

实验一 监控程序与汇编语言程序设计实验

一、实验要求

- 1、实验之前认真预习，明确实验的目的和具体实验内容，设计好主要的待实验的程序，做好实验之前的必要准备。
- 2、想好实验的操作步骤，明确通过实验到底可以学习哪些知识，想一想怎样有意识地提高教学实验的真正效果。
- 3、在教学实验过程中，要爱护教学实验设备，认真记录和仔细分析遇到的现象与问题，找出解决问题的办法，有意识地提高自己创新思维能力。
- 4、实验之后认真写出实验报告，重点在于预习时准备的内容，实验数据，实验过程、遇到的现象和解决问题的办法，自己的收获体会，对改进教学实验安排的建议等。善于总结和发现问题，写好实验报告是培养实际工作能力非常重要的一个环节，应给以足够的重视。

二、实验目的

- 【1】学习和了解TEC-XP16教学实验系统监控命令的用法；
- 【2】学习和了解TEC-XP16教学实验系统的指令系统；
- 【3】学习简单的TEC-XP16教学实验系统汇编程序设计。

三、实验注意事项

（一）实验箱检查

- 【1】连接电源线和通讯线前TEC-XP16实验系统的电源开关一定要处于断开状态，否则可能会对TEC-XP16实验系统上的芯片和PC机的串口造成损害。
- 【2】五位控制开关的功能示意图如下：

上方	单步	手拨	硬布线	联机	8位
下方	连续	内存	微程序	脱机	16位

【3】几种常用的工作方式【开关拨到上方表示为 1，拨到下方为 0】

工作方式	拨动开关
连续、硬布线、联机、16位	00110
连续、微程序、联机、16位	00010
单步、手动、硬布线、联机、16位	11110
单步、手动、微程序、联机、16位	11010
脱机运算器实验、16位	1XX00

（二）软件操作注意事项

【1】用户在选择串口时，选定的是PC机的串口1或串口2，而不是TEC-XP16实验系统上的串口。即选定的是用户实验时通讯线接的PC机的端口；

【2】如果在运行到第五步时没有出现应该出现的界面，用户需要检查是不是打开了两个软件界面，若是，关掉其中一个再试；

【3】有时若TEC-XP16实验系统不通讯，也可以重新启动软件或是重新启动PC再试；

【4】在打开该应用软件时，其它的同样会用到该串口的应用软件要先关掉。

（三）联机通讯失败自检

如果上述的硬件和软件的操作都正确，联机却依旧失败，可以进行如下测试：

【1】测试PC机的串口是否能正常工作，或是换一台PC或换同一台PC的另一个串口再试，在换串口时要将TEC-XP16实验系统断电，换完后重新启动实验系统和软件；

【2】检查机器上的元器件插接是否正确(建议用户对照能够正常通讯的实验系统进行详细检查)，有没有被学生动过，尤其是扩展内存和扩展I/O接口时，芯片方向是否插对，片选信号有没有连接；

【3】检查相应的短路子是否连接正确；

【4】建议教师预留一台运行正常的TEC-XP16实验系统备用，机器出问题后可以对照检查。

四、实验步骤

1. 准备一台串口工作良好的PC机；
2. 将TEC-XP16放在实验台上，打开实验箱的盖子，确定电源处于断开状态；
3. 将黑色的电源线一端接220V交流电源，另一端插在TEC-XP16实验箱的电源插座里；
4. 取出通讯线，将通讯线的9芯插头接在TEC-XP16实验箱上的串口“COM1”或“COM2”上，另一端接到PC机的串口上；
5. 将TEC-XP16实验系统左下方的五个黑色的控制机器运行状态的开关置于正确的位置，在这个实验中开关应置为：00110【连续、内存、硬布线、联机、16位】，其它实验相同；
6. 打开电源，船形开关和5V电源指示灯亮。
7. 在PC机上运行PCEC16.EXE文件，根据连接的PC机的串口设置所用PC机的串口为“1”或“2”，其它的设置一般不用改动，直接回车即可。（具体步骤附后）

8. 按一下“RESET”按键，再按一下“START”按键，主机上显示：

TEC-2008 CRT MONITOR

Version3.0 2007.10

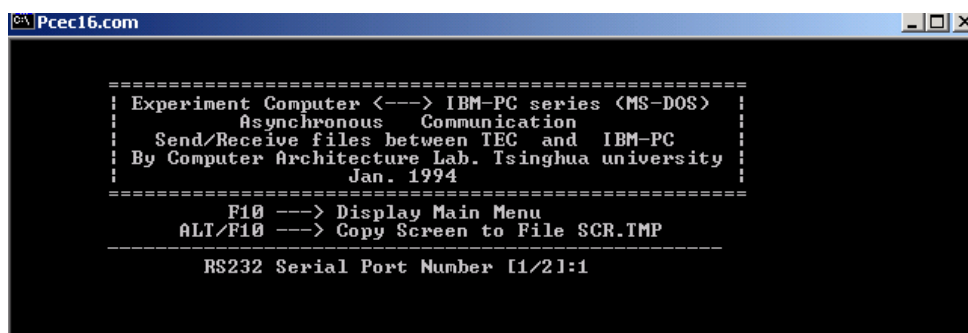
Computer Architectur Lab., Tsinghua University

Copyright Jason He

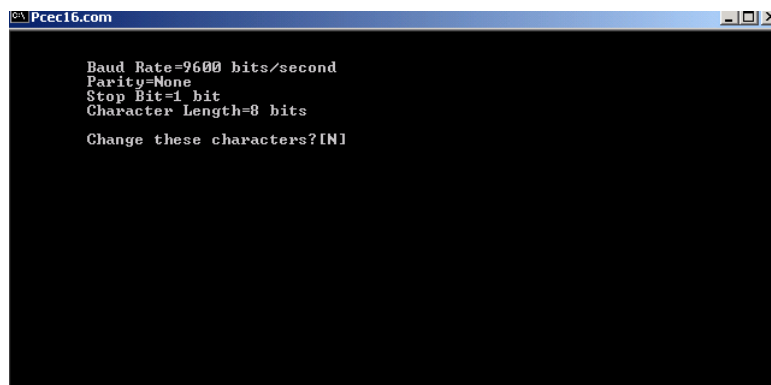
>

五、仿真终端软件的操作步骤

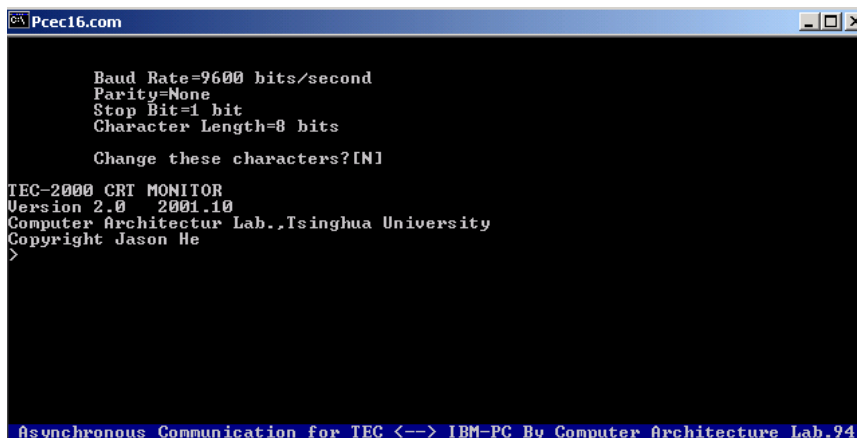
- 【1】在PC机上建一个文件夹TEC-XP16；
- 【2】取出配套的用户盘，将应用程序PCEC16拷贝到用户机器硬盘上该文件夹里；
- 【3】双击PCEC16图标，出现如图所示的界面：



【4】系统默认选择串口1，用户可根据实际情况选择串口1或是串口2(这里的串口指的是和TEC-XP16教学实验系统相连的PC机的串口)，按回车后出现如图界面：



【5】图中是系统设定的一些传输参数，建议用户不要改动，直接回车。按一下“RESET”按钮放开后再按一下“START”按钮，出现界面如图所示：



【6】此时表明TEC-XP16机器联机通讯正常。

六、实验内容

【1】学习联机使用TEC-XP16教学实验系统和仿真终端软件PCEC；

【2】使用监控程序的R命令显示/修改寄存器内容、D命令显示存储器内容、E命令修改存储器内容；

【3】用“A”命令输入例1-例6，用“G”命令或“T”命令运行并观察结果，请在实验报告中给出各例的运行结果。

注：实验给出的例子，都是用监控程序的A命令完成输入源汇编程序的。在涉及到汇编语句标号的地方，不能用符号表示，只能在指令中使用绝对地址。使用内存中的数据，也由程序员给出数据在内存中的绝对地址。显而易见，对这样的极短小程序矛盾并不突出，但很容易想到，对很大的程序，一定会有较大

的困难。

在用A命令输入汇编源语句的过程中，有一定用机经验的人，常常抱怨A命令中未提供适当的编辑功能，这并不是设计者的疏漏，因为我们并不准备在这种操作方式下支持设计较长的程序，这种工作应转到提供了交叉汇编程序的PC机上去完成。相反的情况是，输入上述一些小程序，用监控程序的A命令完成，往往比用交叉汇编完成更简捷。

附：例子程序

例1 设计一个小程序，从键盘上接收一个字符并在屏幕上输出显示该字符。

【1】在命令行提示符状态下输入：

A 2000↵；

屏幕将显示：

2000：

输入如下形式的程序：

2000: IN 81 ; 判键盘上是否按了一个键

2001: SHR R0 ; 即串行口是否有了输入的字符

2002: SHR R0

2003: JRNC 2000 ; 未输入完则循环测试

2004: IN 80 ; 接收该字符

2005: OUT 80↵；在屏幕上输出显示字符‘6’

2006: RET↵；每个用户程序都必须用RET指令结束

2007: ↵；(按回车键即结束输入过程)

注：在十六位机中，基本I/O接口的地址是确定的，数据口的地址为80，状态口的地址为81。

【2】用“G”命令运行程序

在命令行提示符状态下输入：G2000↵

执行上面输入的程序，结果是什么？

注：该例建立了一个从主存2000H地址开始的小程序。在这种方式下，所有的数字都约定使用16进制数，故数字后不用跟字符H。每个用户程序的最后一个语句一定为RET汇编语句。因为监控程序是选用类似子程序调用方式使实验者的程序投入运行的，用户程序只有用RET语句结束，才能保证程序运行结束时能正确返回到监控程序的断点，保证监控程序能继续控制教学机的运行过程。

例2 设计一个小程序，用次数控制在终端屏幕上输出‘0’到‘9’十个数字符。

【1】在命令行提示符状态下输入：A 2020✓

屏幕将显示：

2020:

从地址2020H开始输入下列程序：

```
2020:MVRD R2, 000A    ; 送入输出字符个数
2022:MVRD R0, 0030    ; “0”字符的ASCII码送寄存器R0
2024:OUT 80           ; 输出保存在R0低位字节的字符
2025:DEC R2           ; 输出字符个数减1
2026:JRZ 202E         ; 判10个字符输出完否, 已完, 则转到程序结束处
2027:PUSH R0          ; 未完, 保存R0的值到堆栈中
2028:IN 81            ; 查询接口状态, 判字符串行输出完成否,
2029:SHR R0           ;
202A:JRNC 2028         ; 未完成, 则循环等待
202B:POP R0           ; 已完成, 准备输出下一字符并从堆栈恢复R0的值
202C:INC R0           ; 得到下一个要输出的字符
202D:JR 2024          ; 转去输出字符
202E:RET
202F: ✓
```

该程序的执行码放在2020H起始的连续内存区中。若送入源码的过程中有错，系统会进行提示，等待重新输入正确汇编语句。在输入过程中，在应输入语句的位置直接打回车则结束输入过程。

【2】用“G”命令运行程序

在命令行提示符状态下输入：G2020

执行结果为？

例3 从键盘上连续打入多个属于‘0’到‘9’的数字符并在屏幕上显示，遇到非数字字符结束输入过程。

【1】在命令行提示符状态下输入：A 2040✓

屏幕显示如下：2040：从地址2040H开始输入下列程序：

```
(2040)  MVRD R2, 0030      ; 用于判数字字符的下界值
        MVRD R3, 0039      ; 用于判数字字符的上界值
(2044)  IN 81               ; 判键盘上是否按了一个键,
        SHR R0              ; 即串行口是否有了输入的字符
        SHR R0
        JRNC 2044           ; 没有输入则循环测试
        IN80                ; 输入字符到R0
        MVRD R1, 00FF
        AND R0, R1          ; 清零R0的高位字节内容
        CMP R0, R2          ; 判输入字符 $\geq$ 字符'0' 否
        JRNC 2053           ; 为否, 则转到程序结束处
        CMP R3, R0          ; 判输入字符 $\leq$ 字符'9' 否
        JRNC 2053           ; 为否, 则转到程序结束处
        OUT 80              ; 输出刚输入的数字字符
        JMPA 2044           ; 转去程序前边2044处等待输入下一个字符
(2053)  RET
```

【2】在命令行提示符状态下输入：G 2040✓

执行结果为？

例4 计算1到10的累加和。

【1】在命令行提示符状态下输入：

A 2060✓屏幕将显示：

2060：从地址2060H开始输入下列程序：

```
(2060)  MVRD R1, 0000      ; 置累加和的初值为0
        MVRD R2, 000A      ; 最大的加数
        MVRD R3, 0000
(2066)  INC R3              ; 得到下一个参加累加的数
        ADD R1, R3          ; 累加计算
        CMP R3, R2          ; 判是否累加完
```

JRNZ 2066 ; 未完, 开始下一轮累加

RET

【2】在命令行提示符状态下输入: G 2060✓

注: 运行过后, 可以用R命令观察累加器的内容。R1的内容为累加和。

执行结果为?

例5 设计一个有读写内存和子程序调用指令的程序, 功能是读出内存中的字符, 将其显示到显示器的屏幕上, 转换为小写字母后再写回存储器原存储区域。

【1】将被显示的6个字符 ‘A’ ~ ‘F’ 送入到内存20F0H开始的存储区域中。

在命令行提示符状态下输入:

E 20F0✓ 屏幕将显示:

20F0 内存单元原值:

按下列格式输入:

20F0 内存原值: 0041 内存原值: 0042 内存原值: 0043

内存原值: 0044 内存原值: 0045 内存原值: 0046✓

在命令行提示符状态下输入:

从地址2080H开始输入下列程序:

(2080) MVRD R3, 0006 ; 指定被读数据的个数

MVRD R2, 20F0 ; 指定被读、写数据内存区首地址

(2084) LD RR R0, [R2] ; 读内存中的一个字符到R0寄存器

CALA 2100 ; 指定子程序地址为2100, 调用子程序, 完成显示、
转换并写回的功能

DEC R3 ; 检查输出的字符个数

JRZ 208B ; 完成输出则结束程序的执行过程

INC R2 ; 未完成, 修改内存地址

JR 2084 ; 转移到程序的2086处, 循环执行规定的处理

(208B) RET

从地址2100H开始输入下列程序:

(2100) OUT 80 ; 输出保存在R0寄存器中的字符

MVRD R1, 0020

ADD R0, R1 ; 将保存在R0中的大写字母转换为小写字母

STRR [R2], R0 ; 写R0中的字符到内存，地址同LOD所用的地址

(2105) IN 81 ; 测试串行接口是否完成输出过程

SHR R0

JRNC 2105 ; 未完成输出过程则循环测试

RET ; 结束子程序执行过程，返回主程序

【2】在命令行提示符状态下输入：G 2080✓

屏幕显示运行结果为？

【3】在命令行提示符状态下输入：D 20F0✓

20F0H~20F5H内存单元的内容为？

例6 设计一个程序在显示器屏幕上循环显示95个(包括空格字符)可打印字符。

【1】在命令行提示符状态下输入：A 20A0✓

屏幕将显示：20A0：

从地址20A0H开始输入下列程序：

A 20A0 ; 从内存的20A0地址单元开始建立用户的第一个程序

20A0: MVRD R1, 7E ; 向寄存器传送立即数

20A2: MVRD R0, 20 ;

20A4: OUT 80 ; 通过串行接口输出R0低位字节内容到显示器屏幕

20A5: PUSH R0 ; 保存R0寄存器的内容到堆栈中

20A6: IN 81 ; 读串行接口的状态寄存器的内容

20A7: SHR R0 ; R0寄存器的内容右移一位，最低位的值移入标志位C

20A8: JRNC 20A6 ; 条件转移指令，当标志位C不是1时就转到20A6地址

20A9: POP R0 ; 从堆栈中恢复R0寄存器的原内容

20AA: CMP R0, R1 ; 比较两个寄存器的内容是否相同，相同则标志位Z=1

20AB: JRZ 20AE ; 条件转移指令，当标志位Z为1时转到20AE地址, 结束程序

20AC: INC R0 ; 把R0寄存器的内容增加1

20AD: JR 20A4 ; 无条件转移指令，一定转移到20A4地址

20AE: RET ; 子程序返回指令, 程序结束

【2】在命令行提示符状态下输入：G 20A0✓

运行过后，可以观察到显示器上会显示出所有可打印的字符。

七、实验结果分析与总结

1. 执行例1输入的程序，结果是什么？
2. 执行例2输入的程序，结果是什么？
3. 执行例3输入的程序，结果是什么？
4. 执行例4输入的程序，结果是什么？
5. 执行例5输入的程序，屏幕显示运行结果是什么？在命令行提示符状态下输入：D 20F0↵ 20F0H~20F5H内存单元的内容分别是什么？
6. 执行例6输入的程序，结果是什么？

八、思考题

若把例2中的IN 81, SHR R0, JRNC 2028三个语句换成4个MVR R0, R0语句，该程序执行过程会出现什么现象？试分析并实际执行一次。

提示：该程序改变这三条语句后，若用T命令单条执行，会依次显示0~9十个数字。若用G命令运行程序，程序执行速度快，端口输出速度慢，这样就会跳跃输出。

在命令行提示符状态下输G 2020，屏幕显示09。

类似的，若要求在终端屏幕上输出'A'到'Z'共26个英文字母，应如何修改例1中给出的程序？请验证之。