

基于 NDK 的 DSP 网络设计

周恒国, 黄方

(电子科技大学 资源与环境学院, 四川 成都 611731)

摘要: 针对 TI 的 TMS320C6000 系列 DSP 芯片网络系统进行设计, 对比 OSI 模型阐述了 NDK 的 TCP/IP 模型, 并给出了网络硬件接口设计。根据 NDK 结构模型, 设计网络软件编程。对从事 DSP 网络设计的人员有一定指导作用。

关键词: DSP; NDK; TCP/IP; STACK

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2013)02-0054-03

DSP Network design based on NDK

Zhou Hengguo, Huang Fang

(School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: This paper aims at TI TMS320C6000 DSP chip network system design, and contrasting the OSI model describes the NDK TCP/IP model, and gives the network hardware interface design. According to the NDK structural model, designs the process of network software programming. It has a certain extent significance for engaging at DSP network design.

Key words: DSP; NDK; TCP/IP; STACK

由于网络的高速发展, 嵌入式平台上的网络通信已经成为一个必要的通信模块。TI 公司(Texas Instruments)的 C6000 系列 DSP 芯片有很多都在其片上集成了以太网接口, 如 TMS320C6747、TMS320C6748 等。结合配套的 TCP/IP 协议栈, 可使基于数字信号处理技术的因特网终端的网络连接成本降低 50% 以上。而此协议栈与 NDK (Network Developer's Kit) 结合在一起使用, 可以缩短软件开发周期。

本文针对 TMS320C6748 芯片, 对其网络硬件进行设计, 并基于 NDK 进行网络软件设计。

1 网络模型

国际标准化组织(ISO)于 20 世纪 70 年代提出了开放式通信系统互联参考模型 OSI (Open System Interconnection Reference Model), 旨在使各种计算机在世界范围内互联的网络标准框架。它划分为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层 7 个层次。物理层定义了所有电子及物理设备的规范, 其中特别定义了设备与物理媒介之间的关系, 包括了针脚、电压、线缆规范、集线器、中继器、网卡、主机适配器(在 SAN 中使用的主机适配器)以及其他设备的设计定义; 数据链路层的功能在于管理物理层的比特数据, 并且将正确的数据传送到没有传输错误的路线中;

网络层为数据传送的目的地寻址, 再选择出传送数据的最佳路线; 传输层用于控制数据流量, 并且进行调试及错误处理, 以确保通信顺利。而传送端的传输层会为分组加上序号, 方便接收端把分组重组为有用的数据或文件; 会话层用于为通信双方制定通信方式, 并创建、注销会话(双方通信); 表示层能为不同的客户端提供数据和信息的语法转换内码, 使系统能解读成正确的数据。同时, 也能提供压缩解压、加密解密; 应用层能与应用程序界面沟通, 以达到展示给用户的目的, 常见的协定有 HTTP、HTTPS、FTP、TELNET、SSH、SMTP 和 POP3 等。

OSI 是一个定义良好的协议规范集, 但过于庞大复杂。因此, DARPA (Defense Advanced Research Projects Agency) 开发了 TCP/IP 模型, 它采用更少的层(主要是把应用层、表示层、会话层合并为应用层), 更适应现实的协议。OSI 模型与 TCP/IP 模型对比如图 1 所示。



图 1 OSI 模型与 TCP/IP 模型

网络与通信 Network and Communication

2 硬件设计

2.1 模块介绍

TMS320C6748 的网络模块主要包括 EMAC 控制模块、EMAC 模块和 MDIO 模块,其网络功能模块图如图 2 所示。

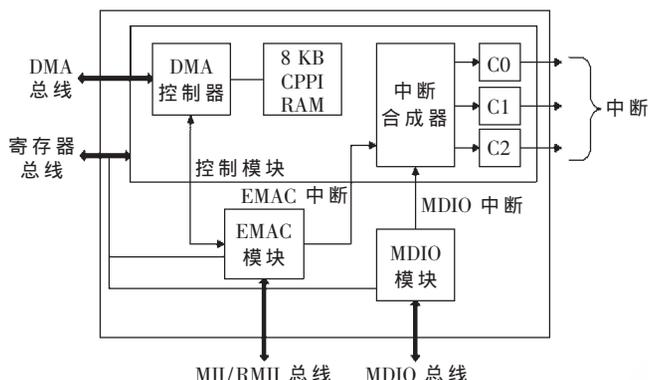


图 2 网络模块框图

EMAC 控制模块是设备处理器与 EMAC、MDIO 模块之间的主要接口,它控制网络设备的中断并采用 8 KB 的内部 RAM 来容纳 EMAC 缓冲区描述符(CPPIRAM)。

MDIO 模块的全称是数据输入/输出管理模块(Management Data Input/Output Module),采用 802.3 串行管理接口来查询和控制多达 32 个使用共享双总线的以太网 PHY 设备。主机软件通过 MDIO 模块配置每个基于 EMAC 模块的物理层自适应参数,检索自适应参数的值,并且配置使 EMAC 模块正常运作的参数。概括地说,MDIO 模块负责管理与 EMAC 相连的所有 PHY 芯片,包括对 PHY 芯片进行枚举、配置和器件状态检测等。

EMAC 模块的全称是以太网媒体访问控制模块(Ethernet Media Access Controller Module),它提供处理器与网络之间的高效接口,负责以太网数据的发送与接收。TMS320C6748 上的 EMAC 模块支持 10 Mb/s 和 100 Mb/s 模式,工作在半双工或全双工模式下,同时具有硬件流控制及服务质量保证(QoS)支持。

2.2 通信接口硬件电路

TMS320C6748 支持媒体独立接口 MII(Media Independent Interface)与简化媒体独立接口 RMII(Reduced Media Independent Interface)。以 MII 为例,其接口电路设计如图 3 所示。

MII 接口信号线定义为:

(1)MII_TXCLK:发送时钟信号。所发送的时钟是一个连续的信号,作为发送操作的定时基准,MII_TXD 和 MII_TXEN 信号依赖于这个时钟信号。它由 PHY 产生,在 10/100 Mb/s 模式下启用,当系统工作在 10 Mb/s 时为 2.5 MHz,工作在 100 Mb/s 时为 25 MHz。

(2)MII_TXD[3-0]:发送数据线。包含 4 条数据线,是 4 位数据的集合。其中 MTDX0 是最低有效位(LSB)。信号根据 MII_TXCLK 同步,并且只有当 MII_TXEN 使能时才有效。

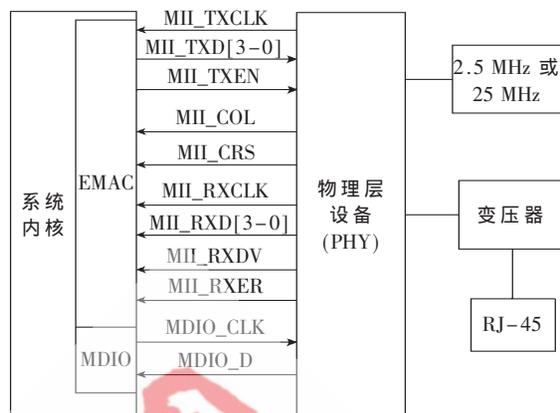


图 3 MII 接口电路

(3)MII_TXEN:发送使能信号。发送使能信号表示 MII_TXD 生成半字节的数据可发送给 PHY,由 MII_TXCLK 信号线驱动。

(4)MII_COL:网络冲突监测信号。在半双工模式下,当检测到网络上的冲突时该位被置位,它会一直保持直到冲突消除,并且该信号不一定与 MII_TXCLK 和 MII_RXCLK 同步。在全双工模式下,该位被用于硬件流量传输的控制。当该位有效时,停止数据的传输,如果数据包正在传输的过程中,将完成本次传输。如果该位被设置为低电平,将不启用硬件流量传输控制。

(5)MII_CRS:载波感应信号。在半双工模式下,如果 PHY 正在发送或者接收数据,该位会被置位。同样它与 MII_TXCLK、MII_RXCLK 不同步。在全双工模式下,MII_CRS 应保持低电平。

(6)MII_RXCLK:接收时钟信号。接收时钟也是一个连续的信号,作为接收操作的定时基准,MII_RXD 和 MII_RXEN 信号都依赖于这个时钟信号。它是由 PHY 产生,在 10/100 Mb/s 模式下启用,当系统工作在 10 Mb/s 时为 2.5 MHz,工作在 100 Mb/s 时为 25 MHz。

(7)MII_RXD[3-0]:接收数据线。

(8)MII_RXDV:接收数据使能。

(9)MII_RXER:接收出错信号。

(10)MDIO_CLK:MDIO 时钟。MDIO 的数据源于系统模块 MDIO。该时钟频率由 MDIO 控制寄存器的 CLKDIV 位来设置。

(11)MDIO_D:管理数据输入输出。

3 软件设计

3.1 NDK 结构介绍

网络开发者套件(NDK)是一个开发 TI 嵌入式处理器网络应用平台,但目前仅限于 TMS320C6000 DSP 系列和 ARM 处理器。NDK 的设计目的是为开发者提供一个完整的 TCP/IP 功能环境,缩短开发周期,降低开发难度。NDK 通过编程接口与本地操作系统(DSP/BIOS)和底层硬件相互隔离,如图 4 所示^[1]。本地操作系统(DSP/BIOS)被抽象成一个操作系统适应层(OS Adaptation Layer),底

网络与通信 Network and Communication

层硬件被抽象成一个硬件抽象层(Hardware Abstraction Layer),两个抽象层的函数库分别为 OS.LIB 和 HAL.LIB,它们是 NDK 与本地操作系统和底层硬件的接口^[2]。图 5 为 TCP/IP 协议栈的结构^[3]。

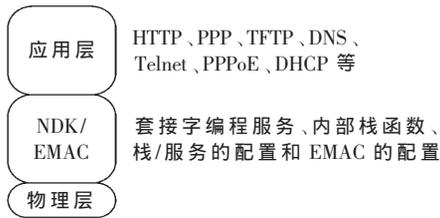


图 4 NDK 模型

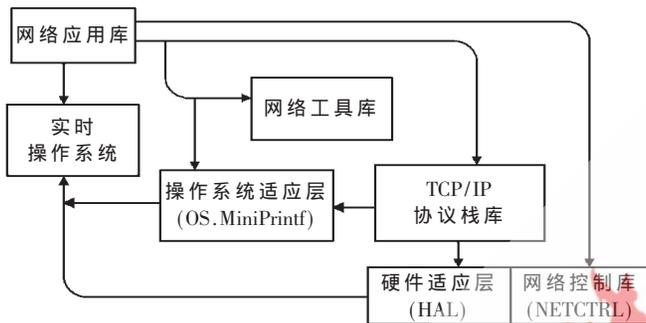


图 5 TCP/IP 协议栈结构

STACK 库是主要的 TCP/IP 网络功能库,它包括了从底层链路层到上层套接口层的所有功能,该库基于 DSP/BIOS 操作系统。NETCTRL 库在整个协议栈中起关键作用,协调操作系统和底层硬件驱动,管理所有网络事件。NETTOOL 库提供配置网络的各种服务^[4]。

3.2 NDK 编程设计

在网络程序执行之前,需对开发平台进行板极初始化,并对网络寄存器进行配置,将其作为一个钩子函数,在 DSP 启动时加载。而 TCP/IP 协议栈需要设置成一个主线程,并设置较高的优先级等级(可设置成等级 5)。初始化及线程任务设置需要在 DSP/BIOS 中添加。要调用 NDK 库中函数,需要添加以下库:stack.lib、hal_timer_bios5.lib、hal_ser_stub.lib、hal_userled_stub.lib、os_bios5.lib、nettool.lib、netctrl.lib、miniPrintf.lib 以及板极驱动库^[3]。

在主线程中,TCP/IP 协议栈需要首先进行配置并初始化,具体步骤如下:

(1)分配一个网络参数配置,主要包括主机名、IP 地址、网关、子网掩码、域名和 DNS 服务器地址等。

(2)调用 NETCTRL 库函数 NC_SystemOpen(),并用 Cfg-New()函数创建一个新的配置。

(3)调用一个配置入口函数,将第一步中分配的网络参数添加进去。如:

```
CfgAddEntry(hCf, CFGTAG_SYSINFO, CFGITEM_DHCP_HOSTNAME, 0, strlen(HostName), (UINT8 *)HostName, 0);
```

经过以上步骤,TCP/IP 协议栈配置完成,然后调用 NC_NetStart()函数启动网络程序。注意,它的返回值是一个关闭代码。通常该系统的关闭方式会决定协议栈是

否重新启动(例如改变配置后需要自动重启),一般有以下代码:

```
do
{
    rc=NC_NetStart(hCf, NetworkOpen, NetworkClose,
                  NetworkIPAddr);
} while(rc>0);
```

如果返回值大于 0 就重新启动网络。其中 NetworkOpen 函数的功能是设置网络协议、连接端口号、连接数以及网络连接时的回调函数。可配置如下:

```
static void NetworkOpen()
{
    hdsp_tcp=DaemonNew(SOCK_STREAMNC, 0, 3350,
                      dtask_tcp, OS_TASKPRINORM, OS_TASKSTKNORM, 0, 3);
}
```

在工程设计中,如果需要多线程使用网络收发数据,需要在线程中共享网络文件描述符,这样在新线程中可以像在网络主线程中一样使用 recv()和 send()函数来收发数据。一般来说可以使用如下代码:

```
fdOpenSession(TaskSelf());
fdShare(s);
```

另外,TI 提供了 recvnc()和 recvncfrom()函数,不需要拷贝数据就能直接接收,使得网络接收数据速率大大提高,特别是网络吞吐量比较大的应用中效果尤为显著。如果对发送和接收缓存区有要求,需要在网络回调函数中调用 setsockopt()函数进行设置^[5]。

经过测试,基于 NDK 的 DSP 网络运行非常稳定、速度快、占用系统资源较少,对于需要在 DSP 上发送大量数据的场合非常方便。虽然采用 NDK 开发 DSP 网络有很多优点,但必定要占用一定的系统资源,并且优先级比较高,因此实时性要求非常高的项目中,应采用其他方案。

参考文献

- [1] Texas Instruments. Getting started with the C6000 network development kit. SPRAAX4[A]. 2008.
- [2] 刘钊江,郑红,吴兴华,等.基于 TIC6000 系列 DSP 的网络开发研究[J].仪器仪表用户,2010,30(3):60-62.
- [3] Texas Instruments. TI network developer's kit(NDK) v2.21 User's Guide. SPRU523H[A]. 2012.
- [4] 梁迅.基于 NDK 的 DSP 网络编程[J].计算技术与自动化,2005,24(9):79-81.
- [5] Texas Instruments. TI network developer's kit(NDK) v2.21 API Reference Guide. SPRU524H[A]. 2012.

(收稿日期:2012-10-19)

作者简介:

周恒国,男,1986 年生,硕士研究生,主要研究方向:检测技术与自动化装置。

黄方,女,1966 年生,副教授,主要研究方向:环境监测及水污染控制。