

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 编程基础**

**实验时间： 2018-3-30，18:30-21:50 实验地点： 南一楼804室xx号实验台**

**指导教师： xx 专业班级：计算机科学与技术201607班**

**学 号： U201614700 姓 名：**

**同组学生： 无 报告日期： 2018年3月30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）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

                    日期：

**目录**

[1 实验目的与要求 1](#_Toc510112451)

[2 实验内容 1](#_Toc510112452)

[3 实验过程 1](#_Toc510112453)

[任务1 1](#_Toc510112454)

[3.1.1 设计思想及存储单元分配 1](#_Toc510112455)

[1. 流程图 1](#_Toc510112456)

[i. 源程序 2](#_Toc510112457)

[ii. 实验步骤 2](#_Toc510112458)

[iii. 实验记录 2](#_Toc510112459)

[2. 体会 3](#_Toc510112460)

[参考文献 4](#_Toc510112461)

# 实验目的与要求

1. 掌握汇编源程序编辑工具、汇编程序、连接程序、调试工具TD的使用；
2. 理解数、符号、寻址方式等在计算机内的表现形式；
3. 理解指令执行与标志位改变之间的关系；
4. 熟悉常用的DOS功能调用；
5. 熟悉分支、循环程序的结构及控制方法，掌握分支、循环程序的调试方法；
6. 加深对转移指令及一些常用的汇编指令的理解。

# 实验内容

任务1. 《80X86汇编语言程序设计》教材中 P31的 1.14题。

任务2. 《80X86汇编语言程序设计》教材中 P45的 2.3题。

任务3. 《80X86汇编语言程序设计》教材中 P45的 2.4题的改写。

# 实验过程

## 任务1

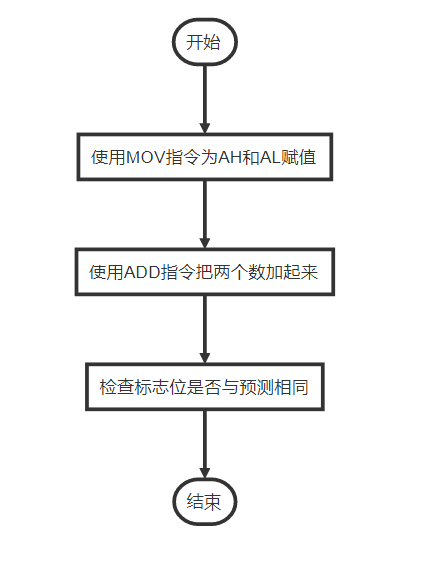
任务详细内容：已知二进制数x1和x2的值，求x1和x2的补码的和，并指出符号，判断是否发生了溢出和进位。

### 设计思想

设计思路：任务一的内容是对两个数进行加减运算，指出结果的符号，可以使用AX寄存器的高低位存放两个数的值，运算即可。

寄存器分配：AH存放x1，AL存放x2。

### 流程图



### 实验步骤

1. 打开dosbox，将masm的等文件目录挂载为c，输入c:切换到c盘。
2. 输入td打开td的窗口。
3. 直接编写要执行的汇编代码：首先把数据x1和x2的补码使用MOV指令到AX寄存器的高位和低位。
4. 使用ADD AH, AL命令计算AL加AH并将结果存放在AH中。
5. 观察此时的标志位SF, OF, CF, ZF，记录下来和自己实验前预测的对比并分析原因。

### 实验记录

实验环境条件：i5 6200U，2.8GHz，8GB内存；dosbox命令行窗口；visual studio 2017；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

首先是实验使用的数据x1，x2，它们的补码表示，如表1-1所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X1 | X2 | 和 | SF | OF | CF | ZF |
| 33H | 5AH | 8DH | 1 | 1 | 0 | 0 |
| D7H | A3H | 7AH | 0 | 1 | 1 | 0 |
| 65H | A3H | 08H | 0 | 0 | 1 | 0 |

接下来测试在CPU上执行汇编代码的真实情况，

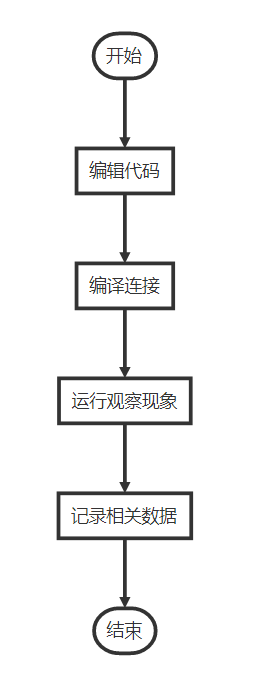
## 任务二

任务二是运行课本上45页题2.3的汇编代码，观察现象。

### 设计思想

由于课本上已经给出了代码，我只需要将代码写入文本文件保存并使用ml编译链接即可运行，使用td打开程序进行运行。

### 流程图



### 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:DATA

START:

MOV AX, DATA

MOV DS, AX

; code after here

MOV SI, OFFSET BUF1

MOV DI, OFFSET BUF2

MOV BX, OFFSET BUF3

MOV CX, OFFSET BUF4

MOV CX, 10; count number

LOPA: MOV AL, [SI]; let ax = BUF1[0]

MOV [DI], AL

INC AL; add 1

MOV [BX], AL

ADD AL, 3; add 3

MOV DS:[BP], AL

INC SI

INC DI

INC BP

INC BX

DEC CX

JNZ LOPA; check jmp or not

;stop here

MOV AH, 4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 编写代码：使用visual studio编写汇编代码并保存为asm文件。
2. 打开dosbox，挂载d:\github\asm\asm为c盘。
3. 输入c:切换到c盘，输入ml mission2.asm完成编译和连接。
4. 输入td mission2运行程序。
5. 单步运行程序，分别记录执行到“MOV CX，10”和“INT 21H”之前的(BX), (BP),(SI),(DI)各是多少，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。
6. 修改程序：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。

### 实验记录

使用visual studio编写代码并保存到d:\github\asm\asm文件夹，运行dosbox执行编译连接后运行程序。记录如下

## 实验三

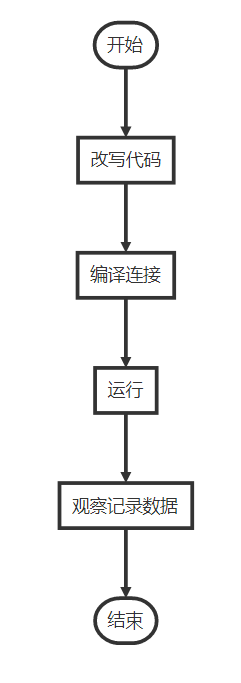
对教材45页2.4题代码的改写，实现使用32位寄存器变址寻址方式运行，记录数据区前40个字节的内容。

### 设计思想

本任务为使用32位寄存器变址寻址运行程序，记录数据段前40个字节的内容，所以只需要将4个寻址使用的寄存器替换为32位的即可。

变址寻址的实现：使用EBX存放BUF1的地址，BUF2的地址位EBX+10，BUF3地址为EBX+20，BUF4地址为EBX+30，就可以改写为变址寻址。

### 流程图



### 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:DATA

START:

MOV AX, DATA

MOV DS, AX

; code after here

MOV ESI, OFFSET BUF1

MOV CX, 10

LOPA: MOV AL, [ESI]

MOV [ESI + 10], AL; according memory order

INC AL

MOV [ESI + 20], AL

ADD AL, 3

MOV DS:[ESI + 30], AL

INC ESI

DEC CX

JNZ LOPA

;stop here

MOV AH, 4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 实验步骤和任务二相同。

### 实验记录

记录主要是记录代码运行前后数据区前40个字节的内容。

# 体会

# 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)