系统结构 组成 实现

计算机系统结构：计算机系统的软、硬件的界面；即机器语言程序员所看到的传统机器级所具有的属性

从单纯依靠指令级并行转向开发线程级并行和数据级并行。

虚拟机：由软件实现的机器

程序员所看到的计算机属性，即概念性结构与功能特性

广义的系统结构定义：指令集结构、组成、硬件

分类

Flynn分类法、冯氏分类法

3，7

一 定量分析技术

1经常性发生事件；2amdhl定律（加快某一部件部分的速度）及依赖因素：性能改进的递减规则--整体性能评价；

3cpu性能评价（程序的总时钟周期/时钟频率）：三个参数

cpi（总时钟周期数/指令条数ic）：平均每条指令的时钟数 理想：小于等于1

指令集结构，组织，实现，生产等相关

cpu时间=

ic 不同的指令类型 cpi×ic求和

指令系统不同，软硬设计都不同

系统结构选择设计方案：条件分支（判断条件、目标地址转移）

a变为两条指令：设置条件码触发器--0/1（触发指令） 根据触发器进行转移（条件转移指令--执行速度快） ；多了触发器，程序代码变长（触发器0/1 两条指令），ic大，指令系统简单

b直接比较转移 ；一条指令执行时间长，cpi大，指令集规模大--硬件结构（电路规模）复杂--主频下降--时钟周期时间加长

4程序局部性原理 访问地址往往在一段连续空间，存储器地址非随机--将用信息即当前或附近信息

二性能评测

1执行时间与性能反比，吞吐率

执行时间多种定义：全部时间，cpu时间（不算等待时间，更多是硬件性能）--分为用户cpu、系统cpu时间。

2基准测试程序 用到各个硬件测到各方面 专门生产测试程序套件--SPEC系列 多版本

多语言多类型整数程序浮点程序……图形系统

事务处理性能测试基准程序--事务处理：数据库访问更新等

3比较 总时间、平均值、加权执行时间 数学、风湿

三发展

器件技术发展：cpu主频，现在难以再提高；转向系统结构发展（未来）：cpi提高周期减少

现在所有商用计算机：冯诺依曼系统结构 定义了计算机结构，不断改进但是仍在此框架下

新的突破，第五代计算机：非冯诺依曼 纳米……

存储程序计算机结构 1指令驱动：预先程序 存储程序 逻辑执行2冯特点：数据和指令放在同一存储器，不加以区别；一维编址；指令流产生控制流，控制流产生数据流--顺序到乱序流出执行完程序

指令集简单，硬件和编译简单很多

运算器为中心到存储器为中心

3,14

ch2 （easy 3lessons）指令集结构

1分类：

累加器：操作数和结果放在一个寄存器（与通用寄存器不同）

堆栈：只能从栈顶进行

和存储器有关的指令只有load store--在RR型中

隐含寻址：指令最短，无需操作数

累加器： loadA--放入累加器，addB隐含寻址，storeC--累加器内容放入存储器

指令字长，条数ic，快慢（访问时间cpi）都不一样。硬件……定性来分析。

对编译器而言，寄存器分配更容易有效例如A×B-C×D-E×F

寄存器地址比存储器地址短，字长短

ARM：设计系统结构的公司 苹果手机 ipad基于arm（RR） 苹果电脑基于x86（RM）

允许有在存储器上，但未必从存储器中取操作数。

通用寄存器分类依据：操作数个数及来源

优缺比较：指令执行周期尽可能相近的时候可以用流水线技术--RR

2寻址方式：

立即数寻址（load和alu使用频度很高，采用立即数寻址，立即数的取值范围8-16位，一般我们赋初始值都不大，不能再大，否则字长太长）和偏移寻址（4-8位来选择偏移量大小）最为经常

寻址方式也是软硬件相关，可硬件直接实现若无则编程序。

3功能设计：

很早时计算机系统结构设计就是指令集结构设计--主要。

设计的基本要求：规整性（对硬件的使用比较均匀，存储单元无特殊对待）

如果寄存器只有R7能够变址寻址，编译在寄存器分配时，将变量a放入R7寄存器，编译时发现R7只能用变址，需要把a转移，多余了指令。带来麻烦。

cisc：300-500条指令，格式丰富，长短不一，时间不一，功能复杂。发展到一定程度后，性能不能再增加。

risc：Patterrson（计算机系统结构的教材）

作业：计算机系统结构第二版张晨曦 王志英编著 1(5)名词解释任选五个，5，6，8

3,21

1根据cisc缺点设计risc--设计原则：RR 只有load store访问存储器，硬连逻辑，cpi小于等于1，指令集结构简单对编译优化--生成比较高效的目标代码--的要求比较高，流水线。

MIPS系列处理器--分支指令是条件寄存器方式--简单，高性能芯片--famous

共同点：123

2重点介绍控制类指令，比其他类型指令复杂，其他多相似

过程调用和过程返回，函数调用、子程序调用、过程调用都可以--无条件，只要目标地址

统计指令类型的使用频率--整形平均和浮点平均--分支指令（最麻烦，频率最高）--3种方式

讲cpi ic 定量分析时提到两种方式，x86系列条件码--寄存器若干位--涉及很多情况判断，根据运算结果的最后一步来设置各个位，无论后面有无条件判断，如果有则根据此寄存器判断。

--优缺：顺序严格，运算指令必须完成。

比较与分支：复杂

pc相对寻址方式：偏移量一般位数比较小4～8位

过程调用：一般直接寻址。与无条件不一样：1要保存返回地址--一个寄存器来保存或者压到堆栈，压到堆栈利用递归调用。2访问所调用的子程序的寄存器--如果和本程序已执行的一样，则需要保存好--三种方式，本程序保存寄存器或所调用程序保存寄存器或编译保存，不管谁来保存都会有多余--事先约定好。现代计算机都无保存，由编译来做。变量分配（寄存器地址分配）是编译来分配--生成目标代码时来保存。应用程序很少涉及因为是高级语言来编。

指令设计的四个方面：寻址方式，功能设计，操作数类型和大小，指令格式。

3操作数类型和大小：

数据表示--计算机不能识别则需要编译时将一条高级语言指令变成一段程序，所有数据类型中一子集由硬件直接实现，可直接识别--软硬分配：这个是系统结构问题，软硬分工的权衡

数据结构：数据类型的逻辑结构和物理结构关系以及算法

两种方式：操作码指定类型（现在大多是这种），类型多的话操作码编码复杂规模变大。

带标志符的数据表示：在数前带上数据描述符--动态检测操作数类型（少有）

操作数的进制和大小比较合适--统计比例

4指令格式

定长操作码，变长操作码Huffman

地址码：0123地址

寻址方式：两种：编码于操作码中间（操作码的功能就要求很高，适合寻址方式种类少时，loadstore时采取）或每个地址前带有地址描述符（加长字长，种类多，操作数个数多，否则编入操作码则操作码太复杂）

5考虑因素：1寄存器个数、寻址方式数目--平均字长和目标代码大小

2简单且便于硬件实现且表达信息多

3指令字长一般字节整数倍，一般比机器字长一样或短不会长

编码格式三种方式：定长（risc）、可变（奔腾系列1～12字节）、混合（平衡目标代码长度和译码时间，规定为几种，差异不太大）--现代都是这三种中某种

6MIPS指令集结构

64位 R0～R31 32个通用寄存器 32个浮点寄存器

利用某几种寻址方式创造其他方式

按字节编址 边界对齐

一个数据访问两次比较慢，要求同一数据只能放在一个单元不能跨存储单元，但是浪费较大

现代很多采用后一种，存储空间资源不再如当初稀缺。空间换时间，保证访问速度。这种信息存放叫信息的整数边界原则。--必须是字节宽度的整数倍 再写入时先判断是否能够是倍数 地址不断+1+1

操作码：指出操作码类型和寻址方式?

i类指令（？有不用的）

load rs+immeidate -->rt

store

分支 寄存器相对寻址rs(条件)+immeidate rt不用

R类指令

J

7ch3 流水线 主要cpu

顺序执行，下一章乱序执行--判断的原则算法以及硬件实现

向量--数组 向量处理机

流水线概念 浮点加法流水线，ALU上，访存上 不仅用在cpu设计上

级数

时空图