指令系统

25个 ZJU 之

① 指令格式
② 数据类型和格式
③ 操作数存放方式
④ 程序可访问寄存器个数、位数和编号
⑤ 存储空间大小和编址方式
⑥ 寻址方式
⑦ 指令执行过程的控制方式等

指令系统（指令集）是 ISA 核心部分

指令格式

零地址 | OP |

- 无操作数指令：空指令、停机指令、转移等
- 堆栈计算机运算类指令

一地址 | OP | A_1 |

- 单操作数指令：$OP(A_1) \to A_1$，++、--、求反、求补等
- 双操作数指令（隐含的地址）

若指令长32位，OP 8位，A_1 24位，直接寻址范围为 $2^{24}=16M$

二地址 | OP | A_1 | A_2 |

$(A_1)\ OP\ (A_2) \to A_1$

OP 8位，A_1、A_2 12位

直接寻址范围 $2^{12}=4k$

三地址 | OP | A_1 | A_2 | A_3 |

$(A_1)\ OP\ (A_2) \to A_3$

直接寻址范围 $2^8=256$

四地址 | OP | A_1 | A_2 | A_3 | A_4 |

$(A_1)\ OP\ (A_2) \to A_3$

A_4 = 下一条指令地址

$2^6=64$

定长操作码指令集

OP n位 最大能表示 2^n 条指令

扩展操作码指令集

扩展原则：
- 短码不能是长码的前缀
- 操作码不能重复

指令操作类型

- 数据传送 mov、load、store
- 算术逻辑运算 add、sub、cmp、mul、div、inc（自增1）、dec（自减1）、and、or、not、xor
- 移多位 shift
- 转移 jmp、branch、call、ret、trap
- 输入输出

指令寻址方式
- 指令寻址
- 数据寻址

指令寻址（找下条要执行的指令）
- 顺序寻址 PC+1
- 跳跃寻址 转移指令

数据寻址

- 隐含寻址
- 立即数寻址
- 直接寻址 $EA=A$
- 间接寻址 $EA=(A)$ 一次间接
- 寄存器寻址 $EA=R_i$
- 寄存器间接寻址 $EA=(R_i)$
- 相对寻址 $EA=(PC)+A$
- 基址寻址 $EA=(BR)+A$ BR基址寄存器面向OS
- 变址寻址 $EA=(IX)+A$ IX变址寄存器面向用户

    基址寻址中(BR)不变，A变

    变址寻址中(IX)变，A不变

- 堆栈寻址

机器级代码

汇编指令格式
- AT&T 只能小写字母
- Intel 大小写不敏感

AT&T

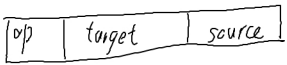
op	source	target

%reg $imm (mem)

disp(base, index, scale) 例 8(%edx, %eax, 2)

= M[edx + eax*2 +8]

Intel



reg imm [mem]

[edx + eax*2 +8] = M[edx + eax*2 +8]

常用指令 (Intel)

数据传送

mov

push ① esp-4 ② 压栈 } 过程调用

pop ① 出栈 ② esp+4 }

算术、逻辑运算

add、sub

inc、dec 自增1、自减1

imul 带符号整数乘

idiv 带符号整数除 edx:eax 为被除数

只有一个操作数 为除数

结果商运往 eax，余数运往 edx

and、or、xor

not 取反

neg 取负

shl、shr 逻辑左、右移

shl 操作数 移位位数

sal、sar 算术移位

控制流指令

jmp

je、jne、jz、jg、jge、jl、jle 条件跳转

cmp、test 设置 PSW

call、ret 过程调用、返回

过程调用机器级表示

调用者保存寄存器 eax、ecx、edx

被调用者保存寄存器 ebx、esi、edi

每个过程有自己的栈帧

选择语句机器级表示

根据PSW中的条件码作为选择依据

ZF、OF、SF、CF

cmp 等价于sub，但只改变PSW

test 等价于and，只改变PSW

循环语句机器级表示

使用条件测试+跳转组合实现

CISC和RISC

CISC 复杂指令系统计算机

如 x86

RISC 精简指令系统计算机

如 ARM、MIPS

	CISC	RISC
指令集	复杂	简单
指令条数	一般大于200条	一般小于100条
指令字长	不固定	定长
可访存指令	不加限制	load、store
各指令执行时间	相差较大	大多数一个周期内完成
各指令使用频率	相差较大	都常用
通用寄存器数量	少	多
目标代码	难以用优化编译生成高效目标代码程序	采用优化编译程序，较高效
控制方式	微程序	组合逻辑
指令流水线	可实现	必须实现