lab01 - CountingZero

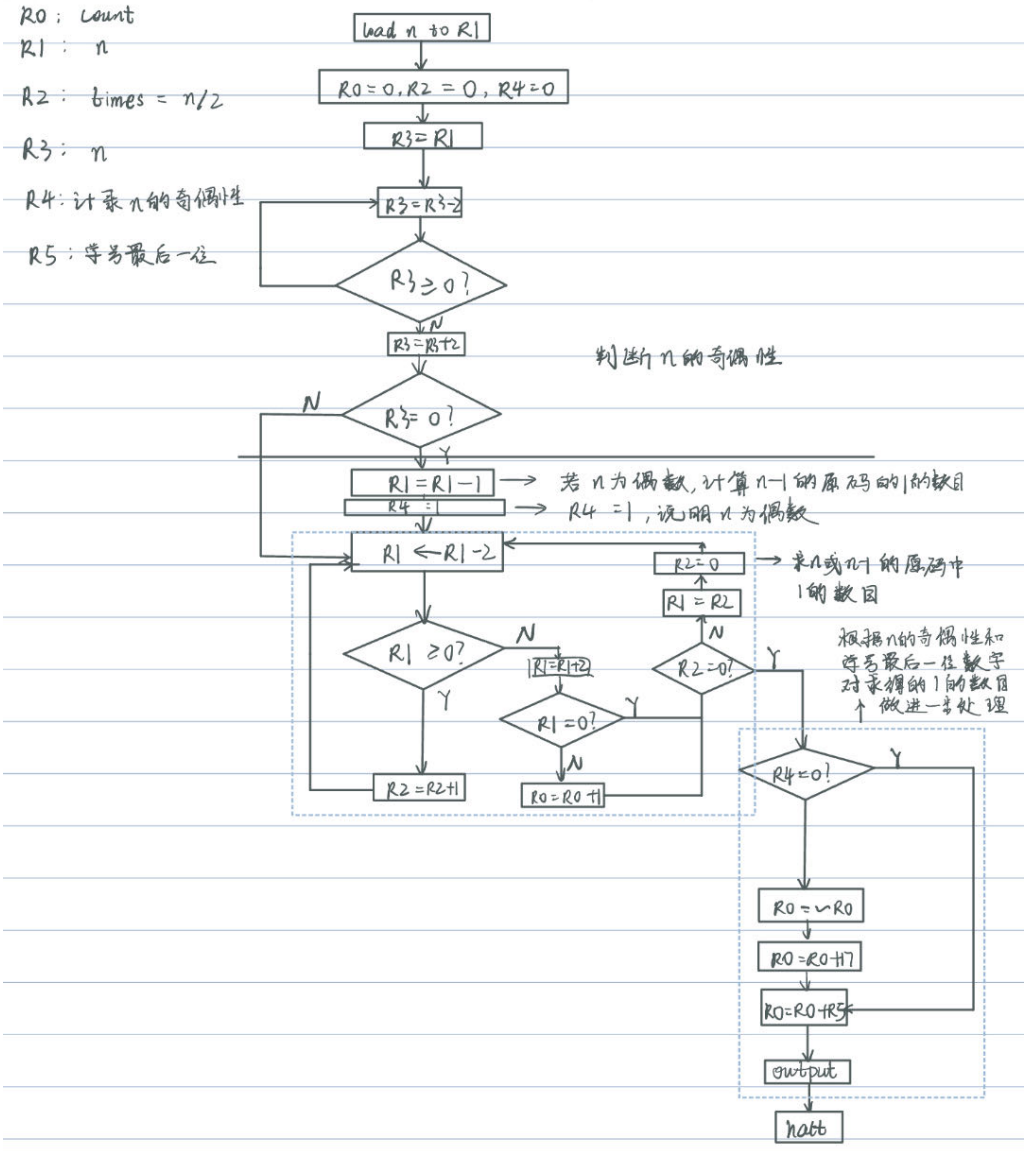
PB22081571 薄震宇

解题思路

求二进制码中0的个数可以模拟短除法来设计实验。 n为奇数时求原码中0的个数，但是由于原码长度为16位，而短除法求出的位数未知，所以如果直接求0的个数还需要一个量来表示短除法求得的位数长度。 n为偶数时求-n的补码的0的个数，如果先求原码再转补码就会很麻烦。而-n的补码的数值位与n-1的反码的数值位相同，其符号位又一定是1，这样-n的补码的0的个数就等于n-1的原码的1的个数。 于是可以求n或n-1的二进制码的1的位数count，对于奇数，16-count即为二进制码中0的个数；而对于偶数count即为所求。最后再加上学号的最后一位即可。

流程图

根据上述解释可作出如下流程图：



代码编写

先按照这个思路编写C语言代码测试正确性：

```c
#include<stdio.h>
int main()
{
    int n,count=0,num,isEven=0;
    printf("请输入n:\n");
    scanf("%d",&n);
    if(n%2==0){
        n--;//如果n为偶数，计算n-1的原码的1的数目，即为n的补码的0的树目
        isEven=1;
    }
    while(n){
        if(n%2) count++;//也可写作count+=n%2;
        n=n/2;
    }
    if(!isEven){
        count=16-count;
    }
    printf("The number of 0 is:%d\n",count);
    printf("请输入学号最后一位数字:\n");
    scanf("%d",&num);
    count+=num;
    printf("The final result is %d\n",count);
    return 0;
}
```

运行结果如下：

```
F:\ICSH> cmd /C "c:\Users\bzy\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe --stdin=Microsoft-MIEngine-In-pxa2v1mu.5dj --stdout=Microsoft-MIEngine-Out-1115hcnp.txj --stderr=Mic
rosoft-MIEngine-Error-eg2cs0up.fx2 --pid=Microsoft-MIEngine-Pid-atedf2iz.l1s --dbgExe=F:\mingw64\bin\gdb.exe --int
erpreter=mi "
请输入n:
5
The number of 0 is:14
请输入学号最后一位数字:
1
The final result is 15

F:\ICSH> cmd /C "c:\Users\bzy\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe --stdin=Microsoft-MIEngine-In-bqmgsss4.njz --stdout=Microsoft-MIEngine-Out-sszxjx3n.zmn --stderr=Mic
rosoft-MIEngine-Error-jpoc2gcj.oio --pid=Microsoft-MIEngine-Pid-gezsnu02.5vd --dbgExe=F:\mingw64\bin\gdb.exe --int
erpreter=mi "
请输入n:
100
The number of 0 is:4
请输入学号最后一位数字:
1
The final result is 5
```

结果正确，于是可以编写机器码

机器码如下：

```
0011 0000 0000 0000     ;start the program at loaction x3000
0010 001 011111111      ;从x3100中取n放到R1中，PCoffset=x3100-x3001=255
0010 101 011111111      ;从x3101中去学号最后一位放到R5中
0101 000 000 1 00000    ;R0清零，记录n或n-1的二进制表示中1的位数
0101 010 010 1 00000    ;R2清零，存放n/2的值
```

```
0001 011 001 1 00000    ;R3中也存放n的值
0101 100 100 1 00000    ;R4清零，记录n的奇偶性，R4=1时为偶数，R4=0时为奇数
0001 011 011 1 11110    ;R3=R3-2
0000 011 111111110      ;如果R3非负，跳转到R3=R3-2
0001 011 011 1 00010    ;R3=R3+2
0000 001 000000010      ;如果R3大于0即R3=1,说明n为奇数，不需要减1，跳过后面两条指令
0001 001 001 1 11111    ;如果R3=0，说明n为偶数，需要计算n-1的原码中1的数目即为-n的补码
中0的数目
0001 100 100 1 00001    ;R4赋值为1，以表示n为偶数
0001 001 001 1 11110    ;R1=R1-2
0000 100 000000010      ;如果R1<0，跳转至R1=R1+2
0001 010 010 1 00001    ;R2=R2+1
0000 111 111111100      ;无条件跳转至R1=R1-2
0001 001 001 1 00010    ;R1=R1+2
0000 010 000000001      ;R1=0则跳转至判断R2是否为0
0001 000 000 1 00001    ;R1不等于0则等于1，说明余数为1，R0自加1
0001 010 010 1 00000    ;R2=R2+0,改变nzp使其与R2中的值相关
0000 010 000000011      ;R2=0则count计算完毕，跳转至后续处理过程
0001 001 010 1 00000    ;R1=R2
0101 010 010 1 00000    ;R2=0
0000 111 111110100      ;无条件跳转至R1=R1-2
0001 100 100 1 00000    ;R4=R4+0,改变nzp使其与R4中的值相关
0000 001 000000011      ;R4>0说明n为偶数，R0中的值即为-n的二进制补码中0的数目,跳过后面
三条指令
1001 000 000 111111     ;R0取反
0001 000 000 1 00010    ;R0=R0+2,即将原R0中的数目取反加1再加1
0001 000 000 1 01111    ;R0=R0+14，分两次完成了16-R0,由于立即数范围的限制，第一步将R0
取反加2而不是加1
0001 000 000 0 00 101   ;R0=R0+R5
0011 000 011100011      ;将R0中的结果存到x3102中
1111 0000 00100001      ;Output
1111 0000 00100101      ;halt
```

这里面共使用R0,R1,R2,R3,R4,R5五个寄存器，它们的功能如下：

R0：存储n或n-1的二进制码中1的位数，设为count，最后在根据n的奇偶性对count进行调整：

- n为偶数时用16减去count，需要注意的是，由于立即数只有五位且为有符号数，所以无法用立即数来表示16。在代码中我采取了先将count取反加2而不是加1，再用15减去count的方式，也可以用另一个寄存器先存17，再将count取反，然后将这两个寄存器中的数相加。
- n为奇数时count即为-n的二进制补码中0的个数

最后再将count加上学号最后一位，我将其存储在R5中

R1：存储n的值

R2：用于记录n/2，模拟短除法

R3：用于存储n，在计算n的奇偶性时会改变n，所以用两个寄存器存了n的值，当然也可以在得出奇偶性后再次从x3100中取一次n，但考虑到从内存中取数很慢且寄存器数量尚且充足，所以我选择了将R1中取得的n再赋给R3，用两个寄存器来存储

R4：表示n的奇偶性，n为偶数时R4中的值为0，n为奇数时R4中的值为1

R5：存储学号的最后一位

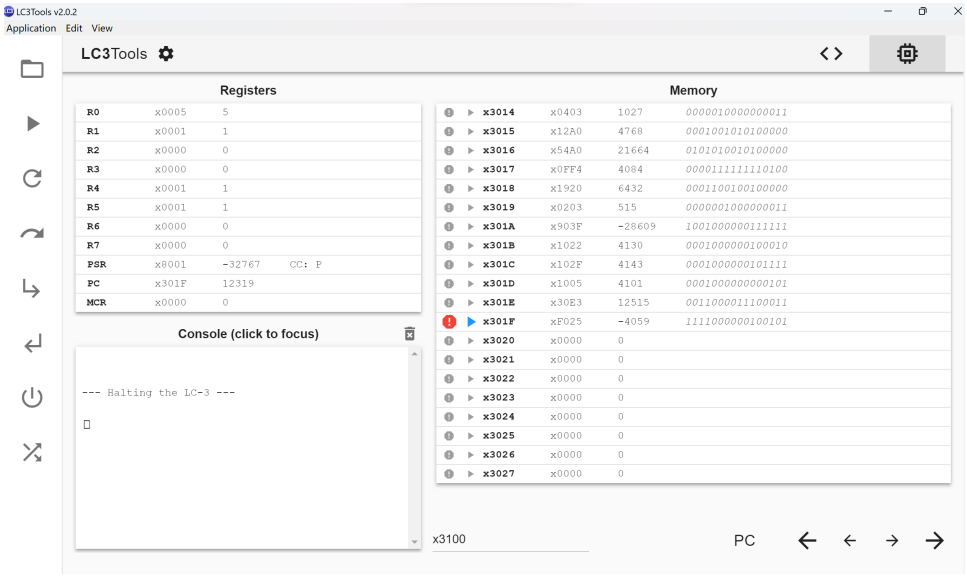
与C语言代码不同的是，机器码中无法直接求出n%2，n/2等等。所以需要不断的用n减去2，直至n<0时，将n再加上2，通过判断此时n是0还是1来判断n是奇数还是偶数。但是这样做会改变n的数值，所以代码中除了使用R1来存储n，还将n赋给了R3。在求n/2时，则使用了另一个寄存器R2来存储n/2，n-2>0时，R2自加1。

总的来说，本题可分为三大步：

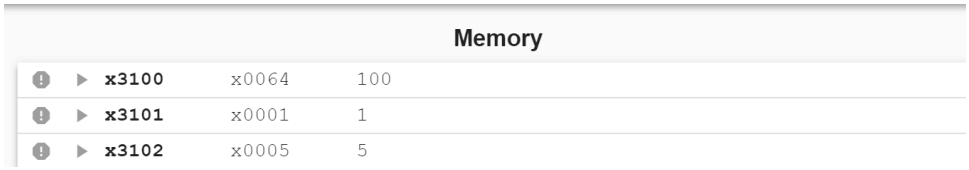
1. 判断n是奇数还是偶数，如果n是偶数将n减1.计算n-1的二进制码中的1的数目
2. 求n或n-1的二进制码中的1的数目count
3. 根据n的奇偶性对count进行处理，如果n是奇数，则用16减去count；如果n是偶数则不处理。最后再将count加上学号最后一位数字1即得最终结果，存入x3102中

运行结果

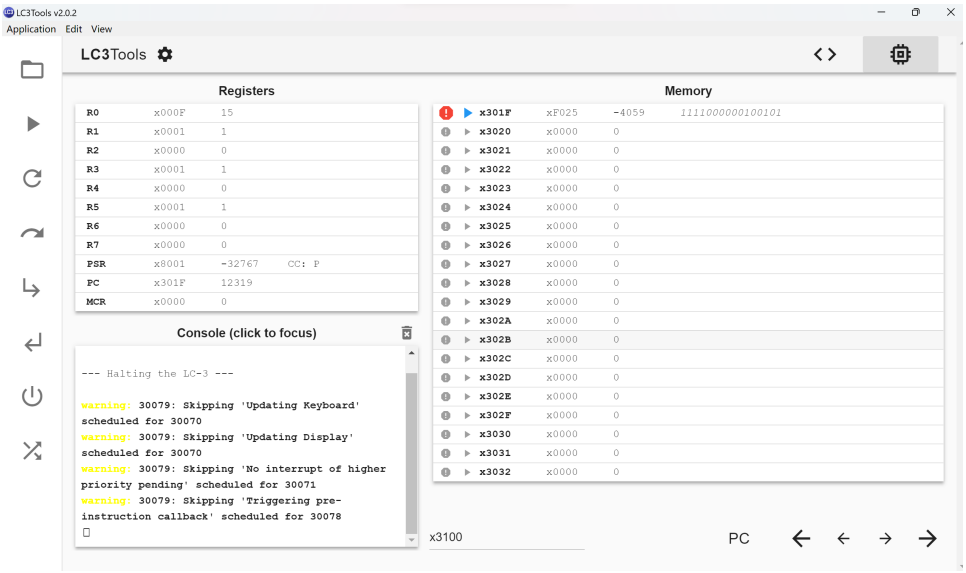
- 用题目中的例子，n为偶数100时，运行结果如下： 在halt命令前停止可得：



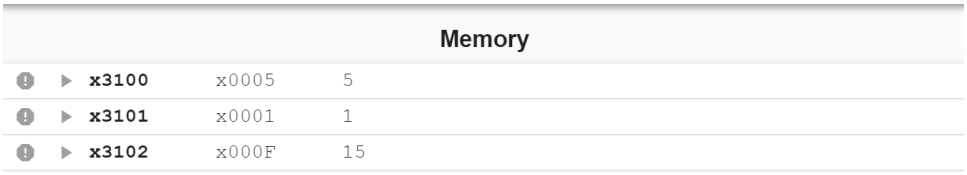
可以看到此时R0中存储的值为5，我的学号最后一位是1，所以答案5正确。此时内存的x3102表示的位置也存储最终的结果5:



- n为奇数5时，运行结果如下：

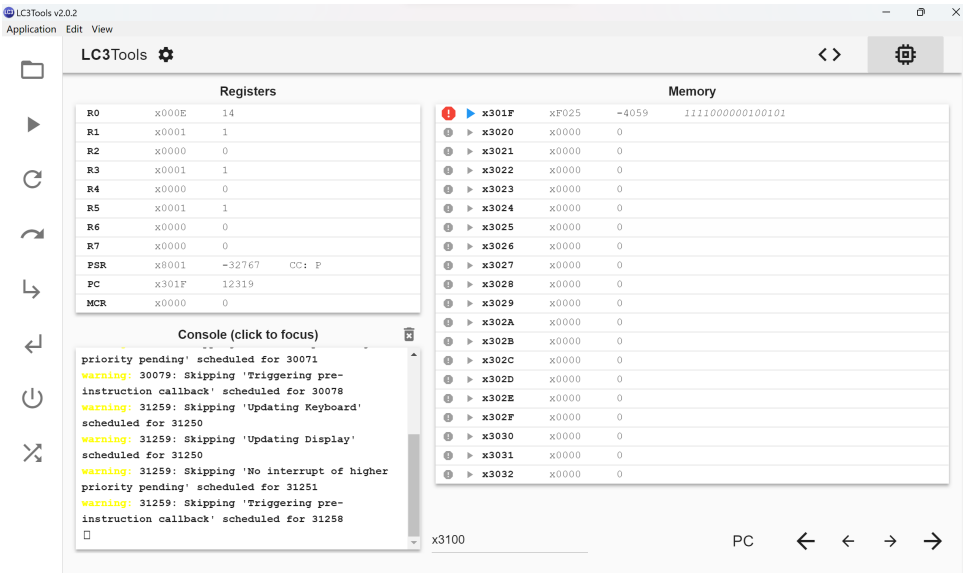


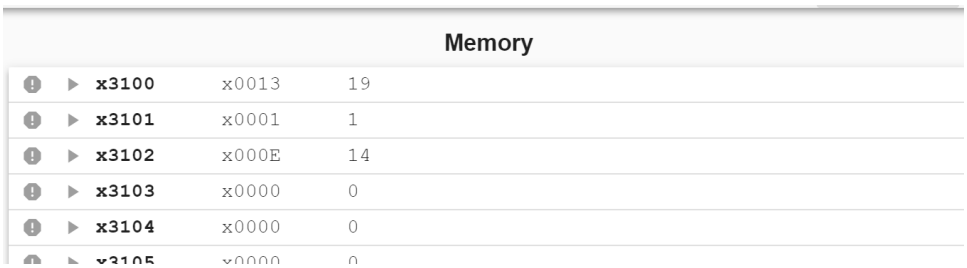
可以看到此时R0中存储的值为15，答案也正确。此时内存的x3102表示的位置也存储最终的结果15：



再随便运行几个数来测试：

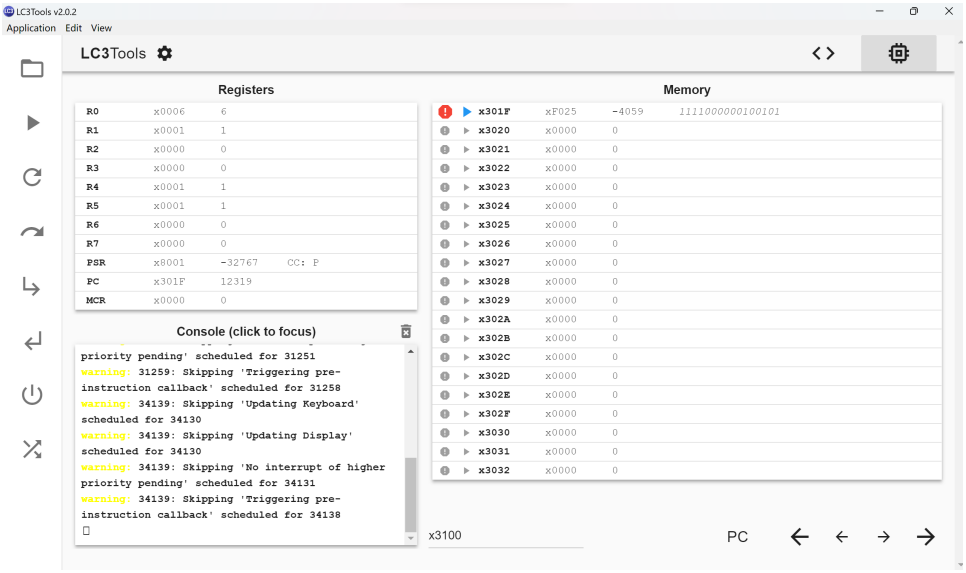
- 令n=19，n为奇数，它的二进制编码为0000 0000 0001 0011，其中0的数目为13，加上学号最后一位后结果应为14。此时的运行结果为：

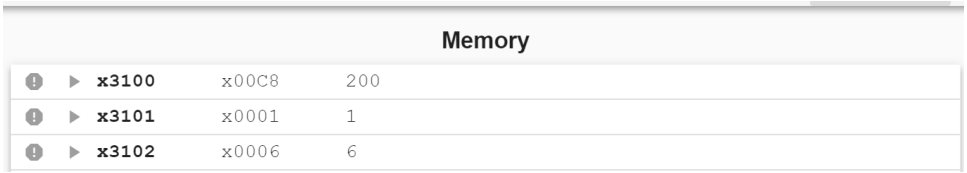




结果正确！

- 令n=200，n为偶数，它的二进制编码为`0000 0000 1100 1000`，-n的补码为`1111 1111 0011 1000`，其中0的数目为5，加上学号最后一位后结果应为6。此时的运行结果为：





结果正确！

综上，机器码编写正确