

DM642 上 TCP/IP 协议的实现及性能测试

Implementation and Performance Analysis of TCP/IP Stack Based on DM642

(清华大学 电子工程系) 杜文 沈勇 唐昆

Du, Wen Shen, Yong Tang, Kun

摘要:高性能 600MHz 的 TMS320DM642 是可编程的数字多媒体处理器,为了了解其对于多媒体数据通信的支持,本文针对 DSP 芯片上的网络开发工具 NDK 进行了研究,测试了其在 UDP 传输过程中的 CPU 效率,对其在不同传输速率和二级缓存大小的条件下的表现给出了比较。说明了 DM642 是一款很适用于多媒体通信的数字信号处理器。

关键词:DSP;DM642;NDK;TCP/IP;UDP;多媒体通信

中图分类号:TP393.1 **文献标识码:**A

Abstract: 600MHz TMS320DM642 is an excellent Digital Signal Processor. To support its function of multimedia data transmission, our work focuses on the Network Development Kits- NDK on it. We have a test on DM642 CPU's efficiency of transmission of UDP data package; make comparison of the performance under different hardware conditions, including different transmission speed and different size of L2 cache. DM642 performs well in multimedia communications area.

Key Words: DSP;DM642;NDK;TCP/IP;UDP;Multimedia Communications

随着互联网技术的发展,嵌入式的多媒体终端日益普及,在嵌入式芯片上多媒体通信的研究逐渐成为热门的课题。多媒体尤其是视频图像的实时通信有数据量大,延时要求严格等特点,而嵌入式芯片在处理能力、存储容量上与通用芯片都有差距,因此能否利用有限的资源实现高效率的通信协议,是嵌入式芯片能否实现多媒体通信的关键。

TI(Texas Instrument)公司是世界领先的 DSP 制造商,它们的 C6000 系列芯片在嵌入式芯片市场获得了巨大的成功,DM642 就是该系列的最新产品。而新推出的 NDK 开发套件在 DM642 上实现了高效率的 TCP/IP 协议栈,我们选择 DM642 为平台,研究和分析了 NDK 实现的高效率通信协议。

1 开发平台和测试环境的介绍

1.1 TI DM642 DSP 开发平台

TI 公司的 DM642 是一款专门面向多媒体应用的专用 DSP。该 DSP 内部时钟高达 600MHz,8 个并行运算单元,最大处理能力达到 4800MIPS,外部总线时钟 100MHz。DM642 的芯片集成了 64 个 32bit 的通用寄存器,能够在—个时钟周期内处理 4 个 16bit 的乘法和 8 个 8bit 的乘法。为了增强多媒体功能,芯片还集成了 3 个高精度、可配置的视频端口,10/100Mbps 的以太网 MAC,面向音频应用的多通道音频串口

(McASP),66MHz 32bit 的 PCI 以及拥有 64 个通道的增强型 DMA 等接口。这些都使得 DM642 DSP 特别适用于音视频的处理和通信。

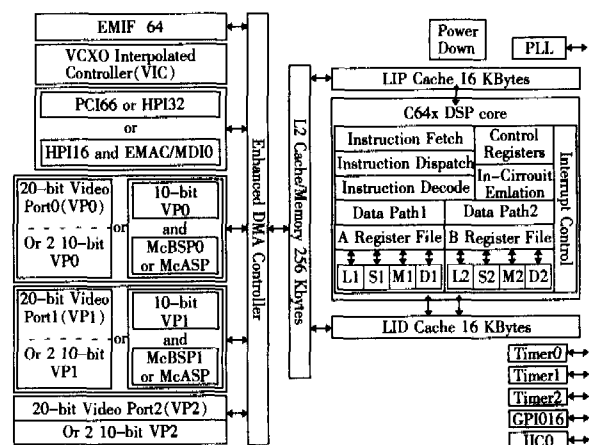


图 1 DM642 DSP 处理器

DM642 的高性能还得益于 DSP 内部的两级高速缓存 (cache) 的结构设计,芯片的第一级缓存包括 16KByte 的程序缓存 (L1P) 和 16KByte 的数据缓存 (L1D),第二级缓存 (L2) 有 256KByte,程序空间和数据空间是共用的。它可以设置成存储单元 (memory),高速缓存 (cache),或者是这两者的结合,具体的分配可以由程序员配置。

CPU 只对 L1 的数据进行访问,程序代码和数据必须经过外部存储器到 L2, L2 到 L1 的逐级搬移才能被

杜文:讲师

基金项目:国家自然科学基金资助项目(60272020)

CPU 访问。L2 存储单元和外部设备的数据交换由功能强大的增强型 DMA 控制器控制,因此在 CPU 处理片内的数据时可以通过 EDMA 把片外的数据倒入片内,达到同步工作以提高效率。由于内部存储器的工作频率与 DSP 内部时钟同频,而远远高于片外存储器的工作频率,这就解决了 DSP 外部时钟频率小于内部时钟频率的问题。有实验表明,合理利用两级缓存配合低工作频率的外部存储器,系统的效率能够达到全部使用高工作频率的内部存储器的 80%到 90%。

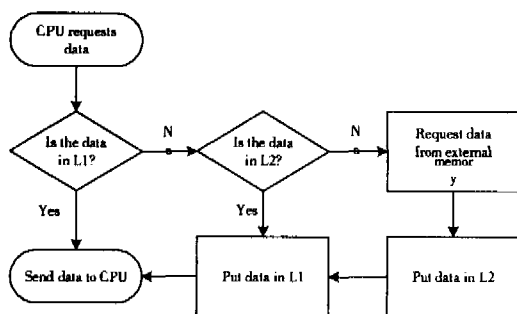


图 2 DM642 两级缓存工作原理

同时 DM642 可与 TI 的 C64x DSP 目标代码完全兼容,这大大降低了客户的系统成本,简化了开发进程。

闻亭(Wintech)公司是 TI 的第三方合作伙伴,该公司推出的 DM642EVM 板是一个基于 DM642 芯片的低成本的独立开发平台,可满足视频设计人员利用最新音频和视频算法套件快速进军市场的需求,适用于 VoIP、视频点播(VOD)、多通道数字录像应用,以及高质量的视频编解码解决方案。板上有一个 600MHz 的 DM642 芯片,还集成了 4MB Flash,32MB 100MHz SDRAM,另外还有音视频采集和输出端口,10/100M 以太网接口,仿真器的 JTAG 接口。EVM 板将高性能的 DSP 内核与集成的音频、视频及连接性选项进行了完美的结合。我们的 TCP/IP 协议栈就是在该板上实现的。

1.2 TCP/IP 网络通信协议栈简介

随着近几年网络技术的发展,以 TCP/IP 为代表的通用网络技术已经成为网络通讯的标准。

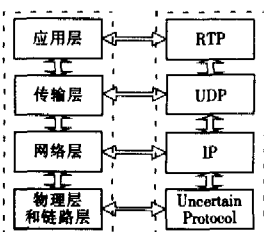


图 3 RTP/UDP/IP 协议栈



图 4 NDK 测试环境示意图

上图是典型的 IP 分层协议栈,底层协议对通信的影响主要表现在 MTU(Maximum transmission unit)的大小对上层打包的制约,例如以太网的 802.3 协议和无线网的 801.1 协议对包长都有限制,否则就会丢包,因此在上层协议的实现时要注意这个问题。IP 是最为成功、应用最为广泛的网络层协议,它提供一种尽力传

送的服务。在传输层,IP 网络通常使用两种协议,即 TCP 和 UDP。TCP 提供一个面向字节的,有保证的传输服务。在差错控制方面,主要使用重传和超时等机制。由于它的延时不可预测性,并不适合实时通信业务。UDP 提供一个简单的不可靠的数据报服务。UDP 报头中包含校验和,能够用来检测和丢弃包含误码的包。它适合进行实时通信,本文就是研究 DM642 用 UDP 传输数据的效率。RTP 是典型的基于 UDP/IP 的应用层传输协议,它是面向会话的,一个会话与一个传输地址(IP 地址+UDP 端口)相关联。一个 RTP 包包括 RTP Header,可选的 Payload Header,及 Payload。

2 TI 通信开发套件 NDK

2.1 NDK 开发套件的高效率设计

为了加速其高档 DSP 的网络化进程,TI 公司结合其 C6000 系列芯片推出了 TCP/IP NDK (Network Developer's Kit)开发套件。

NDK 主要的组件包括:(1)支持 TCP/IP 协议栈程序库。其中主要包含的库有:支持 TCP/IP 网络工具的库,支持 TCP/IP 协议栈与 DSP/BIOS 平台的库,网络控制以及线程调度的库(包括协议栈的初始化以及网络相关任务的调度)(2)示范程序。其中主要包括 DHCP/Telnet 客户端,HTTP/数据服务器示范等。(3)支持文档包括用户手册、程序员手册和平台适应手册。

NDK 采用紧凑的设计方法,实现了用较少的资源耗费来支持 TCP/IP。从实用效果看,NDK 仅用 200~250K 程序空间和 95K 数据空间即可支持常规的 TCP/IP 服务,包括应用层的 telnet、DHCP、HTTP 等。为了最大限度地减少资源消耗,TI 为其 NDK 采用了许多特殊的技巧,重要的有:(1)UDP socket 和 RAW socket 不使用发送或接收缓冲区;(2)TCP socket 使用发送缓冲区,接收缓冲区依配置文件而定;(3)低层驱动程序与协议栈之间通过指针传递数据,不对包进行复制拷贝;4、设置专门的线程清除存储器中的碎片和检查存储器泄露。因此,NDK 很适合目前嵌入式系统的硬件环境,是实现 DSP 联网通信的重要支撑工具。

NDK 的开发环境是 TI 的开发工具 CCS(code composer studio)。它包含有实时操作系统 DSP/BIOS 和主机与目标板之间的实时数据交换软件 RTDX。

2.2 NDK 的配置和使用

在 CCS 下使用 NDK 需要在以下几点上做特别处理:

(1)设置 DSP/BIOS

PRD 设置主时钟。硬件抽象层的时钟驱动需要一个 100ms 启动一次的 PRD 函数作为主时钟,函数名是 llTimerTick()。

HOOK 为 TCP/IP 协议栈设置保存的空间。OS 库的任务调度模块需要调用 hook 来保存和调用 TCP/IP 协议栈的环境变量指针,这两个 hook 函数是 NDK_hookInit() 和 NDK_hookCreate()。

(2) 包含文件和库文件

请注意编译时需要包含库文件和文件路径,一般默认为 c:\ti\c6000\ndk\inc

(3) CCS 工程编译时的链接顺序

CCS 一般按照特定的顺序来链接目标函数和库文件,NDK 是对这个链接顺序很敏感的,错误的顺序可以导致重复定义符号甚至不正确执行等错误。为避免这个情况,可以在 CCS 里选择 Link Order" -> "build options 对话框,将文件按照一定顺序添加并且将库文件添置到连接顺序的最后,推荐的顺序为:NETCTRL.LIB,HAL_XXX.LIB,NETTOOLLIB,STACKLIB 和 OS.LIB。

在初始化启动协议栈之前,要为其分配一块工作内存(SDRAM),命令是 _mmBulkAllocSeg(EXTERN1)。还要调用 fdOpenSession()来初始化文件指针向量表,否则创建 socket 的时候将出现错误。

我们将发送/接收设置定义为一个任务,在创建任务句柄以前,我们应该用 NC_SystemOpen()打开网络功能并进行设置,在系统关闭前也要进行相应的处理。

使用 NDK 提供的 socket API 函数需要注意下面一些问题:(1)NDK 中对 socket API 通过一个文件指针接口与操作系统相连接,因此要调用文件指针向量表初始化和关闭函数对文件系统进行相应操作。(2)NDK 中并没有提供 windows API 中强大的 select 函数,但是可以用 fdselect 实现一些相应的工程。可以相互对应得 API 函数还有 NDK 中的 fdclose 和标准的 close, NDK 中的 fderror 和标准的 errno。(3) NDK 提供了很多网络工具的支持的函数,比如和 DNS 相关的一些函数,可以代替标准 API 中的 getpeername, gethostname 等。另外还有关于 ICMP 的一些函数可以用来支持组播,但是只支持作为组播用户,不能支持作为组播服务器。

3 NDK 传输 UDP 数据包效率测试及性能分析

3.1 测试平台结构

我们研究了在 NDK 下 CPU 对 UDP 数据包发送接收的效率,这个测试分成两部分:一部分是测试从 DM642 向 PC 机发送 UDP 数据包时,在不同的传输速率和不同的 L2 cache 大小时的 CPU 占用率,另一部分是测试 DM642 接收从 PC 机发送来数据包时,在不同的传输速率和不同的 L2 cache 大小时的 CPU 占用率。我们所使用的工具是在 CCS 下的 NDK 提供的 socket API 函数和在 visual studio 下提供的 winsocket API。图 4 是测试环境的示意图。

3.2 测试平台的配置和实现

由于接收和发送程序十分相似,我们仅以发送程序举例。创建发送数据的程序为一个任务,在 DSP/BIOS 中,任务对象就是被 TSK 模块管理的线程。TSK 模块根据任务的优先级和当前的执行状态动态的调

度。DSP/BIOS 总共有 15 个任务优先级可以使用,并且提供了一组函数来操纵任务对象,包括建立、删除、设置任务对象。任何任务对象都处于下面几种状态之一:运行态,就绪态,阻塞态,终止态。

在这个工程中,我们在网络控制的程序中进行任务的创建,图 5 是创建任务的流程图:

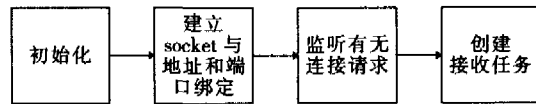


图 5 传输任务创建流程图

其中创建任务的语句为:TaskCreate (tsk_udp, "udp_video", 5, 0x1000, peer_addr, 12345, 12345)。理论上,可以通过设置两个 task 的方法来增加数据传输的速率,但是注意这两个 task 应该用不同的端口进行传送。任务调度的应用程序为:

```

static void tsk_udp( IPN IPAddr, int PeerPort, int LocalPort)
{
    .....
    // 创建 socket
    s = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
    .....
    // 设置要绑定的地址端口属性
    bzero( &sinl, sizeof(struct sockaddr_in) );
    sinl.sin_family = AF_INET;
    sinl.sin_len = sizeof( sinl );
    sinl.sin_port = htons(LocalPort);
    //绑定 IP 地址和端口
    if( bind( s, (PSA) &sinl, sizeof(sinl) ) < 0 )
    { goto exit_tsk; }
    //设定目的地址端口属性
    bzero( &sinl, sizeof(struct sockaddr_in) );
    sinl.sin_family = AF_INET;
    sinl.sin_len = sizeof( sinl );
    sinl.sin_addr.s_addr = IPAddr;
    sinl.sin_port = htons(PeerPort);
    .....
    // 分配工作缓冲区
    if( ! (pBuf = mmBulkAlloc( 1024 )) )
    { goto exit_tsk; }
    // 一下开始发送数据
    for(;;)
    { // 填充发送数据的缓冲区
      *(int*)pBuf=send_udp_count++
    }
    // 发送数据
    if( sendto( s, pBuf, 1000, 0, &sinl, sizeof(sinl) ) < 0 )
    { goto exit_tsk; //break; }
    // 清空数据区
    mmZeroInit( pBuf, (uint)test );
    //设置发送数据率
    TaskSleep(8); // 1Mbit/s
}
.....
  
```

测试里面有两个关键参数需要设置,一个是发送(接收)的数据率和 DM642 内部第二级缓存的大小。收发数据率可以通过改变任务挂起的时间间隔长度来改变。系统函数 TaskSleep(n)表示每隔 n 毫秒执行一次发送,我们设定每次发送 1000 Byte 的数据,这样 TaskSleep(8)表示 1Mbit/s 的传输速率,TaskSleep(4)表示 2Mbit/s 的传输速率,以此类推。

而 L2 cache 大小的改变可以通过以下语句来设置:

CACHE_setL2Mode(CACHE_64KCACHE)表示设置了 64K L2 Cache;CACHE_setL2Mode(CACHE_128KCACHE)表示设置了 128K L2 Cache,以此类推。

3.3 测试结果和性能分析

我们在 DM642 评估版上,采用标准的 recvfrom 函数进行数据接收,以无连接的 UDP 协议与 windows PC 进行相互传输。对不同传输速率和不同大小的二级缓存下 CPU 的占用率进行了比较。

CPU 占用率=空闲周期可完成的低优先级任务/执行传输任务时可完成的低优先级任务

其中接收和发送数据均设为每次 1000 Byte,评测结果在下面四个图表里面显示出来。

表 1 由 DM642 发送 UDP 数据包的 CPU 占用率(%);表 2:由 DM642 接收 UDP 数据包的 CPU 占用率(%)

传输速率 (Mbit/s)	64KCache	128KCache	256KCache
0.4	0.29	0.21	0.19
0.8	0.58	0.45	0.38
2	1.1	1.02	0.96
4	2.64	2.26	1.88
8	5.11	4.38	3.64
16	9.86	8.25	6.89

传输速率 (Mbit/s)	64KCache	128KCache	256KCache
4	0.2	0.13	0.14
0.8	0.35	0.28	0.27
2	0.82	0.7	0.67
4	1.62	1.34	1.34
8	3.65	2.69	2.68

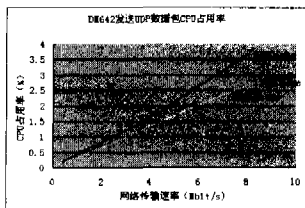


图 6 DM642 发送 UDP 数据包 CPU 占用率比较图

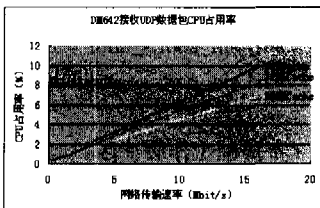


图 7 DM642 接收 UDP 数据包 CPU 占用率比较图

从上面的比较可以看出 DM642 发送和接收数据包时的 CPU 占用率均随着网络传输速率的增加而提高,而且基本上呈线性关系。因为收发数据是对数据简单的搬移,它的复杂度是随着数据的增加而线性增长的,在高速缓存一定得情况下 CPU 的占用率线性增加。

而第二级缓存的大小对 CPU 的占用率也有影响,一般而言是 L2 cache 越大,CPU 占用率越小,而且随着收发数据率的变大而显得更加明显,这个得益于 DM642 两级缓存的工作原理和强大的 DMA 功能。

L2 cache 增大带来的另一个影响是 CPU 片内存容量的减少,使得片内能放下的代码段和数据段就

比较少,这样反而会减缓程序的运行速度,这在处理复杂的编解码程序,数据段和代码段比较多时尤为明显,这就需要程序员根据实际情况统筹安排合理配置。

4 总结与展望

为了研究在嵌入式芯片上进行高速多媒体通信的可行性,本文讨论了 TI DM642 上 NDK 设计的原理,并进行了与 PC 机实时数据通信的测试。测试表明 TI 公司推出的 NDK 套件在 DM642 芯片上实现了高效率的 TCP/IP 传输协议,即使高达 16Mbit/s 的传输速率,CPU 的占用率也不到 10%,这使得它完成多路视频传输也绰绰有余,这是 DM642 能够广泛应用于各种多媒体通信设备和终端的有力保证。再配合 DM642 芯片强大的多媒体处理功能,使得其在多媒体通信市场上有广阔的前景。值得我们去研究和关注。

参考文献:

- [1]TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor Data manual
- [2]TMS320DM642 Evaluation Module,Technical Reference
- [3]Texas Instruments TMS320C6000 TCP/IP Network Developer's Kid Programmer's Reference Guide
- [4]李素侠,段友祥. 嵌入式 TCP/IP 协议的分析与研究[J]微计算机信息 2005,7:52-53
- [5]W.Richard.Stevens. TCP/IP Illustrated. 北京:机械工业出版社 作者简介:杜文(1973.3-),男,清华大学讲师,主要研究方向:多媒体通信和终端。沈勇(1981.5-),男,清华大学硕士研究生,主要研究方向:嵌入式系统上的视频编码和多媒体通信。唐昆(1945.10-),男,清华大学教授。(100084 清华大学电子工程系 微波与数字通信技术国家重点实验室)杜文

(投稿日期:2005.9.11) (修稿日期:2005.10.21)

(接 50 页) 使 PCR-CE 微流体芯片系统在实际应用中更加方便。

参考文献:

- [1]王田苗,嵌入式系统设计与实例开发—基于 ARM 微处理器与 uC/OS-II 实时操作系统,2002
 - [2]黄元峰,李育清,姜生元.基于 μ C/OS 的嵌入式系统应用开发研究[J].微计算机信息,2006:2,100-103
 - [3]Yongsheng Ding, and coworkers,Fuzzy Self-Tuning Immune Feedback Controller for, 0-7803-5877-5/00 02 000 IEEE.
- 作者简介:王海峰(1978.7-)汉族,男,硕士,专业:控制理论控制工程;张玉林:(1947-)汉族,男,教授,博导,研究方向:微机电系统,电子束
- Author brief introduction:**Wang,Haifeng (1978.7-),Male,Jinan, city,Shandong,province,Postgraduate,major:control science and engineering;Zhang,Yulin (1947-), Male,Shandong province,Professor,Major:MEMS,ELECTRON BEAM

通讯地址:

(250061 济南市经十路 73 号山东大学控制学院电子束研究所)王海峰

(投稿日期:2005.9.10) (修稿日期:2005.10.21)