

计算机系统概论 A 实验报告



实验题目：_____实验三_____

学生姓名：_____牛庆源_____

学生学号：_____PB21111733_____

完成日期：_____2022. 12. 13_____

【实验题目】

给定字符串长度 N（存放在 x3100）和字符串 S（起始地址为 x3101），求得最长的连续字符组成的子串的长度，存放在 x3050。

【实验目的】

1. 掌握汇编语言各种指令的写法，包括操作指令 ADD 和 AND，地址搬移指令 LD 和 ST 以及控制指令。
2. 熟练使用 BR 指令。

【实验原理】

1. R0 存 N(位置为 x3100), R1 是一个指针, 存 S(初始位置为 x3101), R2 存当前 R1 指向的字符。
2. R3 存 R1 指向的下一个字符。
3. R7 存最长子串的长度（最后放在 x3050 中），R6 存当前相同子串的长度，R5 为临时寄存器，存过程量，比如 R6 的补码，或者是 R7-R6 的结果。
4. 先进行初始化，将 R0, R1, R2 均初始化。
5. 字符串长度-1，如果减到 0，则跳转到输出步骤。
6. 取下一个字符，与前一个字符比较，相同则当前子串长度+1，与最长子串长度比较，如果大于它，则更新最长子串长度。
7. 将 R2 赋值为当前的 R3，回到第 5 步。
8. 输出步骤为将最长子串长度加一（由于是从 0 开始计算的），再存入要求的地址即可。

【实验步骤】

1. 初始地址。
2. 初始化 R0 和 R1 到指定地址。
3. 取 R2 为 R1 指向的字符。
4. 字符串长度-1，如果等于 0 则已经到达字符串最后，跳转到输出。
5. 移动指针位置指向下一个字符，用 R3 获取下一个字符。
6. 对 R2 取反，再与 R3 相加，如果 R2 和 R3 相同，则结果为 `xffff`，这个结果再取反为 `x0000`，即如果相同则结果为 `x0000`。
7. 如果相同则将当前子串长度 $R6+1$ ，取其负数的补码存在 R5 中，再将 $R7+R5$ 即 $R7-R6$ 存在 R5 中。若 R5 为非正，则需要更新 R7 为 R6 现在的数字，再回到步骤 4；否则不需要，直接更新 R2 内容为 R3，再回到步骤 4。
8. 如果步骤 6 结果为不相同，则将当前子串长度清 0，回到步骤 4。
9. 输出步：由于 R7 和 R6 从 0 开始计，所以将 $R7+1$ ，再输出在指定的位置。

【实验结果】

1. 遇到的最多的 bug 是控制指令取 nz 还是 n 或者是 z，进行调试后即消除 bug
2. 对于每一步的跳转都必须有清楚的意识，不然很容易弄混。

1. .asm 文件截图如下：

1		.ORIG	x3000	
2		LDI	R0, NUM	;R0初始化为N
3		LD	R1, DATA	;R1为指针，初始化为DATA首地址
4		LDR	R2, R1, #0	;R2为R1指向的字符
5				
6	TEST	ADD	R0, R0, #-1	;每次检测char之后，N--
7		BRz	OUTPUT	;如果减到0则跳转到输出
8				
9	GETCHAR	ADD	R1, R1, #1	;移动指针位置
10		LDR	R3, R1, #0	;获取下一个字符
11		NOT	R2, R2	
12		ADD	R2, R2, R3	;如果R2和R3相同，则R2结果为xffff
13		NOT	R2, R2	;如果R2和R3相同，则R2结果为x0000
14		BRnp	FALSE	;如果不相同
15				
16		ADD	R6, R6, #1	;存储当前连续相同字符长度结果的R6++
17		NOT	R5, R6	
18		ADD	R5, R5, #1	;得到-R6的补码存在R5中
19		ADD	R5, R7, R5	;R7-R6存在R5中
20		BRnz	UPDATE	;如果R7<R6
21		BRnzp	MOVE	;更新当前字符和检查是否到字符串尾部
22				
23	FALSE	AND	R6, R6, #0	;否则R6清零
24		BRnzp	MOVE	;更新当前字符和检查是否到字符串尾部
25				
26	UPDATE	ADD	R7, R6, #0	;将当前R7更新为最大的长度
27		BRnzp	MOVE	;更新当前字符和检查是否到字符串尾部
28				
29	MOVE	LDR	R2, R1, #0	;R2赋值为当前指针所指的字符（R3）
30		BRnzp	TEST	;检查是否到字符串尾部
31				
32	OUTPUT	ADD	R7, R7, #1	;最终结果+1（因为从0开始计数的）
33		STI	R7, RESULT	;存在指定位置
34		TRAP	x25	;HALT

36	RESULT	.FILL	x3050
37	NUM	.FILL	x3100
38	DATA	.FILL	x3101
39		.END	

2. 运行测试样例的结果如下:

[illegible]