

中国科学技术大学计算机学院
《计算机系统概论》报告



实验题目: Lab 02

学生姓名: 高楚晴

学生学号: PB18111688

完成日期: 2019.11.19

实验目的

使用十六进制编辑器，用机器码编写LC-3程序，制作一个1位算术右移计算器。该16-bits数存放在R0中，并且最终结果存放在R0中，R7的值在执行过程中不被改变。程序从x3000开始，以HALT指令结束。

设计思路

与左移可以直接通过一次加法实现乘2操作不同，右移也就是除2的计算较为繁琐，因此直接使用按位移动的方式完成。

其基本原理为：先用一个R1存放移位前的R0值，由一个x8000的最高位mask变量与原数值实行与运算实现算术右移的符号位保持功能。对于其他位，用R2和R3分别存放一个独热编码，分别用于检测原R0中该位是否为1和向R0的低一位赋值。若R0中当前检测位为0，则跳过赋值步骤，其中赋值由加法运算实现。

每次循环后R2通过加法实现左移一位，随后R3左移一位。当检测完R1（即R0原值）最高位后，R2从x4000变为x8000，此次循环对应R3产生溢出，此时利用其条件码为Z检测终止条件，直至程序结束。

代码讲解

16进制机器码

```
3000    ;    .ORIG x3000
1220    ;R1存放R0原本的数值
240B    ;R2起mask作用，保留最高位
5042    ;R0通过AND保持原来符号位，用于存放移动后的数
54A0    ;R2用于按位运算
14A1    ;对原数字从第二位开始
1682    ;R3对应移动前所处位数
0405    ;执行到最高位，结束
5643    ;按位运算，检测当前为是否为0
0401    ;当前位为0，跳过加法运算
1002    ;将移位后数字对应的低一位位置1（加法）
1482    ;移位操作向更高一位处理
0FF9    ;循环进行下一位的运算
F025    ;HALT
8000    ;x8000用于实现最高位截取
```

汇编语言

为直观显示，给出相同功能的汇编语言如下。

```
.ORIG    x3000
ADD R1,R0,#0    ;R1存放R0原本的数值
LD  R2,HIGH    ;R2起mask作用，保留最高位
AND R0,R1,R2    ;R0保持原来符号位，用于存放移动后的数
AND R2,R2,#0    ;R2用于按位运算
ADD R2,R2,#1    ;对原数字从第二位开始
LOOP1 ADD R3,R2,R2    ;R3对应移动前所处位数
```

```

BRZ FINAL
AND R3,R1,R3      ;按位运算，检测当前为是否为0
BRZ LOOP
ADD R0,R0,R2      ;将该位置1
LOOP ADD R2,R2,R2  ;移位操作向更高一位处理
BR LOOP1
FINAL TRAP x25
HIGH .FILL x8000
.END

```

调试分析

由于本次实验任务仅为对一个16-bits位宽的二进制数进行处理，因此时间复杂度总在常数时间内，在空间复杂度方面，本次实验动用了R0~R3四个寄存器，并额外用了一个地址空间存放x8000。

编译发现无错误。进一步进行代码测试。

```

Starting Pass 1...
Pass 1 - 0 error(s)
Starting Pass 2...
Pass 2 - 0 error(s)

```

代码测试

初次完成代码后进行调试，为了测试符号位保持是否正确，分别给出正负数测试案例各一，为了检验其鲁棒性，再用零进行测试，结果分别如下。

R0	x7392	29586	R4	x0000	0	PC	x300C	12300
R1	x7392	29586	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

正数:0111 0011 1001 0010(x7392) -> 0011 1001 1100 1001(x39c9)

R0	x39C9	14793	R4	x0000	0	PC	x300C	12300
R1	x7392	29586	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

负数:1000 1010 1001 1100(x8a9c) -> 1100 0101 0100 1110(xc54e)

R0	x8A9C	-30052	R4	x0000	0	PC	x300C	12300
R1	x8000	-32768	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

R0	xC54E	-15026	R4	x0000	0	PC	x300C	12300
R1	x8A9C	-30052	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

R0	x0000	0	R4	x0000	0	PC	x300C	12300
R1	x8A9C	-30052	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

R0	x0000	0	R4	x0000	0	PC	x300C	12300
R1	x0000	0	R5	x0000	0	IR	x0405	1029
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x6666	26214	CC	Z	

零

执行结果与理论值一致。

实验总结

通过本次实验了解了逻辑右移与算术右移的区别，练习了mask的用法，同时通过实际操作在理论课程讲述之前了解了TRAP指令对寄存器的改变，通过试图改变（但最终没有成功）HALT执行前对R7的修改在调试中对TRAP指令特别是HALT的存储和实现有了更好地掌握，在实验过程中，通过阅读LC3winguide掌握了在HALT之前打断点这种调试常用手段。

附录

PB18111688_高楚晴_Lab2.bin => 代码二进制文件