



**实验报告**

**课程名称**  **计算机组成原理**

**学生学院 计算机学院**

**专业班级 计算机科学与技术5班**

**指导教师 叶林锋**

**学生姓名 黄国航**

**学 号 3116004636**

2018年6 月 4 日

目录

[实验一：基础汇编语言程序设计实验 4](#_Toc515831519)

[1实验目的 4](#_Toc515831520)

[2实验设备及器材 4](#_Toc515831521)

[3实验说明和原理 4](#_Toc515831522)

[4实验内容 5](#_Toc515831523)

[5实验步骤 5](#_Toc515831524)

[6 思考题 9](#_Toc515831525)

[7实验心得 9](#_Toc515831526)

[实验二：脱机运算器实验 10](#_Toc515831527)

[1实验目的 10](#_Toc515831528)

[2实验设备与器材 10](#_Toc515831529)

[3实验说明和原理 10](#_Toc515831530)

[4实验内容 11](#_Toc515831531)

[5实验步骤 11](#_Toc515831532)

[6 思考题 13](#_Toc515831533)

[7实验心得 13](#_Toc515831534)

[实验三：存储器部件教学实验 14](#_Toc515831535)

[1实验目的 14](#_Toc515831536)

[2实验设备与器材 14](#_Toc515831537)

[3实验说明和原理 14](#_Toc515831538)

[4实验内容 15](#_Toc515831539)

[5实验步骤 15](#_Toc515831540)

[6 思考题 18](#_Toc515831541)

[7实验心得 19](#_Toc515831542)

[实验四：组合逻辑控制器部件教学实验 20](#_Toc515831543)

[1实验目的 20](#_Toc515831544)

[2实验设备与器材 20](#_Toc515831545)

[3实验说明和原理 20](#_Toc515831546)

[4实验内容 21](#_Toc515831547)

[5实验步骤 22](#_Toc515831548)

[6实验心得 26](#_Toc515831549)

[实验五：微程序控制器部件教学实验 27](#_Toc515831550)

[1实验目的 27](#_Toc515831551)

[2实验设备与器材 27](#_Toc515831552)

[3实验说明和原理 27](#_Toc515831553)

[4实验内容 28](#_Toc515831554)

[5实验步骤 29](#_Toc515831555)

[6思考题 33](#_Toc515831556)

[7实验心得 33](#_Toc515831557)

[实验六：输入/输出接口扩展实验 34](#_Toc515831558)

[1实验目的 34](#_Toc515831559)

[2实验设备与器材 34](#_Toc515831560)

[3实验说明和原理 34](#_Toc515831561)

[4实验内容 35](#_Toc515831562)

[5实验步骤 35](#_Toc515831563)

[6思考题 36](#_Toc515831564)

[7实验心得 37](#_Toc515831565)

[实验七：中断实验 38](#_Toc515831566)

[1实验目的 38](#_Toc515831567)

[2实验设备与器材 38](#_Toc515831568)

[3实验说明和原理 38](#_Toc515831569)

[4实验内容 39](#_Toc515831570)

[5实验步骤 39](#_Toc515831571)

[6实验心得 45](#_Toc515831572)

[实验九：FPGA芯片实现非流水线的CPU系统(综合实验) 46](#_Toc515831573)

[1实验目的 46](#_Toc515831574)

[2实验设备与器材 46](#_Toc515831575)

[3实验说明和原理 47](#_Toc515831576)

[4实验内容 47](#_Toc515831577)

[5实验步骤 47](#_Toc515831578)

[6实验心得 49](#_Toc515831579)

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 基础汇编语言程序设计

# 实验一：基础汇编语言程序设计实验

## 1实验目的

* 学习和了解TEC-XP+教学实验监控命令的用法;
* 学习和了解TEC-XP+教学实验系统的指令系统;
* 学习简单的TEC-XP+教学实验系统汇编程序设计。

## 2实验设备及器材

* 工作良好的PC机;
* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

实验原理在于汇编语言能够直接控制底层硬件的状态，通过简单的汇编指令查看、显示、修改寄存器、存储器等硬件内容。

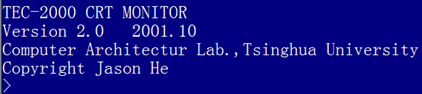
实验箱正如一集成的开发板，而我们正是通过基础的汇编语言对开发板进行使用和学习，过程中我们不仅需要运用汇编语言的知识，还需要结合数字逻辑中所学的关于存储器、触发器等基本器件的原理，通过串口通讯，实现程序的烧录，实验箱与PC端的通讯。

## 4实验内容

* 1. 学习联机使用TEC－XP+教学实验系统和仿真终端软件PCEC；
  2. 学习使用WINDOWS界面的串口通讯软件；
  3. 使用监控程序的R命令显示/修改寄存器内容、D命令显示存储内容、E命令修改存储内容；
  4. 使用A命令写一小段汇编程序，U命令反汇编输入的程序，用G命令连续运行该程序，用T、P命令单步运行并观察程序单步执行情况。

## 5实验步骤

1. 准备一台串口工作良好的PC机器;
2. 将TEC-XP+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. 将黑色的电源线一段接220V交流电源,另一端插在TEC-XP+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-XP+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-XP+实验系统左下方的六个黑色的控制机器运行状态的开关置于正确的位置,再找个实验中开关应置为001100(连续、内存读指令、组合逻辑、联机、16位、MACH),
6. 控制开关的功能在开关上、下方有标识;开关拨向上方表示"1",拨向下方表示"0","X"表示任意,其他实验相同;
7. 打开电源,船型开关盒5V电源指示灯亮;
8. 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",其他的设置一般不用改动,直接回车即可; （8）按一下"RESET"按键,再按一下"START"按键,主机上显示：



1. 实验示例：

【例1】设计一个小程序，从键盘上接收一个字符并在屏幕上数出显示该字符。

<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 ：判断键盘上是否按了建

2001：SHR R0 ：即串行口是否有输入的字符

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

在命令提示符状态下输入：

G 2000↓

执行上面输入的程序

光标闪烁等待输入，用户从键盘入字符后，屏幕会显示该字符。

该例建立了一个从主存2000H地址开始的小程序。这种方式下，所的数字都约定使用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。

测试如下图：



**【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

**代码输入：**

A 2020

2020 MVRD R2，000A ；送入输出字符个数

MVRD R0，0030 ；“0”字符的ASCII码送寄存器R0

OUT 80 ；输出保存在R0低位字节的字符

DEC R2 ；输出字符个数减1

JRZ 202E ；判10个字符输出完否，已完，则转到程序结束处

PUSH R0 ；未完，保存肋的值到堆栈中

2028 IN 81 ；查询接口状态，判字符串行输出完成否，

SHR R0

JRNC 2028 ；未完成，则循环等待

POP R0 ；已完成，准备输出下一字符并从堆栈恢复R0的值

INC R0 ；得到下一个要输出的字符

JR 2024 ；转去输出字符

202E RET

该程序的执行码放在2020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，等待重新打入正确汇编语句。在输入过程中，在应打入语句的位置直接打回车则结束输入过程。

**结果显示：**



**【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.

**代码输入：**



**结果显示：**



**【例4】**计算1到10的累加和。

**代码输入：**



**结果显示：**



## 6 思考题

问题：若把IN 81，SHR R0，JRNC 2028三个语句换成3个MVRR R0，R0语句，该程序执行过程会出现什么现象？

答：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，由于程序运行速度快，端口串行输出的速度很慢，这样新会丢失某些输出的字行；如，在命令行提示符状态下输入G 2020命令后，屏幕可能的显示为059或069，而不是023456789全部的10个字符。

## 7实验心得

这是第一次计组的上机实验，初步了解到TEC-XP+教学系统的概况，通过仿真终端软件PCEC以及WINDOWS界面的串口通讯软件，结合对汇编基本指令的学习，让我们熟悉计组的实验工具和流程，为后续的实验打下基础。

对汇编基本指令的学习，让我对汇编语言以及嵌入式系统方向产生了兴趣，并且通过汇编语言的学习，发现在高级语言中简单的加法或变量申请，在底层实现的过程是相当复杂的。使用汇编语言直接控制硬件，让我们亲身感受到一个简单的指令操作硬件时，硬件所发生的变化，貌似回到了70年代的计算机学习一样，而且在实验一的学习过程中，懂得了一些基于实验箱的汇编小程序，与单片机的使用有很多相似之处，十分有趣。

另外，实验一过程中使用绝对地址，直接访问内存的数据，让我们脱离操作系统的封装，更好地体会到直接控制硬件的感觉。然而，对于大程序的编写，基本的汇编语言的编程难度比起高级语言高了不少，但汇编的学习的确促进了我们对计算机的认识。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 脱机运算器实验

# 实验二：脱机运算器实验

## 1实验目的

* 深入了解AM2901运算器的功能与具体用法；
* 深化运算器部件的组成、设计、控制与使用等知识。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

本次实验脱离PC端，主要通过实验箱进行。因此这个实验更像数据逻辑的电路实验。脱机运算器实验，是指让运算器从教学计算机整机中脱离出来，此时，它的全部控制与操作均需通过两个12位的微型开关来完成，这就谈不上执行指令，只能通过开关、按键控制教学机的运算器完成指定的运算功能，并通过指示灯观察运算结果。

## 4实验内容

在脱机方式下，对于给定指令分析其执行过程中运算的步骤，通过对AM2901运算器反需控制信号的设置，使之完成运算，并核对运算结果。

## 5实验步骤

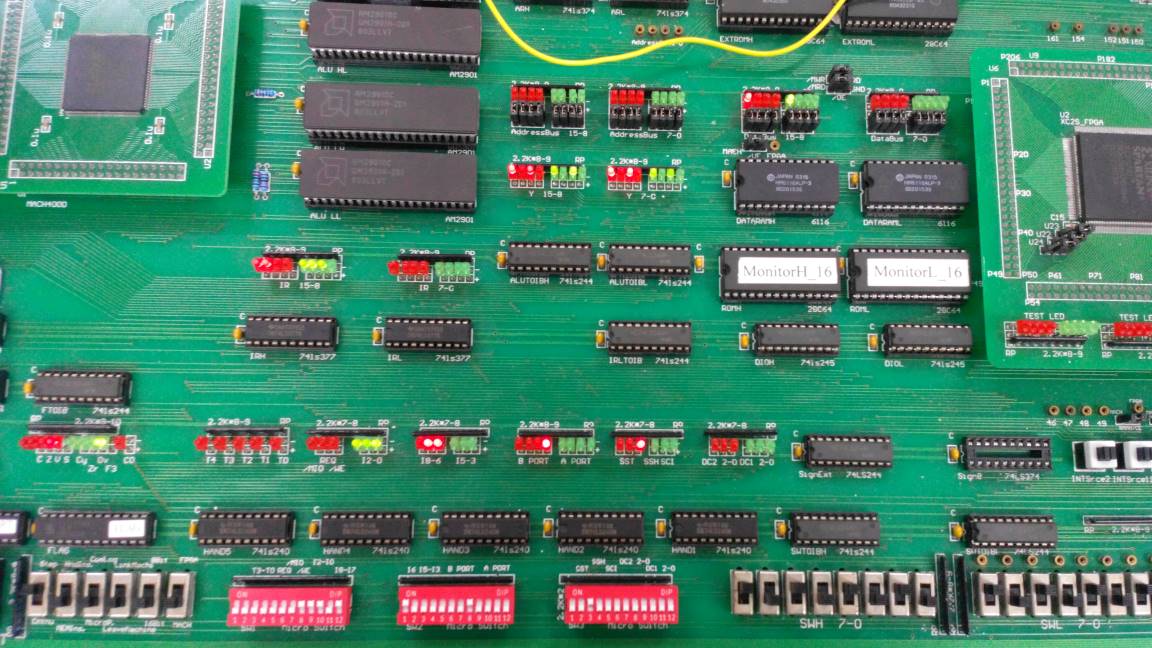
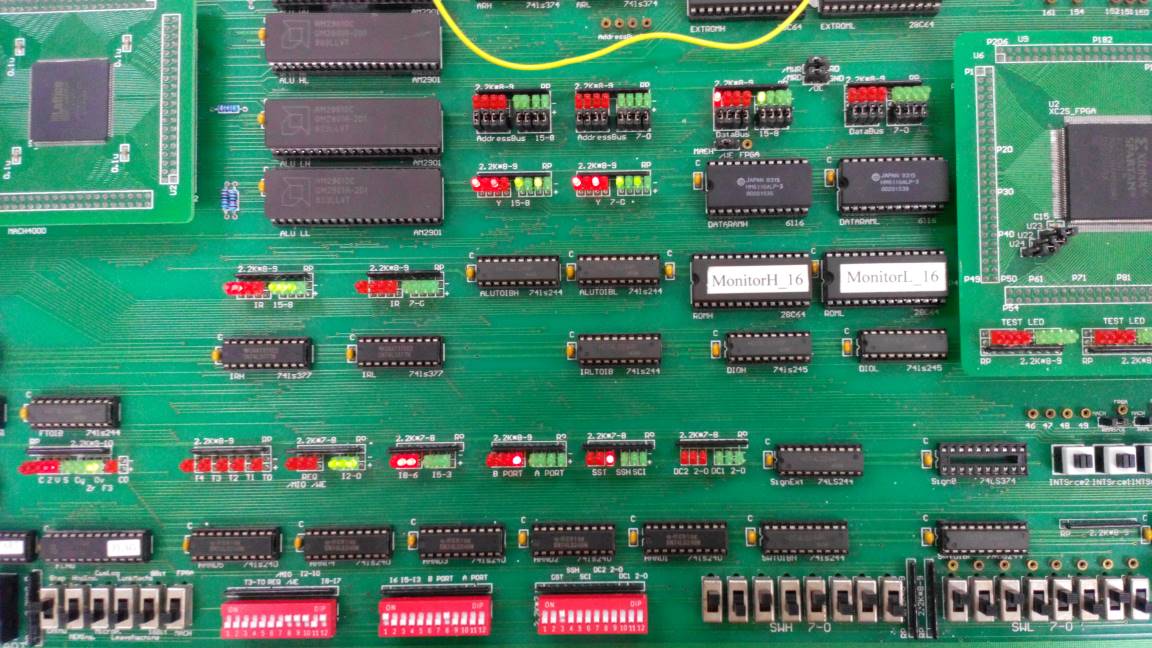
在脱机方式下，通过设置 SW1、SW2、SW3 各微码字段和数据开关，可实现多种运算，以实现AAAAH－>R1为例说明操作过程。

1. 按照下表中的微码和数据开关，对运算器进行设置。操作步骤如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 微动开关 | | | | | | | | 数据开关 |
| I8-I6 | I5-I3 | I2-I0 | SST | SSH | SCi | B | A | D15-D0 |
| 011 | 000 | 111 | 011 | 00 | 00 | 0001 | 不用 | AAAA H |

1. 将教学机左下方的6个拨动开关置为1XX000（单步、16位、脱机、 CACH）；先按一下RESTET按键，再按一下START按键，进行初始化。
2. 通过16个数据开关设置立即数AAAAH。
3. 通过SW1、SW2、SW3设置各微码。
4. 按一次START键，立即数XXXX H置入R1，通过显示灯察看按START键后的输出。若要进行其它操作：
5. 重新设置SW1、SW2、SW3、数据开关，通过显示灯观察输出。
6. 然后按START键执行操作，通过显示灯观察按下START键后的输 出，检查运算结果是否正确。
7. 完成下表的各种运算，记录按START键前和按START键后的ALU输出及标志位C、Z、V、S的值。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 运算 | **I8—I0** | SST | SSH | SCi | B | A | 压STEP前 | | | 压STEP后 | |
| ALU输出 | | CZVS | ALU输出 | CZVS |
| D1+0—＞R0 | 011000111 | 001 | 00 | 00 | 0000 | 不用 | 01h | | 随机 | 01h | 0000 |
| D2+0—＞R1 | 011000111 | 001 | 00 | 00 | 0001 | 不用 | 10h | | 0000 | 10h | 0000 |
| R0 + R1—＞R0 | 011000001 | 00l | 00 | 00 | 0000 | 0001 | 11h | | 0000 | 21h | 0000 |
| R0-R1—＞R0 | 011001001 | 001 | 00 | 0l | 0000 | 0001 | 01t1 | | 0000 | F1h | 1000 |
| R1-R0—＞R1 | 011001001 | 001 | 00 | 0l | 000l | 0000 | 0Fh | | 1000 | 0Eh | 1000 |
| R0 ∨ Rl—＞R0 | 011011001 | 001 | 00 | 00 | 0001 | 0000 | 0FH | 1000 | | 0EH | 1000 |
| R0 ∧ R1—＞R0 | 011100001 | 001 | 00 | 00 | 0000 | 0001 | 01H | 1000 | | 01H | 1000 |
| R0~~∨2~~Rl—＞R0 | 011110001 | 001 | 00 | 00 | 0000 | 0001 | 0EH | 1000 | | 01H | 1000 |
| ¬ (R0~~∨~~R1)—＞R0 | 011111001 | 001 | 00 | 00 | 0000 | 000l | FEH | 1000 | | 0EH | 1001 |
| 2\*R0—＞R0 | 111010011 | 001 | 00 | 00 | 0000 | 不用 | FEH | 100l | | FCH | 0001 |
| R0／2—＞0 | 101000011 | 001 | 00 | 00 | 0000 | 不用 | FCH | 0001 | | 7EH | 0001 |



## 6 思考题

问题：分析比较各指令按START键前和按START键后的值，是否有不同并解释。

前后结果有些不同，按”START”按键之前,ALU输出的是计算结果,参照ALU的操作周期的时序可知A,B口数据锁存是在时钟的下降沿,通用寄存器的接收是在低电平,所以要想寄存器接收ALU的计算结果必须按一次”START”按键.

## 7实验心得

初步了解运算器芯片Am2901的结构，本次实验初步知道它能实现R+S、S-R、R-S三种算术运算功能，和五种逻辑运算功能。选择ALU的八种运算（三种算术运算、五种逻辑运算）功能中的一种。是通过三位功能选择码I5、I4、I3实现。选择送入ALU的两个操作数据 R和S的组合关系（数据来源）。是用I2、I1、I0三位操作数选择码控制实现的。选择运算结果或有关数据以什么方式送往何处, 这主要指通用寄存器组和Q寄存器执不执行接收操作或移位操作, 以及向芯片的输出信息Y提供的是什么内容。这是通过I8、I7、I6三位结果选择码控制实现的。因此我们通过了解AM2901运算器的逻辑功能,懂得通过查表用指令微码来实现一些简单的逻辑功能.

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 存储器部件教学实验

# 实验三：存储器部件教学实验

## 1实验目的

* 熟悉ROM芯片和RAM芯片在功能和使用方法等方面的相同和差异之处。学习用编程器设备向EEPROM芯片内写入一批数据的过程和方法。
* 理解并熟悉通过字、位扩展技术实现扩展存储器系统容量的方案。
* 了解静态存储器系统使用的各种控制信号之间正常的时序关系。
* 了解如何通过读、写存储顺的指令实现对58C65 ROM芯片的读、写操作。
* 加深理解存储器部件在计算机整机系统中的作用。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

内存储器原理：内存储器是计算机中存放正在运行中的程序和相关数据的部件。在教学计算机存储器部件设计中，出于简化和容易实现的目的，选用静态存储器芯片实现内存储器的存储体，包括唯读存储区和随读写存储区两部分，ROM存储区选用4片长度8位、容易8KB的58C65芯片实现，RAM存储区选用2片长度8位、容量2KB的6116芯片实现，每2个8位的芯片合成一组用于组成16位长度的内存字，6个芯片被分成3组，其地址空间分配关系是：0-1777h用于第一组ROM，固化监控程序，2000-2777h用于RAM，保存用户程序和用户数据，其高端的一些单元作监控程序的数据区，第二组ROM的地址范围可以由用户选择，主要用于完成扩展内存容量的教学实验。

地址总线的低13位送到ROM芯片的地址线引脚，用于选择芯片内的一个存储字。用于实现存储字的高位字节的3个芯片的数据线引脚、实现低位字节的3个芯片的数据线引脚分别连接在一起接到数据总线的高、低位字节，是实现存储器数据读写的信息通路。数据总线要通过一个双向三态门电路与CPU一侧的内部总线IB相连接，已完成存储器、接口电路和CPU之间的数据通讯。

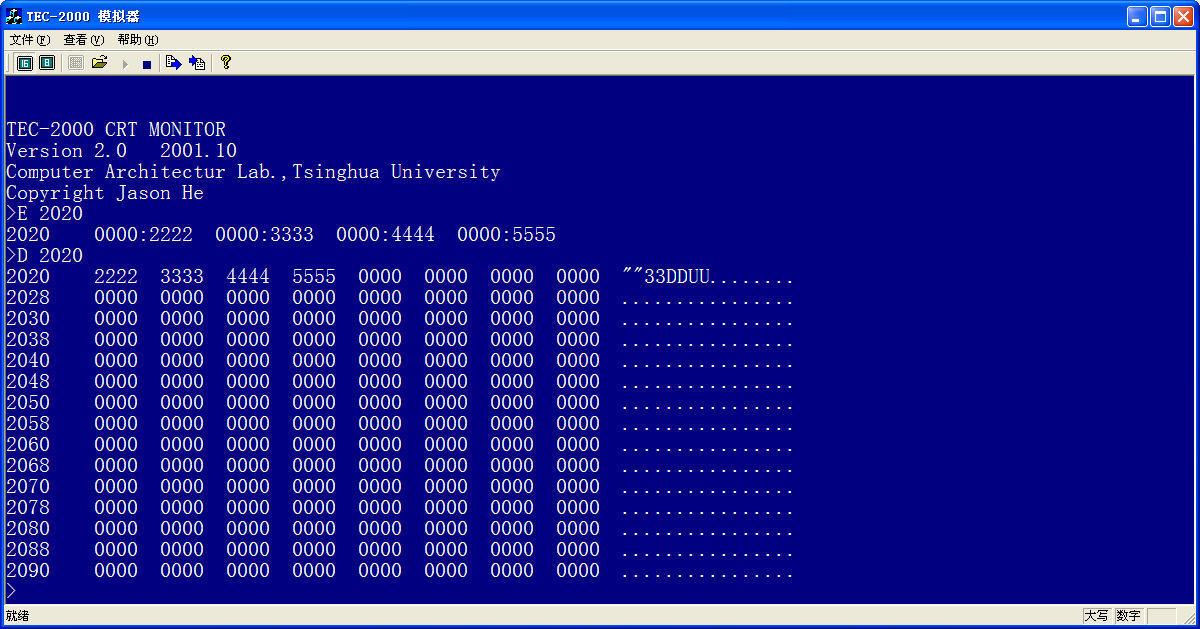
## 4实验内容

1. 要完成存储器容量扩展的教学实验，需为扩展存储器选择一个地址，并注意 读写和OE等控制信号的正确状态。
2. 用监控程序的D、E命令对存储器进行读写操作，比较RAM（3116）、EEPROM（28系列芯片）、EPROM（27系列芯片）在读写上的异同。
3. 用监控程序的A命令编写一段程序，对RAM（6116）进行读写，用D命令查看结果是否正确。
4. 用监控程序的A命令编写一段程序，对扩展存储器EEPRM（28系列芯片）进行读写，用D命令查看结果是否正确；如不正确，分析原因，改写程序，重新运行。

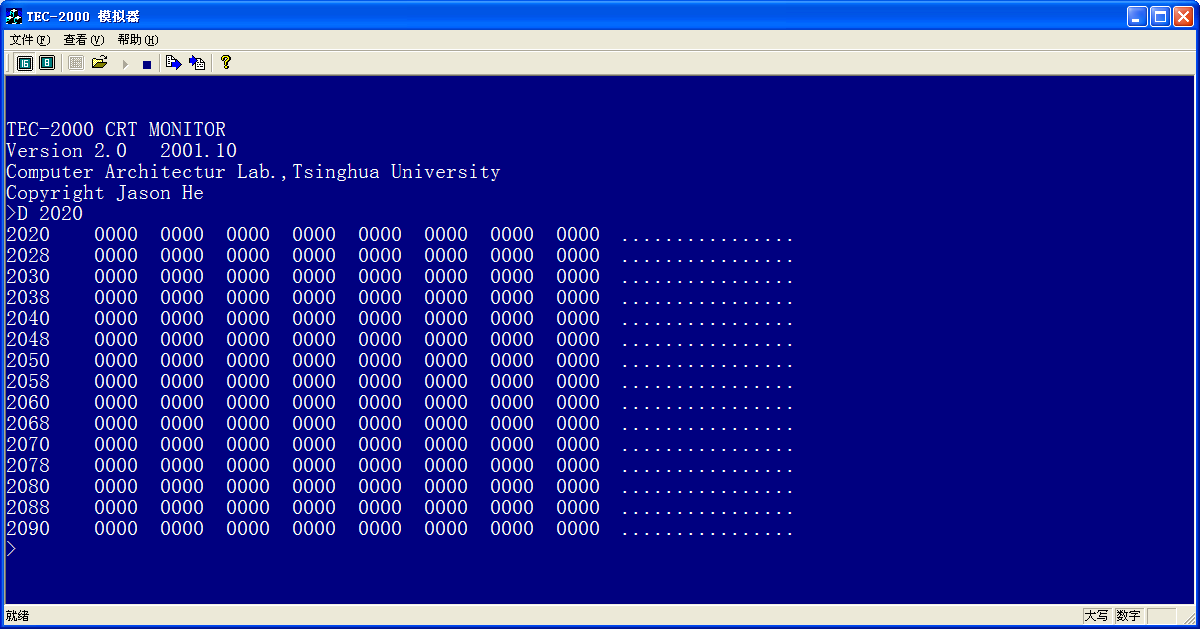
## 5实验步骤

1、RAM中的内容在断电后会消失，重新启动实验机后会发现内存单元的值发生了改变。

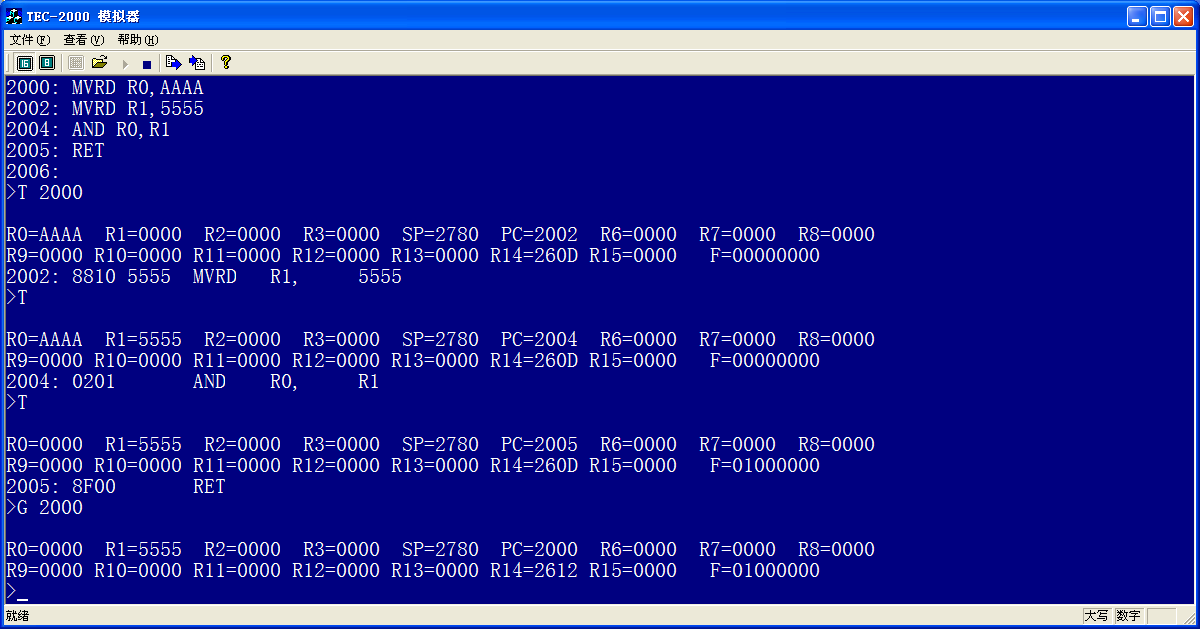
(1)用E命令改变内存单元的值并用D命令观察结果。



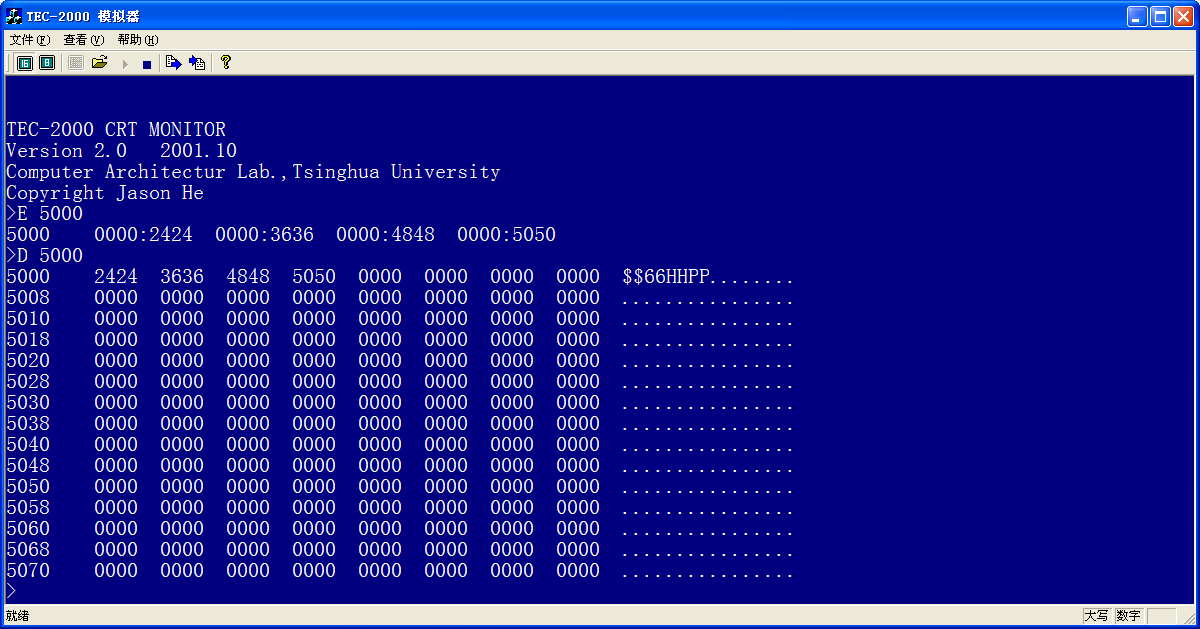
断电重启后：



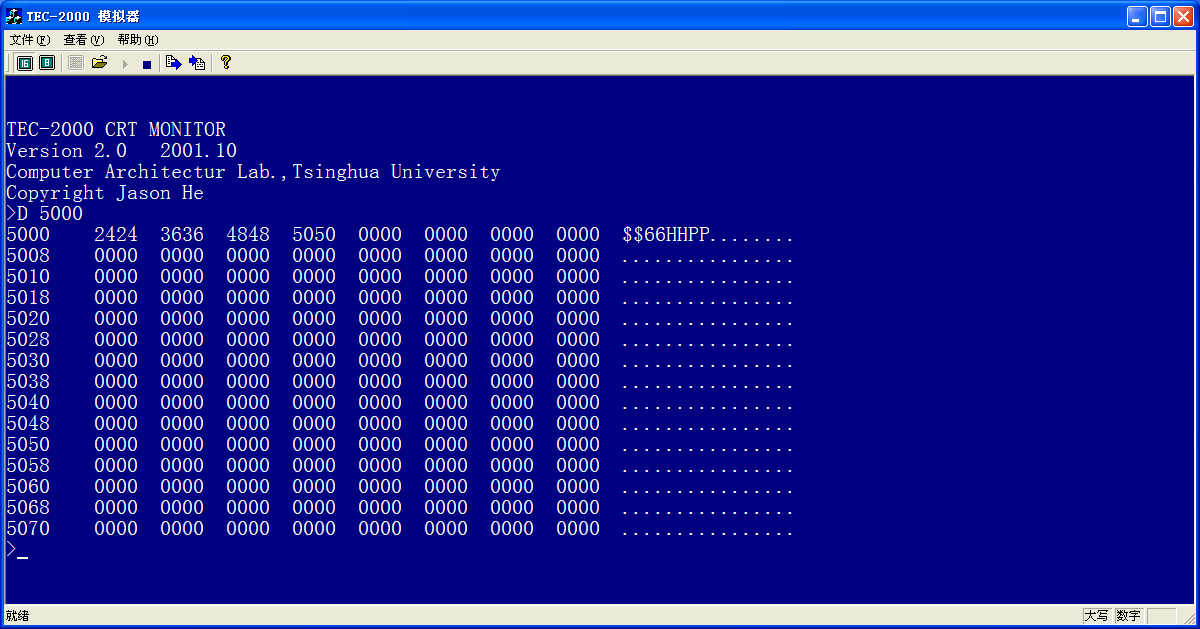
（2）用A命令输入一段程序，执行并观察结果。

2、EEPROM的内容断电后可保存，重新启动会发现内存单元的值没有发生改变。

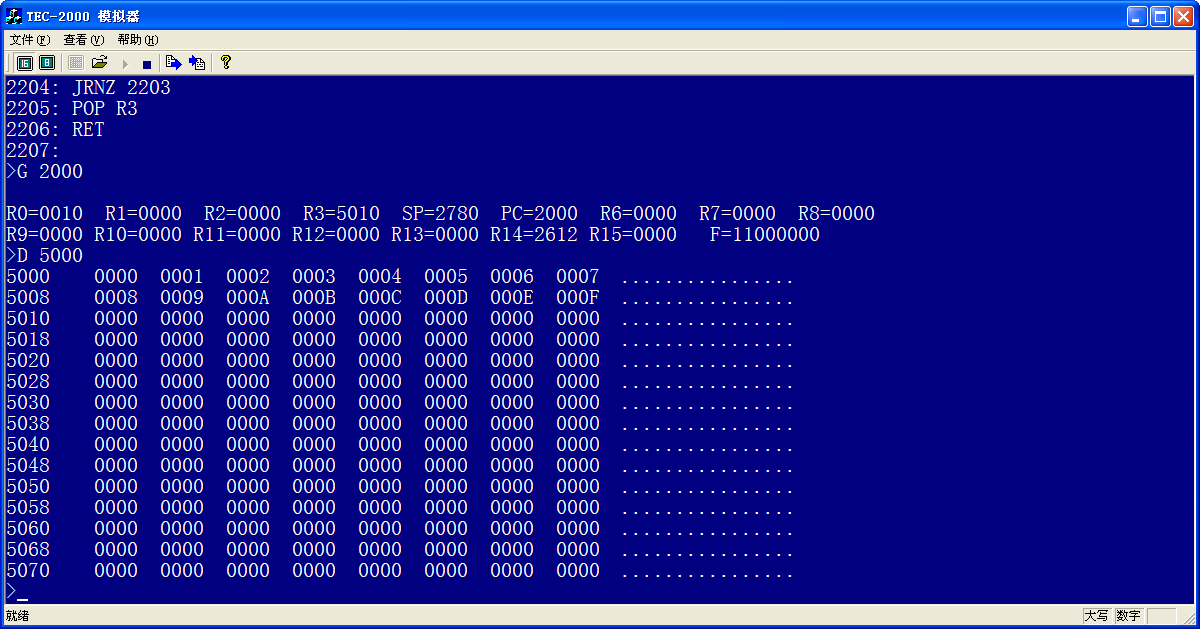
（1）用E命令改变内存单元的值并用D命令观察结果。



断电重启后：



（2）在5000H~500FH单元中依次写入数据0000H、0001H、…000FH。

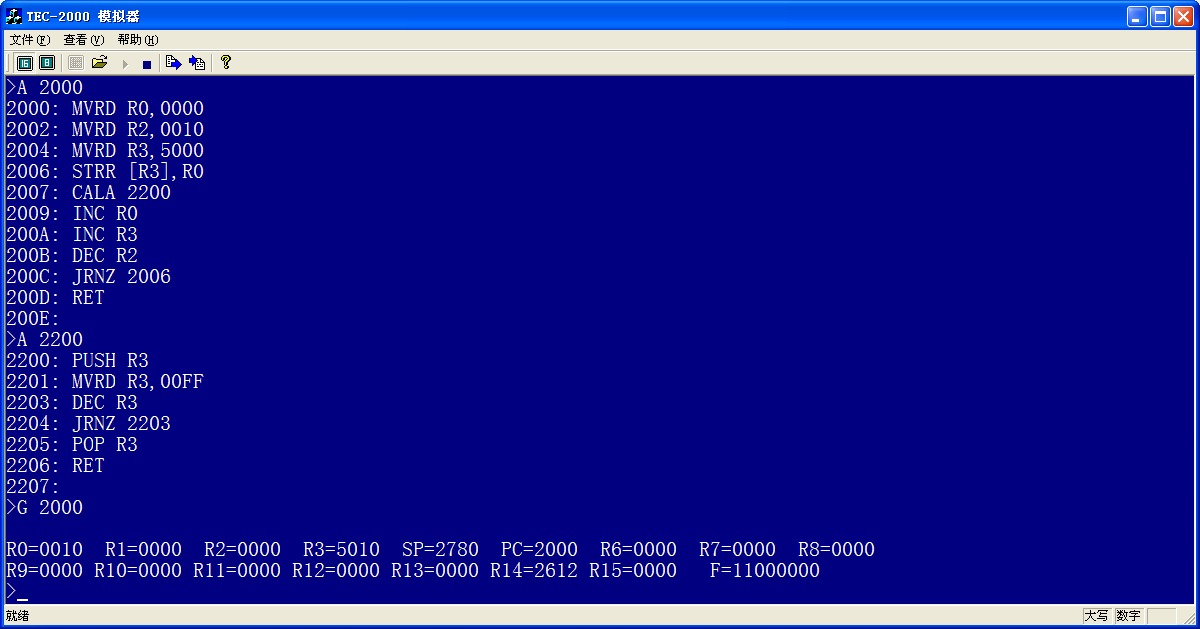


## 6 思考题

（1）为何能用E命令直接写AT28C64B的存储单元，而A命令则有时不正确？

答：E命令是储存寄存器指令A时监控器指令，而E直接多个程序写入AT28C64B的存储单元，写入的速度快，A命令只能是一次写入执行一条程序，是延迟指令，所以用E命令直接写AT28C64B的存储单元，而A命令则有时不正确。

（2）将延时改短，将程序中的R3的内容赋成00FF，再看运行结果。



## 7实验心得

能用E命令直接写AT28C64B的存储单元，而不用A命令，修改延时子程序，将其延时改短，可将延时子程序中的R3的内容赋成00FF或0FFF等，再看运行结果时，发现因为当写入双字节指令的低位时会出错，这也是不能用A命令而只能用E命令去写存储单元的原因。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 组合逻辑控制器部件教学实验

# 实验四：组合逻辑控制器部件教学实验

## 1实验目的

通过看懂教学计算中已经设计好并正常运行的几条典型指令的功能、格式和执行流程，然后自己设计几条指令的功能、格式、和执行流程，并在教学计算机上实现、调试正确。其最终要达到的目的是：

* 深入理解计算机控制器的功能、组成知识。
* 深入地学习计算机各类典型指令的执行流程。
* 指令格式、寻址方式、指令系统、指令分类等建立具体的总体概念。
* 学习组合逻辑控制器的设计过程和相关技术

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

控制器设计是学习计算机总体组成和设计的最重要部分。要在TEC-XP16教学计算机完成这项实验，必须清楚地懂得：

1. TEC-XP+教学机的组合逻辑控制器主要由MACH器件组成。
2. TEC-XP+教学机上已实现了29条基本指令。
3. 应了解监控程序的A命令只支持基本指令，扩展指令应用E命令将指令代码写入到相应的存储单元中；不能用T、P命令单步调试扩展指令，只能用G命令执行有扩展指令的程序。
4. 要明白TEC-XP+教学机支持的指令格式及指令执行流程分组情况；理解TEC-XP+教学机中已经设计好并正常运行的各类指令的功能、格式和执行流程、也包括控制器设计的实现中的具体路线的控制信号的组成。
5. 要明确自己要实现的指令功能、格式、执行流程设计中必须遵从的约束条件。

为了完成自己设计几条指令的功能、格式和执行流程，并在教学计算机上实现、调试正确的内容，具体过程包括：

1. 确定指令格式和功能，要受到教学机已有硬件的约束，应尽量与已实现指令的格式和分类办法保持一致。
2. 划分指令执行步骤并设计每一步的执行功能，设计节拍状态的值，应参照已实现指令的处理办法来完成，特别注意的是，读取指令的节拍只能用原来已实现的，其他节拍的节拍状态也应尽可能地与原用节拍的状态保持一致和相近。
3. 在指令流程表中真写每一个控制信号的状态值，基本上是个查表填数的过程，应特别仔细，并有意识地体会这些信号的控制作用。
4. 在给出来mach的源文件中添加扩展指令的控制信号的逻辑表达式，编译适配后下载到MACH器件中。
5. 写一个包含你设计的指令的程序，通过运行该程序检查执行结果的正确性，初步判断你的设计是否正确；如果有问题，通过几种办法查出错误并改正，继续调试，直到完全正确。

## 4实验内容

1. 完成控制器部件的教学实验，主要内容是由学生自己设计几条指令的功能、格式和执行流程，并在教学机上实现、调试正确。
2. 首先是看懂TEC-XP教学机的功能部件组成和线路逻辑关系，然后分析教学计算机中已经设计好并正常运行的几条典型指令的功能、格式和执行流程。
3. 设计几条指令的功能、格式和执行流程，并在教学计算机上实现、调试正确。
4. 单条运行指令。查看指令的功能、格式和执行流程。
5. 用监控程序的A、E命令编写一段小程序，观察运行结果。

## 5实验步骤

1. 接通教学机电源。
2. 将教学机左下方的6个拨动开关置为111100.
3. 按一下“RESET”按键。
4. 通过16位的数据开关SWH、SWL置入16位的指令操作码。
5. 在单步方式下，通过指示灯观察各类型基本指令的节拍。
6. 选择基本指令的A组指令中的ADD指令，观察其节拍流程： 1）、置拨动开关SW＝00000000 00000001； 2）、按RESET按键；节拍指示灯T3 ~T0显示1000； 3）、按START按键；节拍指示灯T3~T0显示0000； 4）、按START按键；节拍指示灯T3~T0显示0010； 5）、按START按键；节拍指示灯T3~T0显示0011；
7. 选择基本指令的B指令中的PUSH指令，观察其节拍流程：
8. 1）、置拨动开关SW＝10000101 00000000； 2）、按RESET按键；节拍指示灯T3 ~T0显示1000； 3）、按START按键；节拍指示灯T3~T0显示0000； 4）、按START按键；节拍指示灯T3~T0显示0010； 5）、按START按键；节拍指示灯T3~T0显示0110； 6）、按START按键；节拍指示灯T3~T0显示0100；
9. 选择基本指令的D组指令中的CALA指令，观察其节拍流程：
10. 1）、置拨动开关SW＝11000110 00000000； 2）、按RESET按键；节拍指示灯T3 ~T0显示1000；3）、按START按键；节拍指示灯T3~T0显示0000； 4）、按START按键；节拍指示灯T3~T0显示0010； 5）、按START按键；节拍指示灯T3~T0显示0110； 6）、按START按键；节拍指示灯T3~T0显示0100； 7）、按START按键；节拍指示灯T3~T0显示0111； 8）、按START按键；节拍指示灯T3~T0显示0101；
11. 单步方式下，通过指示灯观察各类基本指令的控制信号。
12. 选择指令的A组指令中的SHR指令，观察其执行过程中控制信号的变化，分析其作用。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **节拍** | **指令** | **编码** | **/MIO** | **REQ** | **/WE** | **A** | **B** | **Sci** | **SSH** | **I8-6** | **I5-3** | **I2-0** | **SST** | **DC1** | **DC2** |
| **1000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **011** | **001** | **001** | **000** | **000** | **111** |
| **0000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0010** |  |  | **0** | **0** | **1** | **0000** | **0000** | **00** | **0** | **001** | **000** | **000** | **000** | **000** | **001** |
| **0011** | **SHR** | **0000 1001** | **1** | **0** | **0** | **0000** | **0001** | **00** | **0** | **101** | **000** | **011** | **101** | **000** | **000** |

1. 选择基本指令的B组指令中的JMPA指令，观察其执行过程中控制信号的变化，分析其作用。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **节拍** | **指令** | **编码** | **/MIO** | **REQ** | **/WE** | **A** | **B** | **Sci** | **SSH** | **I8-6** | **I5-3** | **I2-0** | **SST** | **DC1** | **DC2** |
| **1000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **011** | **001** | **001** | **000** | **000** | **111** |
| **0000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0010** |  |  | **0** | **0** | **1** | **0000** | **0000** | **00** | **0** | **001** | **000** | **000** | **000** | **000** | **001** |
| **0110** | **JMPA** | **1000**  **0000** | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0100** | **JMPA** | **1000**  **0000** | **0** | **0** | **1** | **0000** | **0101** | **00** | **0** | **011** | **000** | **111** | **000** | **000** | **000** |

1. 选择基本指令的D组指令中的CALA指令，观察其执行过程中控制信号的变化，分析其作用。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **节拍** | **指令** | **编码** | **/MIO** | **REQ** | **/WE** | **A** | **B** | **Sci** | **SSH** | **I8-6** | **I5-3** | **I2-0** | **SST** | **DC1** | **DC2** |
| **1000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **011** | **001** | **001** | **000** | **000** | **111** |
| **0000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0010** | **CALA** | **1100**  **1110** | **0** | **0** | **1** | **0000** | **0000** | **00** | **0** | **001** | **000** | **000** | **000** | **000** | **001** |
| **0110** | **CALA** | **1100**  **1110** | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0100** | **CALA** | **1100**  **1110** | **0** | **0** | **1** | **0000** | **0000** | **00** | **0** | **000** | **000** | **111** | **000** | **000** | **000** |
| **0111** | **CALA** | **1100**  **1110** | **1** | **0** | **0** | **0000** | **0100** | **00** | **0** | **011** | **001** | **011** | **000** | **000** | **011** |
| **0101** | **CALA** | **1100**  **1110** | **0** | **0** | **0** | **0101** | **0101** | **00** | **0** | **010** | **000** | **010** | **000** | **001** | **000** |

1. 在以上几步实验的基础上，选择几条扩展指令，设计出扩展指令的节拍 和每拍对应的控制信号。
2. 选择扩展指令的A组指令中的RCR指令，观察其节拍流程：    1）、置拨动开关SW＝00101011 00010000； 2）、按RESET按键；节拍指示灯T3 ~T0显示1000； 3）、按START按键；节拍指示灯T3~T0显示0000； 4）、按START按键；节拍指示灯T3~T0显示0010； 5）、按START按键；节拍指示灯T3~T0显示0011；
3. 选择扩展指令的C级指令中的LDRA指令，观察其流程：    1）、置拨动开关SW＝11100100 00000000； 2）、按RESET按键；节拍指示灯T3 ~T0显示1000； 3）、按START按键；节拍指示灯T3~T0显示0000； 4）、按START按键；节拍指示灯T3~T0显示0010； 5）、按START按键；节拍指示灯T3~T0显示0110； 6）、按START按键；节拍指示灯T3~T0显示0111； 7）、按START按键；节拍指示灯T3~T0显示0101；
4. 设计几条扩展指令的控制信号如下：
5. 选择扩展指令ADC、STC、JRS、LDRX和JMPR。
6. 根据设计的控制信号的表格用ABEL语言编写MACH的逻辑表达式。
7. 将编好的程序MACHC.JED下载到MACH芯片内。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **节拍** | **指令** | **编码** | **/MIO** | **REQ** | **/WE** | **A** | **B** | **Sci** | **SSH** | **I8-6** | **I5-3** | **I2-0** | **SST** | **DC1** | **DC2** |
| **1000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **011** | **001** | **001** | **000** | **000** | **111** |
| **0000** |  |  | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0010** |  |  | **0** | **0** | **1** | **0000** | **0000** | **00** | **0** | **001** | **000** | **000** | **000** | **000** | **001** |
| **0011** | **ADC** | **00100000** | **1** | **0** | **0** | **SR** | **DR** | **10** | **0** | **011** | **000** | **001** | **001** | **000** | **000** |
|  | **JRS** | **01100100** | **1** | **0** | **0** | **0101** | **0101** | **00** | **0** | **0S1** | **000** | **101** | **000** | **010** | **000** |
|  | **STC** | **01101101** | **1** | **0** | **0** | **0000** | **0000** | **00** | **0** | **001** | **000** | **000** | **100** | **000** | **000** |
|  | **JMPR** | **01100000** | **1** | **0** | **0** | **SR** | **0101** | **00** | **0** | **011** | **000** | **100** | **000** | **000** | **000** |
| **0110** | **LDRX** | **11100101** | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
|  | **STRX** | **11100110** | **1** | **0** | **0** | **0101** | **0101** | **01** | **0** | **010** | **000** | **011** | **000** | **000** | **011** |
| **0111** | **LDRX** | **11100101** | **0** | **0** | **1** | **SR** | **0000** | **00** | **0** | **001** | **000** | **101** | **000** | **000** | **011** |
|  | **STRX** | **11100101** | **0** | **0** | **1** | **0000** | **DR** | **00** | **0** | **011** | **000** | **111** | **000** | **000** | **000** |
| **0101** | **LDRX** | **11100110** | **0** | **0** | **0** | **0000** | **DR** | **00** | **0** | **001** | **000** | **011** | **000** | **001** | **000** |

1. 用教学机已实现的基本指令和扩展的几条指令编写程序并运行，测试扩展的几条指令是否正确。
2. ADC指令

代码输入：

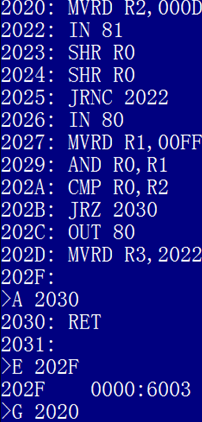
 

结果显示：



1. 测试JMPR

代码输入：

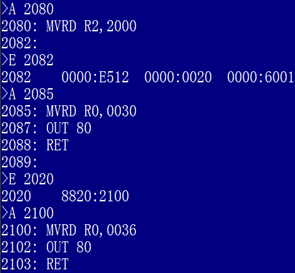


结果显示：



1. 测试LDRX、STRX

代码输入：



结果显示：



## 6实验心得

经过一段时间汇编语言的学习，发现我们的实验箱与8686架构的汇编指令的二进制表示代码并不相同。而通过学习ADD,SHR,OUT,MVRD,JRC,RET,CALA等指令，对硬件运行流程以及汇编指令的控制和调试都有了新的进步。另外实验箱的A命令只支持基础命令，而不支持扩展指令，当需要用到扩展指令时，要用E命令输入，并且通过G命令执行。这是后期实验需要注意的地方。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 微程序控制器部件教学实验

# 实验五：微程序控制器部件教学实验

## 1实验目的

* 全面掌握计算机各部件组成及相互联接关系。
* 深入理解计算机微程序控制器的功能、组成知识。
* 深入的学习计算机各类典型指令的执行流程。
* 对指令格式、寻址方式、指令系统、指令分类等建立具体的总体概念。
* 学习微程序控制器的设计过程和相关技术。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

控制器设计是学习计算机总体组成和设计的最重要部分。要在TEC-XP16教学计算机完成这项实验，必须清楚地懂得：

1. TEC-XP+教学机的微程序控制器主要由微程序定序器AM2910、产生当前微地址和下地 址的微控存和MACH器件组成。
2. TEC-XP+教学机上已实现了的全部基本指令和留给用户实现的19条扩展指令的控制信号都是由微控存和MACH给出的。
3. 应了解监控程序的A命令只支持基本指令，扩展指令应用E命令将指令代码写入到相应 的存储单元中；不能用T、P命令单步调试扩展指令，只能用G命令执行有扩展指令的 程序。
4. 要明白TEC-XP+教学机支持的指令格式及指令执行流程分组情况；理解TEC-XP+教学 机中已经设计好并正常运行的各类指令的功能、格式和执行流程、也包括控制器设计的 实现中的具体路线的控制信号的组成。
5. 要明确自己要实现的指令功能、格式、执行流程设计中必须遵从的约束条件。为了完成自己设计几条指令的功能、格式和执行流程，并在教学计算机上实现、调试正确的内容，具体过程包括：
6. 确定指令格式和功能，包括确定要用的操作码,指令中的其它字段的内容分配与使用， 要受到教学机已有硬件的约束，应尽量与已实现指令的格式和分类办法保持一致。
7. 按新指令的功能和格式、设计指令的执行流程。划分指令执行步骤并设计每一步的执行功能，设计微地址和下地址的取值，应参照已实现指令的处理办法来完成。
8. 在指令流程表中真写每一个控制信号的状态值，基本上是个查表填数的过程，应特别仔细，并有意识地体会这些信号的控制作用。
9. 将设计好的微码，装入控制存储器的相应单元。
10. 写一个包含你设计的指令的程序，通过运行该程序检查执行结果的正确性，来初步判断你的设计是否正确；如果有问题，通过几种办法查出错误并改正，继续调试，直到完全正确。

## 4实验内容

1. 完成控制器部件的教学实验，主要内容是自己设计几条指令的功能、格式和执行流程，并在教学计算机上实现、调试正确。
2. 首先是看懂TEC-XP+教学计算机的功能部件组成和线路逻辑关系，然后分析教学计算机中已经设计好并正常运行的几条典型指令的功能、格式和执行流程，注意各操作功能所对应的控制信号的作用。
3. 设计几条指令的功能、格式和执行流程，并在教学机上实现、调试正确。
4. 单条运行指令。查看指令的功能、格式和执行流程。
5. 用监控程序的A、E命令编写一段小程序，观察运行结果。

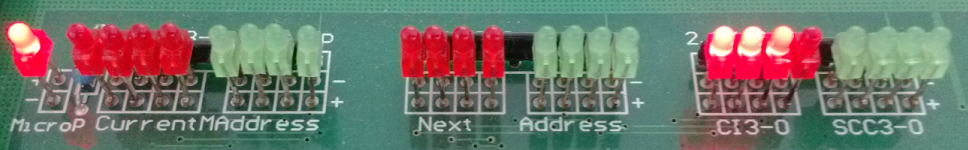
## 5实验步骤

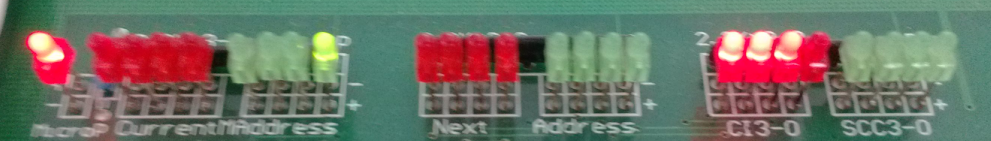
1. 接通教学机电源。
2. 将教学机左下方的六个拨动开关置为110100.
3. 按一下“RESET”按键。
4. 通过16位的数据开关SWH、SWL置入指令操作码。
5. 在单步方式下，通过指示灯观察各类基本指令的微码。

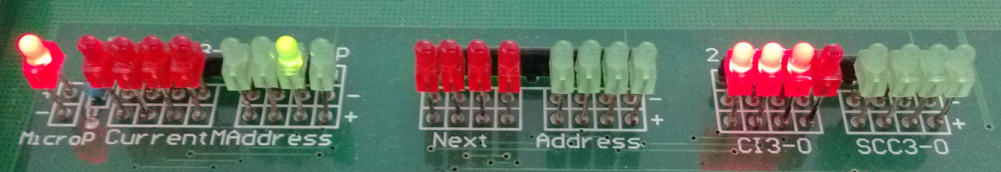
示例：选择基本指令的A组指令中的ADD指令，观察其流程

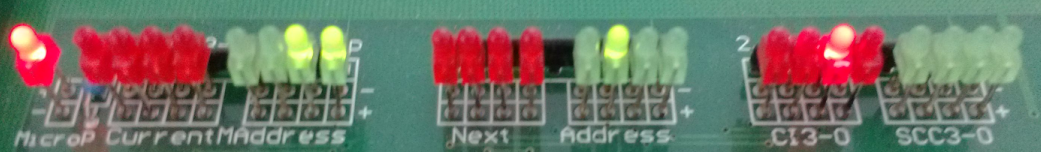
1. 置拨动开关SW＝00000000 00000001；
2. 按RESET按键，指示灯Microp亮，其它全灭。
3. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000，微址和下址的指示灯全灭；
4. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000——微址指示灯显示0000 0001，下址的指示灯全灭。
5. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000——微址指示灯显示0000 0010，下址的指示灯全灭。
6. 以上三步为公共操作，其它指令同；
7. 按START按键；指示灯CI3~0、SCC3~0显示0010 0000——微址指示灯显示0000 0011，下址的指示灯显示0000 0100。
8. 按START按键；指示灯CI3~0、SCC3~0显示0011 0000——微址指示灯显示0000 0100，下址的指示灯显示为0011 0000。
9. 按START按键；指示灯CI3~0、SCC3~0显示0011 0000——微址指示灯显示0011 1010，下址的指示灯显示0011 1010。
10. 按START按键；指示灯CI3~0、SCC3~0显示0011 0010——微址指示灯显示0011 1010，下址的指示灯显示0000 0010。

**实验箱部分实现过程如下：**









1. 在连续方式下，用A命令键入程序并运行。

示例：计算1到10的累加和

**代码输入：**



**结果显示：**



1. 设计几条指令的功能、格式和执行流程，设计每条微指令各字段的具体编码值，包括控制码的各字段、下地址字段、形成下址用到的条件码。
2. 扩展几条指令，确定各步的控制信号。
3. 将扩展好的控制信号添加到给出的MACH程序中，编译生成JED的熔丝图文件，写入MACH内的寄存器中。

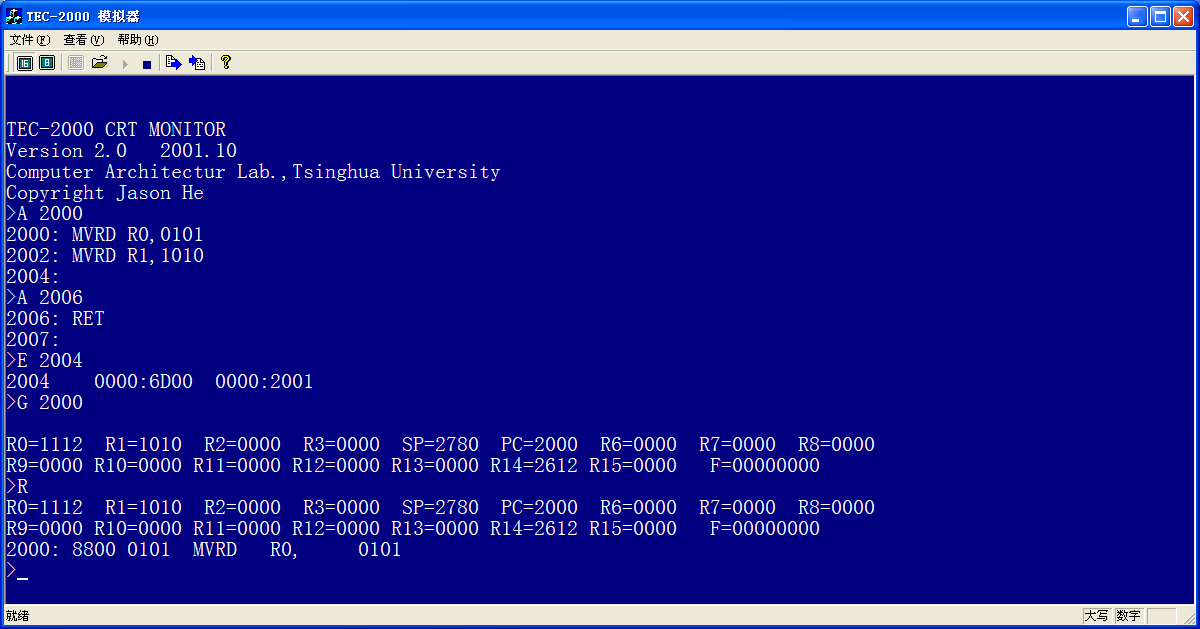
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | 操作功能 | 微址 | 下址 | CI3-0 | SCC3-0 | MRW | I2-0 | I8-I6 | I5-3 | B | A | SST | SSH  SCI | DC2 | DC1 |
| ADC | DR+SR+CF->DR | 50 | 30 | 0011 | 0000 | 100 | 001 | 011 | 000 | 0000 | 0000 | 001 | 010 | 000 | 000 |
| STC | STC | 57 | 30 | 0011 | 0000 | 100 | 000 | 001 | 000 | 0000 | 0000 | 100 | 000 | 000 | 000 |
| LDRA | PC->AR  PC+1->PC | 5B | 00 | 1110 | 0000 | 100 | 011 | 010 | 000 | 0101 | 0101 | 000 | 001 | 011 | 000 |
|  | MEM->AR | 5C | 1C | 0011 | 0000 | 001 | 111 | 001 | 000 | 0000 | 0000 | 000 | 000 | 011 | 000 |
| CALR | SP-1->SP,AB | 64 | 00 | 1110 | 0000 | 100 | 011 | 011 | 001 | 0100 | 0000 | 000 | 000 | 011 | 000 |
|  | PC->MEM | 65 | 00 | 1110 | 0000 | 000 | 100 | 001 | 000 | 0000 | 0101 | 000 | 000 | 000 | 001 |
|  | SP->PC | 66 | 30 | 0011 | 0000 | 100 | 100 | 011 | 000 | 0101 | 0000 | 000 | 000 | 000 | 000 |

1. 在单步方式下，通过指示灯观察各类扩展指令的微码。

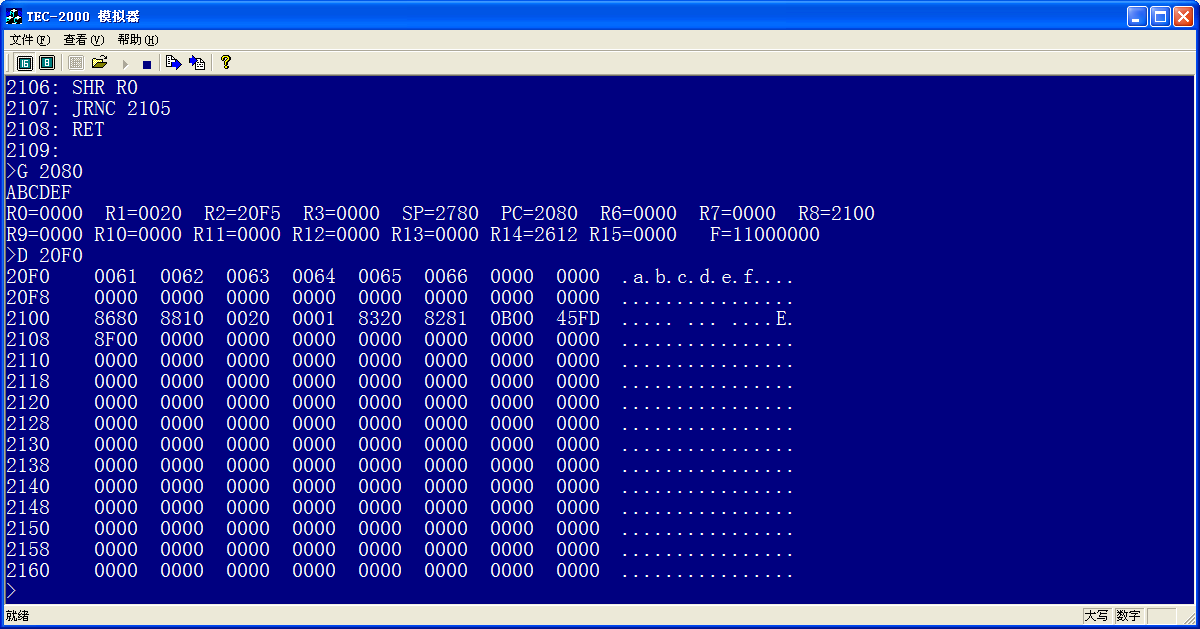
示例：选择基本指令的A组指令中的ADC指令，观察其流程。

1. 置拨动开关SW＝00100000 00000000；
2. 按RESET按键，指示灯Microp亮，其它全灭。
3. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000，微址和下址的指示灯全灭；
4. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000——微址指示灯显示0000 0001，下址的指示灯全灭。
5. 按START按键；指示灯CI3~0、SCC3~0显示1110 0000——微址指示灯显示0000 0010，下址的指示灯全灭。
6. 以上三步为公共操作，其它指令同；
7. 按START按键；指示灯CI3~0、SCC3~0显示0010 0000——微址指示灯显示0000 0011，下址的指示灯显示0101 0000。
8. 按START按键；指示灯CI3~0、SCC3~0显示0011 0000——微址指示灯显示0101 0000，下址的指示灯显示为0011 0000。
9. 按START按键；指示灯CI3~0、SCC3~0显示0011 0010——微址指示灯显示0011 0000，下址的指示灯显示0011 1010。
10. 按START按键；指示灯CI3~0、SCC3~0显示0011 0000——微址指示灯显示0011 0001，下址的指示灯显示0000 0010。
11. 用A、E键入程序连续运行。
    1. 示例1：测试ADC指令。

**代码输入：**

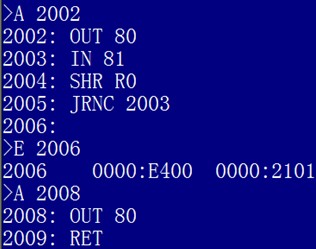


b) 示例2: 测试CALR指令:



c)示例3：测试LDRA指令

**代码输入：**



**结果显示：**



## 6思考题

在 A 命令下能否直接输入新指令？为什么？

答：不能，A 命令只持基本指令，扩展应用指令应用 E 命令将指令代码写入到相 应的存储单元中

## 7实验心得

本次实验让我们对指令的格式、寻址方式以及指令系统有了一个总体概念，特别是对于扩展指令的认识与使用得到了加深，并且可以自主实现自设微码调用微指令，初步掌握了微程序控制器的设计过程。

此外，我们通过对指令的执行，更为深入地知晓汇编典型指令的执行流程，有了感性的认识，减少了学习的枯燥。本次实验需要用到的单步运行指令较多，慢慢地习惯了对指令的调试以及内存或寄存器的查询。还有，当我们使用扩展指令时，需要用E命令输入。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 输入/输出接口扩展实验

# 实验六：输入/输出接口扩展实验

## 1实验目的

* 了解串行接口与计算机主机之间的数据传送方。
* 理解串行接口芯片的内部组成的传送数据的运行方式。
* 了解串行接口在投入运行之前必须执行的初始化操作的作用及完成初始化操作的具体方案。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

1. 串行通信是指外设和计算机间使用一根数据信号线,数据在一根数据信号线上按位进行传输，每一位数据都占据一个固定的时间长度。
2. 这种通信方式使用的数据线少，在远距离通信中可以节约通信成本，当然，其传输速度比并行传输慢。相比之下，由于高速率的要求，处于计算机内部的CPU与串口之间的通讯仍然采用并行的通讯方式，所以串行口的本质就是实现CPU与外围数据设备的数据格式转换（或者称为串并转换器），即当数据从外围设备输入计算机时，数据格式由位 (bit)转化为字节数据；反之，当计算机发送下行数据到外围设备时，串口又将字节数据转化为位数据。
3. 串行端口的本质功能是作为CPU和串行设备间的编码转换器。当数据从 CPU经过串行端口发送出去时，字节数据转换为串行的位。在接收数据时，串行的位被转换为字节数据。

## 4实验内容

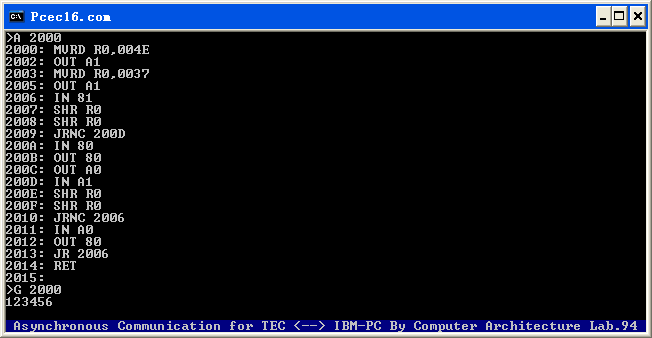
1. 为扩展I/O口选择一个地址，即将与COM2接口相连的8251的/CS与标准有I/O/CS的一排插孔中的一个相连。
2. 将COM2口与终端或另一台运行有PCEC16的PC机的串口相连。
3. 用监控程序的A命令，编写一段小程序，先初始化COM2口，再身COM2口发送一些字符，也可从COM2口接收一些字符，或实现两个串口的通信。

## 5实验步骤

1. 为扩展I/O口选择一个地址：将与COM2口相连的8251的/CS与标有I/O/CS的插孔中地址为A0-AF的一个相连。
2. 将教学机COM1口与一台PC机相连，在PC机上启动PCEC16.EXE。
3. 断开COM1与PC的串口线，将其连接到另一台PC机或同一个串口，同样启动PCEC16.EXE。
4. 用另一根串口线将COM2口和第一台PC或同一台PC的另一个串口相连。
5. 在与COM1相连的PCEC上输入程序，这是主PCEC可以输入输出，和COM2连接的是从PCEC只作输出。
6. 用A、E命令编程进行COM2口的操作。

以下代码为两个PC机串行通讯代码：

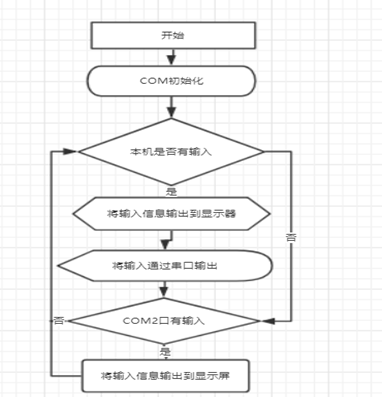
从2000H单元开始输入下面的程序



该程序完成两台教学计算机的第2个串行接口扩展操作并完成串口初始化，启动两台教学机，都运行这个程序，则两个键盘的输入同时显示在两个屏幕上，实现的是双机的双向通讯功能。

## 6思考题

画出程序的流程图



## 7实验心得

通过本次实验了解串行接口与计算机主机之间的数据传送方式。 一开始因为 没有把接线联接好，进行的并不顺利，一直不能实现两台电脑之间接口的联接， 不过当真正实现操作时，心里好高兴，我们几次的努力的结果，这次实验对于我 来说又有了另一种收获， 同学之间的协作是很重要的， 不仅仅能增加彼此的交流， 也能通过双方一起讨论解决问题，快速找出答案。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 中断实验

# 实验七：中断实验

## 1实验目的

* 加深理解计算机系统中断的工作原理及处理过程。
* 学习和掌握中断产生、响应、处理等技术。
* 掌握中断服务子程序的编写要点，进行一次硬、软件的综合调试。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

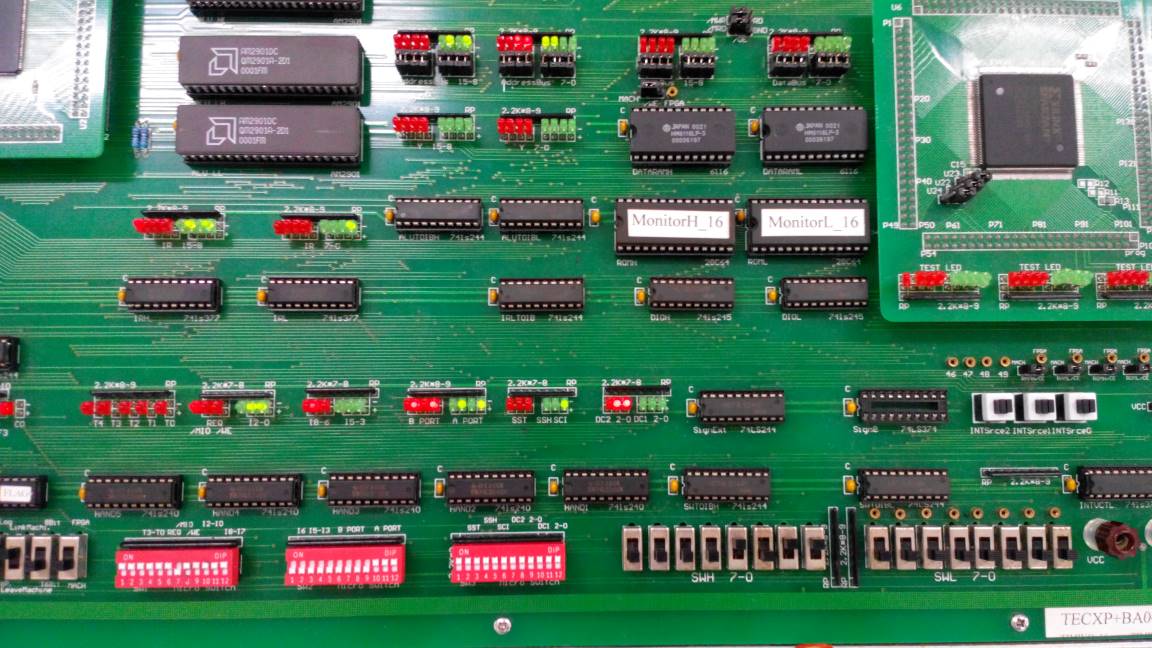
1. 要求中断隐指令中执行关中断功能，如果用户中断服务程序允许被中断，必须 在中断程序中执行EI开中断命令。
2. 教学机的中断系统共支持三级中断，由三个无锁按键确定 从右到左依次为一、二、三级中断，对应的P1、P0的编码分别是01、10、11，优先级也依次升高，这决定 了它们的中断向量，为XXX4、XXX8、XXXC。可以看到，每级中断实际可用空间只有四个字节，故这个空间一般只存放一条转移指令，而真正的用户中断服务程序则存放在转移指令所指向的地址。
3. 用户需扩展中断隐指令、开中断指令、产中断指令、中断返回指令及其节拍。

## 4实验内容

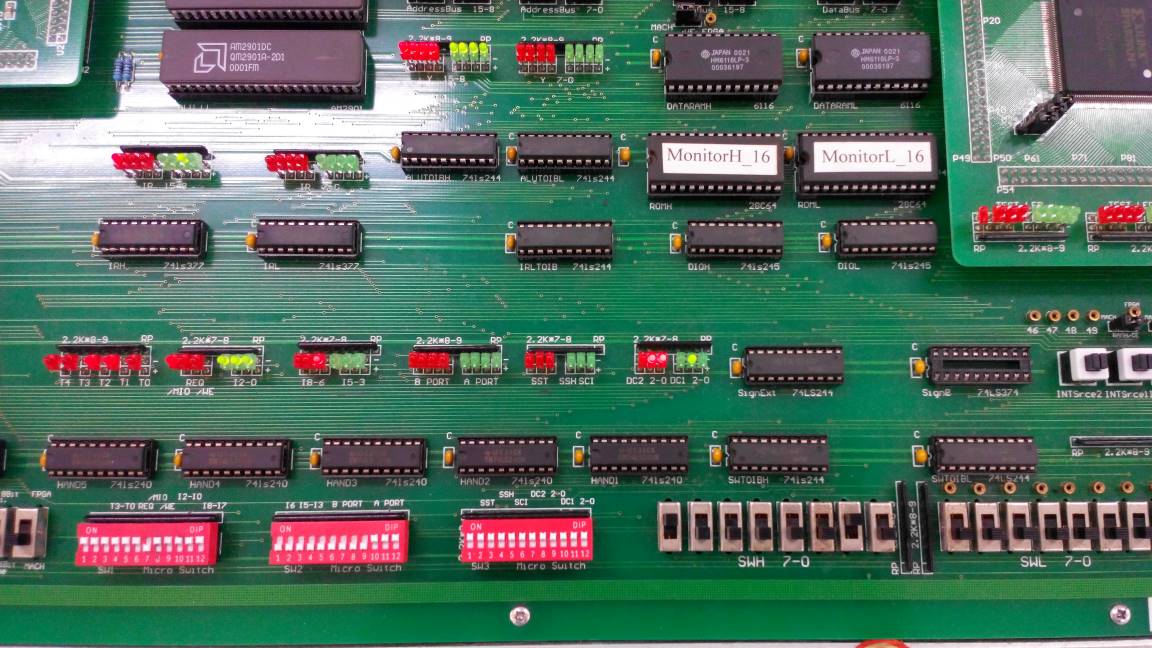
1. 扩展中断隐指令，为中断隐指令分配节拍，中断隐指令用到12个节拍，为了和一般指令相区别，应将其节拍T3设计为1.
2. 扩展开中断指令EI、关中断指令DI、中断返回指令IRET。
3. 确定中断向量地址。中断微量的高12位由开关确定为（0001001000000）。三级中断对应的中断微量为2404H、2408H、240CH。当有中断请求且被响应后，将执行存放在该中断的中断微量所指向的内存区的指令。
4. 真写中断微量表。在上述2404H、2408H、240CH地址写入三条JR转移指令，JR指令的OFFSSET是-128~127之间，但在PCEC16中输入时，用户不需要计算偏移量，直接输入要转向的绝对地址即可。
5. 编写中断服务程序。中断服务程序可以放在中断微量表之后，中断服务程序可以实现在程序正常运行时在计算机屏幕上显示与优先级相对应的不同字符。
6. 写主程序。可编写一死循环，要求先开中断。

## 5实验步骤

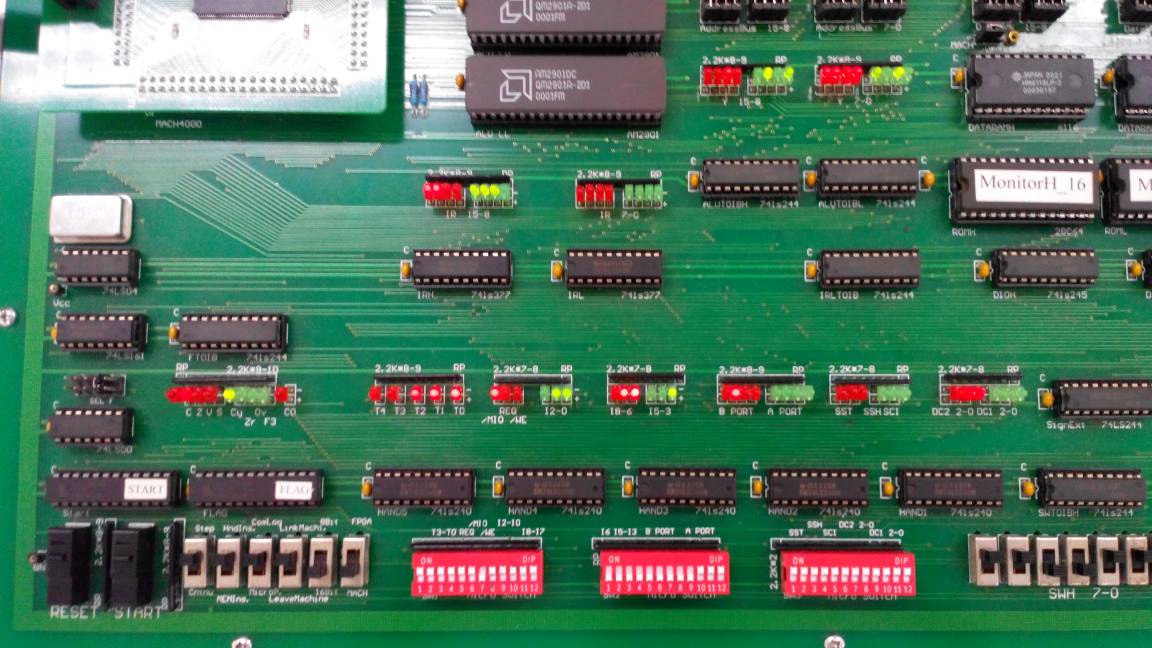
1）选择基本指令的A组指令中的SHR指令，观察其节拍流程：



2）选择基本指令的B组指令中的IN指令，观察其节拍流程：



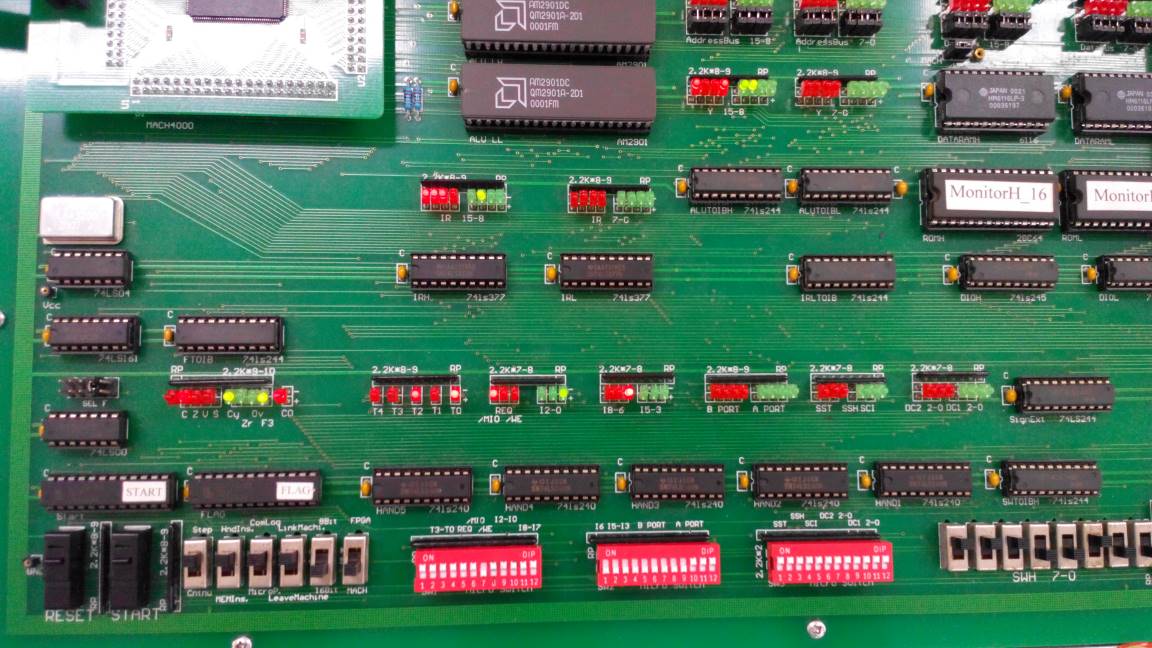
3）选择基本指令的D组指令中的CALA指令，观察其节拍流程：



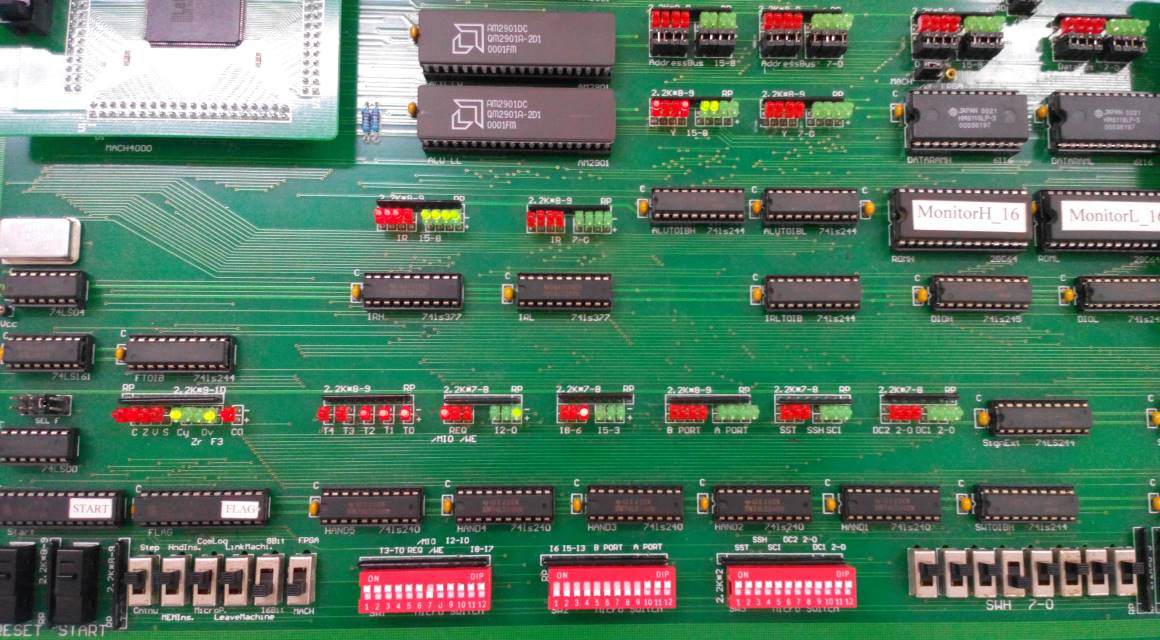
4）选择扩展指令的A组指令中的SBB指令，观察其节拍流程：



5）选择扩展指令的C组指令中的LDRA指令，观察其节拍流程：



6）选择扩展指令的D组指令中的IRET指令，观察其节拍流程：



7）填写中断向量表

* 1. 将数据开关的高12位设置成：0010 0100 0000，即选择三级中断的中断向量为2404H、2408H、240CH。
  2. 中断向量一共有16位，高12位由数据开关SWH7-0和SWL7-4决定；后四位有无锁按键决定，分别为01、10、11，所以中断向量的16位为2404 、2408 、240C。
  3. 向量表指令填写：

（2404）JR 2420

（2408）JR 2430

（240C）JR 2440

8）编写中断程序

用A、E命令从2420H单元开始输入下面和程序：（标有\*的语句表示要用E命令输入）

2420：PUSH R0

2421:PUSH R3

2422:MVRD R3,31

2424:JR 2450

2430:PUSH R0

2431:PUSH R3

2432:MVRD R3,32

2434:JR 2450

2440:PUSH R0

2441:PUSH R3

2442:MVRD R3,33

2444:JR 2450

\*2450:EI

2451:MVRD R0,0042

2453:CALA 2200

2455:MVRD R0,0049

2457:CALA 2200

2459:MVRR R0,R3

245A:CALA 2200

245C:IN 81

245E:SHR R0

245F:JRNC 245C

2460:IN 80

2461:MVRD R00045

2463:CALA 2200

2465:MVRD R0,0049

2467:CALA 2200

2469:MVRR R0,R3

246A:CALA 2200

246C:POP R3

246D:POP R0

\*246E:IRET

9）编写子程序

2200：PUSH R0

2202:SHR R0

2203:JRNC 2201

2204:POP R0

2205:OUT 80

2206:RET

10）编写主程序

\*2000:EI

2001:MVRE R0,0036

2003:CALA 2200

2005:MVRD R0,4000

2007:DEC R0

2008:JRNZ 2007

2009:JR 2001

200A:RET

11）运行主程序，等待、响应中断。

在命令行提示符状态下输入：G 2000

12）结果显示与截图：



屏幕将连续显示“6”。在程序执行过程中按下教学机右下方任意一个无锁按键。此时，教学机转向执行本级中断服务程序，在屏幕上显示BI以及按下的键对应的中断优先级。在接收键盘一个字符后，显示该字符并退出当前级的中断服务程序，恢复中断现场，接着执行断点处的程序。若在接收字符之前，又有更高一级的中断请求，则教学机转向执行高一级的中断服务程序，执行完后接着执行低级中断，然后 退出 执行主程序。需要注意的是若当前中断为高级中断，则不会响应低级中断简单的中断服务程序。

## 6实验心得

由于本次实验课前，计算机组成原理课堂上进行了关于中断机制，中断向量表的学习，因此在本次实验课上得心应手。

在学习中断时，应学会主动制造错误，如除法溢出等出发中断，然后逐步了解其响应过程，以及通过中断向量表调用内存中自己所写入的程序代码，从而加深对中断整个过程的了解。

从单片机中的学习，清楚明白到中断机制的存在为底层编程予以极大方便和灵活性，因此熟悉中断机制是我们深入学习嵌入式系统的前提。

gdut

计算机学院计算机科学与技术专业 5班 学号3116004636

姓名 黄国航 协作者\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师评定\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_实验题目 FPGA芯片实现非流水线的CPU系统(综合实验)

# 实验九：FPGA芯片实现非流水线的CPU系统(综合实验)

## 1实验目的

* 进一步熟悉教学计算机的指令格式、指令编码、寻址方式和指令功能等内容。
* 进一步熟悉教学计算机的总体组成和各个部件的功能，理解控制器部件在计算机整机中的关键作用；
* 进一步理解和指令执行步骤的划分方案；
* 进一步熟悉教学计算机的硬连线控制器各个控制命令的控制功能，学习用VHDL语言描述节拍发生器和控制信号产生部件的功能。
* 进一步理解与熟悉在TH-union教学计算机控制器中处理原有指令和扩展指令的方案，提高对控制器功能能描述的理解程序。

## 2实验设备与器材

* TEC-XP+教学实验系统和仿真终端软件PCEC。

## 3实验说明和原理

用VHDL语言来描述的CPU的源码文件，经过专用工具软件的编译和综合后，经专用烧录器，把程序下载到FPGA芯片中，就得到了能够正常运行的CPU系统。

## 4实验内容

控制器实验可以在两个层次上进行：

1. 第一个层次属于观察、验证性的实验，即通过多种方式，察看教学计算机指令的执行步骤、运行结果、各组控制信号在每一个执行步骤中的状态、指令之间的衔接等有关内容。这个层次的实验，重点在于学懂教学计算机中已有的设计结果，把实现基本指令的VHDL语言程序中的语句描述与教学机的运行结果对应清楚。
2. 第二个层次是学生进行自己的设计与实现新的扩展指令的实验，即在教学机系统已有指令的基础上，由学生自己添加若干条新的指令进行，包括定义指令格式、功能。划分指令执行步骤和确定每一步的功能，确定每一执行步骤使用的全部控制信号的状态值，使用VHDL语句把新的设计结果描述正确并添加到已有的源程序代码中去，编译、下载并调试正确，写一个包含已有指令和刚刚实现的指令的小程序，检查运行结果的正确性，若发现错误则找出原因并设法改正，直到全部正确为止。

## 5实验步骤

1. 按前述的步骤准备好实验机，连接好串口线和电源线，打开PCEC1.6EXE的仿真界面；将六个功能开关置为00X101；
2. 将TIMING GAL左方的波动开关置于“FPGA”端；
3. 确认标有“DataBus 15-8”和“DataBus 7-0”的数据总线的指示灯正文的插针断开；
4. 确认标有“AdressBus 15-8”和“AdressBus 7-0”的地址总线的指示灯下方的插针断开；
5. 将提供的带彩线的FPGA的下载线并口一端和计算机的并口连接，彩线一端按红色在左边的位置和大板上电源模块下方的一溶解度插针接好；
6. 打开实验机的电源；
7. 在PC机上打开ISE的软件；
8. 打开软件的下载界面，选择SERIAL方式，添加器件CPU.BIT，进行下载；
9. 下载完成关闭下载界面，启动PCEC界面，注意实验机不要断电。
10. 按一下“RESET”按键，再按一下“START”按键，主机上显示：

TEC-2000 CRT MONITOR

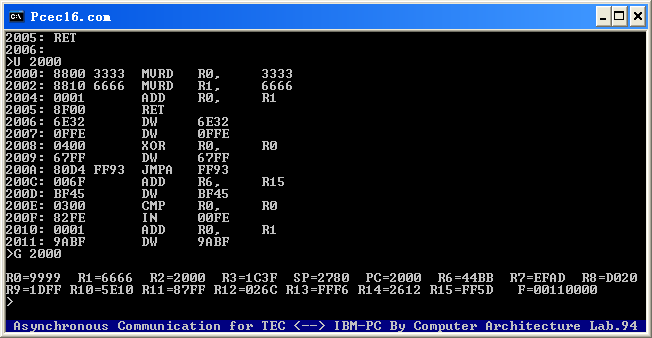
Version 1.0 April 2001

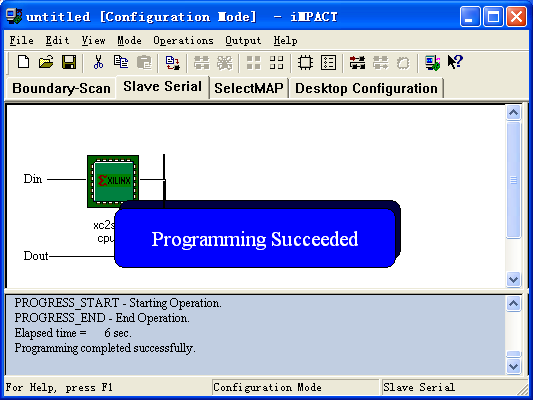
Computer Architectur Lab,Tsinghua University

Programmed by He Jia

>

1. 在FPGA构成的CPU的控制下将汇编语言程序设计的内容重新作一遍。







## 6实验心得

由本次实验可知，FPGA支持基本汇编指令集。经过基本汇编指令微码的对比可知，其实对于FPGA、8086架构、单片机、ARM或者是最近的的灵动系统CPU，它们都有各自的汇编指令集，而FPGA系统则在指令系统、使用的软件资源等方面与小规模期间构成的实验箱兼容，能够使软件资源得到充分的应用，同时提高动手和学习能力，尽管不同的CPU架构之间为知识的应用也带来了麻烦。