

计算机系统概论 A 实验报告



实验题目：_____ 实验一

学生姓名：_____ 牛庆源

学生学号：_____ PB21111733

完成日期：_____ 2022. 11. 5

【实验题目】

起始地址为 x3000，将存放在 x3100 的 A（数值从 0x0001 到 0x7fff）的低 B 位（B 存在 x3101 中）中的 1 的数目输出在地址 x3102 中。

【实验目的】

1. 掌握机器码各种指令的写法，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. 对 PC 相对寻址模式产生认识，熟练使用控制指令写条件或者是循环语句。

【实验原理】

1. R0 存 A（位置为 x3100），R1 存 B（位置为 x3101），R2 存结果（位置为 x3102），R3 用于判断 A 的被判断位数是否为 1（初始被置为 1），R4 为 R3 与 A 进行 AND 之后的结果（如果非 0，则表示 A 的当前被判断位为 1）。
2. 运行过程中每判断过一位，B 减 1，R3 的 1 左移一位，循环判断直到 B 为 0 为止。
3. 例如如 A 为 10110，B 为 3 时，首先让 R3 存 1，此时 R3 为 00001。
 - (1) R3 和 A 与之后的结果为 00000，存入 R4，R4 为 0，则存放结果的 R2 不变（如果是 R4 非 0，则 $R2 + 1$ ），将 $R3 * 2$ （直接使用 ADD 即可，目的是将 1 左移一位），R3 变为 00010，将还需判断的位数 B 减 1 得到 B 为 2。
 - (2) 循环（1）的操作，直到 B 为 0 为止。
 - (3) 将 R4 的结果存入指定地址即可。

【实验步骤】

1. 首先，输入起始地址 x3000，即第一行为 0011 0000 0000 0000。
2. 将 R0 地址为 x3100，即在 PC: x3000 的基础上加 x0100，用 LD 指令在 PC 的基础上移动即可，第二行为 0010 000 011111111。
3. R2, R3 置零，则第三四行为 0101 010 010 1 00000 和 0101 011 010 1 00000。
4. R3 初始值为 1，第五行为 0010 001 011111100。
5. 由分析，控制指令是由 R1 控制，所以在初始化的最后目的寄存器指向 R1。R1 地址为 x3101，用 LD 指令在 PC: x3004 的基础上移动即可，第六行为 0010 001 011111100。
6. 如果初始的 R1 (B) 为 0，则直接跳转到存入结果的地址即可，目前未知存入结果语句的地址，预留出第七行为 0000 010
*****。
7. R3 与 R0 (A) 取 AND 存入 R4，第八行为 0101 100 000 0 00 011。
8. 判断 R4 是否为 0，是则结果 R2 不增加，否则增加。于是第九行第十行分别为 0000 010 000000001 和 0001 010 010 1 00001。
9. 判断完这一位，R3 左移一位，第十一行为 0001 011 011 0 00 011。
10. 已经判断了一个 bit 的 A，则 B 减一，第十二行为 0001 001 001 1 11111。
11. 判断 B 是否为 0，是则下一步，如果是正数则跳回第八行判断 A 的下一位是否为 1，从第十三行跳到第八行需要前移五个地址位，于是第十三行为 0000 001 111111010。

12. 将结果 R2 存入目的地址 x3102, 第十四行为 0011 010 011110101。
13. 停止程序运行, 第十五行为 1111 0000 0010 0101。
14. 再看第七行预留的, 第七行应该直接跳到第十四行结果赋值的指令中, 所以要加七个地址位, 则第七行的*应该为 000000111, 于是第七行为 0000 010 000000111。

【实验结果】

1. .bin 文件截图如下:

lab1.bin			
1	0011 0000 0000 0000		; 起始地址
2	0010 000 011111111		; R0地址为x3100, 表示A
3	0101 010 010 1 00000		; R2置零 (存放结果)
4	0101 011 011 1 00000		; R3置零 (用于判断A的被判断位数是否为1)
5	0001 011 011 1 00001		; R3置为1
6	0010 001 011111100		; R1地址为x3101, 表示B
7	0000 010 000000111		; R1(B)为0, 则直接跳转到存放结果
8	0101 100 000 0 00 011		; R3与R0(A)取AND存入R4(判断该位是否为1)
9	0000 010 000000001		; 如果R4非零, 则进行下一步, 否则跳过下一步
10	0001 010 010 1 00001		; R2加一 (结果位数加一)
11	0001 011 011 0 00 011		; R3左移一位 (从1开始一直*2)
12	0001 001 001 1 11111		; R1(B)减1 (B bits已经判断了一个bit)
13	0000 001 111111010		; 判断R1(B)是否为0 (是否判断完), 若否则循环
14	0011 010 011110101		; R2存入x3102
15	1111 0000 0010 0101		; 停止执行

2. 运行测试样例的结果如下：

选择评测实验

☒ lab1 ☐ 自定义

测试样例，样例之间以逗号分割

13:4, 167:4, 32767:13

代码文本

```
0001 011 011 1 00001 ;R3直为1
0010 001 011111100 ;R1地址为x3101, 表示B
0000 010 000000111 ;R1(B)为0, 则直接跳转到存放结果
0101 100 000 0 00 011 ;R3与R0(A)取AND存入R4(判断该位是否为1)
0000 010 000000001 ;如果R4非零, 则进行下一步, 否则跳过下一步
0001 010 010 1 00001 ;R2加一(结果加一)
0001 011 011 0 00 011 ;R3左移一位(从1开始一直*2)
0001 001 001 1 11111 ;R1(B)减1(B bits已经判断了一个bit)
0000 001 11111010 ;判断R1(B)是否为0(是否判断完), 若否则循环
0011 010 011110101 ;R2存入x3102
1111 0000 0010 0101 ;停止执行
```

调试模式



如果指令数过多，无法显示日志，可以尝试减小 单样例最大指令数 的值

评测

机器码评测

3 / 3 个通过测试用例

- 平均指令数: 48.333333333333336
- 通过 13:4, 指令数: 30, 输出: 3
- 通过 167:4, 指令数: 30, 输出: 3
- 通过 32767:13, 指令数: 85, 输出: 13