

基于飞腾 E2000 处理器网卡驱动程序的设计与实现

张 帅¹ 李若玮¹ 武佳宇¹

ZHANG Shuai LI Ruowei WU Jiayu

摘 要

飞腾腾珑 E2000 系列处理器是飞腾公司推出的最新一代高能效处理器内核，其网络控制器采用了可支持千兆以太网传输能力的全新自研方案，目前行业从业人员对该网络控制器的适配经验较为欠缺。针对上述问题，为实现在该系列处理器上网络控制器的驱动适配以能够正常使用其以太网通信功能，对该处理器的网络控制器中硬件工作原理与描述符结构进行了分析，基于现有的网卡驱动框架设计并实现了一套网卡驱动程序。经可靠性测试，上述网卡驱动程序运行稳定，能够在搭载飞腾 E2000 系列处理器的开发板上长时间稳定地进行以太网通信工作。

关键词

飞腾腾珑 E2000；网络控制器；网卡驱动框架；网卡驱动程序；可靠性测试；以太网通信

doi: 10.3969/j.issn.1672-9528.2024.01.009

0 引言

飞腾腾珑 E2000 系列处理器（以下简称 FTE2000）是飞腾公司基于 ARMv8a 指令集架构^[1]自主研发的高能效嵌入式处理器，根据内核的数量与规格不同分为 E2000Q、E2000D、E2000S 三种系列。FTE2000 处理器集成了两种类型的自主研发 64 位通用处理核心，包括了采用乱序多发射超标量体系结构的高性能处理器核 FTC664 与采用了顺序架构的低功耗处理器核 FTC310，可以运行在 64 位与 32 位工作模式下。作为飞腾公司近期推出的最新一代产品，该系列处理器在云终端、电力、轨道交通、国防军工等领域具有较大的应用潜能。

相较于飞腾的前几代处理器产品而言，FTE2000 不再使用传统的 GMAC 网络控制器^[2]，而是采用了全新的自研技术方案与网络设备型号以进一步提高其产品性能，这使得开发人员无法沿用过去的网卡驱动程序，而是必须重新设计一套网卡驱动程序以适配基于该系列处理器所生产的板卡。

通常情况下，网卡驱动程序只需实现主机端（即 PC 机端）与搭载着 FTE2000 处理器的板卡端（又称目标机端）能够进行最基本的数据收发即可^[3]，TCP/UDP 等协议可由上层应用程序进行实现。虽然网卡驱动程序所实现的功能较为单一，但却对数据收发的稳定性与可靠性有着较高的要求^[4-6]。本文提出并实现了一套对 FTE2000 板卡进行网络控制器适配的网卡驱动程序方案，并对该程序进行了可靠性测试，以证明其在 FTE2000 板卡上以太网通信功能的稳定性与可靠性。

1 FTE2000 网络设备工作基本原理

1.1 硬件原理

FTE2000 的网络控制器包含了支持百兆与千兆的以太网接口，主要实现 AXI 总线与以太网接口的互相转换。该网络控制器以队列的形式进行网络数据包的接收与发送，当根据硬件手册为 FTE2000 板卡配置好正确的 PHY 芯片工作模式、MAC 配置寄存器、网络控制寄存器与 DMA 通道寄存器等寄存器值后，FTE2000 板卡端与 PC 机端即可进行正常的网络数据通信。当 PC 通过 ping 命令对目标机进行网络通信时，目标机内部的基本工作原理如图 1 所示。

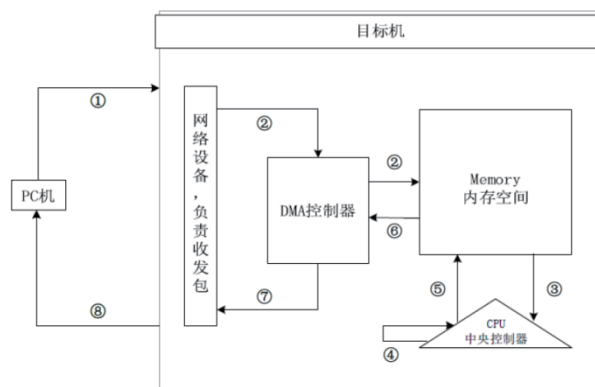


图 1 目标机端内部工作流程示意图

其中，图 1 中的各步骤对应的数字编号含义如下。

（1）主机发送数据包给目标机。

（2）目标机网络控制器收到数据包后，根据以太网 DMA 接收描述符中的相应信息将数据包直接写入预设的内存缓冲区地址中。

(3) 中央控制器运行网卡驱动程序, 从内存中的指定地址将收到的数据包读取出来。

(4) 中央控制器对收到的数据包进行加工, 并生成用于响应主机端的应答包。

(5) 中央控制器执行网卡驱动程序中的发送数据包程序, 将应答包写回内存。

(6) 中央控制器将应答包的内存地址写给 DMA 控制器, 即填充对应的以太网 DMA 发送描述符信息。

(7) 网络控制器根据以太网 DMA 发送描述符信息, 从内存的对应地址中取出应答包数据, 并发送应答包。

(8) 目标机应答 PC 机, 完成一次网络通信。

1.2 以太网 DMA 描述符结构

网络控制器中的接收/发送 FIFO 与内存之间的以太网数据包传输是通过以太网 DMA 描述符的形式, 并经由 DMA 控制器来完成的, 该过程的实现不需要 CPU 的参与。在通常情况下, 以太网 DMA 描述符被分为接收描述符与发送描述符两大类。在接收数据包阶段, 接收 FIFO 中的数据可以不经过 CPU 参与而直接被传输至接收描述符指向的缓冲区当中; 在发送数据包阶段, DMA 控制器可不经 CPU 参与直接将发送描述符指向的缓冲区数据传输到发送 FIFO 当中。因此, 描述符结构的确定是网卡驱动程序的设计与实现中最为重要的一环^[7-8]。

FTE2000 的接收描述符与发送描述符结构所占空间大小一致, 但部分字段含义存在较大差异。在 AARCH32 运行模式下, FTE2000 的描述符结构由两个 UINT32 类型的数据所组成, 分别为描述符的地址字与状态字; 在 AARCH64 运行模式下, 则是由四个 UINT32 类型的数据所组成, 其中后两个 UINT32 数据为 64 位高位地址的扩展字。

为了同时兼容 AARCH32 运行模式与 AARCH64 运行模式, FTE2000 的以太网 DMA 描述符结构体定义如下。

```
typedef struct fte2000_dma_desc
{
    volatile UINT32 bdAddr;
    volatile UINT32 bdSts;
#ifdef __64BIT__
    volatile UINT32 bdExt1;
    volatile UINT32 bdExt2;
#endif /* __64BIT__ */
} FTE2000_DMA_DESC;
```

其中, 第一个 UINT32 类型的数据 bdAddr 为描述符的地址字, 第二个 UINT32 类型的数据 bdSts 为描述符的状态字, bdExt1 与 bdExt2 仅在 AARCH64 运行模式下编译生效, 它们均为内存地址高 32 位的扩展字。本文仅对两种执行模式均使用到的状态字与地址字进行详细介绍。

对于接收描述符而言, bdAddr 的第 0 位为 ownership 位,

在网卡驱动程序进行初始化时会将其置为 0, 当接收到网络数据包时, 该位会由网络控制器自动置为 1, 当 CPU 处理完毕该数据包时, 网卡驱动程序会将其重新设置为 0; 第 1 位为 wrap 位, 该位标记了接收描述符队列中的最后一个接收描述符; 其余高 30 位则代表了接收描述符指向的缓冲区起始地址。bdSts 的第 0 至 12 位表明了接收到的数据包长度; 第 14 与 15 位分别代表数据帧的开始位与结束位。

对于发送描述符而言, bdAddr 的第 0 至 31 位表示发送描述符指向的缓冲区数据起始地址。bdSts 的第 0 至 13 位表示应答包的数据长度; 第 15 位被置位则标志当前发送数据包的最后缓冲区已到达; 第 30 位标记了发送描述符队列中最后一个发送描述符; 第 31 位为 ownership 位, 在网卡驱动程序初始化时会将其置为 1, 在发送数据包时则将其清零表明此发送描述符可由网络控制器进行处理, 当数据包被正确发送后, 该位会由网络控制器置为 1, 此时网卡驱动程序会将其重新设置为 0。

2 网卡驱动程序的实现

2.1 网卡驱动框架

根据现有的网卡驱动程序框架, 本文将 FTE2000 板卡网卡驱动程序的实现分为四个部分, 即网络设备初始化部分、网络设备接收数据包部分、网络设备发送数据包部分、中断模式下的网络收发包部分。在上述框架下, 网络设备收包阶段与发包阶段一次只对一个数据包进行处理。此外, 将接收包转换为应答包的加工过程, 以及中断模式下中断服务程序的挂接与中断线的使能由上层应用程序加以实现, 本文不再对其进行介绍。

2.2 网络设备初始化

网络设备初始化阶段的工作流程如图 2 所示。

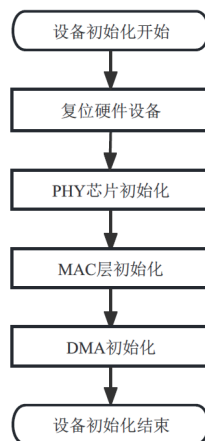


图 2 网络设备初始化工作流程示意图

从图 2 中可以看出, 网络设备的初始化部分共分为四个阶段, 分别为复位硬件设备、PHY 芯片初始化、MAC 层初

始化、DMA 初始化。

复位硬件设备阶段的主要作用是将网络设备相关的寄存器进行复位，保证设备处于停止工作状态。在该阶段所进行的操作主要包括禁用设备的发送与接收功能、禁止设备产生发送或接收中断、清除设备全部中断状态等操作。

PHY 芯片初始化的主要功能是扫描当前硬件所使用的 PHY 号，判断当前 MII 类型与分频系数，设置 PHY 芯片的自协商模式，并根据是否自协商来确定设备的速度与双工模式。

MAC 层初始化主要进行 MAC 地址的设置、对 MAC Control 与 MAC Config 寄存器进行配置、根据 MII 类型与速度双工模式对设备硬件其他相关的寄存器的部分位进行设置的操作。

DMA 初始化主要是对 DMA Config、receive_q_ptr 与 transmit_q_ptr 寄存器进行设置，并对以太网 DMA 描述符进行初始化。在这个阶段，需要明确接收与发送描述符的结构以及用于存放这些描述符的数据结构。在通常情况下采用链式或环形队列的数据结构对描述符进行存放^[9]，由于环形队列具有节约空间且开销较小的优点^[10]，因此本文使用环形队列作为存放描述符的数据结构。

在对描述符进行初始化时，需分别构建发送环形描述符队列与接收环形描述符队列用于存放发送与接收描述符。除此之外，需明确这两个队列的最大描述符个数。对于接收描述符而言，在初始化阶段要遍历接收环形描述符队列并为每个接收描述符的地址字预设一段缓冲区域地址用于存放网络数据包，清空地址字的 ownership 位，置位队列中最后一个描述符地址字的 wrap 位，并将状态字的值置为 0；对于发送描述符而言，在初始化阶段需遍历发送环形描述符队列并将每个描述符的地址字清零，将状态字的 ownership 位置 1，并置位队列中最后一个描述符状态字的 wrap 位。

2.3 接收数据包

目标机接收数据包阶段的工作流程如图 3 所示。

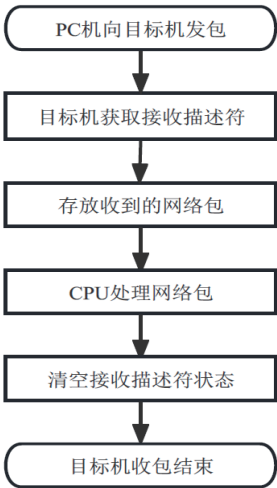


图 3 目标机接收数据包工作流程示意图

当 PC 机向目标机发送数据包进行通信时，目标机端的网络控制器在顺利进行初始化后会收到该数据包。网络驱动程序的主要任务是引导硬件将收到的数据包存放在内存的对应地址，由 CPU 读取内存处理该数据包，转换成对应的应答包发送给 PC 机。

因此，在目标机接收数据包过程中，驱动程序首先要根据接收描述符环形队列的当前工作指针从该队列中获取一个可用的接收描述符，与此同时更新队列的工作指针。在获取到接收描述符后，需判断该描述符的 ownership 位是否为 1，如果该位为 1 则表明硬件已经将收到的数据包以 DMA 的形式写入到该接收描述符所指定的内存空间中，否则则表明硬件未将数据包存放至内存中。若数据包已被存放至内存中，网络驱动程序则通过内存拷贝的方式将接收描述符所指向的缓冲区地址中的数据取出，并读接收描述符状态字的第 0 至 12 位获取收到的网络包长度，并以输出型参数的方式将网络包内容与长度信息传递给上层应用程序以完成由接收数据包到应答数据包的逻辑转换。在最后，网络驱动程序需将接收描述符的 ownership 位做清零操作，以供网络设备进行循环使用。

2.4 发送数据包

目标机发送数据包阶段的工作流程如图 4 所示。

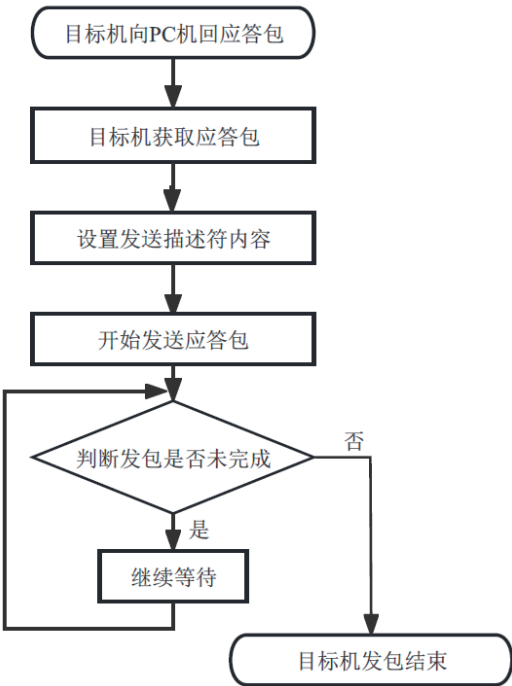


图 4 目标机发送数据包工作流程示意图

CPU 将收到的网络包转换为应答包后，上层应用程序会将存放应答包的缓冲区首地址与长度以输入型参数的方式传给网卡驱动程序的发送数据包接口。该接口的主要功能是将

应答包的缓冲区地址与长度填入发送描述符的对应字段中,并启用寄存器的某些位开始进行数据发送。

在目标机发送数据包过程中,驱动程序首先要获取存放应答包的缓冲区首地址与长度信息并通过发送描述符队列的当前工作指针获取该队列中第一个可用的发送描述符。获取完毕后,将应答包缓冲区首地址填入发送描述符地址字中,并将长度信息填入发送描述符状态字的第 0~13 位中。在将上述字段设置完毕后,驱动程序使能 MAC control 寄存器的“开始发送”位,通知硬件可以开始发送数据包了,与此同时循环读取发送状态寄存器,以判断当前应答包是否发送完成,若以发送完成则释放存放应答包的缓冲区地址并更新发送描述符队列的当前工作指针,并清空发送描述符状态字的 ownership 位,此时目标机的发包任务已经结束。若当前应答包未发送完成,则需持续等待该包发送完成。

2.5 中断模式下的网络收发包

当网卡驱动程序应用于操作系统中时,网络的收发包动作均应以中断的方式进行响应,因此驱动程序需要考虑中断模式下网络控制器如何进行收发包。

在中断模式下进行收发包工作时,首先应当明确网卡控制器的中断向量号,并根据中断向量号来挂接对应的中断服务程序。本文默认采用了 FTE2000 的第 0 路网的 q0 队列,通过查询该处理器的软件编程手册,明确了网口 MAC0 的 q0 队列采用的中断向量号为 87。

其次是中断服务程序的设计,当目标机收到网络数据包时,在中断控制器驱动正常适配的前提下,硬件会上报 87 号中断并改变其网卡的中断状态寄存器。因此,在中断服务程序的设计中,会对网卡中断状态寄存器进行循环判断,直到该寄存器的第 1 位 receive_complete 位被置 1 后退出循环,并执行 2.3 节中所提到的接收数据包接口。当该接口被执行完毕后,上层应用程序将会自行调用数据包转换接口与发送数据包接口以完成一次网络数据包的通信工作。

3 验证与测试

为确保本文所设计并实现的网卡驱动程序运行时的稳定性,本文采用了烤机的方式在搭载了 FTE2000Q 与 FTE2000D 处理器的来自不同厂商所生产的四块标准开发板上对驱动程序进行了可靠性测试。在运行数小时后,实验结果表明本文所提网卡驱动程序运行稳定可靠,未出现丢包现象或网络断连问题。

4 结语

FTE2000 系列处理器作为最新一代处理器平台,其优秀的功耗与低廉的价格使其在未来具有较大的应用场景,但其

自研的网络控制器方案使得目前行业内对其进行适配的经验较为有限。本文在研究了该处理器寄存器结构、网络控制器硬件原理与以太网 DMA 描述符结构后,在现有的网卡驱动程序框架基础上对该处理器的网络控制器进行了适配。经可靠性测试验证,本文所提驱动程序运行稳定可靠,可以用于大型应用软件或 Linux 等操作系统之上,也可以为行业内其他从事人员适配该处理器的网络驱动程序提供相关经验作为参考。

参考文献:

- [1] ARM Architecture Reference Manual[EB/OL]. (2020-08-12) [2023-06-26]. <https://documentation-service.arm.com/static/5f8dacc8f86e16515cdb865a>.
- [2] 周浩,戴华东,邵立松.飞腾处理器网络控制器的驱动设计和实现[J].计算机光盘软件与应用,2013,16(19):66-67+70.
- [3] 梅涛,黄堡垒,徐晓光.嵌入式系统网卡驱动模型设计与实现[J].电子科技,2016,29(4):111-115.
- [4] 张前,张钰尧,王宏伟.基于 PowerPC440 处理器的网卡驱动的设计与实现[J].信息通信,2019(03):125-126.
- [5] JAVED F,AFZAL M K,SHARIF M ,et al.Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: a comparative review[J]. IEEE communications surveys and tutorials,2018,3(20): 2062-2100.
- [6] 张前,张钰尧,杨静远.基于 P2040 处理器的以太网驱动程序的设计与实现[J].信息技术与信息化,2023(7):55-58.
- [7] 任蕊.基于 Linux 的 DMA 网卡驱动程序设计[D].哈尔滨:黑龙江大学,2020.
- [8] 王君怡.Linux 网络设备驱动优化的研究[D].武汉:湖北工业大学,2018.
- [9] 易正利.基于 Linux 的千兆网卡驱动程序实现及数据传输效率优化[D].北京:北京邮电大学,2011.
- [10] 孙明刚.一种使用环形缓存和环形队列实现 UDP 高效并发的方法[J].中国新技术新产品,2016(11):6-7.

【作者简介】

张帅(1997—),男,山西朔州人,硕士,助理工程师,研究方向:嵌入式操作系统驱动程序开发。

李若玮(1998—),女,陕西西安人,硕士,助理工程师,研究方向:嵌入式操作系统体系结构开发。

武佳宇(1992—),男,陕西榆林人,硕士,工程师,研究方向:嵌入式操作系统体系结构开发。

(收稿日期:2023-09-26)