第一章 系统结构基础

※本章主要内容

(1) 计算机系统结构的概念

计算机系统的层次结构，系统结构的定义、分类

(2) 计算机系统的性能评测

性能指标，性能评测(程序/测试/比较)，成本与价格

(3) 计算机系统的设计

设计步骤、设计思路，

定量原理(采用并行性、局部性原理、关注经常性事件、Amdahl定律等)

(4) 计算机系统结构的发展

冯·诺依曼结构及其改进，影响系统结构发展的因素，

系统结构结构中并行性的发展

※总体要求

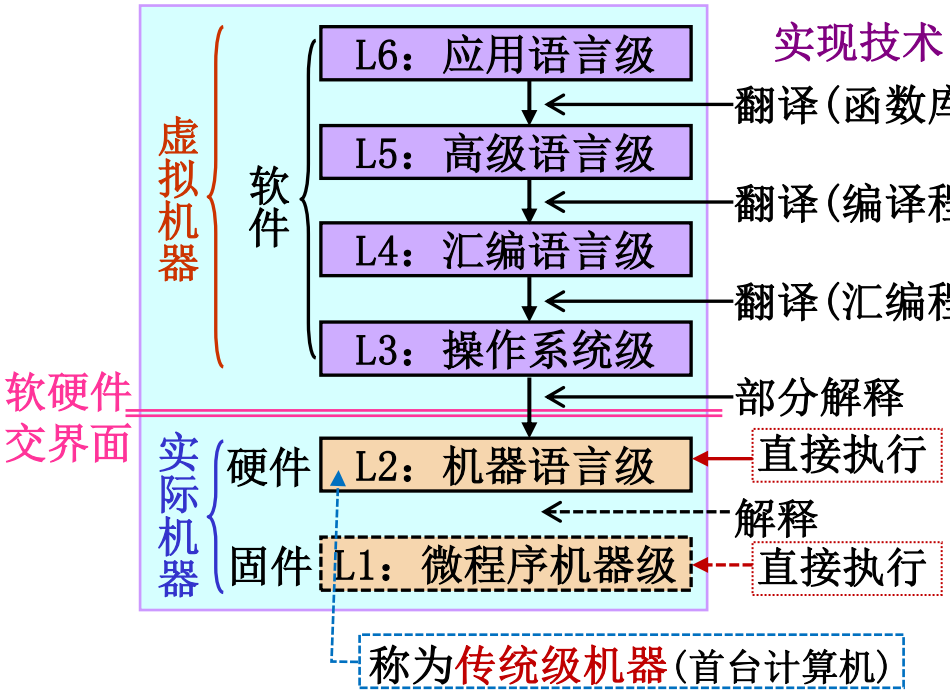
了解系统结构概念，理解量化分析原理，了解并行性开发途径

第1节 系统结构的基本概念

※主要内容：计算机系统的层次结构，系统结构的定义、分类

一、计算机系统的层次结构

计算机系统＝软件＋硬件/固件



思考①：程序员眼里的计算机属性？

语言处理程序(编辑器+编译器+链接器)

＋运行时系统(人机接口+运行器)

[加载&动态链接&执行引擎]

思考②：操作系统级机器的功能是？

(CPU+存储+设备+文件)管理＋人机接口

[软硬件资源]

思考③：为什么会有微程序级机器？

思考④：翻译和解释的特点？能统一吗？

思考①：编程环境(编辑器+编译器+链接器)、运行环境(人机接口+运行器[加载+译码+动态链接+执行引擎])

思考②：OS是计算机系统软硬件资源的管理机构(平台)，为高级语言的使用和实现提供所需的基本操作和数据结构(传统机器所没有的)，故作为一级虚拟机器存在。

思考③：指令系统越来越大，指令功能逐步变强，执行控制日益复杂，通过微程序可简化控制

思考④：解释执行时间长、可移植性好，翻译执行相反；不能，不同需求的性能指标不同

二、计算机系统结构(Computer Architecture)的定义

计算机设计者的任务—指令集设计，功能的组织、逻辑设计与实现

任务达成度的评价—计算机的性能及成本

1、系统结构的定义

*经典定义：　--Amdahl提出

传统级机器程序员(机器语言)所看到的计算机属性，即概念性结构和功能特性 → 即ISA(Instruction Set Architecture)

实质—确定软/硬件的交界面

*广义定义：　--J. L. Hennessy提出

指计算机的结构，包含指令集结构(ISA)(经典CA)、组成(CO)(微CA/宏CA)、硬件(CI)(实现)

宏CA—指多处理机的节点互连、存储器访问、节点交互

依据—①CA设计者关注计算机的性能与成本

②性能与成本涉及计算机的组成及实现

如：执行方式、MEM结构、总线结构、主频等

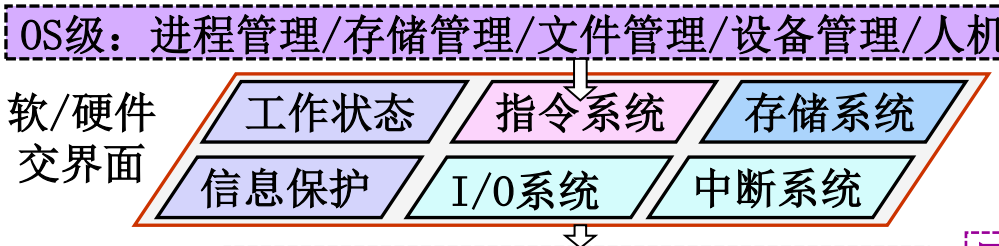
CA设计者就是计算机设计者

思考：串行/流水执行方式∈CA吗？

2、ISA包含的内容

***概念性结构：** 主要为指令系统，涉及硬件子系统、软件管理等

└→程序(指令序列)←┴←结构&原理　　└←面向OS



格式：

OP_i	F_1	A_1	F_2	A_2

功能：$A_D \leftarrow (A_1')\,OP\,(A_2')$

思考：硬件如何触发软件运行？

中断方式执行相应程序

***功能特性：**

指令系统
- 数据表示— 硬件能直接识别和处理的数据类型
- 寻址方式— 数据的最小寻址单元、寻址种类及地址计算规则等
- 指令功能— 指令的数据操作类型、下条指令地址计算方法等
- 指令格式— 表示指令功能的指令字格式及各信息的编码方式

硬件子系统
- 存储系统— 主存的编址单位、地址空间，层次结构等
- I/O系统— 外设的数据传送控制，I/O操作及状态表示等
- 中断系统— 中断&异常的类型、响应机制等

软件管理
- 机器工作状态— 系统态/用户态的定义与切换
- 信息保护— 保护方式、保护机构

示例1—不了解中断系统的响应机制，中断处理程序中带底纹指令不会写

示例2—不了解MEM的层次结构，进程切换代码中TLB及Cache冲刷会忘记

```
SERV_8253 PROC   ; 8253中断处理程序
  PUSH AX        ; 保护现场
  STI            ; 开中断，允许中断嵌套
  IN AL,60H      ; 设备服务
  ……
  CLI            ; 关中断，禁止INTR请求
  POP  AX        ; 恢复现场
  IRET           ; 中断返回
SERV_8253 ENDP
```

3、结构(ISA)与组成及实现的关系

*计算机组成(CO)： 是ISA的逻辑实现

包含内容—CPU、存储系统、总线结构等的组织(设计)

如：部件功能、控制机构、排队与缓冲、预估与预判技术等

*计算机实现(CI)： 是计算机组成的物理实现

包含内容—部件物理结构、器件使用、组装技术等

*三者关系：关系—硬件的总体框架→逻辑实现→物理实现

要求—结构：组成：实现的方案数＝1：m：m*n

示例：	ISA	CO	CI
乘法功能	是否有乘法指令	1位/阵列乘法器	芯片及电路
主存系统	容量&编址方式	速度&措施	器件&电路
总线系统	带宽&时延	线数&传输控制	介质&线距

三、系统结构的分类

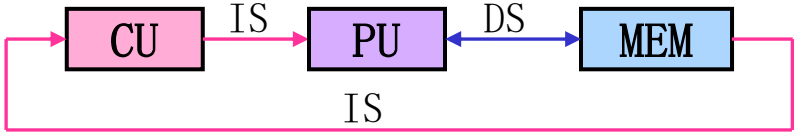
└←指计算机系统的结构，包含ISA、CO及CI

1、弗林分类法

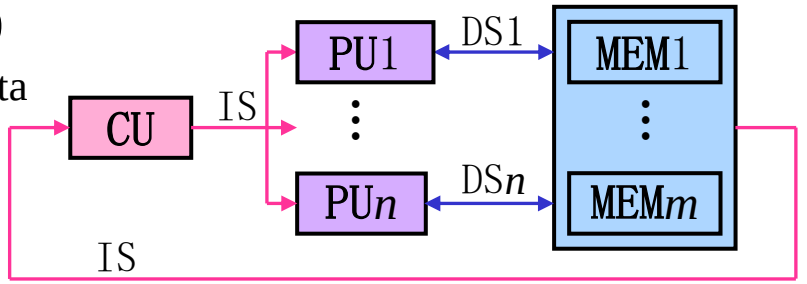
*分类方法：按指令流和数据流的多倍性分类

*分类结果：SISD、SIMD、MISD、MIMD

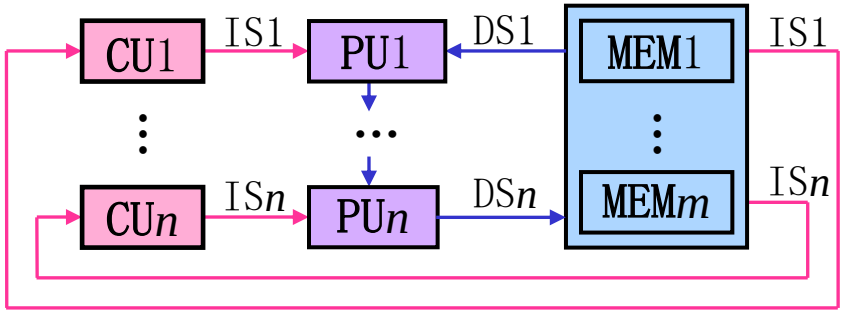
└←Single Instruction Multiple Data



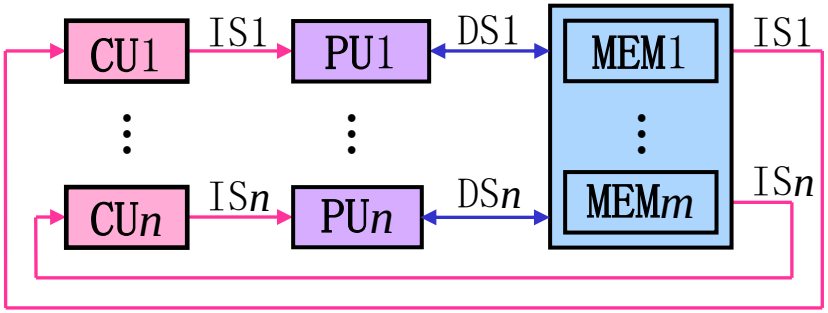
SISD结构(串行处理机)



SIMD结构(并行处理机)



MISD结构(专用处理机)



MIMD结构(多处理机)

*特点：对流水线处理机分类不明确

2、冯氏分类法

*分类方法：按最大并行度分类

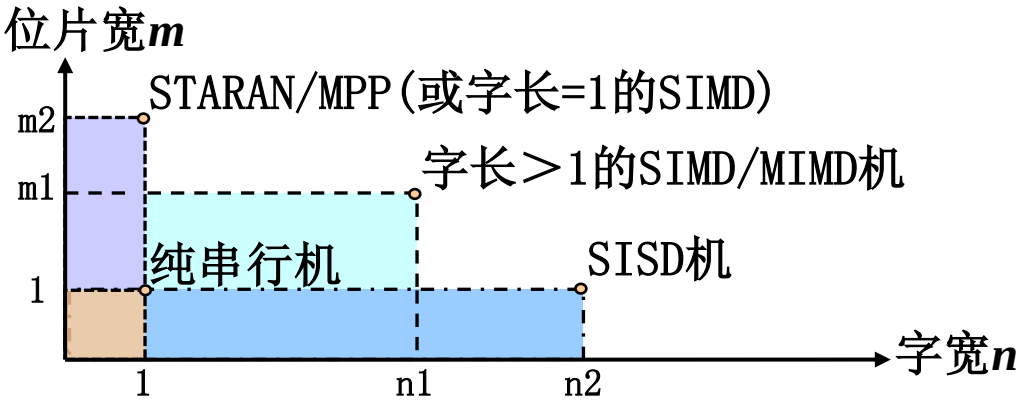
最大并行度—单位时间内能处理的最大二进制位数

即 P_m＝位片宽m×字宽n

一个芯片同时处理的字数→　←一个字中可处理的位数

*分类结果：

字串位串	字并位串	字串位并	字并位并
(n=1，m=1)	(n>1，m=1)	(n=1，m>1)	(n>1，m>1)
(早期纯串行机)	(传统单处理机)	(位级并行机)	(全并行机)
例— EDVAC	8086	STARAN及MPP	Core2



*特点：对流水线分类不明确

3、汉德勒分类法

*分类方法：在3个层次、按并行及流水处理程度分类

层次— PCU（程序控制部件或宏流水） K级

ALU（算逻部件或指令流水） D级

BLC（位级电路或操作流水） W级

描述— T(C)＝<K×K’，D×D’，W×W’>

其中：K为PCU数、K’为宏流水级数(所含PCU数)，K’≤K

D为每个PCU中ALU数、D’为指令流水级数(所含ALU数)，

W为ALU宽度，W’为操作流水线级数(所含ELC套数)

例：某机采用双核CPU，仅1套I/O接口(串行I/O)，每个核含12个ALU级部件，可构成6级指令流水线，字长为64位，部件最多可实现1~16级操作流水，该机的结构可表示为<2*1,12*6,64*16>。

*特点：对并行及流水线的程度有清晰的描述

注：流水采用时间重叠策略，并行采用资源重复策略

第2节 计算机系统的性能评测

※主要内容：性能指标、性能评测，成本与价格

一、性能指标

1、响应时间

外存-主存间I/O时间

指任务从提交→完成的总时间，即 $T_{响应} = T_{CPU} + T_{等待}$

其中，$T_{CPU} = I_N \times CPI \times T_C$，$T_{等待} = T_{I/O} + T_{OS} + \cdots$

*影响 T_{CPU} 的因素：$T_{CPU} = I_N \times CPI \times T_C = I_N \times (p + m \times k) \times T_C$

其中，p—处理时延/指令，m—访存次数/指令，k—访存时延

处理器

系统属性 \ 性能因子	I_N	p	m	k	T_C
指令系统	√	√			
编译技术	√	√①	√②		
PE实现与控制技术		√			√
Cache和内存层次结构				√③	√

思考：表中①②③的原因？

*特点：可反映系统的总体性能，但不易测量(每次可能不同)

思考：①i*4可用乘法或左移指令实现，②变量和放在REG或MEM中，③受映射方式、替换算法、写策略影响

2、吞吐率

指单位时间内能够处理的工作量，即 $TP = (\sum W_i) / T_n$

其中，W_i—任务i的工作量，T_n—完成n个任务的总时间

*表示：常用MIPS、MFLOPS代替 ←工作量的定义未能统一

MIPS(每秒百万次指令)—

$$\text{MIPS} = \frac{\text{程序中指令条数}}{\text{程序执行时间}\times 10^6} \rightarrow \frac{1}{\text{CPI}\times T_C \times 10^6} = \frac{\text{时钟频率}}{\text{CPI}\times 10^6}$$

缺点：不能反映指令功能强弱 (→常用相对MIPS方法)

MFLOPS(每秒百万次浮点运算)—

$$\text{MFLOPS} = \frac{\text{程序中浮点操作次数}}{\text{程序执行时间}\times 10^6}$$

缺点— 仅能反映浮点操作能力

*特点：可反映系统的多任务性能，但无统一指标(ISA不同/不全面)

3、加速比

指现行条件下执行速度相对于基准条件的比值，即 $S = T_0 / T_n$

例如，流水方式相对于串行方式，$S = [n \cdot m\Delta t] / [m\Delta t + (n-1)\Delta t]$

又如，多机系统相对于单机系统，$S(p,n) = T(p,1) / [T(p,n) + H(p,n)]$

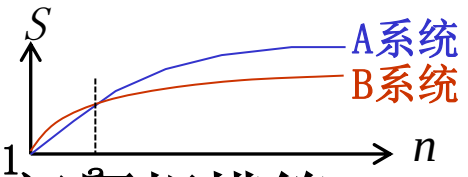
p—问题规模，n—处理器数量，H—通信时间

*特点：利于设计优化的性能评价，但不是绝对值(受参考机影响)

*可扩放性：指系统性能随处理机数n的增长比例，$\Psi = f(n)$

衡量方法— 测量不同n时的加速比

(得到性能可扩放性曲线)



影响因素— 结构、处理机数、存储系统、问题规模等

4、其他指标

效率，RAS(可靠性/可用性/可维护)等

可用性=MTTF/(MTTF+MTTR)，MTTF为平均无故障时间，MTTR为平均修复时间

※性能指标的局限性—不易测量，或无统一标准

二、性能评测

需求—不同应用(如桌面/服务器/嵌入式)关注不同性能(如计算/IO)

需使用不同测试程序←┘

1、基准测试程序

目标—确定不同应用公认的测试程序

***测试程序种类：**

程序类型	程序内容	测试目标	可信度
核心程序	真实程序中的关键代码段	部件	低
小型测试程序	几十行的代码段		
合成测试程序	模拟实际应用特征的程序	系统	
真实的程序	测试各方面性能的程序	系统	高

***基准测试程序：**

由多种真实程序组成、行业认可的套件，涉及多个方面

(如CPU/图形/IO)

例：SPEC基准测试程序集，可面向桌面系统、服务器

2、性能测试

目标—得到面向设计中、已实现系统的有效测试方法

*模拟技术：　　--可面向设计中系统

应用所需的程序

方法— 建立模拟器，模拟系统模型、工作负载模型；

运行模拟程序，获得模拟的性能

如：Cache的$T_A=T_{命中}+F*T_{缺失}$，$T_{命中}=T_{取标记}+T_{比较}+T_{访问}+\cdots$，

根据设计方案，给定每个参数的值，可计算出性能

特点— 需借助分析技术实现

*测量技术：　　--面向已实现系统

方法— 运行基准测试程序(多个)，测量实际的性能

3、性能评价

目标—执行多个测试程序后，得到归一化的结果(相对值)

示例	计算机X	计算机Y	计算机Z
执行程序A(s)	20	10	40
执行程序B(s)	40	80	20

*方法一：总执行时间 --综合度量标准

$S_n = \sum T_i$ 或 $A_n = (\sum T_i)/n$，其中n是测试程序数，T_i为执行时间

问题—工作负载中各程序所占比例可能不同

*方法二：加权执行时间 --精准度量标准

$A_n = \sum W_i T_i$，其中W_i是程序i在测试程组中的权重

问题—工作负载中各程序的执行频度无法给出

(无法统一)

*方法三：归一化执行时间 --相对度量标准 （假设性能是执行速度）

算术平均值法— $A_m = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{T_i}$，结论随参考机而改变

几何平均值法— $G_m = \sqrt[n]{(\prod_{i=1}^{n} R_i)} = \sqrt[n]{(\prod_{i=1}^{n} \frac{1}{T_i})}$，与参考机无关，无法量化

调和平均值法— $H_m = n \Big/ \left(\sum_{i=1}^{n} (1/R_i)\right) = n \Big/ \left(\sum_{i=1}^{n} T_i\right)$，较为精确(以时间为标准)

示例	以X归一化			以Y归一化			以Z归一化		
	X	Y	Z	X	Y	Z	X	Y	Z
程序A	1.00	0.50	2.00	2.00	1.00	4.00	0.50	0.25	1.00
程序B	1.00	2.00	0.50	0.50	1.00	0.25	2.00	4.00	1.00
算术平均值A_m	1.00	1.25	1.25	1.25	1.00	2.13	1.25	2.13	1.00
几何平均值G_m	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
调和平均值H_m	1.00	0.89	0.80	0.80	1.00	0.47	0.80	0.47	1.00

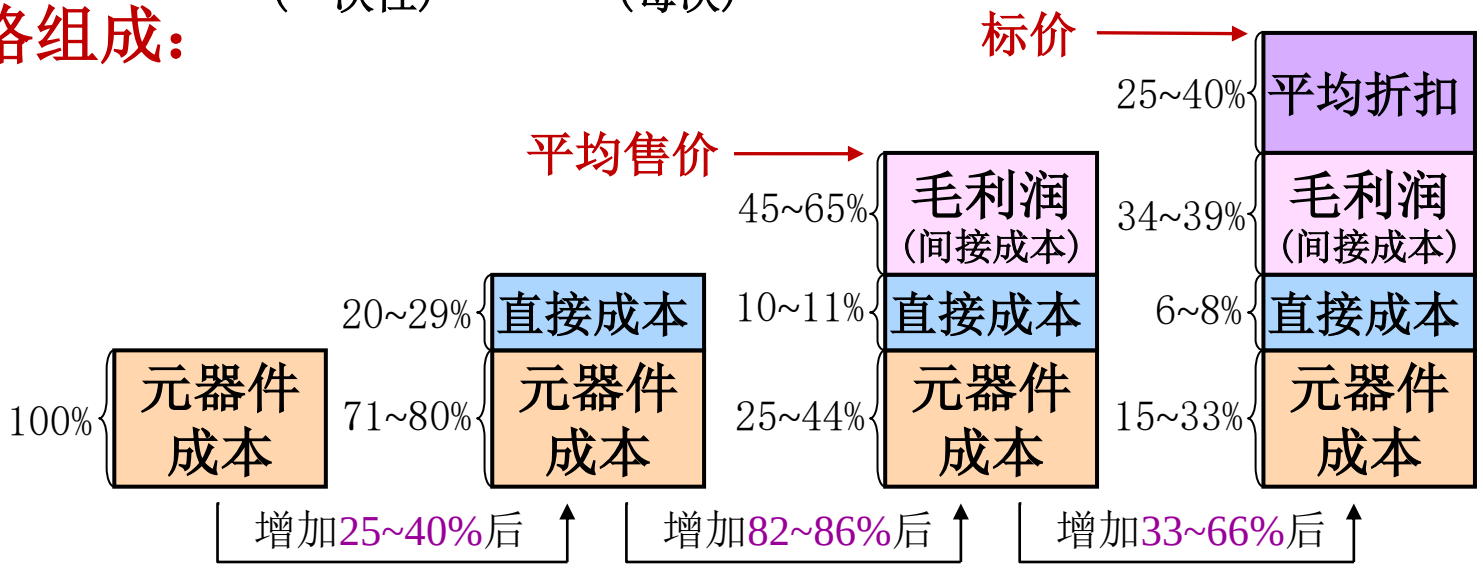
※性能评测的局限性—不同应用的性能无法归一化(程序不同)，
同类应用的性能仅作为参考(非标准化等)

三、成本与价格

*成本组成：开发成本＋生产成本，软/硬件不同

（一次性）　　（每次）

*价格组成：



直接成本—劳务、采购、报废、质量成本(人员培训、生产管理等)

间接成本—研发、市场、销售、设备维护费用，房租、财务、税务及利润

*成本-价格关系：元器件成本(占价格1/6~1/3)影响其他成本，

开发费用(占收入4%~12%)需持续投入

第3节 计算机系统的设计

※主要内容：设计步骤、设计思路，定量原理

一、系统结构设计的步骤 (类似于软件工程)

(1)需求分析

进行需求分析—包括应用环境、语言、OS、外设等方面，及技术经济指标、市场分析等

形成需求说明—设计准则、功能说明、器件性能说明等

└→造价/可靠性/可扩展性/兼容性/速度等方面

(2)具体设计

进行软/硬件功能划分，确定机器级界面(ISA)；

组织(设计)机器级界面的各个方面功能，可考虑几种方案

(3)评价及优化

二、计算机系统设计的思路

*由上向下方法：软件→硬件，适合专用机的设计

特点— 周期长(好几年)，忌需求变化，

不能利用最新软件技术 →形成软、硬脱节

*由下向上方法：硬件→软件，适合通用机的设计

特点— 周期长(好几年)，不能利用最新硬件技术，

软件效率低 →形成软、硬脱节

*从中间开始方法：软/硬件交界面→软件及硬件

特点— 周期短(约1/2)，

能够利用最新软、硬件技术 →主流设计方法

要求— 不断交互、优化设计， →需好的评价工具及方法

对设计人员技术要求高

三、计算机系统设计的定量原理

1、充分利用并行性 --最有效的优化方法

*基本思想：通过并行性来提高性能

*应用举例：指令流水线(操作并行)，SIMD技术(数据并行)，
多核CPU(线程并行)，RAID(I/O并行)

RAID—Redundant Arrays of Independent Disks

2、程序的局部性原理

程序访问局部性—指令和数据访问所呈现出的相对簇聚现象

例— for (int i=0; i<100; i++) Sum=Sum+A[i];

*基本思想：用最近使用信息预测将要使用信息，来提高性/价

*应用举例：层次结构存储系统，替换算法组织

3、重点关注经常性事件 --最重要、最常用的方法

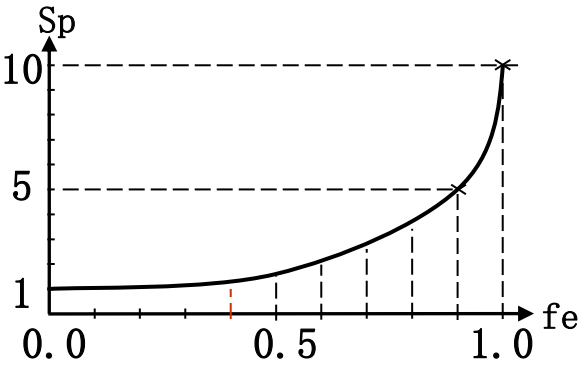
*基本思想：经常性事件优先获得处理权及资源使用权

*应用举例：指令操作码采用扩展编码，高频率指令的CPI较小，优先提高运算不溢出时的性能

4、阿姆达尔(Amdahl)定律 --优化设计的评价方法

*基本思想：优化某部件所获得的性能提高，受限于该部件的使用频率(或所占执行时间的比例)

$$S_p = \frac{\text{改进前运行时间}}{\text{改进后运行时间}} = \frac{T_0}{T_e} = \frac{1}{(1-f_e)+f_e/r_e}$$



例：若ALU操作占程序执行时间的40%，将其加快到10倍后，则程序的执行速度提高S_p＝1/(0.6＋0.4/10)＝1.56倍

*应用举例：预测优化性能(%已知)，寻找瓶颈(%未知、S↑急剧时)

5、利用CPU性能公式

CPU性能公式— $T_{CPU} = I_N \times CPI \times T_C = \sum (I_i \times CPI_i) \times T_C$，

$$CPI = \sum [(I_i / I_N) \times CPI_i]$$

分量特性： I_N ～ISA及编译技术，

CPI～组成及ISA，

T_C～器件及组成

*基本思想： 优化CPU性能公式的某分量

注意：优化时尽量不影响其它分量，否则需测评

*应用举例： 优化指令系统，比较设计方案(分量易测量)

Amdahl定律的%不易测量→┘

第4节 系统结构的发展

※主要内容：冯·诺依曼结构改进，CA发展影响因素，并行性发展

一、冯·诺依曼结构及其改进

1、计算机模型

*硬件结构：5大部件，以运算器为中心

*软件组成：指令序列(用逻辑地址表示)，执行顺序由指令类型决定

*工作方式：程序及数据预先存放在存储器中，
机器工作时，自动、逐条地取出指令并执行

2、性能瓶颈

①指令执行：串行执行方式(下条指令地址由当前指令产生)

②MEM访问：访存频率高(REG容量小)、用逻辑地址访问(便于编程)

③I/O方式：I/O与运算串行(受限于早期模型)

3、结构改进

*优化CPU结构：采用流水线、数据流、多线程技术(指令级并行)，
采用多核CPU技术(线程级并行)

*改进存储系统：采用层次结构(降低T_A)、哈佛结构(并行访问)，
采用多体交叉MEM(提高带宽)，
采用虚拟MEM(优化管理/保护)、优化地址变换
(不访存/变换-访问并行)

*改进I/O方式：采用中断(减小开销)、DMA(IO与运算并行)I/O方式，
提高总线性能(增加宽度&级数/优化传输模式)，
采用网络互连(并行I/O)

*改进指令系统：采用RISC技术(利于并行处理)，
增加数据表示、指令功能(提高代码效率/执行速度)

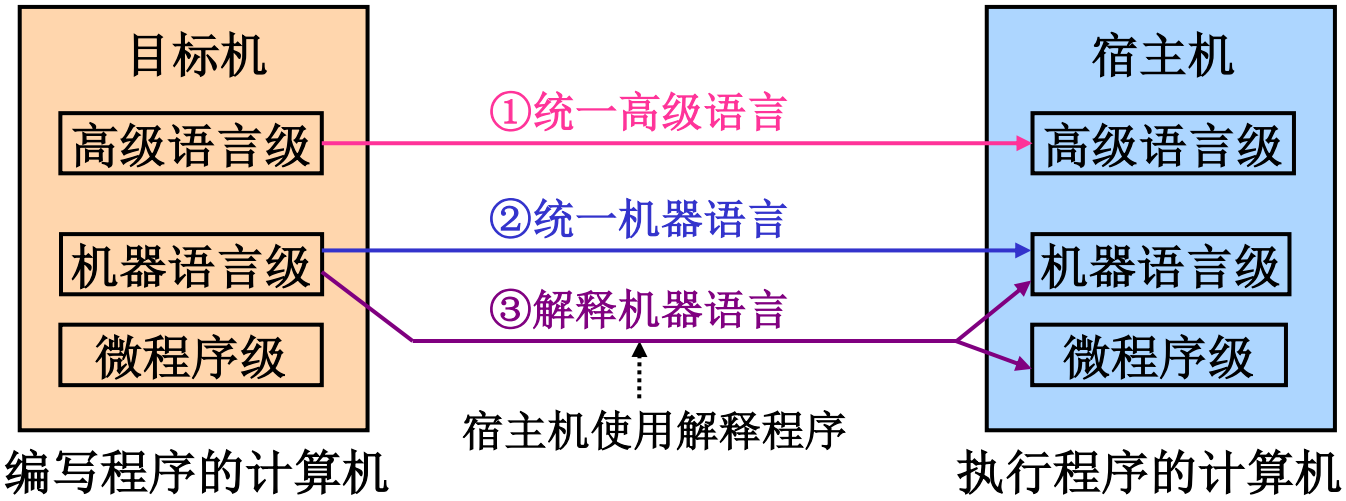
二、影响系统结构发展的因素

1、软件对系统结构的影响

***影响因素：** 主要为软件可移植性

└←以前的程序，可在新的计算机上运行

***解决方法：** 通常有三种

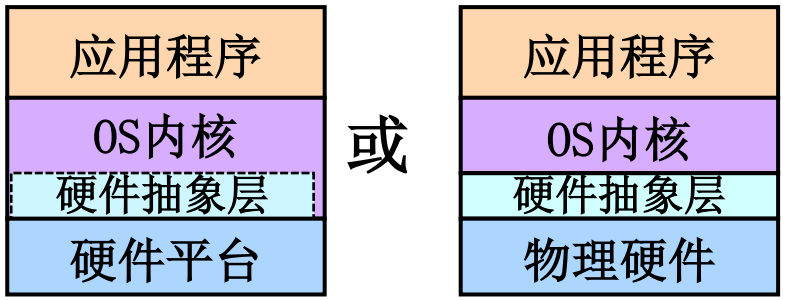


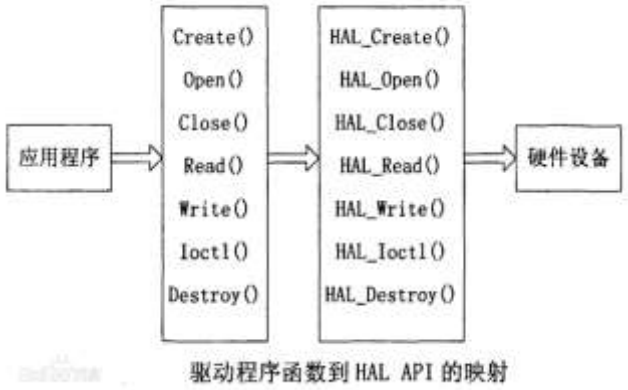
(1)统一高级语言

存在一定困难，可争取汇编语言或接口/技术的统一

*方案1：采用统一的中间语言(如Java)，可通过解释执行以适应不同的系统结构　(解释执行的效率不太理想)

*方案2：采用标准的开放系统(具有可移植性/交互操作性)，用硬件抽象层技术适应不同的系统结构　(要求接口基本保持不变)





驱动程序函数到 HAL API 的映射

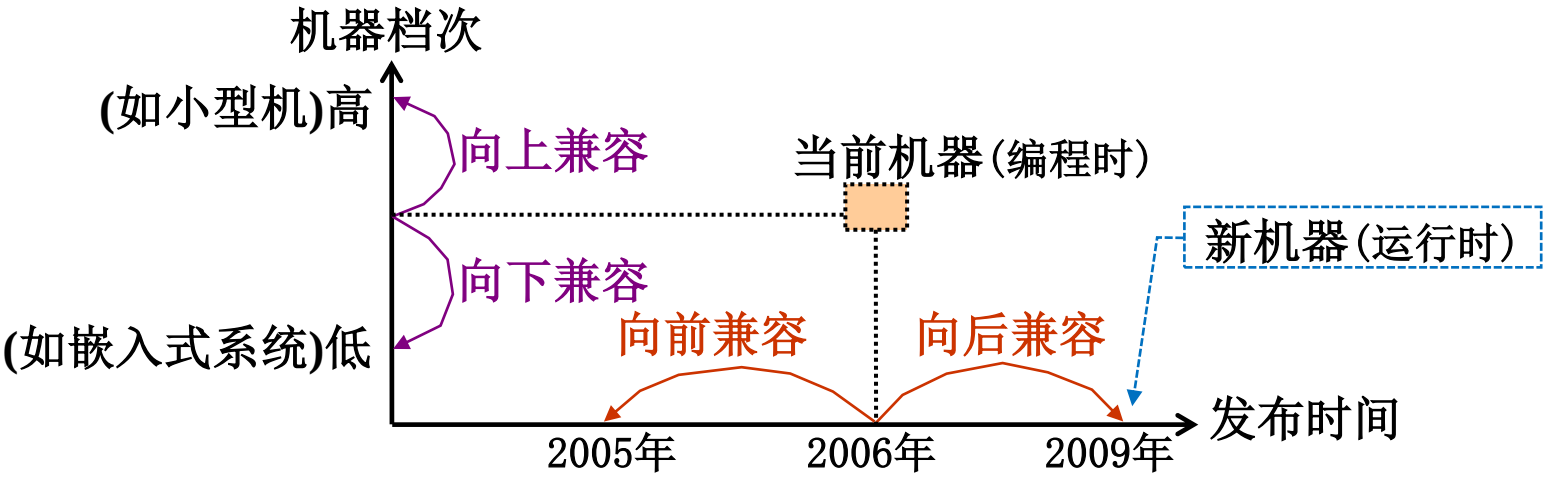
*影响：对系统结构的发展无限制，

要求相关功能的接口一致(→发展/效率受限)

(2) 系列机

通过保持或扩充系统结构，来实现软件可移植性

软件兼容的种类— 向上/向下、向前/向后



对系列机的要求— 保证向后兼容，力争向上兼容

*影响1：可使用新的组成及实现技术，推动了系统结构的发展

（出现兼容产品） 如486DX2的性能是486DX的2倍

*影响2：要求系统结构基本不变，限制了系统结构的发展

如：IA16(117+16)→IA32(+40+3+SSE)→IA64(+EM64T)，只增不减、效率不高

(3)模拟与仿真

通过软件(程序)解释目标机，来实现软件可移植性

*模拟：用机器语言程序解释目标机 ←放在宿主机主存中

└←指令系统/存储系统/IO系统/OS等

*仿真：用微程序解释目标机 ←放在宿主机CU中

*比较：

不同点— 解释程序的语言及存放位置、硬件是否参与解释

相同点— 解释内容含指令系统、存储系统、I/O系统、OS

*影响：对系统结构的发展无限制，

模拟/仿真的系统性能不佳(性能受限)

2、应用对系统结构的影响

*应用对系统结构的要求：性能好、效率高、价格低

现状—不存在对所有应用都高效的计算机

*系统结构的设计：

结果—专用结构：性能好、效率高，价格较高(~市场规模)

通用结构：性能、效率略低，价格较低(~市场定位)

方法—基于需求，选择一种性能&价格较优的设计方案

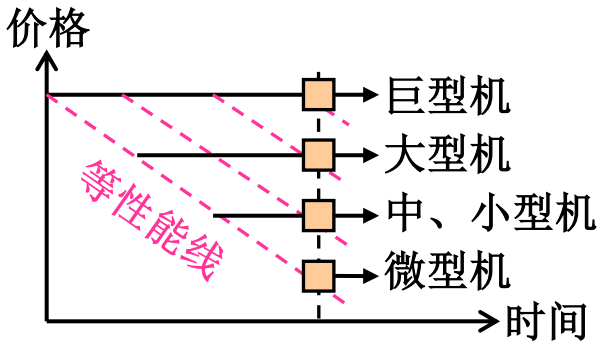
└→结果随时间变化[技术进步→器件发展→价格]

常见选择：

保持价格基本不变，提高性能；

保持性能基本不变，降低价格

*影响：会促进系统结构的发展



3、器件对系统结构的影响

***器件与系统结构的关联：**

器件的性能—决定新结构使用的可能性　　(如Cache)

器件的功能及使用方法—影响系统结构的性能及组织

└→通用片→现场片→半用户片→用户片

器件的性/价—加速了结构的下移速度　　(如SSE)

器件的发展—推动了算法、语言的发展　　(如VHDL)

***影响：** 会推动系统结构的发展

三、系统结构中并行性的发展

1、并行性

指同一时刻或同一时段内完成≥2种工作的特性

特征—有同时性、并发性

(1)并行性的等级划分

类型—有执行程序(功能并行)、处理数据(数据并行)

表示—用并行性粒度(即计算-通信比)表示

执行程序角度	处理数据角度
操作级	字串位串
指令级	字并位串
线程级①	字串位并②
任务或过程级	全并行
作业或程序级	

注：①调度单位为进程/线程
②少见(字长=1位的并行机)

(2)并行性的开发途径

***时间重叠：** 从时间上开发并行性(并发性)，

多个处理过程同时使用同一硬件设备的不同部分

例—流水线技术(操作流水线、指令流水线)

***资源重复：** 从空间上开发并行性(同时性)，

多个处理过程同时使用重复设置的多个硬件设备

例—多CPU、多核CPU

***资源共享：** 从软件上开发并行性(并发性)，

多个任务轮流使用 同一硬件设备

例—网络打印机

2、并行性的发展

(1)单机系统中的并行性发展

*时间重叠：分离及细化部件→指令流水→宏流水(如CPU与IO)
（功能专用化） （处理步骤重叠）

*资源重复：多个操作部件→指令级并行→阵列处理机(SIMD)
（多条指令用空闲部件） （一条指令用相同部件）

*资源共享：分时OS→虚拟机

(2)多机系统中的并行性发展

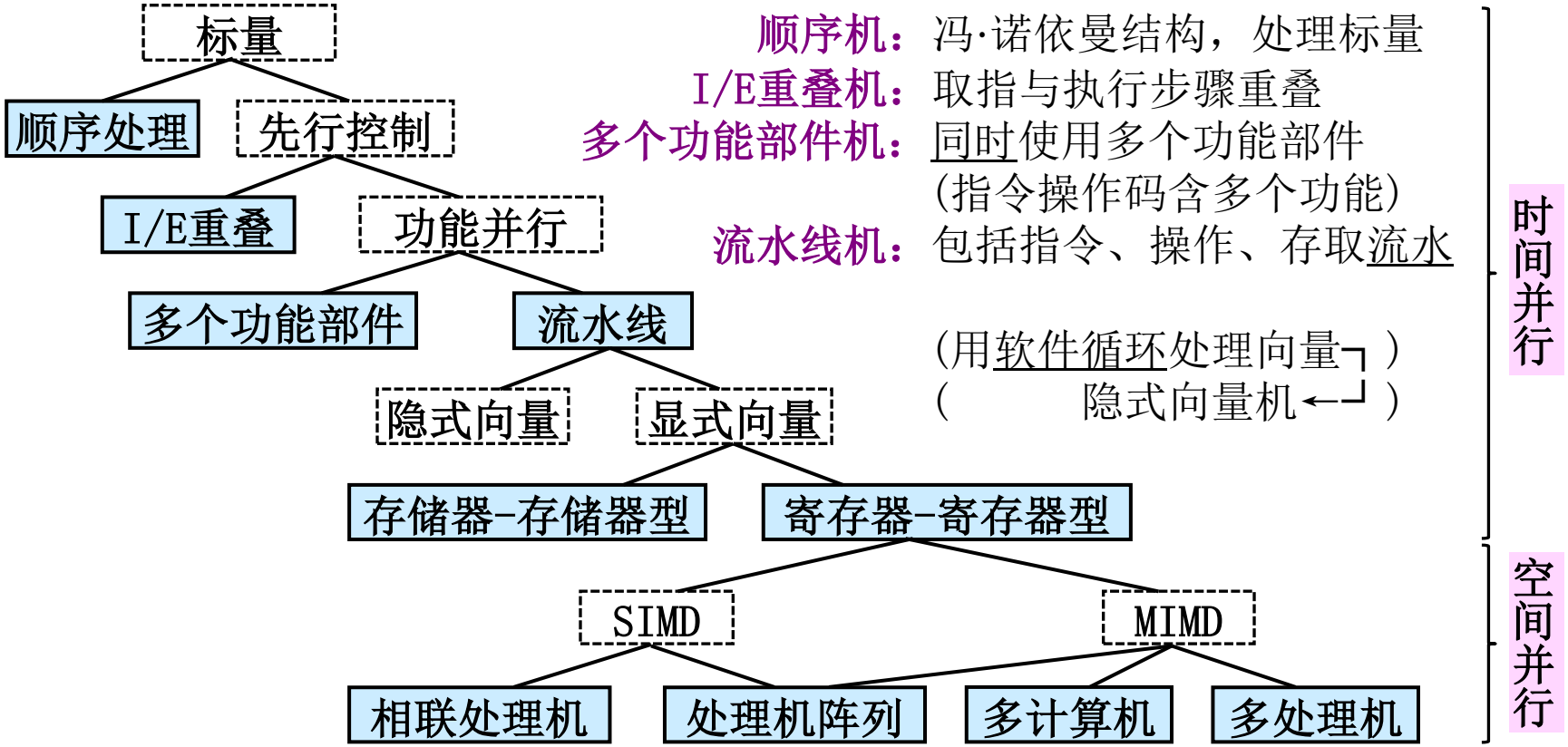
处理机连接方式— 有紧耦合、松耦合2种
（通过MBus） （通过IOBus） （如多CPU、网络计算机）

*时间重叠： 功能专用化→专用处理机→异构型MP系统

*资源重复： 提高可靠性→容错及可重构系统→同构型MP系统

*资源共享： 网络化→分布式系统

※系统结构的发展历程：



向量机：用一套硬件循环操作来处理向量，实现数据流水[硬件循环]

SIMD机：用多套硬件同时操作来处理向量，实现数据并行[字并位并]

MIMD机：用多套硬件同时执行指令(可不同)，实现功能并行[松耦合/紧耦合]

VLIW：

LD/ST1	LD/ST2	FADD	FMUL

第一章课后复习思考题

1、系统结构的经典定义、实质是什么？广义定义？系统结构的研究内容？系统结构与计算机组成、实现的关系如何？

2、系统结构主要有哪几种分类方法？特征是什么？

3、计算机系统性能指标主要有哪些？评测性能的方法有哪些？

4、计算机系统结构设计的5个准则的内容、含义是什么？

5、从软件、应用及器件对系统结构发展的影响角度，分析Intel系列处理器设计者的成功之处与苦衷。

6、开发计算机系统的并行性有哪些途径？