

指令集优秀；硬件设计方便，编译器编译方便，性能更高，成本更低

基本概念：①机器指令：机器能直接识别、执行的某种操作命令

②指令系统：一台计算机中所有机器指令的集合

③系列机：基本指令系统相同，基本系统结构相同的计算机（解决兼容问题）

设计要求：①完备性：指令丰富，功能齐全，使用方便

②高效性：程序空间小，执行速度快

③兼容性：系统向后兼容

④规范性：不同寻址方式支持

⑤一致性：不同数据类型支持

⑥统一性：指令长度和寻址长度一致

指令格式：二进制代码表示指令结构形式，包含以下信息：

→计算机执行什么操作 计算机处理什么数据

操作码字段 寻址方式 地址码字段

→计算机怎样得到待处理的数据

操作数字段 地址码AC → 寻址方式字段：长度与寻址方式种类有关，可能隐藏于操作码字段中

可能有多行

地址码字段：作用影响，长度和寻址方式有关

操作码字段 OP → 每条指令对一个操作码

长度决定指令系统规模

操作码 Length_{OP} = [log_n] 偏移

长度决定指令系统规模

操作码字段扩展 扩展

长度决定指令系统规模

扩展操作码

32位指令

双操作数

OP(8)

A(12)

A(12)

三种不得重叠，否则无法区分编码

→ 双操作数指针数为k，显然 k < 2⁸；除状态，2⁸-k个可用表示其他类型指令

单操作数

OP(20)

A(12)

→ 可用单操作数指针数最多(2⁸-k)·2¹², 2¹²是多条12位的组合

无操作数

OP(32)

2³²

指令字长度选择

→ 指令中包含二进制位数

长度越大→地址码越长，寻址空间越大，占用空间越大，取指越慢

与机器字长度有关（单字、双字、半字）

等长指令：结构简单，控制线路简单，MIPS指令

度长指令：结构灵活，充分利用指令长度，控制复杂

指令操作数长度选择

源操作数 目的操作数

三地址 A1 OP A2 → A3

二地址 A1 OP A2 → A1

单地址 OP A1 → A1

零地址 停机/空操作/关中断

寻址方式、寻找指令/操作数有效地址的方式

1. 指令寻址

①顺序寻址 Mem[PC+I] → IR

寻址方式 程序对应机器指令序列在内存顺序存放

从第一条开始逐条取出执行

实现方式 程序计数器PC对指令地址进行计算

PC存放下一条指令地址，初始值为程序首址

执行一条指令，PC=PC+当前指令字节长度

②跳转寻址（跳转、子程序调用、程序中断返回） IR(A) → PC

实现方式：出现分支、循环、调用，返回会改变程序执行顺序

实现方式：下一条指令地址不是PC+得到的，而是指令本身给出的

跳转的实现方式是重新修改PC的内容，进入取指阶段

2. 操作数寻址：形成操作数有效地址的方法

单地址码 OP I D E: 实际有效地址 S: 实际操作数 S=C(E)

寻址方式 → 开式地址

①立即寻址 ⇒ 地址码字段即为操作数本身 S=D 能快速获取操作数

②寄存器寻址 ⇒ 操作数在CPU内部寄存器中 E=D, S=R(D) 快速获取操作数；地址码短，有利于缩短指令长度

③直接寻址 ⇒ 地址码字段直接给出操作数在内存的地址 E=D, S=C(D) 需访存操作，速度较慢，寻址范围受限于开式地址寻址

④间接寻址 ⇒ D单元内容是操作数地址，D是操作数地址的地址 E=(D), S=(CD)

寻址范围大（由字长决定）；需两次访存，慢，已淘汰

⑤寄存器间接寻址 ⇒ 寄存器D单元内容是操作数的地址，D是操作数地址的地址 E=R(D), S=C(R(D)) 寻址范围大（寄存器张量），编程灵活；一次访存，速度快于间接寻址

⑥相对寻址 ⇒ 指令中的D加上PC的内容作为操作数的地址 E=DT(PC) 例：PC+D+1 执行ID

⑦基址/变址寻址 ⇒ 操作数地址为基址+变址量 基址寄存器一般不修改 E=DT[R(X)], S=CD+R(X)

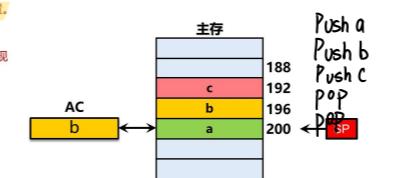
⑧堆栈寻址 ⇒ 堆栈值不变，偏移量

■ 寄存器堆栈

- CPU内部一组串联的寄存器
- 数据的传送在栈顶和通用寄存器之间进行
- 栈顶不动，数据移动，进出栈所有数据都需移动
- 速度快，需要硬件支持，栈容量有限

■ 存储器堆栈 进栈 (AC) → Mem[SP] → 出栈 Mem[SP+1] → AC

- 内存区间做堆栈
- SP---堆栈指示器(栈指针)，改变SP即可移动栈顶位置
- 栈顶移动，数据不动，非破坏性读出
- 栈容量大，栈数目容量均可自定义，软硬件均可实现



设某16位机定义寻址方式如下：

5bits	3bits	8bits	寻址速度与范围	
操作码	寻址方式	形式地址D	实地址E	寻址范围
立即寻址	MOV 000	38H	S=D	0~255 -128~127 第二
寄存器寻址	MOV 001	00	E=D	0~255# Reg 最快 形式地址
直接寻址	MOV 010	200	E=D	0~255 RAM Cell
间接寻址	MOV 011	200	E=(D)	0~2 ¹⁶ -1 RAM Cell 最慢
寄存器间接寻址	MOV 100	01	E=R(D)	0~2 ¹⁶ -1 RAM Cell 主存/寄存器宽
相对寻址	JMP 101	20	E=(PC)+D	PC-128~PC+127
变址寻址	MOV 110	20 100	E=R(X)+D	0~2 ¹⁶ -1 RAM Cell

变址寻址

↓ 变址量 寄存器偏移量

E=(R(X)+D)

例题见 PPT P19

MIPS (Microprocessor without Interlocked Pipeline Stages)

简单的 Load / Store 结构；易于流水线 CPU 设计；易于编译器开发

寻址方式简化；指令操作简单

→ 最快的数据单元，寄存器个速度↑，但面积有限

CPU 中的寄存器：通用寄存器/专用寄存器

MIPS 寄存器由 0~31 共 32 个 32 位通用寄存器；32 位单精度浮点寄存器 f0~f11；232 位乘商寄存器 H 和 L；

程序员 PC：无程序状态寄存器

汇编程序中，变量存放于寄存器/存储器中，本身为二进制数据，其数据类型运算由采用的指令决定

汇编程序中，变量存放于寄存器/存储器中，本身为二进制数据，其数据类型运算由