**Assignment 1 Report**

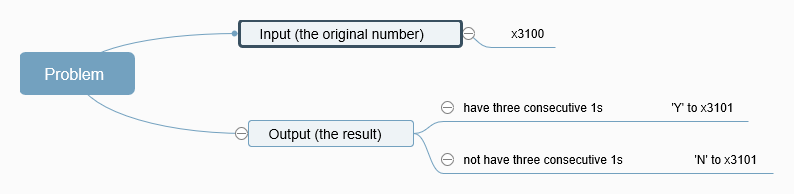
**1.Problem：**

A 16-bit value is stored in memory location x3100 of the LC-3. Your task is to figure out whether or not the 16 bits contain at least three consecutive 1's.

1) If so, you are to store the ASCII code for Y in memory location x3101,

2) Otherwise, put the code for N at the same location.

e.g. If x3100 contains: 0110000110001101, you should store the ASCII code for N in x3101. If the memory content is 0000001101111000, ASCII code for Y should be put in x3101.

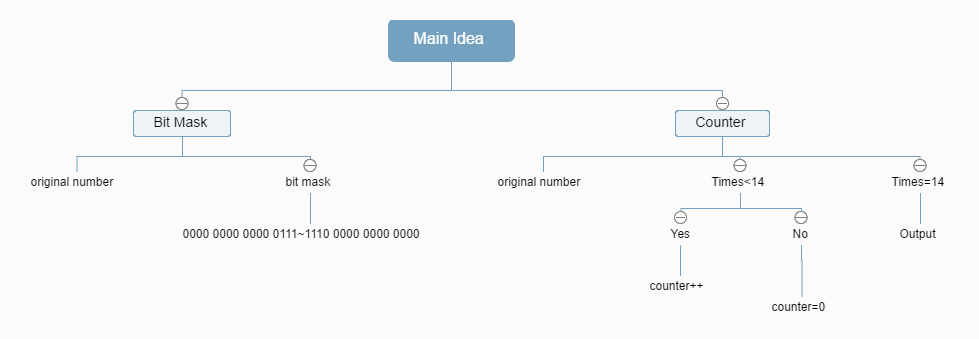


**2.Main Idea:**

I came up with two different main ideas and during the assignment 1 I chose the first idea below:

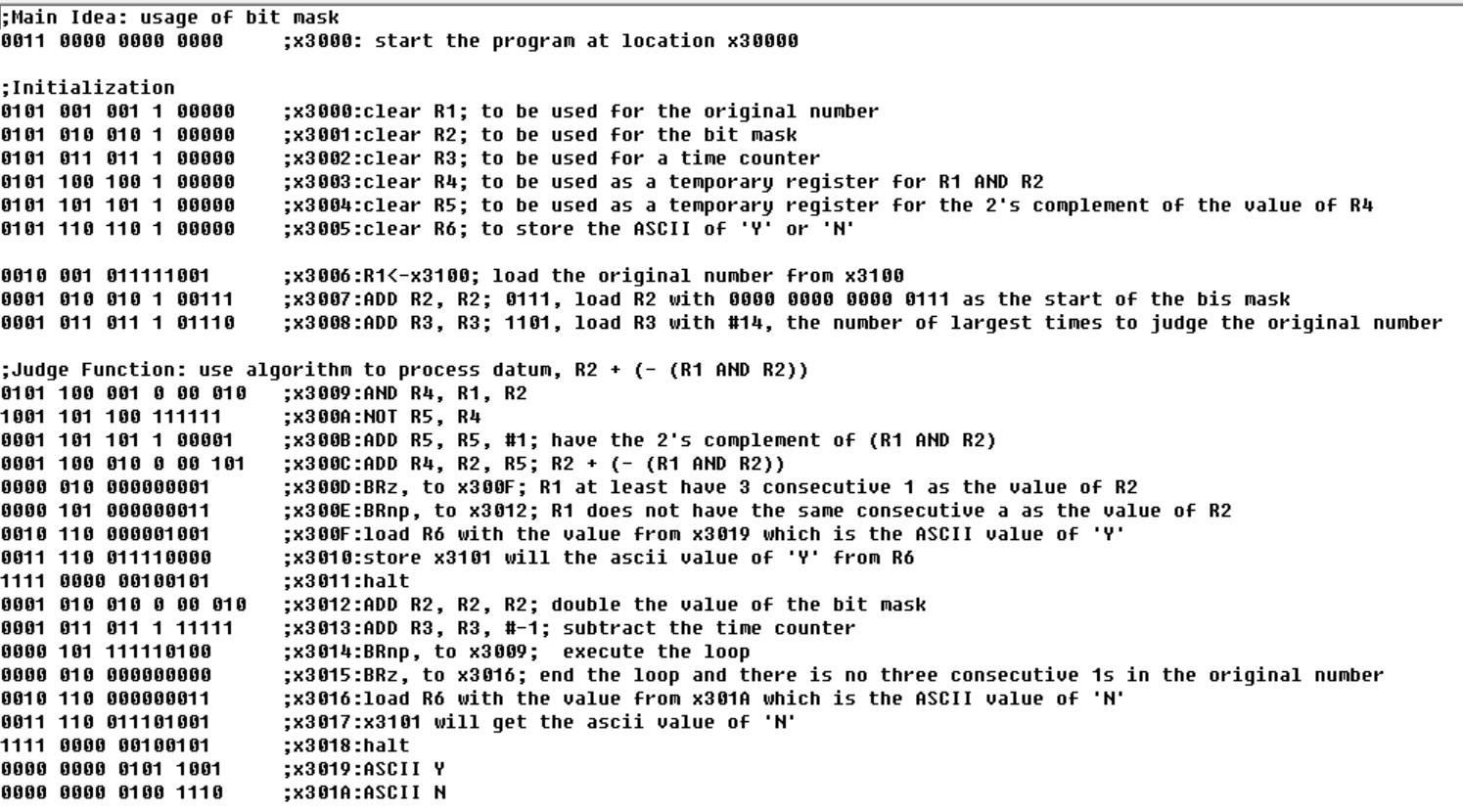
1）**Bit Mask:** use the instruction of AND with values ofthe original number from x3100 and a 16-bit large bit mask which contains three consecutive 1s and others are left 0. The Bit Mask starts from 0000 0000 0000 0111 to 1110 0000 0000 0000, and adds the value of itself every time (the same as double but since there is no multiply instruction in LC-3).

2) **Counter:** to check every bit in the original number from x3100, the counter adds 1 when meets 1 and becomes 0 when its 0. If the counter meets 3 which means the original number contains three consecutive 1s and outputs the ASCII value of ‘Y’ to x3101. After checking the whole number, if the counter never achieves 3, then outputs the ASCII value of ‘N’ to x3101.

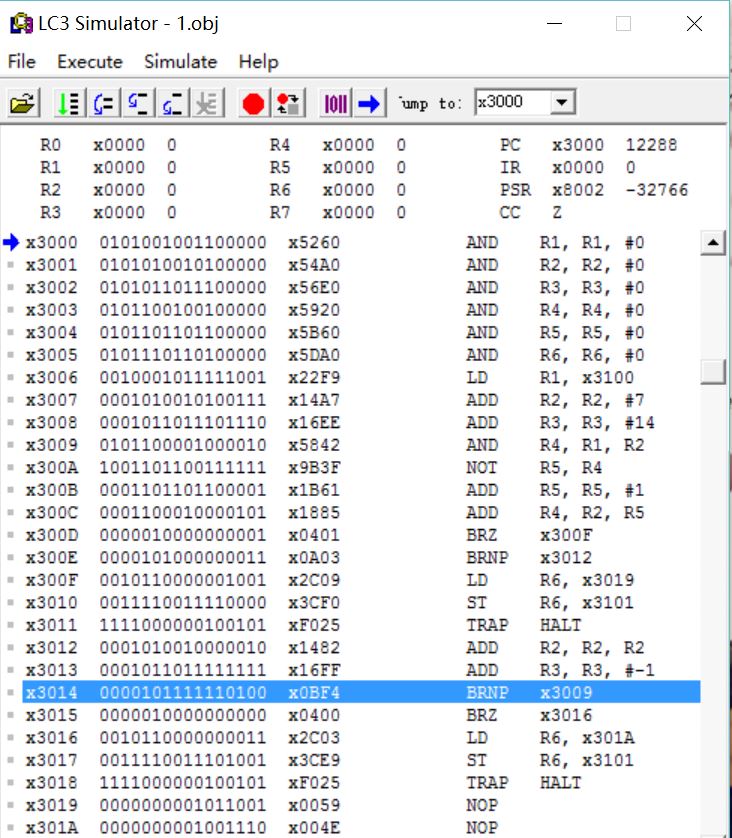


**3.Codes：**

1）Codes with comments:

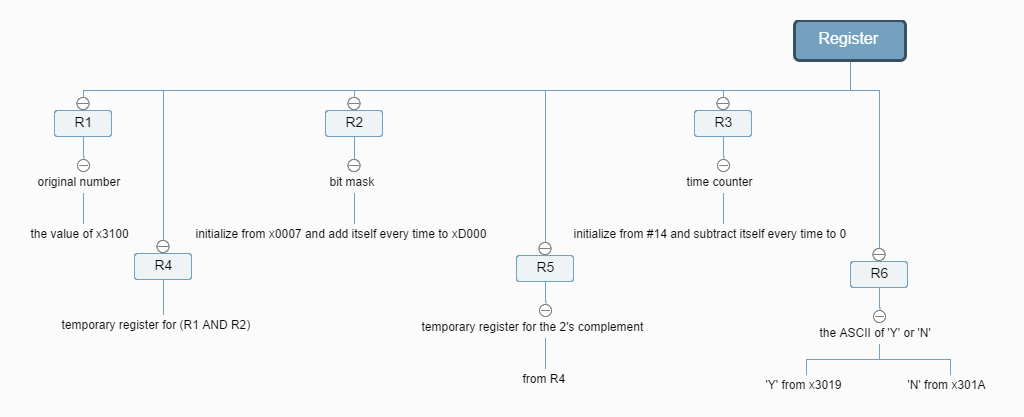


2)Codes in Simulator:



**4.Details about Program：**

1）Register:



2) Initialization:

**a.** initializes R1, R2, R3, R4, R5, R6 first, even though LC-3 will initialize registers automatically, I decided still to initialize them so that the program can check different values of x3100 without restart the program everytime.

**b.0010 001 011111001**

;x3006:R1<-x3100; load the original number from x3100

**c.0001 010 010 1 00111**

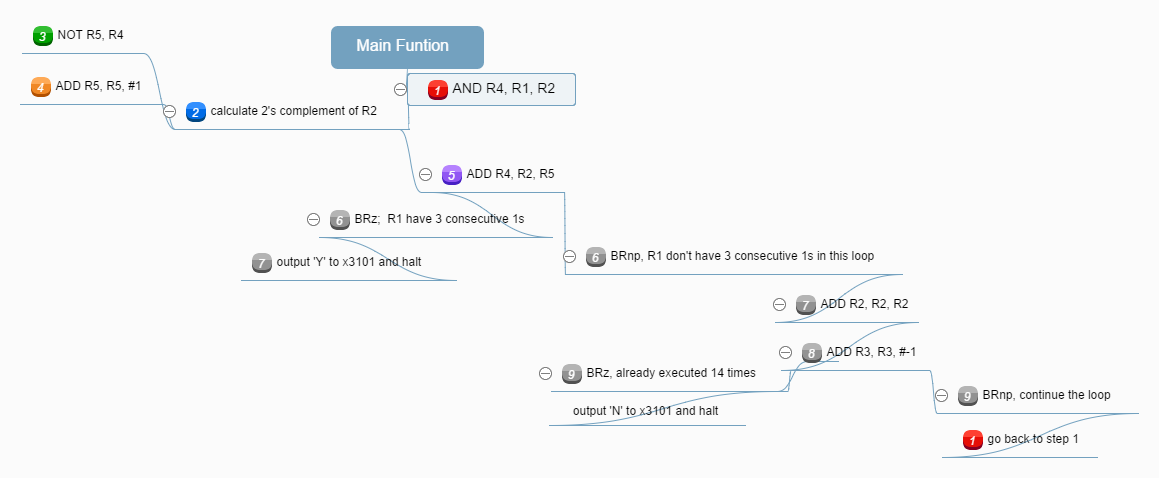
;x3007:ADD R2, R2; 0111; load R2 with 0000 0000 0000 0111 as the start of the bit mask

**d.0001 011 011 1 01110**

;x3008:ADD R3, R3; 1101, load R3 with #14, the number of largest times to judge the original number

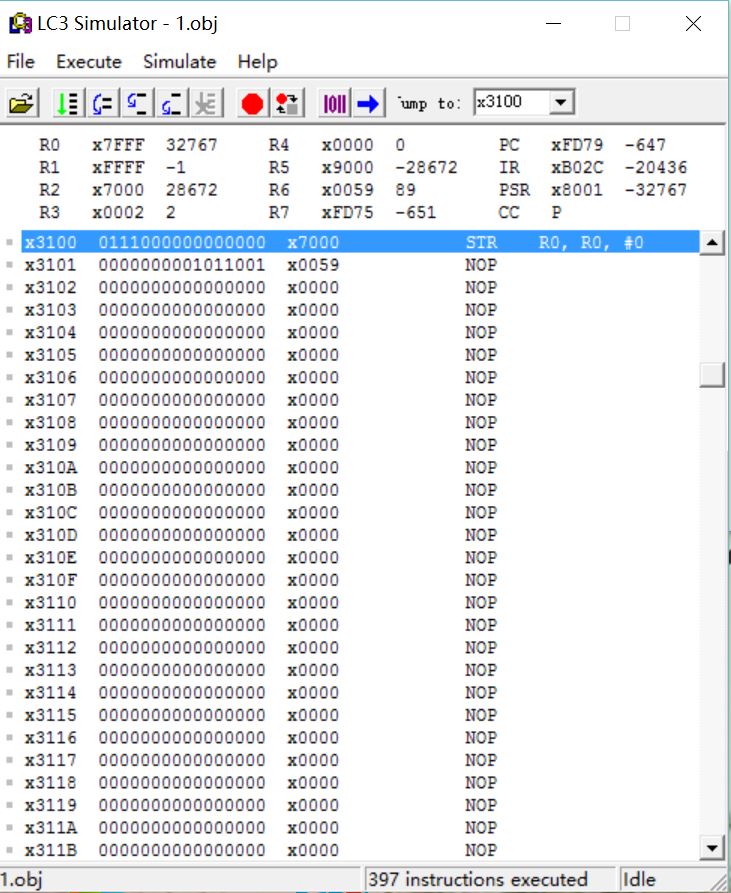
3) Algorithm: R2+(-(R1&R2))

4) Main Function:

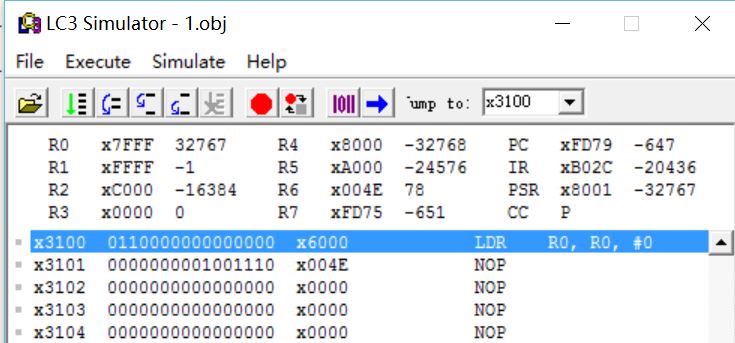


**5.Result：**

1）Yes



2)NO



Before Coding:

