**Lab01**

**1. 算法思想**

实验题目要求在内存的x3100位置存放一个初始的16 bit的0-1串，在x3101指定要求左移的位数，将初始的0-1串左移指定位数后将结果存放到x3102中。

将初始的0-1串看作一个16 bit的二进制数，是第i位的值，则初始的0-1串可表示成

将该二进制数乘以2，即得

可以发现原～左移了1位，假设通过某种方式将移到了最后，即得

将(3)式再乘以2，得到

通过同样的方法将(3)的最高位移到(4)的最后，得到

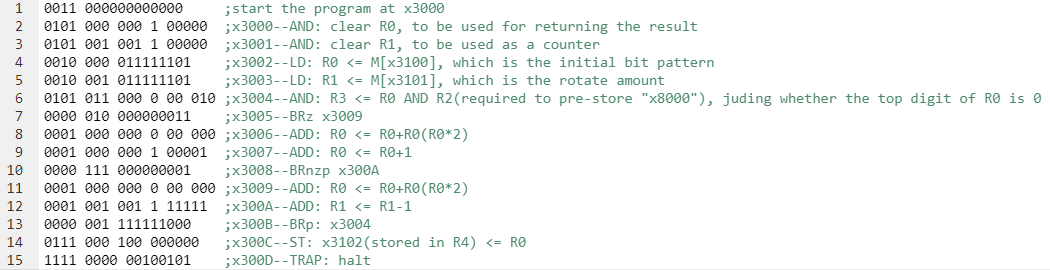
这样，初始的0-1串就可以通过多次乘以2和将最高位放到乘积结果的最后来得到左移指定位数后的0-1串。

在LC-3中，可以将初始的和过程中产生的0-1串放在一个寄存器中，就可以通过ADD指令，将自己与自己相加来实现乘以2。二进制数每次自己与自己相加后，最末位必定为0，因此只要判断当前0-1串的最高位是否为1，若为1，则左移后的0-1串的最末位为1，这时只要在自己加自己后再加1即可实现左移1位；若当前0-1串的最高位为0，则只要自己加自己即可实现左移1位。

判断最高位是否为1可以通过将当前0-1串和1000\_0000\_0000\_0000(存放在另一个寄存器中)相AND的结果来判断，将结果放到一个寄存器中，通过N、Z、P寄存器和BR指令来判断，令BR指令中的nzp为010即可(检查z寄存器)，若最高位为0，则相AND结果为0，BR指令检查z寄存器就能跳转到自己与自己相加但不加1的操作指令；当最高位为1，则相AND的结果不为0，BR指令不跳转，后面紧跟自己与自己相加后再加1的操作指令。

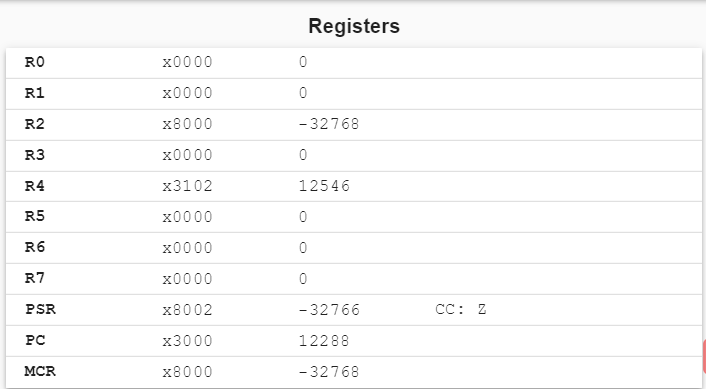
x3101存放的指定位数则取出并放到一个寄存器中作为计数器，每当实现一次左移操作，该寄存器中的值就减1，直到为0为止。将最后得到的结果存放到x3102便完成任务。

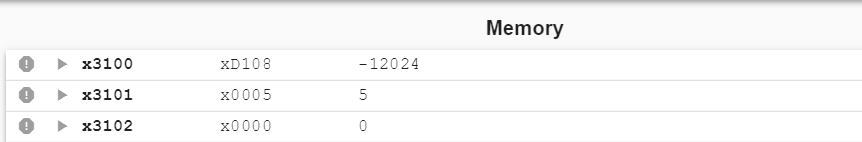
**2. 代码实现**

****

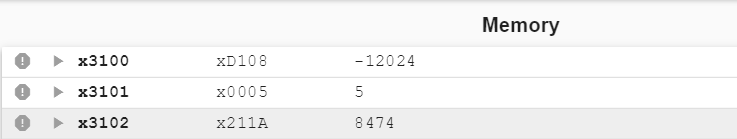
**3. 测试**

取lab01讲义中的初始0-1串：1101\_0001\_0000\_1000(xD108)存放在x3100中，x3101中存放0000\_0000\_0000\_0101()，则最后应看到的结果是x3102中存放着0010\_0001\_0001\_1010(x211A)。在程序运行前还需对一些寄存器进行初始化：R2 <= x8000(用于判断当前0-1串最高位是否为1)，R4 <= x3102(最后存放结果的内存地址，ST指令需要用到)。初始化后的情况如下：





运行后结果如下：



得到了预期结果。