

DM642 嵌入式网络接口开发设计

梁 迅, 熊水东

(国防科技大学光电科学与工程学院, 长沙 410073)

摘要:给出了一个基于 TMS320DM642 DSP 芯片的嵌入式网络实现方案, 实现了在大规模信号处理情况下的 100Mb/s 网络数据传输接口, 给出了硬件连接框图和软件流程。同时对 DM642 网络接口设计过程中的相关问题进行了讨论, 如与硬件兼容问题、初始化问题、中断和同步问题、内存溢出问题等, 并给出了相应的解决办法。

关键词: DM642; TCP/IP; 网络; NDK

Embedded Ethernet Interface Development with DM642

LIANG Xun, XIONG Shui-dong

(College of Optoelectronic Science and Engineering, National University of Defense Technology, Changsha, China 410073)

【Abstract】 An embedded network project based on TMS320DM642 is proposed, which realizes a 100Mb/s network interface in massive signal processing, including hardware connecting means of plot of pane and software flowchart. Some related problems in developing the DM642 network interface are discussed, such as hardware-software compatibility, initialization, interruption and synchronization, out of memory, etc. The ways to solve these problems are put forward too.

【Key words】 DM642; TCP/IP; network; NDK

1 概述

随着网络技术的飞速发展, 嵌入式设备接入网络已经成为一种趋势, IP 电话终端、IP 网关、IP 视频会议、IP 机顶盒, 网络摄像、工业网络控制等等, 都需要 DSP 芯片同时具有数字信号处理和网络传输的功能, 在军事用途方面, 传感器需要通过网络来实现信息的实时获取和远程控制, 从而争取战场的主动。就目前的技术来讲, TCP/IP 协议是被广泛支持的一种协议, 在嵌入式设备中实现了 TCP/IP 协议, 就可以很方便地将设备接入网络中。

本文在德州仪器公司 TMS320C6000 系列芯片的基础上实现了一套高速信号处理——网络传输系统, 如图 1 所示。该方案以 TMS320C6713 浮点 DSP 作为浮点信号处理的主芯片, 同时采集多路 AD 数据并进行信号处理, 而使用 TMS320DM642 定点 DSP 实现网络接口, 通过网络收发芯片 (LXT971ALC) 将处理结果传送到计算机。两个 DSP 芯片之间使用 FIFO (IDT72V263) 进行数据通信。该系统实现了 100Mb/s 的网络收发性能, 并达到了实时数据传输的要求。本文根据在实践中积累的一些经验, 对 DM642 的嵌入式网络接口在硬件设计和软件编写进行了介绍, 并对软件编写上的一些问题进行了讨论, 提出了相应的解决办法。

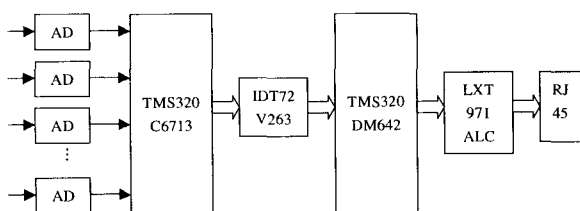


图 1 信号处理——网络传输系统

2 DM642 的网络接口硬件

在 DM642 没有出现之前, 实现 C62x 系列和 C67x 系列 DSP 的网络接口需要以网络控制芯片为核心的独立电路模块 (一般为网络子卡), 如第三方提供的 MX98728 网络卡和 LAN91C111 网络卡^[1]等, 通过一定的接口协议与 TI 开发板相连, 但购买子卡在设计上不够灵活, 且价格较贵, 如果独立开发则在硬件和软件上都比较复杂, 周期也较长。DM642 芯片出现后, 由于其自带网络控制器 EMAC, 在硬件上只需连接一个网络收发芯片 Intel LXT971ALC 即可, 如图 2 所示, 大大简化了开发过程, 因此成为高速嵌入式网络连接很好的选择。

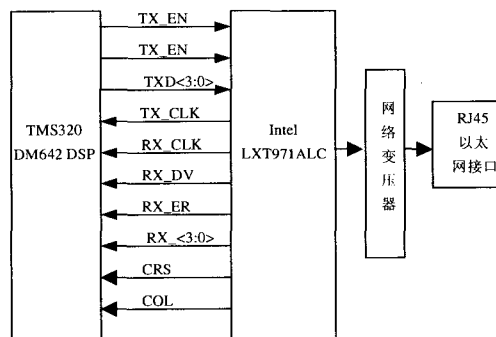


图 2 DM642 网络接口电路

Intel LXT971ALC 的作用只在于物理层网络数据包的收

作者简介: 梁 迅(1980-), 男, 博士研究生, 主研方向: 光电工程, 信号处理; 熊水东, 博士、讲师

收稿日期: 2006-10-22 **E-mail:** liang428@163.com

发,而网络的设置和控制都在 DM642 本身自带的 EMAC 模块中,通过软件来完成^[2]。TI 公司提供的 DM642EVM 评估板采用了上述接口方式,它提供了一些资料可供参考^[2,3]。

3 软件设计

TI 公司为 C6000 系列 DSP 提供了高效而又稳定的操作系统——DSP BIOS 以及相应的集成开发环境 CCS(code composer studio),并在此基础上开发了一套网络开发程序套件(network developer's kit, NDK)^[1],将网络协议封装起来作为 API 供用户调用,大大提高了网络开发的效率。该 API 函数遵循了国际上通用的 socket 编程标准,在 Windows 和 Unix、Linux 上开发的网络代码,稍加改动就可以应用到 DSP 中去。

3.1 软件流程

在本系统中,DM642 主要完成两个任务:FIFO 数据的接收和数据的网络发送。FIFO 数据存满时,会通过硬件中断通知 DM642,DM642 开始 EDMA 方式的数据传输,在不占用 CPU 资源的情况下将 FIFO 数据传入内存中指定的位置。此时将通过 SEM 标志通知网络发送程序,将该数据取走并进行网络发送。因此整个程序按照功能大体分为 3 部分:初始化程序对硬件中断和网络进行了初始化和设置;EDMA 中断程序中通知网络发送程序,该数据已经准备好;在网络发送程序中对数据进行打包发送,各个部分的流程如图 3。

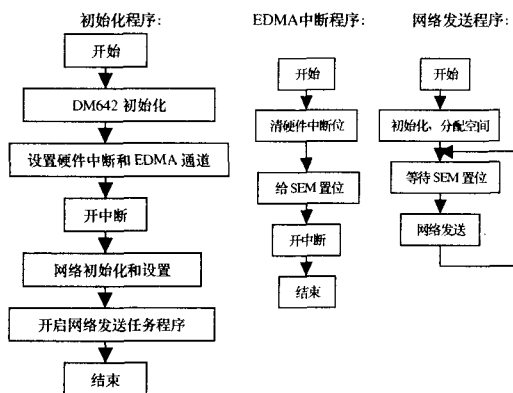


图 3 网络程序流程

3.2 开发中的一些问题

在 NDK 的 API 基础上编写网络通信程序,开发周期短,效率高,虽然如此,在开发过程中,还是遇到了一些问题,下面分别加以讨论。

3.2.1 程序与硬件兼容问题

在开发过程中,最好又最快的方法是在例子程序的基础上进行修改。TI 提供了 DM642EVM 开发板和基本网络功能的示例程序,方便了开发。但是其中有些函数只适用于该板卡,硬件稍加改动就不能很好地运行。因此给开发带来了软硬件的兼容问题:

(1)地址空间分配问题。

根据开发者不同的需要,板上使用的存储芯片大小及地址分配不尽相同,因此须在 CCS 中设置*.cdb 文件,在 MEM 选项中重新分配存储空间。对于网络开发,DM642 本身自带的 256KB 内存(L2 存储区)是不够的,必须外接 RAM。DM642 的 256KB L2 存储空间(L1 是内部高速缓存)即可以做 L2 缓存,也可以做内部 RAM,但在网络应用情况下,至少开辟 64KB L2 空间作为缓存,推荐值为 128KB 或 256KB。缓存越大,网络性能越好^[4]。下面的程序用来设置 L2 缓存大小,一

般写在程序初始化部分:

```

CACHE_clean(CACHE_L2ALL, 0, 0); //清 L2 缓存
CACHE_setL2Mode(CACHE_128KCACHE); //设置 128KB L2
//缓存
    
```

本方案中使用了 128KB 内部缓存,目的在于留下一部分内部空间接受 EDMA 传输的 FIFO 数据。

(2)I²C EEPROM 问题

另外,在 EVM642 评估板上装有 EEPROM(24WC256),使用 I²C 的接口方式,用来存储一些常用的参数。在示例程序中,网络的物理地址就保存在 EEPROM 中,在程序初始化的部分读入该地址:

```

EVM642_EEPROM_read(0x00007F00,(Uin32)bMacAddr, 8);
    
```

如果自己开发的电路中觉得 I²C ROM 没有必要,也可以省掉该部分,那么可以直接在程序中给 MAC 地址赋值,例如:

```

bMacAddr[0]=0x12; bMacAddr[1]=0x34; bMacAddr[2]=0x56;
bMacAddr[3]=0x78;
bMacAddr[4]=0x9a; bMacAddr[5]=0xbc;
    
```

如果没有 I²C ROM,这一步是必须的,否则程序无法运行。

3.2.2 程序初始化问题

TI 的例子程序中提供了两个初始化函数:EVMDM642_LED_init()和 EVMDM642_init()。这两个函数只适用于 DM642EVM 板,根据笔者的实践,很可能与板上的 FPGA 有关,而该 FPGA 的内部逻辑没有公开,所以硬件改变后这两个函数就不能够正确运行。而 TI 公司并没有提供它们的源程序,开发者不能进行相应的修改,因此最好不用这两个函数,经过实践证明,在开发板上去掉这两个函数对网络性能影响不大。

在程序初始化部分,对网络性能有较大影响的是以下两个函数:CACHE_enableCaching (CACHE_EMIFA_CE00)和 CACHE_enableCaching (CACHE_EMIFA_CE01),它们的作用在于开辟外部空间作为缓存,CE00 对应外部 RAM 的 0x80000000~0x80FFFFFF 空间,而 CE01 对应外部 RAM 的 0x81000000~0x81FFFFFF 空间。在开发板上的实践证明,这两个函数对网络速度影响极大。实测结果如图 4~图 7。

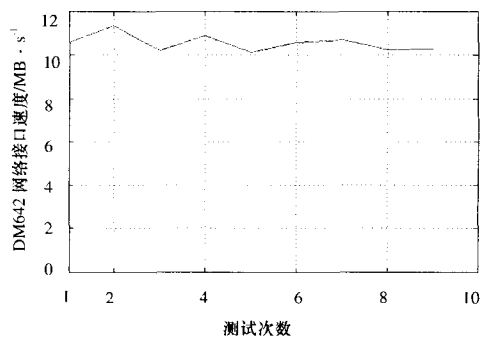


图 4 CE00 和 CE01 都作为缓存的情况

千兆网指速度峰值可以达到约 100Mb/s。换算成字节后,约达到 12 MB/s~13MB/s。从图 4~图 7 可以看出,在 CE00 和 CE01 全部缓存后,速度接近该最大数值。在只有 CE00 缓存时,速度下降了接近一半,而只有 CE01 缓存时,速度则下降了 7、8 倍之多,两者都不缓存时速度与只有 CE01 缓存时相近。由此可见 CE00 是否缓存对网络速度的影响巨大,而 CE01 即是否缓存也对速度有一定的影响。CE00 和 CE01

同为外部 RAM 的空间,对网络性能却有不同的影响,具体原因还有待研究。

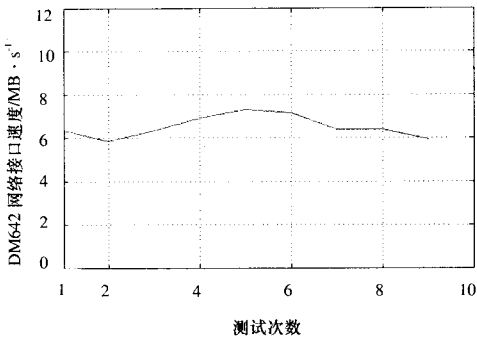


图5 只有 CE00 作为缓存的情况

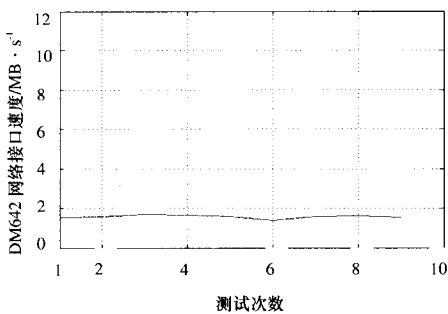


图6 只有 CE01 作为缓存的情况

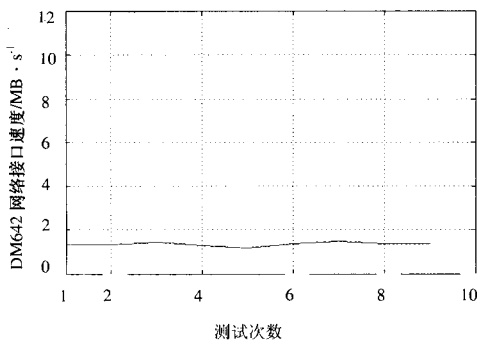


图7 CE00 和 CE01 都不缓存的情况

3.2.3 中断和同步问题

在本方案中,DM642 需要实时地将 FIFO 数据读入、打包、网络发送。它接收 FIFO 的数据是被动的,因此,在 FIFO 数据存满之后,必须通过外部硬件中断来通知 DM642。在该中断中触发 EDMA 采集,在不占用 DSP 资源的情况下,高速的将 FIFO 数据读入内部 RAM 中。而网络程序中的很多函数(如 fdselect(), send()等)带有阻塞性质,不能在中断程序中执行,必须单独开辟任务。因此,存在 FIFO 数据读入和网络数据发送之间的协调同步问题。

FIFO 数据读入和网络数据发送这两个程序进程之间,共享内存的通信方式最为快速,而其中同步问题最为突出。

TI 提供了进程之间相互协调的工具:信号灯(semaphore),它起源于 Unix,是专为解决进程间的同步问题而设计的。EDMA 采集 FIFO 数据完毕后会触发硬件中断,此时数据已经存入内部 RAM 中,在中断程序里需调用 SEM_post(),将信号灯置位。网络发送程序则等待信号灯的有效(SEM_pend()),信号灯置位后开始发送数据。使用这种方式,避免了网络程序对数据有效性的不断查询,提高了效率。

需要特别注意的是,硬件中断(HWI)中必须考虑保存和恢复使用过的寄存器。DSP/BIOS 提供了汇编语言宏 HWI_enter 和 HWI_exit 来执行此项功能,即如果使用汇编语言编写 HWI 程序,在进入中断时须调用 HWI_enter(),而在退出时须执行 HWI_exit()。如果使用 C 语言编写 HWI 服务程序,则可以使用 DSP/BIOS 提供的“HWI Dispatcher”,它会自动执行 enter 和 exit 操作。实践证明,没有设置 dispatcher 的程序很不稳定,一般在运行一二分钟后就会崩溃。

3.2.4 内存溢出问题

开发中还会遇到内存溢出问题。程序运行一会儿后死机,错误标志是:“OOM”,即“out of memory”。这是一种常见问题,但产生的原因却可以很多,一般是由于过于频繁的创建和关闭“socket”,从而将很多“socket”置于“timewait”状态,耗尽了内存。解决的办法是设置 socket 的“SO_LINGER”选项,可以使用如下语句来实现:

```
setsockopt( stepactive, SOL_SOCKET, SO_LINGER, &timeout,
sizeof( timeout ) );
```

其中:stepactive 是目标 socket;timeout 数据结构则是等待的时间,可以设置为 1s 左右,即最迟 1s 后将该 socket 的内存完全释放。

3.2.5 其它常见问题

DM642 作为高端 DSP 芯片,开发起来有一定难度。文献[1](第 44 页)提供了常见问题的列表,比如调用 fdOpenSession()和 fdCloseSession()的问题,PRD 函数中设置 100ms 周期的问题等,资料上的绝大多数问题都会在实践中遇到,因此具有一定的参考价值。

4 结论

DM642 网络接口的硬件相对简单、开发较为方便、资料也比较丰富。在实践过程中绝大多数问题都可以得到很好的解决。实践表明,DM642 的网络接口在全速情况下可以接近 100Mb/s 的速度,而且十分稳定,在大规模嵌入式信号处理当中,将 DM642 作为高速网络数据输出接口,是一个很好的选择。

参考文献

- 1 Texas Instruments. Texas Instruments TMS320C6000 TCP/IP Network Developer's Kit User's Guide[Z]. 2003.
- 2 Texas Instruments. TMS320C6000 DSP Ethernet Media Access Controller (EMAC)/ Management Data Input/Output (MDIO) Module Reference Guide[Z]. 2004.
- 3 Spectrum Digital Inc. TMS320DM642 Evaluation Module Technical Reference[Z]. 2003.
- 4 Texas Instruments. NDK Getting Started Guide[Z]. 2003.
- 5 Texas Instruments. Texas Instruments TMS320C6000 TCP/IP Network Developer's Kit Programmer's Reference Guide[Z]. 2003.