

深圳大学实验报告

课程名称 计算机系统(1)

项目名称 实验 3: 求成绩等级

学 院 数学科学学院

专 业 数计

指导教师 李庚辉

报 告 人 詹耿羽 学号 2023193026

实验时间 2024/5/21

教务处制

一、实验目的与要求

1. 实验目的：

解决以下问题：

- 1) 程序起始地址 x3000
- 2) 在 x4000 处开始存放 16 个学生成绩，成绩为 0~100 的正整数
- 3) 编写程序对 16 个学生的成绩进行排序，从低至高存放在起始地址 x5000 处

4) 计算学生获得成绩等级为 A 的数量 存放在 x5100，获得 B 等级的数量存放在 x5101 处。

成绩等级计算方法：如果一个学生的成绩在 85 分以上同时排名前 25% 则获得 A, 如果一个学生不能获得 A, 他的成绩在 75 分以上同时排名前 50%, 则可获得 B。其它成绩则获得 C。

2. 实验要求

- (1) 熟悉和理解 LC-3 的汇编语言指令格式。
- (2) 掌握在 LC-3 仿真平台下汇编语言代码的设计输入和调试过程。
- (3) 掌握利用汇编语言解决问题的思路，加深对底层硬件的理解。

二、实验内容

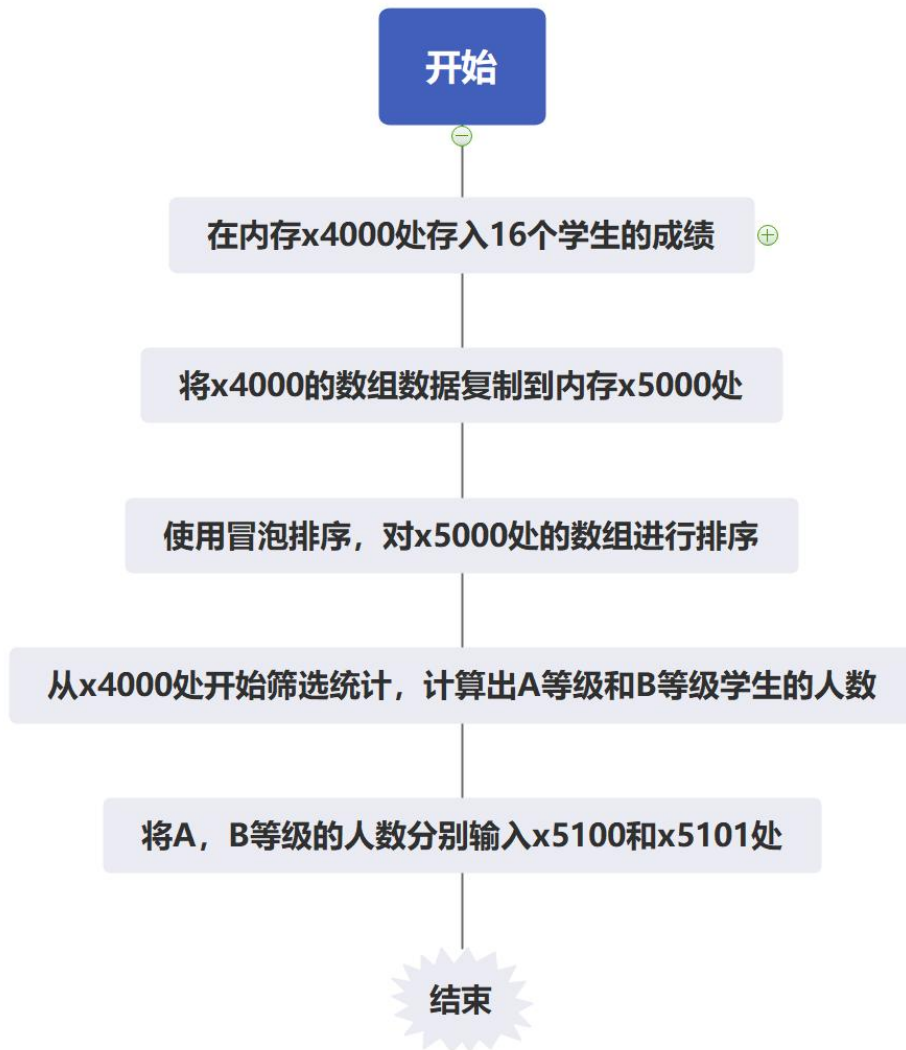
- (1) 分析和理解试验指定的需解决问题。
- (2) 利用 LC-3 的汇编语言代码设计实现相关程序。
- (3) 通过 LC-3 仿真器调试和运行相关程序并得到正确的结果

三、实验步骤与过程

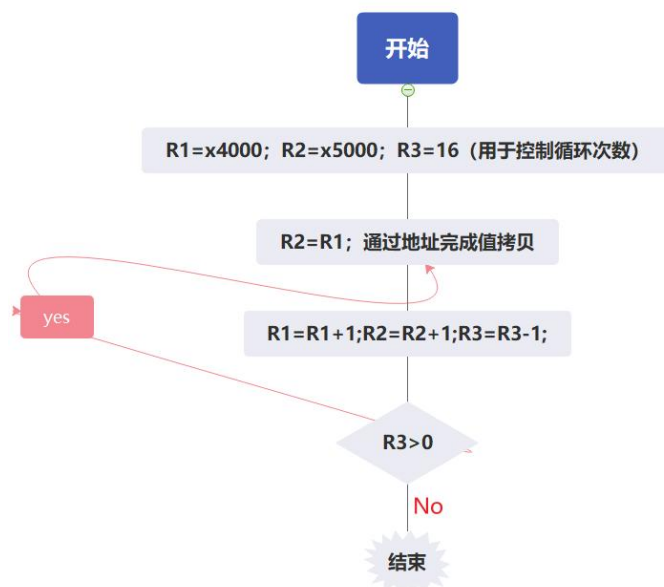
1. 实验思路：

首先要在内存 x4000 处输入 16 个分数的原始数据，并通过一轮循环，将其拷贝到 x5000 处。

在新地址使用冒泡排序降序排序，得出排名。最后对 x5000 处的数组进行筛选，即可统计出学生中得 A 级和 B 级的人数。思维导图如下：



1) 对 16 个数据进行拷贝，画出思路图：



根据思路图，可以写出汇编代码：

```

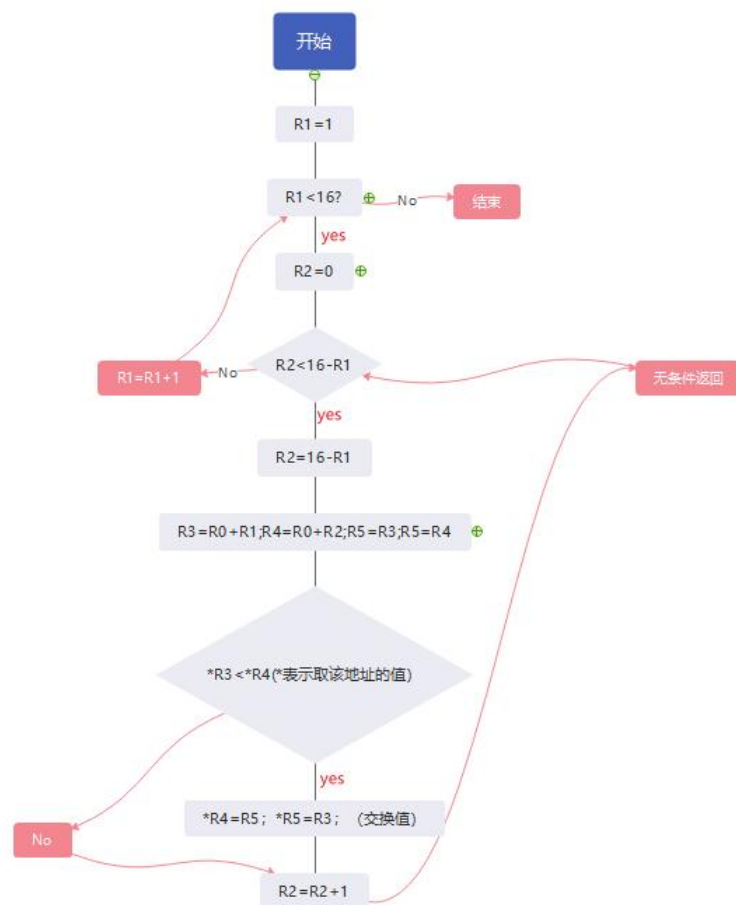
LC3Edit - Untitled *
File Edit Translate Help

[Icons] [B] [X] [asm]

        .ORIG    x3000    ;起始地址
;
;完成对原始数据的拷贝
;
        LD      R1      DATA    ;R1指向源地址
        LD      R2      RESULT   ;R2指向目的地址
        AND     R0      R0      #0    ;R0长期指向目的地址
        ADD     R0      R0      R2
        AND     R3      R3      #0
        ADD     R3      R3      #1
        ADD     R3      R3      #15    ;R3 = 16, 循环次数
;
;循环，do-loop结构，将输入数据拷贝到目的地址
;
INIT_LOOP    LDR      R4      R1      #0    ;R4 = *R1
              STR     R4      R2      #0    ;*R2 = R4
;
              ADD     R1      R1      #1
              ADD     R2      R2      #1
              ADD     R3      R3      #-1
              BRp     INIT_LOOP

```

2) 进行冒泡排序，对 16 个数值进行排列，画出思路图：



为了更好的解释冒泡排序，我使用 C 语言打出了相关代码：

```
1  #include<stdio.h>
2  void Bubble_sort(int arr[], int size)
3  {
4      int j,i,tem;
5      for (i = 0; i < size-1;i ++)//size-1是因为不用与自己比较，所以比的数就少一个
6      {
7          int count = 0;
8          for (j = 0; j < size-1 - i; j++)    //size-1-i是因为每一趟就会少一个数比较
9          {
10             if (arr[j] > arr[j+1])//这是升序排法，前一个数和后一个数比较，如果前数大则与后一个数换位置
11             {
12                 tem = arr[j];
13                 arr[j] = arr[j+1];
14                 arr[j+1] = tem;
15                 count = 1;
16             }
17         }
18         if (count == 0)    //如果某一趟没有交换位置，则说明已经排好序，直接退出循环
19             break;
20     }
21 }
22
23 }
```

根据解释，可以写出汇编代码：

;冒泡排序

;

AND R1 R1 #0

ADD R1 R1 #1 ;i = 1

BRnzp MIDDLE1

LOOP1 AND R2 R2 #0 ;j = 0

BRnzp MIDDLE2 ;

LOOP2 ADD R3 R0 R2

ADD R4 R3 #1

LDR R5 R3 #0 ;R5 <- *R3

```
LDR R6, R4, #0 ;R6 <- *R4
```

```
NOT    R7, R6
```

```
ADD     R7, R7, #1
```

```
ADD     R7, R7, R5
```

```
BRzp    END2
```

```
;
```

```
;exchange arr[j] and arr[j + 1]
```

```
;
```

```
STR R5, R4, #0
```

```
STR R6, R3, #0
```

```
END2    ADD     R2, R2, #1
```

```
MIDDLE2    ADD     R3, R1, R2
```

```
ADD     R3, R3, #-16
```

```
BRn LOOP2
```

```
ADD     R1, R1, #1
```

```
MIDDLE1    ADD     R3, R1, #-16
```

```
BRn LOOP1
```

```
;
```

;排序结束

;以下开始筛选等级

;循环依旧采用 jump-to-middle 结构

;

AND R1 R1 #0 ;i = 0

;

;初始化 R6 = -85 · R7 = -75

;

LD R6 A_GRADE ;

LD R7 B_GRADE ;

NOT R6 R6

ADD R6 R6 #1

NOT R7 R7

ADD R7 R7 #1

;

AND R2 R2 #0 ;cntA = 0 · A 的人数

AND R3 R3 #0 ;cntB = 0 · B 的人数

;

BRnzp MIDDLE3

LOOP3 ADD R4 R1 #-4

BRzp IS_B

;

ADD R4 R0 R1 ;R4 = arr[i]

LDR R4 R4 #0 ;

ADD R5 R4 R6 ;if

BRn IS_B

ADD R2 R2 #1 ;cntA++

BRnzp NEXT

;

IS_B ADD R4 R0 R1 ;R4 = arr[i]

LDR R4 R4 #0 ;

ADD R5 R4 R7 ;if

BRn NEXT

ADD R3 R3 #1 ;cntB++

;

NEXT ADD R1 R1 #1

MIDDLE3 ADD R4 R1 #-8

BRn LOOP3

;

;最后将结果保存到指定位置

;

LD R1 A_CNT

STR R2 R1 #0


```
LD R1 B_CNT
```

```
STR R3 R1 #0
```

```
;
```

```
HALT
```

```
;
```

3) 最后，只需要把相关的值填充入地址中即可：

```
;
DATA .FILL x4000
RESULT .FILL x5000
A_CNT .FILL x5100
B_CNT .FILL x5101
A_GRADE .FILL x0055
B_GRADE .FILL x004B
.END
```

2. 实验过程：

1) 将代码输入 LC3 编译器：

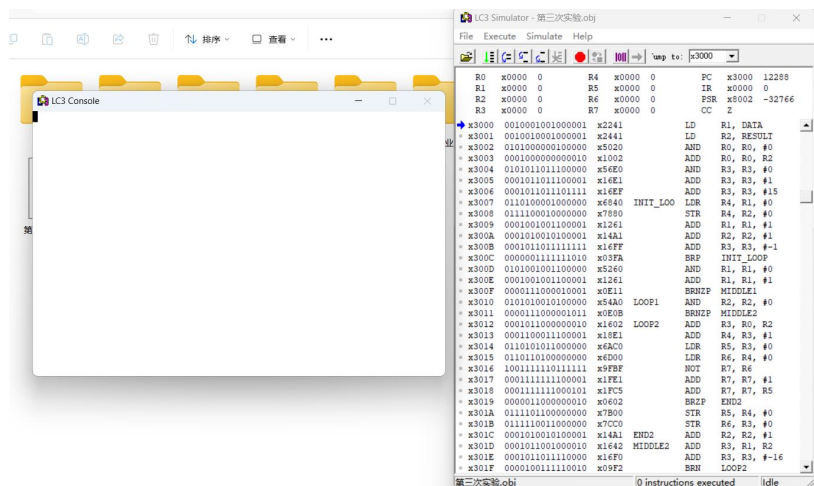
```

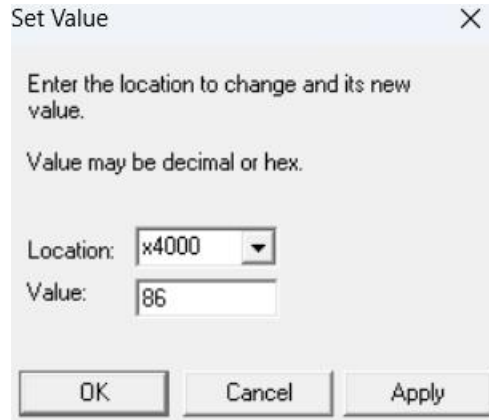
LC3Edit - Untitled *
File Edit Translate Help

;循环依旧采用 jump-to-middle 结构
;
;          AND    R1    R1    #0    ;i = 0
;
;初始化R6 = -85, R7 = -75
;
;          LD      R6    A_GRADE ;
;          LD      R7    B_GRADE ;
;          NOT     R6
;          ADD     R6    R6    #1
;          NOT     R7
;          ADD     R7    R7    #1
;
;          AND     R2    R2    #0    ;cntA = 0, A的人数
;          AND     R3    R3    #0    ;cntB = 0, B的人数
;
;          BRnzp   MIDDLE3
;
;LOOP3     ADD     R4    R1    #-4
;          BRzp    IS_B
;
;          ADD     R4    R0    R1    ;R4 = arr[i]
;          LDR     R4    R4    #0    ;
;          ADD     R5    R4    R6    ;if
;          BRn     IS_B
;          ADD     R2    R2    #1    ;cntA++
;          BRnzp   NEXT
;
;IS_B      ADD     R4    R0    R1    ;R4 = arr[i]
;          LDR     R4    R4    #0    ;
;          ADD     R5    R4    R7    ;if
;          BRn     NEXT

```

2) 编译代码，将生成的 obj 文件放入计算器中，将数据填入 x4000 中：





3) 最后实验运行结束，即可查看结果：

比如我输入 86, 50, 50, 50, 50, 75, 16, 20, 32, 58, 97, 54, 24, 56, 78, 95;

结果如下：

| | | | |
|---------|------------------|-------|------|
| ▪ x50FD | 0000000000000000 | x0000 | NOP |
| ▪ x50FE | 0000000000000000 | x0000 | NOP |
| ▪ x50FF | 0000000000000000 | x0000 | NOP |
| ▪ x5100 | 0000000000000011 | x0003 | NOP |
| ▪ x5101 | 0000000000000010 | x0002 | NOP |
| ▪ x5102 | 0000000000000000 | x0000 | NOP |
| ▪ x5103 | 0000000000000000 | x0000 | NOP |
| ▪ x5104 | 0000000000000000 | x0000 | NOP |
| ▪ x5105 | 0000000000000000 | x0000 | NOP |
| ▪ x5106 | 0000000000000000 | x0000 | NOP |
| ▪ x5107 | 0000000000000000 | x0000 | NOP |
| ▪ x5108 | 0000000000000000 | x0000 | NOP |
| ▪ x5109 | 0000000000000000 | x0000 | NOP |
| ▪ x510A | 0000000000000000 | x0000 | NOP |
| ▪ x510B | 0000000000000000 | x0000 | NOP |
| ---- | ----- | ---- | ---- |

四、实验结论或体会

1.实验结论或体会：

通过这次的实验，我对汇编语言有了更深的认识。虽然在实验过程中遇到不少困难，最终都成功解决。如判断时条件错误，忘记将因用于比较而变成负数的值转回原值等。最后在一次次 debug 和测试中都得到解决。在对汇编语言进行编程时，如果代码较长，理解起各个指令的含义变得困难，此时就需要借助注释来进行提示，注释的作用在本实验中得到了很大体现。此外，我也学会了将高级语言中排序算法应用到汇编语言的能力，这大大降低了在汇编语言中实现相应操作的难度。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。