

ICS 实验一实验报告-LC-3实现

实验目的

- 熟悉lc-3汇编语言
- 掌握lc-3汇编语言模拟环境的使用
- 尝试使用LC-3汇编语言编程
- 掌握汇编语言编程的方法，使用机器语言编程

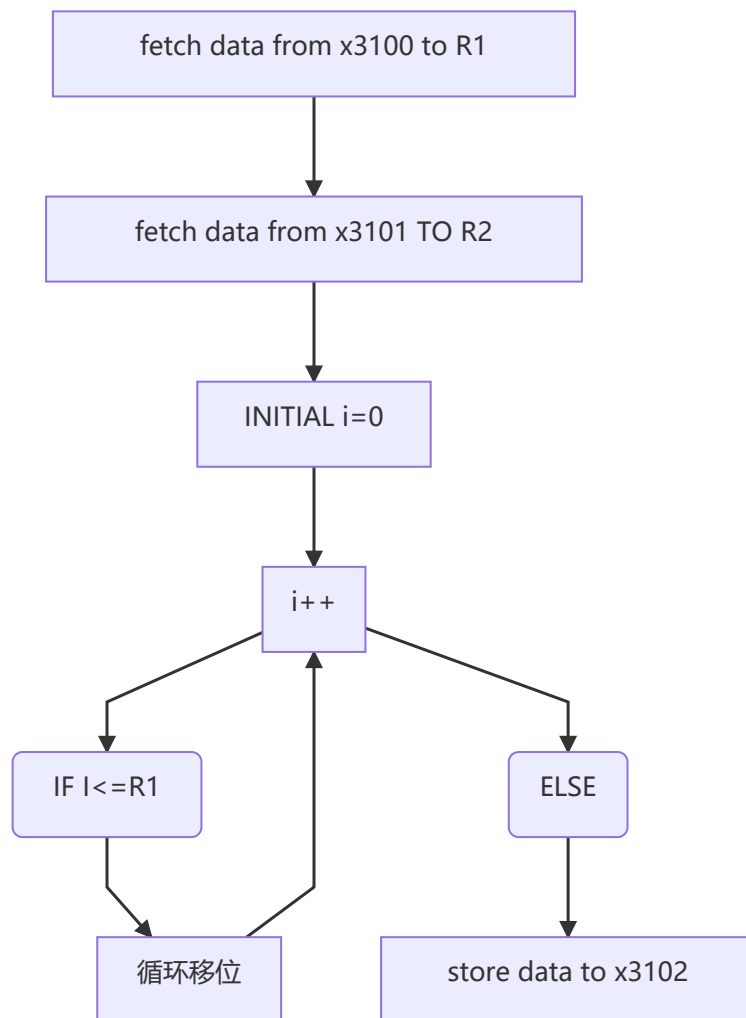
实验环境

- Windows10操作系统
- LC-3模拟器

实验题目分析

- 实验要求实现循环移位，但是LC-3不含有移位指令，这就需要我们利用已有的指令集的指令实现移位操作
 - 移位通过加法实现
- 循环移位可能需要在最低位补1
 - 如何检测是否补1
 - 判断操作数的正负
- 循环实现
 - 循环变量的递增，ADD
 - 循环变量和某个寄存器的值比较：采取异或逻辑运算，如果异或的结果为0，代表两个数相等，利用摩尔定律将异或转化为基本的与运算和非运算以便LC-3汇编指令处理
- 判断
 - 需要提前置标志位

算法概览



汇编语言程序设计

下面是汇编语言写成的代码，机器语言不容易让人直接理解，现在采用汇编语言展示设计

```

.ORIG x3000 ; start the program at location x3000
ADD R4,R4,R4
ADD R4,R4,R4
LD R1,#253 ; store the x3100
LD R2,#253 ; store the x3101
ADD R3,R3,#0 ; clear the R3
loop ADD R3,R3,#1 ; ADD R3
ADD R4,R1,x0 ; Judge whether R1<0
BRN ZERO
ADD R1,R1,R1 ; if R1<0
ADD R1,R1,#1 ; ADD 1'B1 IN THE LOW BIT OF R1
normal ADD R4,R2,x0
ADD R5,R3,x0
NOT R4,R4 ; R4=~R4=~R2
NOT R5,R5 ; R5=~R5=~R3
AND R6,R2,R5 ; R6=(~R3)&R2
AND R7,R3,R4 ; R7=(~R2)&R3
NOT R6,R6
NOT R7,R7
AND R6,R6,R7
NOT R6,R6

```

```

BRNP loop
ST R1,#236      ; store the data to Memory
HALT
ZERO ADD R1,R1,R1
ADD R1,R1,#1
BRNZP normal
.END

```

对一些没加注释的行加以说明

- 第二行，第三行加上两条奇怪的ADD指令是为了让指令到内存取数据不会超过立即数表示的范围
- 从normal到BRNP loop实际上都在实现的是异或运算，利用的是如下公式

- $A \oplus B = \overline{C}D$
- $C = \overline{A}B$
- $D = \overline{B}A$

结果验证

利用样例给出的例子验证，运行前

Registers				Memory			
R0	x0000	0		0	► x3100	xD108	53512
R1	x0000	0		1	► x3101	x0002	2
R2	x0000	0		2	► x3102	x0000	0
R3	x0000	0		3	► x3103	x0000	0
R4	x0000	0		4	► x3104	x0000	0
R5	x0000	0		5	► x3105	x0000	0
R6	x0000	0		6	► x3106	x0000	0
R7	x0000	0		7	► x3107	x0000	0
PSR	x8002	32770	CC: Z	8	► x3108	x0000	0
PC	x3000	12288		9	► x3109	x0000	0
MCR	x8000	32768		10	► x310A	x0000	0
Console (click to focus)				11	► x310B	x0000	0
				12	► x310C	x0000	0
				13	► x310D	x0000	0
				14	► x310E	x0000	0
				15	► x310F	x0000	0

运行后

Registers				Memory			
R0	x0000	0		0	► x3100	xD108	53512
R1	x7FFF	32767		1	► x3101	x0002	2
R2	x0002	2		2	► x3102	x4423	17443
R3	x0002	2		3	► x3103	x0000	0
R4	xFFFF	65533		4	► x3104	x0000	0
R5	xFFFF	65533		5	► x3105	x0000	0
R6	x2FFE	12286		6	► x3106	x0000	0
R7	xFFFF	65535		7	► x3107	x0000	0
PSR	x0002	2	CC: Z	8	► x3108	x0000	0
PC	x0263	611		9	► x3109	x0000	0
MCR	x0000	0		10	► x310A	x0000	0
Console (click to focus)				11	► x310B	x0000	0
				12	► x310C	x0000	0
				13	► x310D	x0000	0
				14	► x310E	x0000	0
				15	► x310F	x0000	0
				16	► x3110	x0000	0
				17	► x3111	x0000	0