Lab1 of ICS

姓名:曾芑润

学号: PB21000009

Lab1 of ICS

Purpose

Principles

Procedure

Result

Purpose

- 1. To write a program in LC-3 machine language that counts how many 1 are in the lower B bits of a given number A, and stores the output in memory.
- 2. To master how to use LC-3 tool.
- 3. To understand the principle of LC-3 code.
- 4. To know the characters of LC-3 code.
- 5. To have a good command of writing LC-3 code.

Principles

• Use as less instructions, memories, time as possible.

先将 A 和 B 赋给 R5 和 R3,从R5的最后一位开始往前,每次检验 R5 的一位,为1则R0++,每次R3--,当 R3=0 时结束程序

详细过程见程序

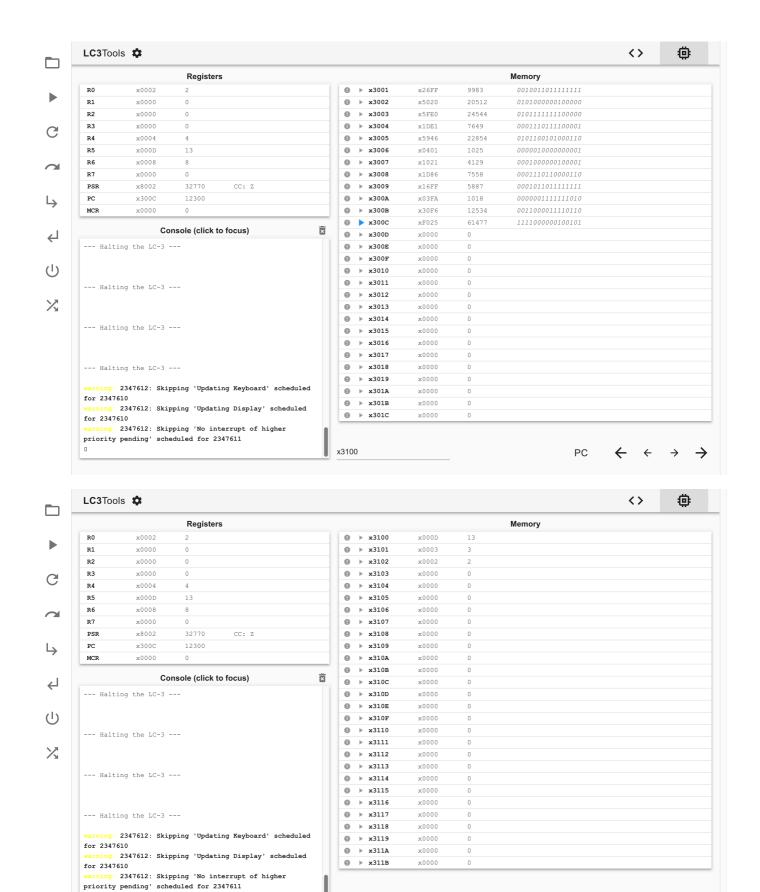
Procedure

```
1 0011 0000 0000 0000 ; 从 x3000 开始
2 0010 101 011111111 ; R5 <- A (位于 x3100)
3 0010 011 0111111111 ; R3 <- B (位于 x3101)
4 0101 000 000 1 00000 ; R0 <- 16'b0 & R0 (R0 <- 0)
5 0101 111 111 1 00000 ; R7 <- 16'b0 & R7 (R7 <- 0)
6 0001 110 111 1 00001 ; R6 <- 16'b1 AND R7 (R6 <- 16'b1)
7 0101 100 101 0 00 110; R4 <- R5 & R6
8 0000 010 0 00000001 ; 当 R4 为 0 时跳过下一步,即 R5 的从右往左数第 i 位 9 0001 000 000 1 00001 ; R0 <- R0 + 1
10 0001 110 110 110 0 00 110; R6 <- R6 * 2
11 0001 011 011 1 11111 ; R3 <- R3 - 1
12 0000 001 111111010 ; 当 R3 > 0 时回跳至 x3005
13 0011 000 011110110 ; 终止程序
```

```
1 0011 0000 0000 0000 ; 从 x3000 开始
   0010 101 0111111111 ;x3000 LD: R5 <- A (位于 x3100)
2
  0010 011 011111111 ;x3001 LD: R3 <- B (位于 x3101)
   0101 000 000 1 00000 ; x3002 AND: R0 <- 16'b0 & R0 (R0 <- 0)
   0101 111 111 1 00000 ; x3003 AND: R7 <- 16'b0 & R7 (R7 <- 0)
   0001 110 111 1 00001 ;x3004 ADD: R6 <- 16'b1 + R7 (R6 <- 16'b1)
   0101 100 101 0 00 110; x3005 AND: R4 <- R5 & R6
7
   0000 010 0 000000001 ;x3006 BR: 当 R4 为 0 时跳过下一步
   0001 000 000 1 00001 ;x3007 ADD: R0 <- R0 + 1
   0001 110 110 0 00 110; x3008 ADD: R6 <- R6 * 2
   0001 011 011 1 11111 ;x3009 ADD: R3 <- R3 - 1 (执行 B 次)
11
   0000 001 111111010 ;x300A BR: 当 R3 > 0 时回跳至 x3005
12
   0011 000 011110110 ;x300B ST: 存储 RO 结果到 x3102
13
   1111 0000 0010 0101 ;x300C 终止程序
```

Result

1. A=13, B=3 exit之前寄存器状态



x3100

РС

 \leftarrow \leftarrow \rightarrow \rightarrow

如上图所示:

初始化程序时:

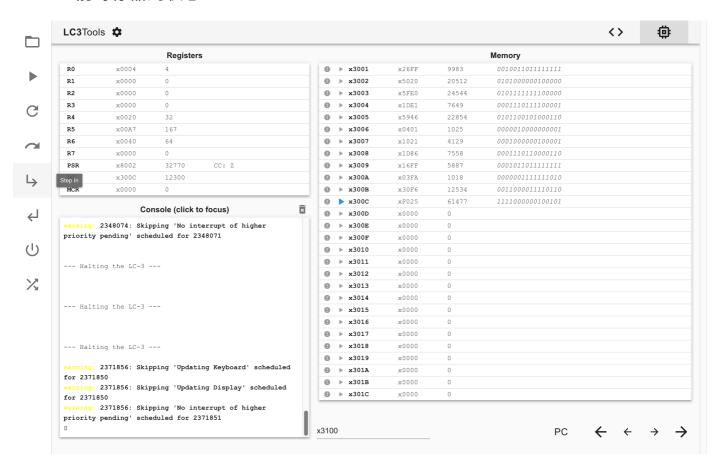
$$R5 = A = 13 = 0b1101$$

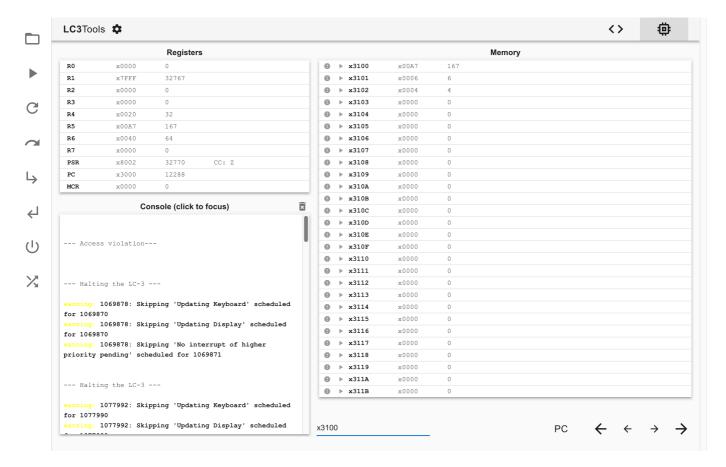
 $R3 = B = 3$
 $R6 = 0b1$

而后每一步:

R6 依次变成 0b10, 0b100, 从此检验了 A 的后 B 位。

2. A=167, B=6 exit前寄存器的状态





3. A = 32767, B = 15 exit之前寄存器状态

