处理器的六个微引擎中，并启动运行。

底层软件的微码流程分为两个部分，其任务分配和以上讨论的六个微引擎的任务分配一致。它也分为两个方向，即 A T M到以太网方向和以太网到 A T M方向。图 5 所示是其微码的软件流程图。

## 5 结束语

本文介绍的基于网络处理器 I X P 1 2 0 0 的以太网上联卡，已经成功应用于 D S L A M设备中，并解决了 D S L A M设备与 I P网络的高速互联问题。经过测试 本卡性能良好，系统运行稳定。