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，b_i是第 i 位的值，则初始的 0-1 串可表示成

$$b_{15} \times 2^{15} + b_{14} \times 2^{14} + \cdots + b_1 \times 2 + b_0 \times 1 \qquad (1)$$

将该二进制数乘以 2，即得

$$b_{14} \times 2^{15} + b_{13} \times 2^{14} + \cdots + b_0 \times 2 + 0 \times 1 \qquad (2)$$

可以发现原b_0～b_{14}左移了 1 位，假设通过某种方式将b_{15}移到了最后，即得

$$b_{14} \times 2^{15} + b_{13} \times 2^{14} + \cdots + b_0 \times 2 + b_{15} \times 1 \qquad (3)$$

将(3)式再乘以 2，得到

$$b_{13} \times 2^{15} + b_{12} \times 2^{14} + \cdots + b_{15} \times 2 + b_{14} \times 1 \qquad (4)$$

通过同样的方法将(3)的最高位移到(4)的最后，得到

$$b_{13} \times 2^{15} + b_{12} \times 2^{14} + \cdots + b_{15} \times 2 + b_{14} \times 1 \qquad (3)$$

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

```
0011 000000000000    ;start the program at x3000
0101 000 000 1 00000 ;x3000--AND: clear R0, to be used for returning the result
0101 001 001 1 00000 ;x3001--AND: clear R1, to be used as a counter
0010 000 011111101   ;x3002--LD: R0 <= M[x3100], which is the initial bit pattern
0010 001 011111101   ;x3003--LD: R1 <= M[x3101], which is the rotate amount
0101 011 000 0 00 010 ;x3004--AND: R3 <= R0 AND R2(required to pre-store "x8000"), juding whether the top digit of R0 is 0
0000 010 000000011   ;x3005--BRz x3009
0001 000 000 0 00 000 ;x3006--ADD: R0 <= R0+R0(R0*2)
0001 000 000 1 00001 ;x3007--ADD: R0 <= R0+1
0000 111 000000001   ;x3008--BRnzp x300A
0001 000 000 0 00 000 ;x3009--ADD: R0 <= R0+R0(R0*2)
0001 001 001 1 11111 ;x300A--ADD: R1 <= R1-1
0000 001 111111000   ;x300B--BRp: x3004
0111 000 100 000000  ;x300C--ST: x3102(stored in R4) <= R0
1111 0000 00100101   ;x300D--TRAP: halt
```

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($5_{(d)}$)，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：

Registers

R0	x0000	0	
R1	x0000	0	
R2	x8000	-32768	
R3	x0000	0	
R4	x3102	12546	
R5	x0000	0	
R6	x0000	0	
R7	x0000	0	
PSR	x8002	-32766	CC: Z
PC	x3000	12288	
MCR	x8000	-32768	

Memory

x3100	xD108	-12024	
x3101	x0005	5	
x3102	x0000	0	

运行后结果如下：

Memory

x3100	xD108	-12024
x3101	x0005	5
x3102	x211A	8474

得到了预期结果。