

# TMS320C6713 与 DM9000A 的接口设计与实现

倪启彦, 石建波

(电子工程学院, 安徽 合肥 230037)

**摘要:** 采用 DAVICOM 公司的以太网控制芯片和 TI 公司的 TMS320C6713 设计了网络通信电路。该设计适用于各种嵌入式设备以及各种信号处理设备中。为各种设备之间的通信提供解决方案。给出了 TMS320C6713 和 DM9000A 的接口电路和软件编程实现。

**关键词:** DSP; 以太网; DM9000A; TCP/IP 协议

中图分类号: TP274.2

文献标识码: B

文章编号: 1674-7720(2011)04-0021-03

## Design and implementation of interface between TMS320C6713 and DM9000A

Ni Qiyang, Shi Jianbo

(Electronic Engineer Institute, Hefei 230037, China)

**Abstract:** Design the circuit bases the principle of the DAVICOM's DM9000A and TI's TMS320C6713. This design is for various embedded and signal processing equipments. It affords a resolvent to communications between equipments. The article give the interface of DM9000A to DSP and the programme.

**Key words:** DSP; Ethernet; DM9000A; TCP/IP protocol

随着嵌入式技术和网络技术的发展及以太网的广泛应用,各种应用设备的网络化成为未来设备发展的一个重要方向。各种嵌入式设备已经成功渗透到生产生活的各个领域,并逐渐朝着网络化、智能化的方向发展<sup>[1]</sup>。以太网因其在实时操作、可靠传输、标准统一等方面的卓越性能及其便于安装、维护简单、不受通信距离限制等优点,已发展成为一种成熟的技术。人们希望通过网络连接各种现场,协调各种设备间的工作,实现设备的网络化控制与管理。AVICOM 公司的 DM9000A 是实现网络通信的芯片,它占用很少的硬件资源,可以在不改变设备其他性能的基础上给设备加入网络功能。开发一个嵌入式通信控制器不仅实现了设备上网的低成本解决方案,同时也提升了设备信息化水平。随着数字信息技术、网络技术和嵌入式系统的高速发展,如视频监控、远程控制、信息家电、网络摄像头都离不开嵌入式设备与网络的结合,嵌入式系统的以太网接口设计已成为目前的一个研究热点。因此嵌入式以太网接口的设计对于嵌入式网络系统的建立是十分必要的。目前, Linux 操作系统已经越来越广泛地被应用于嵌入式设备<sup>[2]</sup>,编写 Linux 下的设备驱动程序十分有意义。

### 1 DM9000A 芯片介绍

DM9000A 是 DAVICOM 公司推出的一款高速以太网接口芯片,其基本特征是:集成 10M/100M 物理层接口;内部带有 16 KB SRAM 用作接收发送的 FIFO 缓存;支持 8/16 bit 两种主机工作模式;通过 HP 认证的 AUTO-Mdix (支持直接互连自动翻转) 功能;支持 TCP/IP 加速(IPv4 check sum offload) 减轻 CPU 负担,提高整机效能;10 ns I/O 读写时间。该电路还集成了 EEPROM 接口,自举时通过 EEPROM 接口输入到芯片中,从而实现自动初始化。在现有以太网控制器芯片中大多数速度是 10 M 的,其传输速率慢,DM9000A 是成本低速度快的以太网控制器芯片,速度 10M/100M 自适应。它被设计为低功耗、高处理性能,而其操作又非常简单,具有通用的处理器接口,可以与多种处理器直接连接,容易完成不同系统的软件驱动开发。数据总线宽度可设置为 8 bit 和 16 bit, IO 口支持 3.3 V~5 V,因此几乎可以和所有处理器连接。DM9000A 以太网控制器遵循 IEEE 颁布的 802.3 以太网传输协议<sup>[3]</sup>。芯片内部集成 4 KB 双字节静态存取存储器,自动收发转换功能使开发变得更加简单,容易完成不同系统的软件驱动开发。

## 2 DM9000A 芯片的硬件接口设计

DM9000A 可以很方便地与目前主流嵌入式 CPU 以 8 bit 或 16 bit 的总线方式连接,本文系统的 CPU 为 TI 公司的 DSP:TMS320C6713(C6713),它是业界流行的一款高性能 32 bit 浮点 DSP,在 255 MHz 时钟主频下,其最高执行速度可达 1 800 MIPS。C6713 拥有丰富的片内资源,常用于数字处理领域,二者的接口设计如图 1。

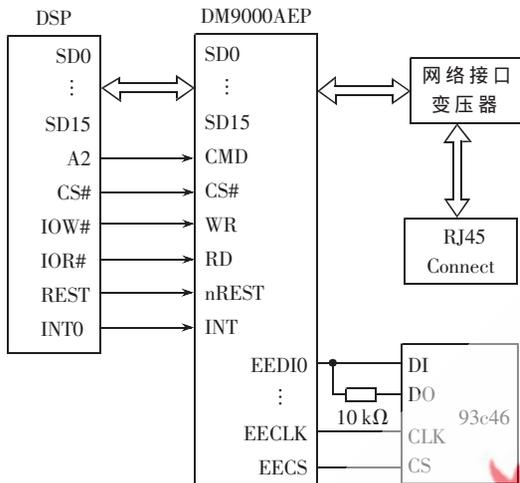


图 1 TMS320C6713 与 DM9000A 的接口

现在的设备都包含数字处理器,DM9000A 可以很容易地应用到各种设备里。由图 1 可以看到,TMS320C6713 与 DM9000A 硬件连接简单。系统上电时,DSP 通过总线配置 DM9000A 内部网络控制寄存器(NCR)、中断寄存器(ISR)等,完成 DM9000A 的初始化。随后,DM9000A 进入数据收发等待状态。该电路还集成了 EEPROM 接口,自举时通过 EEPROM 接口输入配置数据到芯片中,从而实现自动初始化。数据总线 SD0-SD15 可以直接挂在处理器的数据总线上,无需电平转换。如果 EECS 接一个 10 kΩ 的上拉电阻,芯片在上电时将被配置成 8 bit 数据总线模式,此时 SD8-SD15 可以用作 GPIO。再加一个硬件协议栈,可以将其用在没有处理器的设备中。

## 3 DM9000A 芯片的操作

对处理器驱动网卡芯片来说,比较关心的有以下几个引脚:IOR、IOW、CMD(A2)、INT、RST 以及数据引脚 SD0~SD15。IOR 和 IOW 是 DM9000 的读写选择引脚,低电平有效,即在信号的上升沿进行读(IOR)写(IOW)操作。CMD 为命令/数据选择引脚,低电平时读写命令操作,高电平时读写数据操作,将它接到处理器的地址线 A2 上。如 DM9000A 工作的基地址为 0x80000000,则

DM9000A 有两个的地址:命令口地址(0x80000000)和数据口地址(0x80000004)。DM9000A 有多个寄存器,通过先向命令口写寄存器地址,再向数据口写寄存器数据来配置芯片。数据总线 SD0-SD15 则根据数据总线的位数接到处理器数据总线上。收发数据也是通过读写相应的寄存器实现。

图 2 和图 3 分别显示了 DM9000A 的读写时序,由图可知,DM9000A 的读写操作与一般的异步存储器相同。根据芯片资料,编写读写寄存器的子函数。

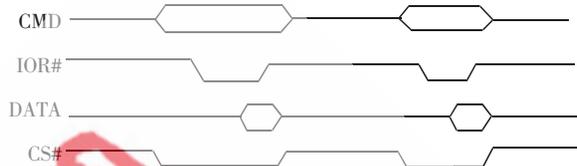


图 2 DM9000A 读时序图



图 3 DM9000A 写时序图

寄存器的读操作:

```
#define DM_INDEX *(volatile unsigned int *)0x8000000
#define DM_DATA *(volatile unsigned int *)0x8000004
int dm9000_reg_read(char reg)
{ DM_INDEX = reg;
  return DM_DATA; }
```

寄存器的写操作:

```
void dm9000_reg_write(char reg, char data)
{ DM_INDEX = reg;
  DM_DATA = data; }
```

## 4 DM9000A 芯片的初始化

对 DM9000A 芯片的初始化,就是向相应的寄存器写入期望的值,为芯片的稳定工作做好准备。表 1 列出各寄存器的名称、地址以及赋参考值,寄存器的具体功能可以参考 DM9000A 的用户手册。

初始化中还要对 PAR(10H--15H)寄存器初始化,PAR 中保存的是芯片物理地址即通常的 MAC 地址,芯片根据网络数据与 PAR 中的数据比较来确定保存或丢弃数据。初始化后,可以通过读 NCR 寄存器来判断芯片是否已经连上网络。

表 1 各寄存器的初始化后的值

寄存器名	GPCR	GPR	ISR	NSR	RCR	TCR	BPTR	FCTR	FCR	SMCR
地址	0x1E	0x1F	0xFE	0x01	0x05	0x02	0x08	0x09	0x0A	0x2F
数值	0x01	0x00	0x3f	0x2c	0x39	0x00	0x3f	0x3a	0xff	0x00

## 5 DM9000A 芯片数据的发送和接收

在传送数据封包之前,需将其封包数据存放在 DM9000A 的内存中。DM9000A 的内部 RAM 地址 0000h~0BFFh 用来存放发送数据包。用户无需关心数据存放的地址,只要向 MWCMD 寄存器写入数据包即可,内部数据指针会自动更新,将 MWCMD 寄存器的数据依次存到内部 RAM。另外还需将要传送封包的大小存放在 TX-PLH 和 TXPLL 寄存器中。之后再将 TCR 的 bit0 设为 1,此时芯片开始进行封包的传送。而在传送完成后,会将传送是否成功的信息放在 TSRI、TSRII 中。

以下为数据发送的函数:datas 为要发送的数据在处理器缓冲区的地址,len(两个字节)为要发送的数据长度(以字节为单位)。

```
void sendpacket(char *datas, unsigned int len)
{
    unsigned int i;
    dm9000_reg_write (IMR, 0x80); //先禁止网卡中断,防止
    在发送数据时被中断干扰。
    DM_ INDEX = MWCMD;           //写命令寄存器
    for(i=0; i<len; i+=2)        //16 bit mode
    DM_DATA = datas[i] | (datas[i+1]<<8);
    dm9000_reg_write(TXPLH, (len>>8) & 0x0ff);
    dm9000_reg_write(TXPLL, len & 0x0ff);
    dm9000_reg_write(TCR, 0x01); //请求发送数据
    dm9000_reg_write(IMR, 0x81);
    //DM9000 网卡的接收中断使能
}
```

数据的发送比较简单,接收相对复杂一点。DM9000 接收到数据包时,会存放于 DM9000 接收内存的 0C00h~3FFFh 中。若是读取位置超过 3FFFh 时,DM9000 会自动将位置移到 0C00h 的位置。在每一个数据包前,会增加 4 B 存放数据包相关资料。接收数据包时首先要读取这 4 个字节来确定数据包的状态,第一个字节“01H”表示接下来的是有效数据包,若为“00H”则表示没有数据包,若为其他值则表示网卡没有正确初始化,需要重新初始化。在读取其他字节之前,必需要确定第 1 个字节是否为“01h”。第 2 个字节则为这个数据包的相关信息,其格式与 RSR 寄存器的格式一样。第 3 和 4 个字节是存放这个封包的长度大小(不包括前四个字节)。

如果接收到的数据包长度小于 60 B,则 DM9000 会自动为不足的字节补上 0,使其达到 60 B。同时,在接收到的数据包后 DM9000 还会自动添加 4 个 CRC 校验字节。可以不予处理。于是,接收到的数据包的最小长度也会是 64 B。

接收数据包的过程:(1)读 MRCMDX 寄存器,判断是否有数据包。(2)如有数据包,读 MRCMD 寄存器,根据数据包的长度信息将整个数据包读取出来。其中第一步需要做两遍,因为第一次读到的值总为 0。

以下为数据包接收的函数:datas 为接收到是数据存

储位置,接收成功返回 1,不成功返回 0。

```
unsigned int receivepacket(unsigned char *datas)
{
    unsigned int i, tem;
    unsigned int status, len;
    unsigned char ready;
    ready = 0; //希望读取到“01H”
    status = 0; //数据包状态
    len = 0; //数据包长度
    /* 以上为有效数据包前的 4 个状态字节 */
    ready = dm9000_reg_read(MRCMDX);
    //第一次读取,一般读取到的是 00H
    ready = dm9000_reg_read(MRCMDX);
    //第二次读取,总能读对 if(ready != 0x01)
    { if(ready != 0x00)
    //若第二次读取到的不是 01H 或 00H,则表示没有初始化成功
    { dm9000_reg_write(IMR, 0x80); //屏幕网卡中断
    DM9000_init(); //重新初始化
    dm9000_reg_write(IMR, 0x81); //打开网卡中断
    }
    return 0;
    }
    /* 以上表示若接收到的第一个字节不是“01H”,则表示
    没有数据包,返回 0 */
    status = dm9000_reg_read(MRCMD);
    len = DM_DATA;
    if(! (status & 0xbf00) && (len < 1522))
    { for(i=0; i<len; i+=2) // 16 bit mode
    { tem = DM_ DATA;
    datas[i] = tem & 0x0ff;
    datas[i+1] = (tem >> 8) & 0x0ff;
    }
    }
    else
    {
    return 0;
    }
}
```

## 6 网络数据传输协议

以上完成了以太网数据包的收发,应用网络传输还需要完整的网络传输协议,TCP/IP 协议为互联网通用协议。TCP/IP 协议族是一整套把各种系统连接在一起并保证数据准确快速传输的规定和格式<sup>[4]</sup>。通常把 TCP/IP 协议族抽象成为一种具有四层结构的模型:链路层、网络层、运输层、应用层。每层各负责一个或一系列独立的功能。根据各种协议的格式编写相应的数据处理函数。在

有操作系统的应用中,要编写网络设备驱动程序。设备驱动程序需要完成网络设备的注册、初始化与注销以及进行发送和接收数据处理,并能针对传送超时、中断等情况进行及时处理<sup>[5]</sup>。

DM9000A 的操作简单,模块独立,不会影响原设备的性能指标,为设备间的通信提供保障。在没有处理器的装备中,93C46 可以正确初始化芯片,硬件协议栈能正确处理收发数据。传输协议有成熟的 TCP/IP 协议,在简单的通信和控制系统中,可以应用简单的数据传输协议,缩短开发周期,提高设备的效率。

#### 参考文献

- [1] 雒珊,尹岗.基于 ARM 的以太网通信控制器的设计[J]. 电子测量技术,2009, 23(10):133-135.  
[2] 赵军.嵌入式平台上 Linux 网络设备驱动程序的开发[J].

电脑知识与技术,2009,5(33):9272-9273.

- [3] 陈峰,彭龔,易彬,等.基于嵌入式技术的以太网接口设计[J].通信技术,2010,43(5):127-129.  
[4] 刘春燕,颜兴建.嵌入式 TCP/IP 协议的设计与实现[J].电脑知识与技术,2010,6(8):1815-1816.  
[5] 周敬琼,周凤星.基于 ARM 的 Linux 网络设备驱动程序开发[J]. 计算机工程与设计,2009,30(22):5124-5127.

(收稿日期:2010-09-11)

#### 作者简介:

倪启彦,男,1986年生,硕士在读,主要研究方向:信号与信息处理。

石建波,男,1986年生,硕士在读,主要研究方向:信号与信息处理。

电子技术应用  
APPLICATION OF ELECTRONIC TECHNIQUE  
www.chinaAET.com