

# 计算系统概论实验报告-lab1

少年班学院 曹高翔 PB20000061

2021 年 11 月 21 日

## 实验名称

温故知新

## 实验目的

在 LC-3 上实现乘法并写出对应程序机器码。

## 实验内容

两个运算数分别放置于 R0 和 R1，将 R0 和 R1 的乘积存储到 R7，其他寄存器状态不做限制（即不限结束状态）。溢出情况参考 C 语言中 16 位 short 型整数的乘法结果。

初始状态：R0 和 R1 存放待计算数，其余寄存器全部为 0。

要求提交两个版本代码，L(Length) 版本尽量编写更少的指令行数，P(Performance) 版本尽量让程序执行更少的指令。

## 实验环境

Windows 11 Home Edition version 21H2, Visual Studio Code, LC3Tools v2.0.2.

## 实验过程

为编写程序方便，本次实验中全部采用先使用 LC-3 汇编编写程序，再手动转换为二进制机器代码的方法。同时，无论是 L 版本还是 P 版本的程序，都会经过如下五组测试数据的验证。

$$1 \times 1 = 1 \quad (1)$$

$$5 \times 4000 = 20000 \quad (2)$$

$$4000 \times 5 = 20000 \quad (3)$$

$$-500 \times 433 = -19892 \text{ (溢出)} \quad (4)$$

$$-114 \times (-233) = 26562 \quad (5)$$

## L 版本

因为本次实验中两个操作数都是整数，可以直接将  $a \times b$  通过循环转换成  $|b|$  次  $a + b$ ，以达到使程序代码最短的效果。汇编代码如下：

```

1  .ORIG x3000
2  LOOP
3      ADD R7, R7, R0
4      ADD R1, R1, #-1
5  BRnp LOOP
6  .END

```

将这段代码翻译成 LC-3 二进制目标代码如下：

```

1  0011 0000 0000 0000
2  0001 111 111 000 000
3  0001 001 001 1 11111
4  0000 101 111111101
5  1111 0000 00100101

```

### P 版本

模拟竖式乘法，将其中一个被乘数每次左移一位，当另一个被乘数的对应位为 1 时加入到结果中。如果令循环的终止条件为被乘数左移一位后为 0，则在最坏情况下（即第二个操作数的末位为 1 时）只需循环 16 次。由于 LC-3 指令集不支持移位指令，我们使用自加（即乘 2）实现左移运算。汇编代码如下：

```

1  .ORIG x3000
2  ADD R2, R2, #1
3  AND R3, R2, R1
4  BRn LOOP
5  ADD R7, R7, R0
6  LOOP
7      ADD R0, R0, R0
8      BRz LEAVE
9      ADD R2, R2, R2
10     AND R3, R2, R1
11     BRz LOOP
12     ADD R7, R7, R0
13  BRnzp LOOP
14  LEAVE
15  .END

```

将这段代码翻译成 LC-3 目标二进制代码如下：

```

1  0011 0000 0000 0000
2  0001 010 010 1 00001
3  0101 011 010 0 00 001
4  0000 010 000000001
5  0001 111 111 0 00 000
6  0001 000 000 0 00 000
7  0000 010 000000101
8  0001 010 010 0 00 010

```

```

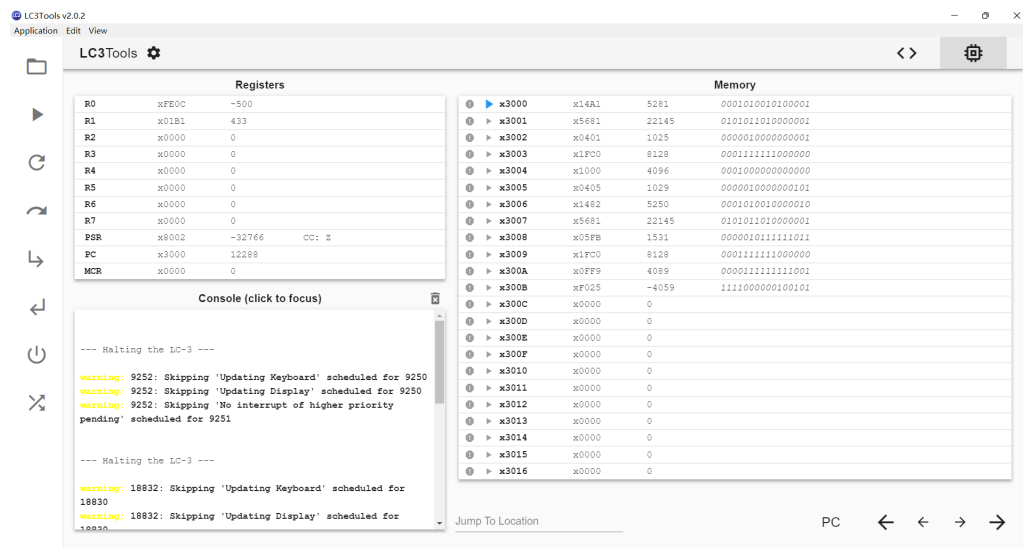
9  0101 011 010 0 00 001
10 0000 010 111111011
11 0001 111 111 0 00 000
12 0000 111 111111001
13 1111 0000 00100101

```

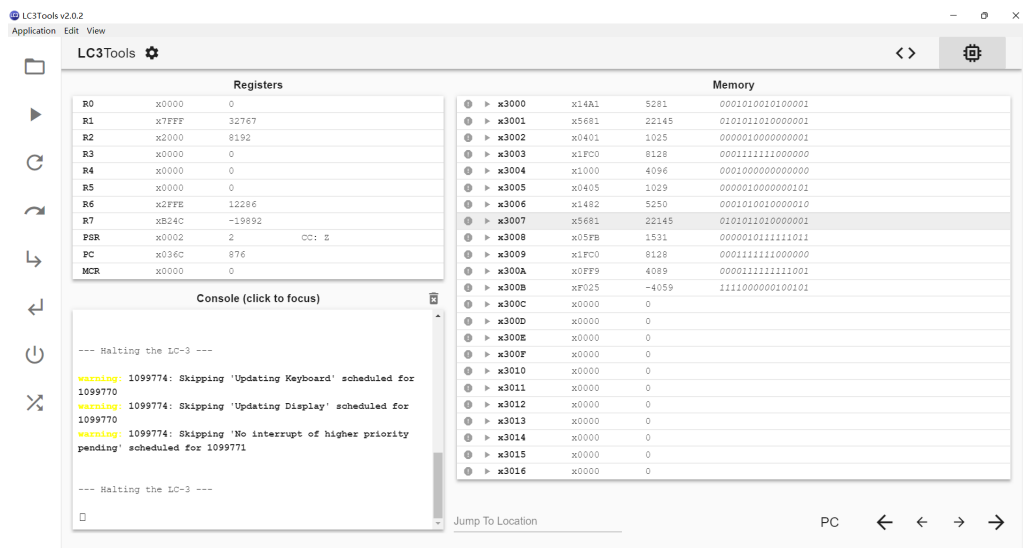
在计算机执行指令数最多的情况下（即两个操作数为 xFFFF 和 xFFFF 的情况下），LOOP 循环被执行 16 次，前 15 次循环完整执行 7 条指令，最后一次循环在执行两条指令后退出。整个程序共执行 112 条指令。

## 正确性验证

本次实验采用 LC3Tools 进行代码的正确性验证。其中一次验证时，程序运行前后 LC3Tools Simulator 界面的截图如下。



(a) 运行前



(b) 运行后