

Introduction to Computer System Organization

SZU Review

Chapter9

LC-3 TRAP子程序总结

一、TRAP指令概述

TRAP是LC-3中用于调用系统子程序（服务例程）的指令，通过陷阱向量（Trap Vector）来指定不同的系统功能。

指令格式：TRAP trapvect8

- 操作码：1111 0000（高8位）
- trapvect8：低8位指定服务例程（范围0x00-0xFF）

二、常用TRAP子程序

LC-3预定义了以下常用TRAP例程：

陷阱向量	名称	功能	输入参数	输出结果
x20	GETC	读取单字符	无	R0 = ASCII字符
x21	OUT	输出单字符	R0[7:0] = 字符	屏幕显示字符
x22	PUTS	输出字符串	R0 = 字符串地址	显示字符串
x23	IN	提示并输入字符	无	R0 = ASCII字符
x24	PUTSP	输出字节字符串	R0 = 字符串地址	显示双字节字符
x25	HALT	停止程序执行	无	停机

三、TRAP执行流程

1. 保存状态：
 - 将PC值存入R7（返回地址）
 - 特权模式切换（若需要）
2. 查找服务例程：
 - 根据trapvect8查陷阱向量表（内存x0000-x00FF）
 - 读取对应地址的内容作为例程入口
3. 执行子程序：
 - 跳转到服务例程执行

4. 返回原程序:

- 通过RET (JMP R7) 指令返回

四、关键特性

1. 双重间接寻址:

- 陷阱向量→向量表地址→服务例程地址

2. 寄存器约定:

- 参数通常通过R0传递
- R7用于保存返回地址

3. 内存映射:

- 陷阱向量表固定在x0000-x00FF

五、示例代码

```
; 输出字符串
LEA R0, MSG      ; 加载字符串地址
TRAP x22         ; 调用PUTS
HALT             ; 停止程序

MSG .STRINGZ "Hello, LC-3!"
```

Attention:

调用TRAP会自动修改R7, 嵌套调用时需要保存R7

PUTS要求字符串以x0000 (NULL) 结尾

IN和GETC的区别:

IN会显示提示符并回显

GETC静默读取不显示

用户可扩展自定义TRAP例程 (需修改陷阱向量表)

关于子程序的两种常见模板,在Chapter7中已预先提到,此处不作额外说明.