Report of Lab01

*---Counting how many 1*

*PB21071416 张郑飞扬*

**Purpose**

Write a program in LC-3 machine language that counts how many 1 are in the lower B(x3101) bits of a given number A(x3100), and stores the output in memory(x3102).

**Principles**

1.Tools

Use ‘LC3Tools-2.0.2.exe’ as a simulator of LC-3.

2.How to justify whether a certain bit is 1 or not

To achieve the function we need, the AND operation is chosen to be   
 the core operation.

In order to justify whether the certain bit is 1, (take the second   
 lowest bit—bit[1] as an example), we can operate ‘A AND x0002’.   
 If the result is 1, we can tell that the second lowest bit (bit[1]) is 1,   
 else it’s 0.

Thus, we get the approach to justify a certain bit.

3.How to count

We just need to set a counter. Every time the result of the AND   
 operation comes out to be 1, we make the counter +1. Once we end   
 the whole loop, the data of the counter is just what we need.

4.How to set the loop

Every time an AND operation is completed and the counter is   
 renewed, it’s a signal to enter the next cycle. But how may cycles do  
 we need to form the whole loop？The answer is B. We can simply   
 use B to determine whether or not to end the loop. Every time a   
 cycle is done. We make B-1. Once B achieves 0, the loop is over.

5.How to renew the number that is to AND with A every cycle

Let’s name this number as ‘C’. We know that in the first cycle,   
 C is ought to be x0001, the second cycle x0002, the third cycle   
 x0004 …… Thus, we can add C to itself each cycle before   
 operate AND with A.

6. Registers to be used

In this program, we use 5 registers. Their functions are listed   
 below:

Register 1: to be used as counter.

Register 2: to be used to store the operand of the AND instruction   
 at each cycle i.e. the number C mentioned above.

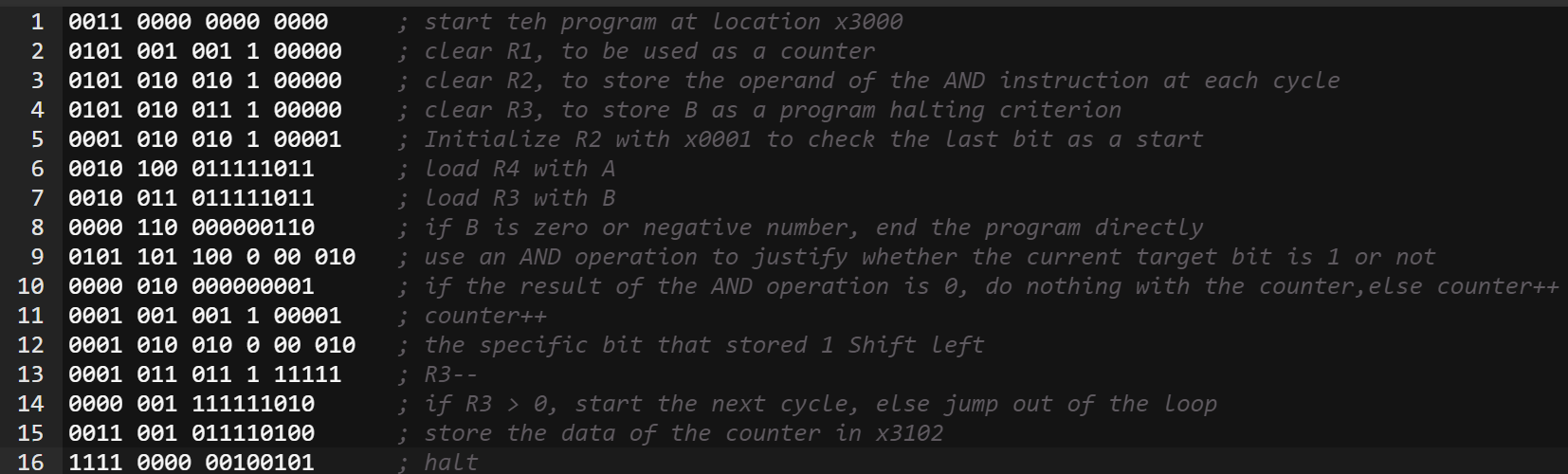
Register 3: to be used to store B at first and minus 1 each cycle.

Register 4: to be used to store A.

Register 5: to be used to store the result of ‘A AND C’ each cycle.

**Procedure**

The program is listed below:



Here is a brief summary of the procedure :

1.Initialization

Start the program at x3000;

Clear registers;

Load R2 with x0001;

Load R3 with B, R4 with A;

2.Justify B

If B is negative or zero, directly end the program;

3.Start the loop

Operate A and C;

If the result is 1, count++, else do nothing;

C= C+C (SHIFT LEFT);

B--;

If B>0, start the next cycle, else jump out of the loop;

4.Store the data of R1(counter) into x3102, End the program.

**Result**

At last we can get the result in memory x3102.

The test data and their result are listed below(decimal):

Number\_A(in x3100) Bit\_B(in x3101) Output(in x3102)

13 3 2

167 6 4

32767 15 15