**实验一：双机接口通讯——实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **班级** |  | **姓名** | |  |
| **学号** |  | **指导教师** | |  |
| **其他**  **组员**  **信息** | **学号** | | **姓名** | |
|  | |  | |
|  | |  | |
|  | |  | |

**一、实验目的**

**1. 掌握串行通讯原理；**

**2. 了解RS-232C串行接口标准及连接方法；**

**3. 掌握16550接口芯片的编程方法。**

**二、实验要求**

**1. 两台装有16550接口芯片的微机通过RS-232C电缆连接，通过编程实现两台微机的串行双工通信；**

**2. 采用异步方式全双工通信，要求每次传送单个字符，波特率因子64，传送字符包含7位数据位，1位停止位；**

**3. 通信双方均采用查询方式；**

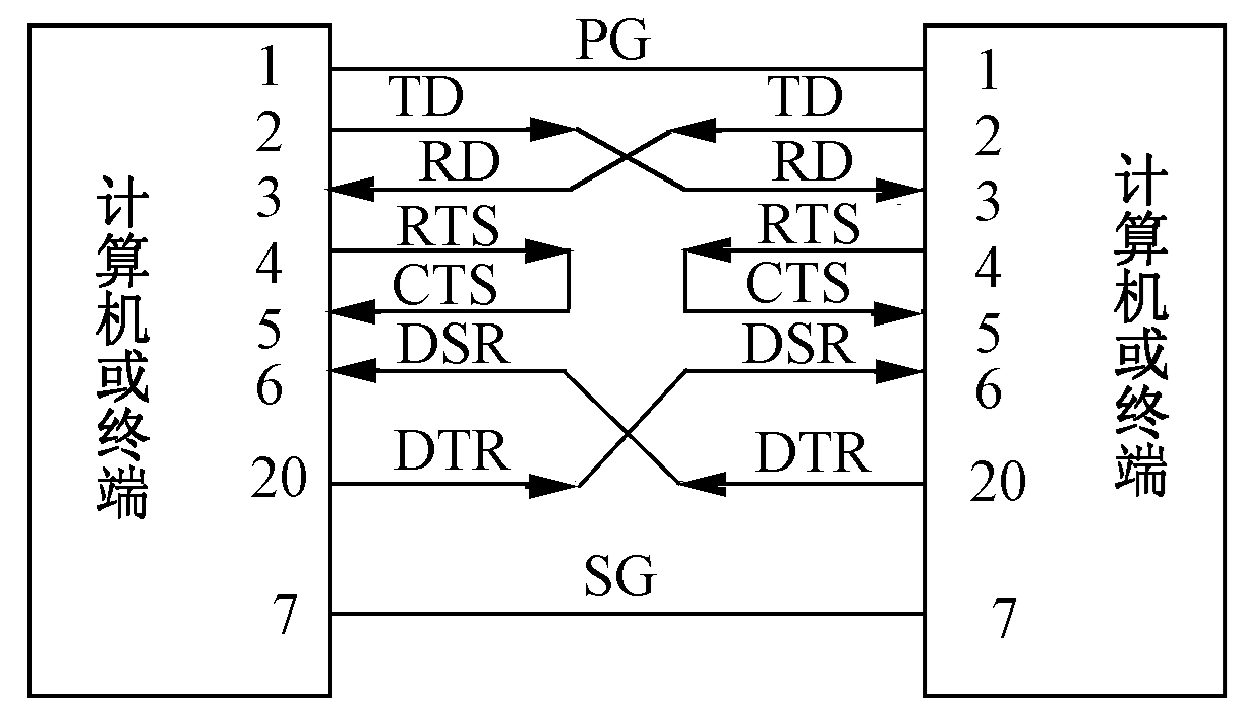
**4. 当输入“\*”号时退出；**

**5. 汇编成exe文件格式；**

**6. 在实验素材SjtxS.asm的基础上完成实验。**

**三、实验环境**

**1. 两台微机的系统：Windows XP with Service Pack 3**

**2. 两台微机通过RS-232C电缆相连，两个数据端口地址分别为：02F8H（COM2）、03f8H(COM1)，两台微机相连接的接口图如下：**

**3. 汇编工具：DosBox、masm.exe、link.exe**

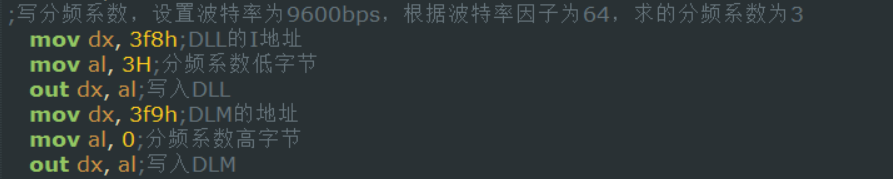
**四、实验步骤**

由于实验素材中的SjtxS.asm已经提供了双机通讯程序的框架代码，所以根据相应注释向该框架中写入缺失部分的代码即可。根据16550接口芯片的工作原理及已知代码可以确定要补充的部分如下：芯片初始化，轮询分别负责收、发的寄存器以做出相应跳转并完成发送字符的任务，完成接收字符的任务。对以上提到的三部分具体实现步骤如下：

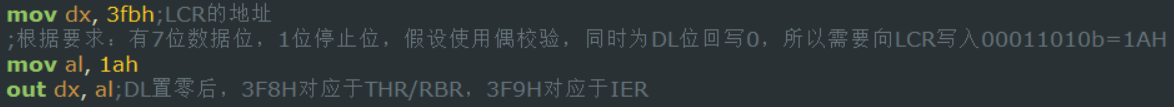
**1. 芯片初始化**

（1）将16550芯片的线路控制寄存器的DL置1，用以开启读写该芯片分频系数的权限。

（2）为16550写入分频系数：

实验中要求波特率因子为64，而未要求波特率，本程序中设定波特率为9600bps，根据公式：分频系数=基准时钟÷(波特率×波特率因子)可得本程序中：分频系数=1843200÷(9600×64)=3；所以要向03F8H端口（DLL）写入分频系数低8位：3H，向03F9H端口（DLM）写入分频系数高8位：0H。

（3）设定异步通信的数据格式，并为16550的LCR的DL位回写0：

根据实验要求，有7位数据位所以设置L1 L0=10；有1位停止位，所以设置S=0；假设使用偶校验，设置ST P PE=011；设置SB=0，推退出间断状态；设置DL=0使得读/写03F8H端口为数据（即读写接收缓冲寄存器和发送保持寄存器）。综上，我们需要向3FBH（LCR）写入1AH（00011010B）。

（4）初始化FIFO控制寄存器：

置FIFO允许，接收器复位，发送器复位的控制字为07H。

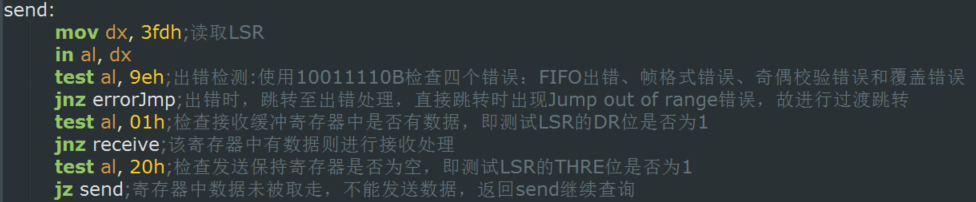
**2. 轮询分别负责收、发的寄存器以做出相应跳转并完成发送字符的任务**

（1）名为send的代码段为一个大循环，在其中要不断完成对THR和RBR寄存器的交替查询，并且在每一次循环开始完成错误检测，这样可以避免在发送或接收字符的程序段中重复检查错误。在send段中最后进行对发送保持寄存器的检测，如果其为空，则自动进入发送字符的代码。

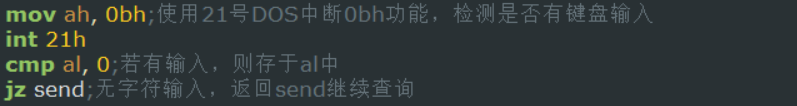
（2）进行出错检测，并检查RBR和THR：

通过检查线路状态寄存器LSR的RFE、FE、PE和OE位以判断是否出错，若检测出错误，则跳至error段进行出错处理，否则继续执行下一步；

通过测试LSR的DR位是否为1，以判断接收缓冲寄存器是否收到数据，若其中有数据，则跳至receive段进行字符的接收，否则继续执行下一步；

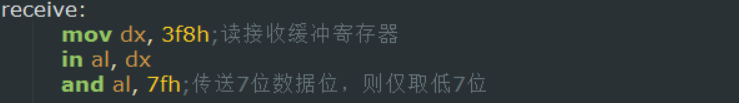
通过测试LSR的THRE位是否为1，以判断发送保持寄存器中的数据是否被取走，若未被取走，则不能发送数据，返回send段开始继续循环查询，否则进行字符的发送。

（3）从键盘读取字符，并进行发送：

 首先利用21号DOS中断0bh功能检测是否有键盘输入，若无输入，则返回send段开始继续循环查询，若有输入首先设置发送字符部分的光标位置，以达到发送字符和接收字符分上下两部分显示的效果；接下来判断输入的是否为回车符，若是则将光标位置设置为行数加1，列数归0；最后将输入的字符发送到THR寄存器。检测键盘输入需要中断，但检测后要自动退出中断，相应代码如下：

（4）在原有的输入“\*”即推出的代码后添加返回send段开始的代码构成大循环，使得程序发送完字符后可以开始新一轮的检查和发送。

**3. 完成接收字符的任务**

（1）首先从接收缓冲寄存器中取得数据，因为根据要求设定的异步通信数据格式，一个字符中仅包含7位数据，所以从接收到的字符中，取得其低7位即可。

（2）设置好接收字符部分的光标行列号，以配合之前的发送字符部分达到分上下两部份显示的效果；接下来判断取得的字符是否为回车或换行符，若是的话则进入setPos段调整光标位置（调整方法同send段的对应部分），否则对接收到的字符进行显示。

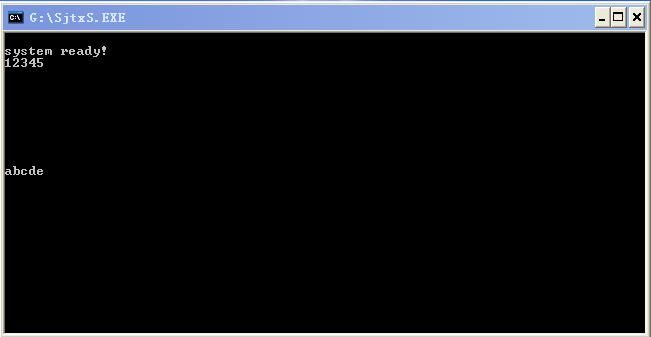
**五、实验结果与分析**

**1. 实验结果**

将改写好的SjtxS.asm汇编成SjtxS.exe，分别复制到已经通过RS-232C电缆连接好的两台微机上并运行。

先在第一台微机上连续输入字符：abcde，在第二台微机上查看收到的字符串是否与之相符；再在第二台微机上输入：12345，然后在第一台微机上检查收到的结果。

在第一台微机上发送的字符串与收到的结果如下：

在第二台微机上发送的字符串与收到的结果如下：

**2. 实验结果分析**

根据实验结果可得，两台微机均能在同一时刻接收对方发送来的字符，并能向对方发送字符，实现了已补全双工通信，符合实验预期。

**六、实验心得与体会**