

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **实验名称：** | | **实验一** | **简单程序设计** | **TD 的使用** | | | |  |  |  |  |  |
| **实验时间：** | | **2016-3-22，14：00-17：30** | | **实验地点： 南一楼 804 室 74 号实验台** | | | | | | | | |
|  | |  |  |  |  | |  | | |  |  |  |
| **指导教师：** | | **张勇** |  | **专业班级：计算机科学与技术 201409** | | | | | | **班** | | |
| **学** | **号：** | **U201414813** | |  | **姓** | **名：** |  | **唐礼威** | |  |  |  |
| **同组学生：** | | **无** |  | **报告日期：** | | |  | **2016 年 3 月 23** | | **日** | | |

**原创性声明**

本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。

特此声明！

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 学生签名： |  |
|  |  |  | 日期： |  |
| 成绩评定 | |  |  |  |
|  |  |  |  |  |
|  | 实验完成质量得分（70 分） | 报告撰写质量得分（30 分） |  |  |
|  | （实验步骤清晰详细深入， | （报告规范、完整、通顺、 | 总成绩（100 分） |  |
|  |  |  |
|  | 实验记录真实完整等） | 详实等） |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

指导教师签字：

日期：

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

**目录**

|  |  |  |  |
| --- | --- | --- | --- |
| [1](#page3) | [实验目的与要求 .......................................................](#page3) | | [1](#page3) |
| [2](#page3) | [实验内容 .............................................................](#page3) | | [1](#page3) |
| [3](#page4) | [实验过程 .............................................................](#page4) | | [2](#page4) |
| [3.1](#page4) |  | [任务 1 .............................................................................................................................](#page4) | [2](#page4) |
| [3.1.1](#page4) | | [设计思想及存储单元分配.........................................................................................](#page4) | [2](#page4) |
| [3.1.2](#page4) | | [流程图.........................................................................................................................](#page4) | [2](#page4) |
| [3.1.3](#page4) | | [源程序.........................................................................................................................](#page4) | [2](#page4) |
| [3.1.4](#page4) | | [实验步骤.....................................................................................................................](#page4) | [2](#page4) |
| [3.1.5](#page4) | | [实验记录.....................................................................................................................](#page4) | [2](#page4) |
| [3.2](#page6) |  | [任务 2 .............................................................................................................................](#page6) | [4](#page6) |
| [3.2.1](#page6) | | [设计思想及存储单元分配.........................................................................................](#page6) | [4](#page6) |
| [3.2.2](#page6) | | [流程图.........................................................................................................................](#page6) | [4](#page6) |
| [3.2.3](#page6) | | [源程序.........................................................................................................................](#page6) | [4](#page6) |
| [3.2.4](#page6) | | [实验步骤.....................................................................................................................](#page6) | [4](#page6) |
| [3.2.5](#page6) | | [实验记录.....................................................................................................................](#page6) | [4](#page6) |
| [3.3](#page8) |  | [任务 3 .............................................................................................................................](#page8) | [6](#page8) |
| [3.3.1](#page8) | | [设计思想及存储单元分配.........................................................................................](#page8) | [6](#page8) |
| [3.3.2](#page8) | | [流程图.........................................................................................................................](#page8) | [6](#page8) |
| [3.3.3](#page8) | | [源程序.........................................................................................................................](#page8) | [6](#page8) |
| [3.3.4](#page8) | | [实验步骤.....................................................................................................................](#page8) | [7](#page8) |
| [3.3.5](#page9) | | [实验记录.....................................................................................................................](#page9) | [7](#page9) |
| [4](#page9) | [体会 .................................................................](#page9) | | [7](#page9) |
| [参考文献 .................................................................](#page10) | | | [8](#page10) |

0

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

**1** **实验目的与要求**

掌握汇编源程序编译工具、汇编程序、连接程序、调试工具的使用；理解数、符号、寻址方式等在计算机内的表现形式；理解指令执行与标志位之间的关系；熟悉常用的 DOS 功能调用。

**2** **实验内容**

任务 1. 《80X86 汇编语言程序设计》教材中 P31 的 1.14 题。

要求：(1) 直接在 TD 中输入指令，完成两个数的求和、求差的功能（可以省略实验报

告中的程序框图）。求和/差后的结果放在(AH)中。

1. 请事先指出执行指令后(AH)、标志位 SF、OF、CF、ZF 的内容。
2. 记录上机执行后的结果，与（2）中对应的内容比较。

(4)求差运算中，若将 A、B 视为有符号数，且 A>B, 标志位有何特点？

若将 A、B 视为无符号数，且 A>B, 标志位又有何特点？

任务 2. 《80X86 汇编语言程序设计》教材中 P45 的 2.3 题。

要求：（1）可以省略实验报告中的程序框图和源程序。

（2）分别记录执行到“MOV CX，10”和“INT 21H”之前的(BX),

(BP),(SI),(DI)各是多少。

（3）记录程序执行到退出之前数据段开始 40 个字节的内容，指出

程序运行结果是否与设想的一致。

(4)在标号 LOPA 前加上一段程序，实现新的功能：先显示提示信息“Press any key to begin!”, 然后，在按了一个键之后继续执行 LOPA 处的程序。

任务 3. 《80X86 汇编语言程序设计》教材中 P45 的 2.4 题的改写。

要求：(1) 实现的功能不变，对数据段中变量访问时所用到的寻址方式中的寄存器改

* 32 位寄存器。
  1. 内存单元中数据的访问采用变址寻址方式。

(3) 记录程序执行到退出之前数据段开始 40 个字节的内容，检查程序运行结果是否与设想的一致。

1

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

(4)在 TD 代码窗口中观察并记录机器指令代码在内存中的存放形式，并与 TD 中提供的反汇编语句及自己编写的源程序语句进行对照，也与任务 2 做对比。（相似语句记录一条即可，重点理解机器码与汇编语句的对应关系，尤其注意操作数寻址方式的形式）。

（5）观察连续存放的二进制串在反汇编成汇编语言语句时，从不同字节位置开始反汇编，结果怎样？理解 IP/EIP 指明指令起始位置的重要性

**3** **实验过程**

**3.1 任务 1**

**3.1.1** **设计思想及存储单元分配**

寄存器分配：一个 AH,运算后的结果存储在 AH 中。

**3.1.2** **流程图**

无

**3.1.3** **源程序**

无

1. **实验步骤**
   1. 使用编辑程序 EDIT.EXE 录入源程序，存盘文件名为 CUBE.ASM。

2. 使用 MASM5.0 或 6.0 汇编源文件。即 MASM CUBE；

* + 1. 观察提示信息，若出错，则用编辑程序修改错误，存盘后重新汇编，直至不再报错为止。
    2. 使用连接程序 LINK.EXE 将汇编生成的 CUBE.OBJ 文件连接成执行文件。
       - LINK CUBE；
  1. 若连接时报错，则依照错误信息修改源程序。之后重新汇编和连接，直至不再报错并生成

CUBE.EXE 文件。

* 1. 执行该程序。即在命令行提示符后输入 CUBE 后回车，观察执行现象。

1. **实验记录**

（包括实验条件、输入/输出、错误和修改等信息的记录）

1、实验环境条件：P3 1GHz，256M 内存；WINDOWS 2000 命令行窗口；EDIT.EXE

2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

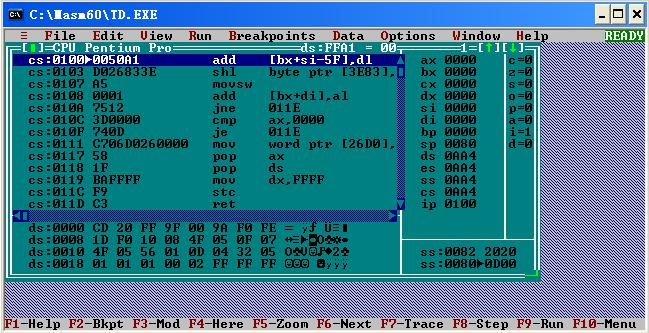
2、求和运算：

mov ah，00110011b add ah，01011010b

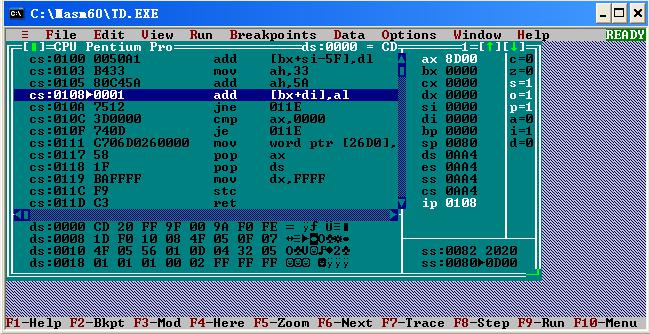
2

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

执行前：



在 add ah，01011010b 下一句设一断点，运行至断点停止执行后：

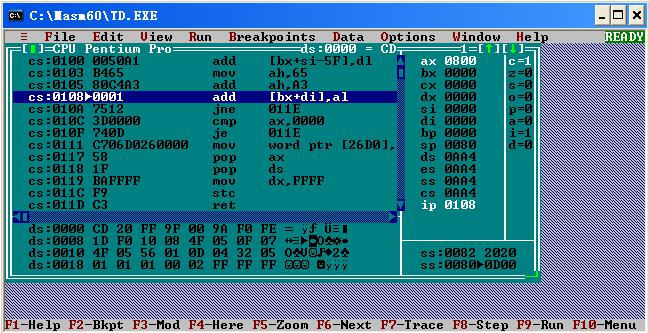


分析结果：右边标志寄存器区 o 变为 1，说明有溢出，c 为 0，说明没有进位

3、求差运算：

mov ah，01100101b add ah，10100011b

执行后：



分析结果：c 变为 1，说明产生进位

3

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

**3.2 任务 2**

**3.2.1** **设计思想及存储单元分配**

用到 4 个数据段 BUF1-BUF4，一开始用 4 个寄存器 SI、DI、BX、BP 依次存储 BUF 第一个值，到后面用循环给 BUF1-BUF4 空间赋以不同的值。

**3.2.2** **流程图**

无

**3.2.3** **源程序**

无

1. **实验步骤**
   1. 使用编辑程序 EDIT.EXE 录入源程序，存盘文件名为 CUBE.ASM。

2. 使用 MASM5.0 或 6.0 汇编源文件。即 MASM CUBE；

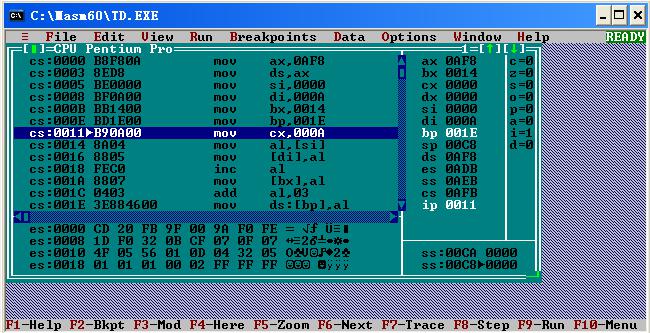
* + 1. 观察提示信息，若出错，则用编辑程序修改错误，存盘后重新汇编，直至不再报错为止。
    2. 使用连接程序 LINK.EXE 将汇编生成的 CUBE.OBJ 文件连接成执行文件。
       - LINK CUBE；
  1. 若连接时报错，则依照错误信息修改源程序。之后重新汇编和连接，直至不再报错并生成

CUBE.EXE 文件。

* 1. 执行该程序。即在命令行提示符后输入 CUBE 后回车，观察执行现象。

1. **实验记录**

1、按 f8 进行单步调试，直到光标移到“MOV CX，10”



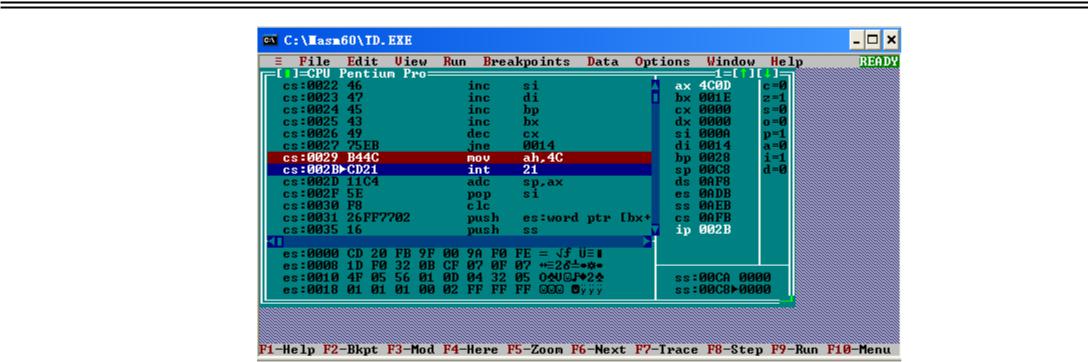
Bx 变为 0014，bp 变为 001E，si 为 0000，di 变为 000A

在 mov ah，4c 一行按下 f2 设置断点，按下 f9，将运行至该行，再按下 f8，执行至“INT

21H”之前

4

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**



Bx 变为 001E，bp 变为 0028，si 为 000A，di 变为 0014

2、记录程序执行到退出之前数据段开始 40 个字节的内容：



依次输出 0、1、2、3、4、5、6、7、8、9 0、1、2、3、4、5、6、7、8、9 1、2、3、4、5、6、7、8、9、10 4、5、6、7、8、9、10、11、12、13

可见与预期结果相符。

3、在代码中改写：在数据段中加入

BUF5 DB 0AH,0DH,’Press any key to begin!$’

再在 LOPA 前加上

LEA DX,BUF5 MOV AH,9 INT 21H

实现 9 号功能调用，在屏幕输出字符串在 LOPA 前加上

MOV AH,1

INT 21H

实现 1 号功能调用，等待从键盘输入一个字符，一旦输入，继续程序执行，截图：



5

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

**3.3 任务 3**

**3.3.1** **设计思想及存储单元分配**

AL 每循环一次+1，再赋给 BUF[ESI]相应的值，寄存器 CX 来控制循环次数。

**3.3.2** **流程图**

无

**3.3.3** **源程序**

|  |  |  |  |
| --- | --- | --- | --- |
| .386 |  |  |  |
| STACK | SEGMENT | | USE16 STACK |
|  | DB | 200 | DUP(0) |
| STACK | ENDS | |  |
| DATA | SEGMENT | | USE16 |
| BUF1 | DB | 0,1,2,3,4,5,6,7,8,9 | |
| BUF2 | DB | 10 | DUP(0) |
| BUF3 | DB | 10 | DUP(0) |
| BUF4 | DB | 10 | DUP(0) |
| DATA | ENDS |  |  |
| CODE | SEGMENT | | USE16 |
|  | ASSUME CS:CODE,DS:DATA,SS:STACK | | |
| START:MOV | | AX,DATA | |
|  | MOV | DS,AX | |
|  | MOV | ESI,0 | |
|  | MOV | CX,10 | |
| LOPA: MOV | | AL,BUF1[ESI] | |
|  | MOV | BUF2[ESI],AL | |
|  | INC | AL |  |
|  | MOV | BUF3[ESI],AL | |
|  | ADD | AL,3 | |
|  | MOV | BUF4[ESI],AL | |
|  | INC | ESI |  |
|  | DEC | CX |  |
|  | JNZ | LOPA | |
|  | MOV | AH,4CH | |
|  | INT | 21H |  |
| CODE | ENDS |  |  |
|  | END | START | |

1. **实验步骤**
   1. 使用编辑程序 EDIT.EXE 录入源程序，存盘文件名为 CUBE.ASM。

|  |  |
| --- | --- |
| 2. | 使用 MASM5.0 或 6.0 汇编源文件。即 MASM CUBE； |
| 3. | 观察提示信息，若出错，则用编辑程序修改错误，存盘后重新汇编，直至不再报错为止。 |

6

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

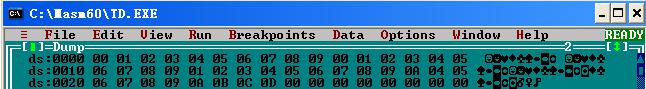
* + 1. 使用连接程序 LINK.EXE 将汇编生成的 CUBE.OBJ 文件连接成执行文件。
       - LINK CUBE；
  1. 若连接时报错，则依照错误信息修改源程序。之后重新汇编和连接，直至不再报错并生成

CUBE.EXE 文件。

* 1. 执行该程序。即在命令行提示符后输入 CUBE 后回车，观察执行现象。

1. **实验记录**

设断点在 int 21 那行，打开 dump 观察数据区的值如下：



对比任务 2 中结果可知符合。

**4** **体会**

这是汇编的第一次上机，总体来说比较顺利，因为在上机之前任务 2 和任务 3 的代码都早已写好，上机主要是熟悉编程和调试环境，学会如何将一个.asm 文件编译成.obj 文件，再将 obj 文件链接成一个 exe 文件，熟悉了 td 的环境及操作过程，对 td 的调试过程有一定的了解，主要是 go to 到指定的一行，和 f2 设置断点，以及单步执行，熟悉了 td 界面的各个区域及里面包含的内容，能够通过观察 td 里面的内容与代码联系起来分析。总体来说，这次实验收获颇丰。

7

**汇** **编** **语** **言** **程** **序** **设** **计** **实** **验** **报** **告**

**参考文献**

1. 王元珍等.80x86 汇编语言程序设计.版本(第 1 版)
2. 王晓虹等.汇编语言程序设计教程.版本(第 1 版)

8