

一种精简 TCP/IP 协议栈的设计与实现

·实用设计·

赵 川,伍瑞卿,樊 丰

(电子科技大学 电子工程学院,四川 成都 611731)

【摘 要】阐述了 TI 的 TMS320DM642 的 EMAC 驱动程序的结构,设计并实现了一个精简协议栈。针对 H.264 视频传输提出了改进方案,以实现更好的视频通信质量。

【关键词】TMS320DM642;EMAC;TCP/IP 协议栈;视频传输

【中图分类号】TP393

【文献标识码】A

Design and Implication of Simplified TCP/IP Stack

ZHAO Chuan, WU Rui-qing, FAN Feng

(School of Electronics Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

【Abstract】The EMAC device driver of TI's TMS320DM642 is described. The design and implementation of a simplified TCP/IP stack are given. Then, a modified scheme aim at the H.264 is proposed to achieve a better video communication quality.

【Key words】TMS320DM642; EMAC; TCP/IP stack; video transmission

1 引言

网络视频传输得到了越来越广泛的应用,如 IP 可视电话、数字摄像机、网络视频会议、视频监控系統、数字视频播放器/点播机等。在很多应用场合,都需要小型的嵌入式系统来实现^[1]。通常,基于 DSP 的协议栈都移植自 LWIP, uIP。而这些协议栈并不针对视频传输设计。视频传输中通常使用 UDP 协议以保证传输的实时性^[2]。但是 UDP 的传输是无连接的,不可靠的。例如,在 H.264 中,如果丢掉解码图像需要的参数集或者 IDR 图像,会造成无法解码的结果。笔者提出了一种针对视频流传输的改进的 UDP 协议,在确保正确解码的同时,保证了较小的开销和较高的效率。

2 TMS320DM642 DSP 开发平台

TMS320DM642(简称 DM642)是 TI 公司的一款专用于数字媒体应用的高性能 32 位定点 DSP 芯片。工作主频高达 720 MHz,处理能力可达 5 760 MI/s(兆指令/秒),外部总线时钟 100 MHz。DM642 集成了 64 个 32 bit 的通用寄存器,能够在一个时钟周期内处理 4 个 16 bit 的乘法和 8 个 8 bit 的乘法。64 bit 宽度的 EMIF 总线,可连接大容量 SDRAM,Flash,ATA 等存储器资源。4 路 PAL/NTSC 标准模拟视频输入,1 路 PAL/NTSC 标准模拟视频输出,以及一个支持 10/100 Mbit/s 自适应网卡的 EMAC/MDIO 模块,通过寄存器配置可提供一定的 QoS 保证^[3]。因此,该芯片非常适合视频数据的处理和传输。

3 网络驱动程序的开发

网络接口控制芯片主要由 EMAC Control,EMAC,MDIO 等模块控制。

3.1 模块主要功能和结构

MAC Control:主要作为 DSP 与 EMAC/MDIO 的接口,提供 4 kbyte 的本地内存空间给 EMAC 的包缓冲描述符;具有总线仲裁、重启 EMAC/MDIO、全局中断使能、中断逻辑控制等功能,与 EMAC 模块一起初始化。

MDIO:使用有限状态机配置和监控物理层设备(PHY)。

EMAC:用于在网络上发送和接收数据包。而开发设备驱动主要是对该模块编程。EMAC 的特点是:1) 作为 DMA 控制器,在 DSP 内部和外部存储空间之间传送数据;2) 连接 PHY 的介质无关接口(MII);3) 8 个接收和 8 个发送通道,提供一定的 QoS 支持;4) 可选的发送 CRC 校验;5) 支持多播、广播和混合模式;6) 硬件流控制。

模块框图如图 1 所示,其中 EMAC/MDIO 通过中断映射到 DSP 的 11 号中断,EMAC 通过 DSP 内存转换控制器读写 DSP 的内部外部存储空间,模块的控制寄存器通过 DSP 配置总线映射到 DSP 存储空间。

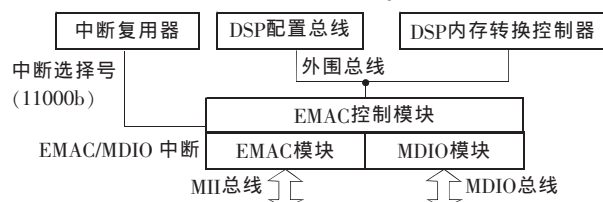


图 1 模块结构框图

3.2 底层驱动的软件结构

EMAC 使用描述符来管理接收和发送缓冲队列。描述符是一个 16 byte 的内存结构,存储在 EMAC Control 模块中,包含数据包的长度,存放位置等信息^[4]。每个描述符表示 1 个以太网包。每次发送或者接收操作完成,EMAC 都会向 DSP 发送中断,调整描述符队列。

以太网包的接收和发送是通过一个缓冲描述符系统实现的。EMAC 提供 256 个可用的描述符,每个接收或发送通道都有自己的通道描述符结构,用来维系描述符链表。设备驱动取出空的描述符以便 EMAC 能够将收到的以太网包的数据写入对应缓存,将缓存中的即将发送的数据送出并且清理对应的描述符以便重复利用。

EMAC 接收和发送数据的流程如图 2 和图 3 所示。

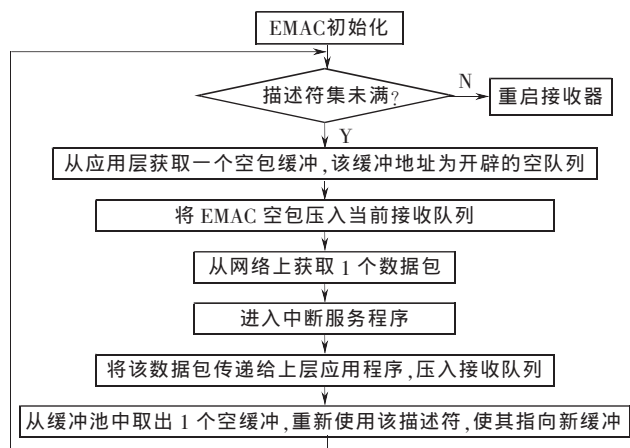


图2 接收流程图

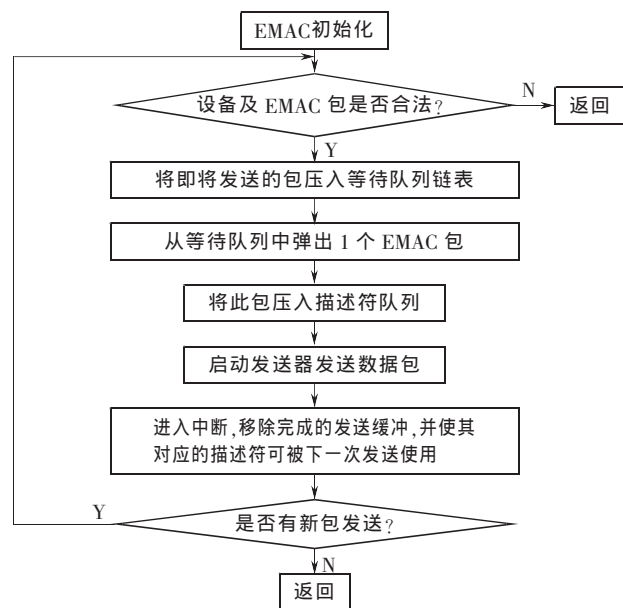


图3 发送流程图

4 精简协议栈的设计与实现

4.1 协议层的结构

嵌入式系统中实现的协议要根据各个系统的特点及功能来设计其独特的 TCP/IP 协议簇,实现与需要有关的部分,而不使用的协议则省略。在局域网上使用的很多功能,比如路由协议、SNMP 都不需要,只使用 ARP、IP、ICMP 和 UDP,可省略应用层协议。在用于视频数据传输的网络中,如果简化协议基于 TCP 的话,不仅开销较大,丢失分组将会造成播放延时的增大^[2]。并且视频流本身具有一定的错误恢复能力,所以本文的协议栈主要以 UDP 的发送和接收为设计和测试对象。

4.2 存储缓冲的管理

TCP/IP 栈和设备驱动使用包缓冲区发送及接收网络包数据。在该协议栈中,开辟 32 个包缓冲区构成的缓冲池,而以太网帧长度限制在 1 514 byte 或 1 518 byte(加入 CRC 校验),因此设置每个缓冲为 1 536 byte=(1 024+512) byte,这样分配可以对齐 Cache 边界,保证冲洗(flush)Cache 时,不会与其他缓冲区发生冲突,并且节省存储空间。32 个包缓冲头使用 EMAC 的 EMAC_Pkt 结构。两段缓冲均存放在片外 SDRAM 中。协议栈的数据处理和底层设备接口使用 sk_buff 的结构,编码如下:

```

struct sk_buff {
    unsigned char pad[2];
    unsigned int truesize; /* 缓存大小 */
    unsigned char *buf;
    unsigned char *data; /* 数据头指针 */
    EMAC_Pkt *pPkt;
    unsigned int len; /* 数据长度 */
};
  
```

其中包含 1 个底层 EMAC 的数据包指针,这样就可以方便地将 EMAC 接收的数据包放入协议栈的缓冲中。协议栈中的缓冲 skb 即是这种结构。

在每次发送 1 个数据包之前,都要首先分配 skb 的存储空间。从 EMAC 的空队列中弹出 1 个 EMAC_Pkt 的数据包,并将 skb 内的这个数据包指针指向它,然后对 skb 进行操作。在释放 skb 时,将这个数据包压入空队列中以便继续使用。

每层协议在封装帧头前,在数据区前预留这些帧头的空间,而数据区域不变。这样数据在各层协议之间传递都使用数据指针,从而避免了数据的重复搬移,以提高效率。当提交到下层协议时,将数据指针 pskb->data 向前移动 1 个帧头长度。而在接收数据时,每层协议去掉帧头,将数据指针向后移动 1 个帧头长度。

由于 I 帧大小一般超过了 1 514 byte,需要加入 IP 分片的功能。首先计算需要分片数,预留该传输层需要的帧头,填入传输层的帧头数据,然后从原始数据复制到 pskb->data,剩下的片预留 IP 层需要的帧头,填入帧头数据后发送。

ARP 的实现需要建立 1 个 ARP 列表,存放网络中主机(DSP 或 PC)的 ARP 信息。在收到 1 个 IP 包后,DSP 首先发送 1 个广播包,查找该报文中源 IP 对应的 MAC 地址,如果不在列表中,发送 ARP 请求,在接收到回应后,将 MAC 地址和对应的 IP 加入列表。

在以太网层,调用 csl 中 EMAC_sendPacket 发送数据。

加入类 BSD 套接字函数,封装底层函数,包括 socket, accept, bind, connect, recv, recvfrom, send, sendto 等,便于应用程序调用。

5 协议栈的改进

5.1 改进的 UDP 协议

鉴于参数集和 I 帧的重要性,为了提高传输的可靠性,保证正确接收端正确解码,并尽量使协议栈简单,采用类似 TCP 的重传机制,流程如图 4 所示。

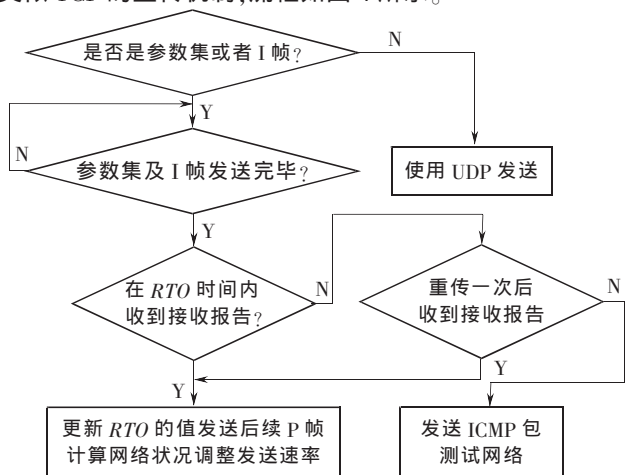


图4 改进的UDP传输

阈值 RTO 采用 TCP 的计算方法^[5]: $RTO = \beta \times RTT$, RTT 为平均往返时延。 $RTT = \alpha \times RTT_{old} + (1 - \alpha) \times RTT_{new}$, RTT_{old} 为上一次的平均往返时间, RTT_{new} 为新的往返时间。往返时间可根据发送时刻的绝对时间和接收到接收报告的绝对时间来计算^[5],每发送 1 个序列(IDR 帧更新)就计算 1 次。在小型网络中,为及时重传关键的数据包,取 $\alpha = 0.25, \beta = 1.5$ 。为测试简便,由几次实验估计,取 RTO 的值为 50 ms。

如果需要更可靠的传输,可以在接收端每收到 1 个数据包就发送 1 个 IP 数据包,内容为成功接收标志位,用于发送端判断是否重传以及接收端剩余的缓存空间,用于发

送端决定是否延迟一小段时间,等待接收端清理出足够的缓存空间再发送数据。

5.2 接收报告的设计

视频流的接收情况需要反馈给发送端以便客户端和服务端的交互,发送速率需要根据接收端接收和解码的速率来进行一些调整以达到匹配,还可以使发送端对出现的问题进行分析和采取一些解决措施。通常的反馈接收报告中设置固定的时间间隔发送接收报告^[1],缺乏灵活性,并且该时间需要大量实验来验证,时间太短会导致网络流量增大,时间太长可能无法及时使发送端获得网络状况并由此控制发送端调整发送速率。因此设计在接收端成功接收到参数集和 I 帧后,或者解码 1 个序列(下一个 IDR 帧到来)后,发送 1 个接收报告。发送端可以根据接收报告检测网络状况及延时抖动,然后相应的调整发送速率。如果在报文中加入各帧解码时间,还可以调整编码的码率,以达到编码和解码的速率的匹配。

为了提高兼容性和扩展性,参照使用 RTCP 报文的头部,以便其他能够解析 RTCP 包的应用程序识别。报头格式如图 5 所示。

V(2 bit)	P(1 bit)	RC(5 bit)	PT(8 bit)	Len(16 bit)
----------	----------	-----------	-----------	-------------

图5 报文头格式

图 5 中, V 为版本号,用于识别报文的合法性。 P 为填充位,为 1 时表示数据包末位有填充位。 RC 为报告位,数据包中所含有的报告数目。 PT 为类型,不同的值代表不同类型的包,可针对不同的用途设计不同内容的报文,以类型区分。例如,用于视频监控(基本档次),用于流媒体播放(扩展档次)等。 Len 为整个数据包的长度,不包括报头长。每种类型的报文有固定的长度,如果不符合可以直接丢弃。报文内容如图 6 所示。

图 6 中, T_n 为第 n 帧的传输时间。 n 由发送端的 I 帧周期决定,或者由接收端设定。丢包的序号由接收端根据发送的 RTP 数据包内容

报头(32 bit)
T0(32 bit)
T1(32 bit)
...
$T_n(32 bit)$
丢包的序号(8 bit)
丢包率(16 bit)
控制标志位(8 bit)

图6 报文内容

决定。丢包率表示从接收到第一个数据包开始的时间内丢掉的数据包总数与累积接收到的数据包的总数的比例。控制标志位为 0 时,不变;为 1 时,提高码率;为 2 时,降低码率;为 3 时,关掉此连接。

6 测试结果

测试使用网线直连 DSP 和 PC,如果 PC 网卡没有自适应交叉线/直通线功能,则必须制作交叉线来进行

测试。

在 PC(IP 地址为 192.168.1.20)上使用控制台 ping 配置的 DSP 的 IP 地址(192.168.1.10),并且使用 wireshark 软件抓取 PC 网卡上的数据包。测试结果如图 7 所示。

1	0.000000	192.168.1.20	192.168.1.10	ICMP	Echo (ping) request
2	0.000100	TexasIns_32:17:42	Broadcast	ARP	Who has 192.168.1.20? Tell 192.168.1.10
3	0.000116	Wistron_ea:05:0c	TexasIns_32:17:42	ARP	192.168.1.20 is at 00:16:d3:ea:05:0c
4	0.000338	192.168.1.10	192.168.1.20	ICMP	Echo (ping) reply
5	5.232653	192.168.1.20	192.168.1.10	ICMP	Echo (ping) request
6	5.232774	192.168.1.10	192.168.1.20	ICMP	Echo (ping) reply
7	6.232901	192.168.1.20	192.168.1.10	ICMP	Echo (ping) request
8	7.149762	192.168.1.10	192.168.1.20	ICMP	Echo (ping) reply
9	7.232665	192.168.1.20	192.168.1.10	ICMP	Echo (ping) request
10	7.232788	192.168.1.10	192.168.1.20	ICMP	Echo (ping) reply

图 7 wireshark 测试结果

首先,PC 网卡向 DSP 发送第 1 个 ping 请求,DSP 接收到后发送 ARP 请求,PC 回应自己 IP 对应的网卡地址,DSP 回应 ping 请求。表明 ARP 协议和 ICMP 协议正确实现。

在实际视频序列传输中,采用 foreman,news 的 QCIF 以及 CIF 序列测试,解码端能够正确解码播放,并且达到设定的帧率。对于误码重传仿真测试,本文不再详述。

7 小结

基于 DSP 的视频传输有着广阔的应用空间,笔者提出了一个精简协议栈的设计和实现方法,以及与 EMAC

底层驱动的连接,并针对 H.264 视频流的特点做了一些改进。实验证明,该协议栈可满足多种应用需求。

参考文献:

- [1] 王力生,梅岩,曹南洋.轻量级嵌入式 TCP/IP 协议栈的设计[J].计算机工程,2007,33(2):246-248.
- [2] 毕厚杰.新一代视频压缩编码标准——H.264/AVC[M].北京:人民邮电出版社,2005.
- [3] Texas Instruments.TMS320C6000 DSP EMAC/MDIO module reference guide (Rev. A)[EB/OL].[2009-10-20]. <http://focus.ti.com/lit/ug/spru628a/spru628a.pdf>.
- [4] 方怀东,陈启美.基于 TMS320DM642 的嵌入式 TCP/IP 协议栈的实现[J].电子技术应用,2006,32(9):25-29.
- [5] 谢希仁.计算机网络[M].4 版.大连:大连理工大学出版社,1989.

作者简介:

赵川(1985-),硕士生,主研 DSP 技术和视频编码技术;

伍瑞卿(1977-),博士生,主研 DSP 技术和多媒体通信;

樊丰(1963-),副教授,硕士生导师,主要研究方向为数字通信系统,数字视频技术。

责任编辑:任健男

收稿日期:2010-02-05

(上接第 81 页)

表 2 两种结构性能对比

硬件结构	数据带宽/(byte/时钟周期)	PE数	全搜索计算周期数	方法 1 计算周期数	方法 2 计算周期数	Slice 数
速度优先	32	256	256	128	80	1 937
资源优先	16	16	4 096	2 048	1 280	736

本文对 H.264 运动估计算法进行优化,提出两种适合硬件实现的搜索方法,表 1 的实验数据表明,在不损失图像质量的同时,明显减少了运算量,提高了速度。同时与硬件相结合,给出速度优先和资源优先的两种结构,两种结构在速度和资源方面各有优势,设计出运动估计的硬件加速器,在 Xilinx XUP Virtex-II PRO 开发板中做了验证,结果表明该加速器能正确完成 H.264 运动估计的数据处理功能。

参考文献:

- [1] 毕厚杰.新一代视频压缩编码标准——H.264/AVC[M].北京:人民邮电出版社,2005.
- [2] YANG K M,SUN M T,WU L. A family of VLSI designs for the motion compensation block-matching algorithm[J].IEEE Trans. Circuits and Systems,1989,36(10):1317-1325.

- [3] CHEN C Y,HUANG C T,CHEN Y H,et al. Level C+ data reuse scheme for motion estimation with corresponding coding orders[J].IEEE Trans. Circuits and Systems for Video Technology,2006,16(4):553-558.
- [4] 王睿,林涛,林争辉,等.一种 H.264 运动估计器的 VLSI 设计[J].微电子学与计算机,2004(11):153-157.
- [5] 黄卫锋,桑红石,郑兆青,等.用于 H.264 的高性能整像素运动估计 VLSI 的设计[J].微电子学,2007(4):260-264.
- [6] CHEN T C,HUANG Y W,CHEN L G. Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture [EB/OL].[2009-12-20]. <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F9255%2F29377%2F01329261.pdf%3Farnumber%3D1329261&authDecision=-203>.
- [7] HSU Meiyun,CHANG Haochieh,WANG Yichu,et al. Scalable module-based architecture for MPEG-4 BMA motion estimation[EB/OL].[2009-12-20].<http://ntur.lib.ntu.edu.tw/bitstream/246246/2007041910021255/1/00921053.pdf>.
- [8] 石磊,林涛,焦孟草. H.264/AVC 硬件解码器设计及其验证策略[J].微电子学,2006(2):16-19.

作者简介:

李本斋(1984-),硕士生,主研视频编码、集成电路设计;

吴中从(1965-),副教授,硕士生导师,主要研究方向为数字信号处理、嵌入式系统技术。

责任编辑:任健男

收稿日期:2010-02-03