

Lab3 Report

PB21111726 杨晓晨

实验目的 读取一段连续的字符串, 寻找其中长度最大的连续子串, 其中连续子串是指字母均相同的连续字符串

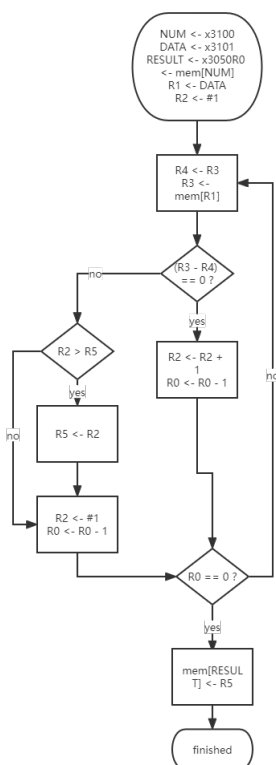
实验原理 给定了字符串的长度, 按长度对字符串遍历, 将前一个字符的 *ASCII* 码与新读取的字符的 *ASCII* 码比较 (即做减法), 如果

1. 相同则当前连续子串长度加 1,
 2. 不同则当前连续子串长度变成 1, 并将前一个记录下来的连续子串长度与已经记录的最大连续子串长度相比较, 如果
 - a. 小于等于则最大连续子串长度不变,
 - b. 大于则最大子串长度更新
- 依次遍历完字符串即可得出最大子串长度.

实验步骤 用寄存器存储实验原理中提到的数据

1. *R0* 存储字符串的长度 *NUM*
2. *R1* 存储字符串存储区域的地址 *DATA*
3. *R2* 存储当前连续子串的长度
4. *R3* 存储字符串中地址 *R1* 存储内容 (*ASCII* 码)
5. *R4* 存储字符串中地址 *R1 - 1* 存储内容 (*ASCII* 码)
6. *R5* 存储最大连续子串长度

则实验的流程图如下:



分为 3 个主要部分:

1. 判断连续子串: 代码如下

```

loop    ADD R4, R3, #0; R4 is pre-data
        NOT R4, R4;
        ADD R4, R4, #1; R4 <- (-R4)
        ADD R1, R1, #1; R1 <- R1 + 1
        LDR R3, R1, #0; R3 <- mem[R1]
        ADD R4, R4, R3; R4 <- R4 + R3
        BRZ add1; predata == data
        BRNP store; predata != data
  
```

2. 存储最大连续子串: 代码如下

```

store   NOT R2, R2;
        ADD R2, R2, #1;
        ADD R6, R2, R5; R5 is max{R2}, use R6 to judge if R2 > R5
        BRN store1; R2 > R5, store R2 in R5
        BRZP minus1
  
```

3. 判断字符串是否遍历完: 代码如下

```

minus   ADD R0, R0, #-1; R0 <- R0 - 1
        BRP loop

minus1   AND R2, R2, #0;
        ADD R2, R2, #1;
        ADD R0, R0, #-1; R0 <- R0 - 1
        BRP loop
  
```

注意当连续子串继续时可以直接对 $R0$ 减 1, 当开始新的连续子串时对 $R2$ 清零置一。

实验结果 对 8 个测试数据, 测试的结果如下:

汇编评测

8 / 8 个通过测试用例

- 平均指令数: 140.25
- 通过 6:aabbbc:3, 指令数: 93, 输出: 3
- 通过 5:ZZZZ:4, 指令数: 74, 输出: 4
- 通过 6:aabaaa:3, 指令数: 97, 输出: 3
- 通过 8:bbbfffa:4, 指令数: 115, 输出: 4
- 通过 10:achdhhdhd:3, 指令数: 167, 输出: 3
- 通过 14:abcdedddfgghhh:4, 指令数: 223, 输出: 4
- 通过 10:aabccddddd:4, 指令数: 149, 输出: 4
- 通过 15:aaabbbccccdffff:4, 指令数: 204, 输出: 4

测试结果均正确，可部分验证程序的正确性