

基于MIP固件实现Neuron芯片的快速并行通信

■ 武汉科技大学 祁明晰 刘明

随着控制、计算机、通信、网络等技术的发展,信息交换沟通的领域正在迅速覆盖从工厂的现场设备层到控制管理的各个层次。现场总线就是顺应这一形式发展起来的新技术。现场总线是当今自动化领域技术发展的热点之一,被誉为自动化领域的计算机局域网。LonWorks总线就是由美国Echelon公司推出的一种现场总线技术,在楼宇自动化、办公自动化、智能家电等领域都有广泛的应用。但是现场总线是一种工业上的底层网络,如何实现节点与上位PC机、底层工作站的通信,特别是大量数据的高速通信,一直是人们研究的方向。本文针对Neuron C神经元芯片3150,详细介绍运用MIP实现高速并行通信的方法。相对于串行通信、并行通信有着明显的速度优势,运用MIP并口通信方式能使传输速度最高达到3.3 Mbps,从而实现了大数据量的快速传输。

1 MIP

MIP是Microprocessor Interface Program的缩写形式。它是一种固件,可使神经元芯片作为专用于通信的辅处理器与另一主处理器连接。MIP将LonTalk协议的上层部分从神经元芯片转移至与神经元芯片相连的主处理器上。Echelon公司提供了3种可用MIP(MIP/20、MIP/50、MIP/DPS)分别用于不同的接口。本文利用MIP/50固件,因为它提供了比MIP/20更大的吞吐量。

2 通信原理与通信方式

将Neuron芯片的11个I/O口全部用于并行通信。Echelon公司提供了三种并口模式,在此仅介绍Slave A模式。在Slave A中I/O0~I/O7定义为8位双向数据线, I/O8口为片选信号线, I/O9口为读/写信号线, I/O10口为握手应答信号线。其硬件连接如图1所示。

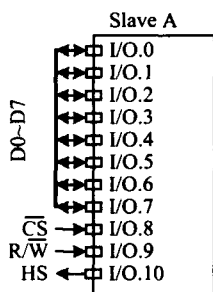
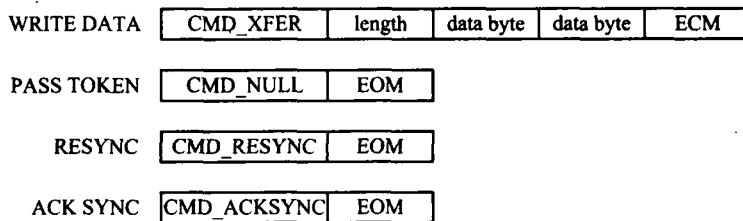


图 1

主处理器与处于Slave A工作方式的Neuron芯片之间的数据传输是通过运用“虚写令牌传递机制”实现的。这一“虚写令牌”如同乒乓球一样在主处理器与Neuron芯片之间相互传递。拥有令牌的一方拥有对数据总线的写控制权,它既可向对方传送数据流又可只不传送数据只将令牌传递给对方。通信的建立由主处理器控制,初始状态下令牌权在主处理器方。片选信号 $\overline{CS}$ 、读/写信号线 $R/\overline{W}$ 都由主处理器控制,握手应答信号线HS则由Slave Neuron芯片给出。一次传输的最大数据量是255字节。在数据传输过程中令牌权始终在发送方,全部数据发送完后发送方才将令牌传递给对方。处于Slave A工作方式的Neuron芯片通过检测读/写信号线($R/\overline{W}$)电平判断令牌权。“虚写令牌传递机制”的命令格式如图2所示。令牌拥有方在一定时间只能发送一种命令,分别是写数据命令、传递令牌命令、要求同步命令(主处理器)、同步应答命令(Slave Neuron芯片)。



注: CMD_XFER=0x01; EOM=任意字节(总是0x00);
 CMD_NULL=0x00; NULL=0x00;
 CMD_RESYNC=0x5A; Length=数据字节长度(不包括EOM);
 CMD_ACKSYNC=0x07; data=实际数据字节。

图 2

3 通信实现

首先声明11个I/O口为并行通信对象,其语法规则如下:

IO_0 parallel slave io_object_name

然后还要为并口通信声明一个结构体:

```
struct paralle_io_interface
{ unsigned length; //数据域的长度,它由应用程序设定
  unsigned data[MAXLENGTH];
  //MAXLENGTH为最大传输缓冲区的大小,最大值为255
```

```
} piofc
```

并口通信主要由以下几个函数完成：

① `io_in_ready()` 当有消息到达，即并口上有数据传送到来，此函数值为 TRUE。此时应用程序须调用 `io_in()` 函数接收数据。

② `io_out_request()` 此函数用于获取令牌。

③ `io_out_ready()` 在调用此函数之前应先调用 `io_out_request()`，当应用程序获取到令牌后，此函数值返回 TRUE。此时，应用程序调用 `io_out()` 将数据发送到并口上。数据发送完后此函数自动将令牌传递给另一方。图 3 给出 Neuron C 通信流程图。

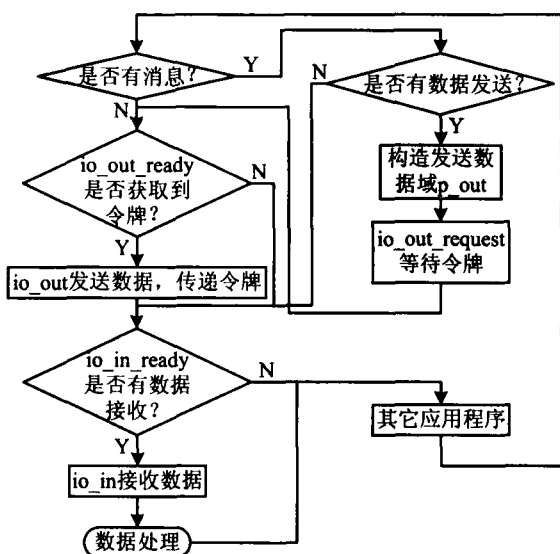


图 3

源程序如下：

```
#define DATA_SIZE 255
IO_0 parallel slave s_bus;
boolean bReceive = FALSE;
```

```
typedef struct
```

```
{
    unsigned int length;
    unsigned int data[DATA_SIZE];
}piofc;
far piofc p_in,p_out;
eeprom onchip unsigned msgcode = 0x0C;
msg_tag mo_parallel;
when(msg_arrives){
    if(msg_in.code==msgcode) {
        p_out.length = msg_in.len + 1;
        p_out.dat[0] = 0x0a;
        memcpy(&(p_out.dat[1]),msg_in.data,msg_in.len);
        io_out_request(s_bus);
    }
}
when(io_out_ready(s_bus))
{
    io_out(s_bus,&p_out);
}
when(io_in_ready(s_bus))
{
    p_in.length=DATA_SIZE;
    io_in(s_bus,&p_in);
    bReceive = TRUE;
}
```

参考文献

- 1 NodeBuilder User's Guide Revision 3. Echelon Corporation 078-0141-01C
- 2 Neuron Chip Data Book. February 1995
- 3 Neuron C Reference Guide Revision 2. Echelon Corporation 078-0141-01B
- 4 Neuron C Programmer's Guide Revision 4. Echelon Corporation 078-0141-01D

从专用RTOS到嵌入式Linux研讨会报道

美国 MontaVista 软件公司和麦克泰软件技术有限公司于 2000 年 6 月 3 日 - 7 日分别在上海、北京、深圳主办了“从专用 RTOS 到嵌入式 Linux”研讨会。会上报告包括从专用 RTOS 到嵌入式 Linux、MontaVista 新产品和开发计划以及麦克泰嵌入式 Linux 应用服务和第 3 方产品介绍。MontaVista 公司合作伙伴——嵌入式计算机硬件厂商 Ampro、Force、Interphase 和嵌入式数据库厂家 Solid 等参展。

MontaVista Linux（原 Hard Hat Linux）由 100% 纯 Linux 集成起来，代码完全开放，不收取运行费，主要应用在电信、手机、PDA 和机顶盒等领域。MontaVista Linux 采用 SMP 多处理器策略的抢占内核实现 Linux 实时性，支持 7 大 CPU 系列和 70 种以上硬件板卡。MontaVista 的产品包括 MontaVista Linux 的嵌入式源码和二进制码、交叉开发平台及 1 套开发工具。麦克泰软件技术有限公司是 MontaVista 公司在中国的授权代理，为开发人员提供全套产品和服务。