指令系统的发展和改进

两种途径和方向

CISC 复杂指令系统计算机 增强原有指令的功能以及设置更为复杂的新指令取代原先由软件子程序完成的功能，实现软件功能的硬化。 将更多的功能交给硬件实现 该方向，指令系统日益复杂和庞大 该放下具有三个优化方向 面向目标程序 面向高级语言 面向操作系统 增加复杂度原因 1.当高级语言取代汇编语言后，就不断增加新的复杂指令来支持高级语言程序的高效实现 2.由于访问主存的速度显著低于访问CPU寄存器的速度，因此在功能相同时不断增加功能复杂的新的指令来取代原先一连串指令完成的功能，将程序软件固化和硬化 3.系列机软件要求向上兼容和向后兼容，使得指令系统不断的扩大和增加，而原有指令又不能消除 RISC 精简指令系统计算机 通过减少指令种数和简化指令功能来降低硬件设计的复杂度，提高指令的执行速度 指令数量少，功能简单 CISC方向发展和改进 方法 增加新指令 增强指令功能，设置功能复杂的指令 增加寻址方式 增加数据表示 面向目标程序 面向目标程序的优化实现改进，就是对已有机器的指令系统进行分析，看哪些功能仍使用基本指令串实现，哪些功能改用新的指令实现，可以提高包括系统软件和应用软件在内的各种机器语言目标程序的实现效率。 该方法既能减少程序所占用的存储空间，减少程序执行时访存次数，缩短指令执行实现，提高程序允许速度又使实现更为容易。 途径 途径1 根据对已有机器的机器语言程序及其执行情况进行分析，统计各种指令及指令串的使用频度加以改进 静态使用频度 程序中统计出的指令及指令串使用频度 按照该频度改进指令系统，着眼于减少目标程序所占有用的存储空间 动态使用频度 在目标程序执行过程中，对指令和指令串统计出的频度 按照该频度改进指令系统，着眼于减少目标程序的执行时间 高频指令 对高频指令可增强指令功能，加快其执行速度，缩短其指令字长 低频指令 对低频指令考虑将其合并到某些高频指令中取 或在今后的系列机中取消 高频指令串 可以增设新指令取代，减少目标程序访存取指令的次数 加快目标程序的执行速度，有效缩短目标程序的长度 途径2 增设强功能复合指令来取代原先由常用宏指令或子村光绪实现的功能，由微程序解释实现，不仅大大提供运算速度，减少程度调用的额外开销，也减少了子程序占用的促主存空间 子程序：双倍长运算、三角函数、开方、指数、2-10进制转换、编辑、翻译等子程序 总结 用新指令替代常用指令串的办法实质上是尽量减少程序中的存、取、传送、转移、比较等不执行数据变换的非功能型指令的使用 让真正执行数据变换的加、减、乘、除、与、或等功能型指令所占的比例提高 面向高级语言 面向高级语言的优化实现改进就是尽可能缩短高级语言和机器语言之间的语义差距，支持高级语言编译，缩短编译程序的长度和编译时间 途径1 对源程序中各种高级语言语句的使用频度进行统计来分析改进，对高频语句增设与之语义差距小的新指令。 但是不同用途的高级语言之间高频使用的语句差异较大，难以做到对各种高级语言都进行优化 所以此方法只适用于用户所使用的高级语言，而且该优化是零碎的，局部的。 途径2 如何面向编译，优化代码生成来改进。 从优化代码生成上考虑，应当增强系统结构的规整性，尽量减少例外或特殊的情况和用法，让所有运算都对称、均匀地在存储单元（寄存器）间进行。 途径3 改进指令系统，使它与各种语言间语义差距都有同等的缩小。 对语言间共同的需求进行优化 途径4 采用让计算机具有分别面向各种高级语言的多种指令系统、多种系统结构的面向问题动态自寻优的计算机系统。 微程序的发展，特别是可写控存的采用，为这种动态结构的计算机的实现提供了可能 途径5 发展高级语言计算机，具有两种形式 让高级语言直接成为机器的汇编语言，通过汇编把高级语言原程序翻译成机器语言目标程序。 称为：间接执行的高级语言机器 让高级语言本身作为机器语言，由硬件或固件逐条进行解释执行，既不用编译也不用汇编。 称为：直接执行的高级语言机器 基本特点是：没有编译 面向操作系统 面向操作系统的优化实现改进的主要目的就是如何通过缩短操作系统与计算机系统结构之间的语义差距，进一步减少运行操作系统的时间和节省操作系统软件所占用的存储空间 途径1 通过对操作系统中常用的指令和指令串的使用频度进行统计分析来改进 改进效果有限 途径2 考虑如何增设操作系统专用的新指令 途径3 把操作系统中频繁使用的，对速度影像大的机构型软件子程序硬化或固化，改用用硬件或微程序解释实现 途径4 发展让操作系统由专门的处理机来执行的功能分布处理系统结构 分布式？ RISC方向发展和改进 CISC的问题 1.指令系统庞大，一般指令在200以上 2.许多指令操作繁杂，执行速度低 3.指令系统庞大，使得高级语言编译程序选择目标指令范围太大，难以优化生成高效机器语言程序，编译程序也太长，太复杂。 4.指令系统庞大，各种指令的使用频度都不太高，且差别大。其中一部分指令的利用率很低 80%指令只在20%的运行时间使用。 设计RISC基本原则 1.确定指令系统时，只选择使用频度很高的那些指令，再增加少量能有效支持操作系

统，高级语言实现及其他功能的指令。一般不超过200条 2.减少指令系统所用寻址方式种类，一般不超过2种。简化指令的格式限制在两种之内，并让全部指令具有相同的长度 3.让所有指令都在一个机器指令周期内完成 4.扩大通用寄存器数，一般不少于32个， 尽量减少访存。所有指令只有存、取指令访存，其他指令一律只对寄存器操作 5.为了提供执行速度，大多数指令都使用硬联控制实现，少数指令才采用微程序实现。 6.通过精简指令和优化设计编译程序，简单、有效地支持高级语言的实现。