



上海大学  
Shanghai University

# 计算机系统结构

上海大学计算机学院



## ■ 授课方式与课程组织

□ 教学平台

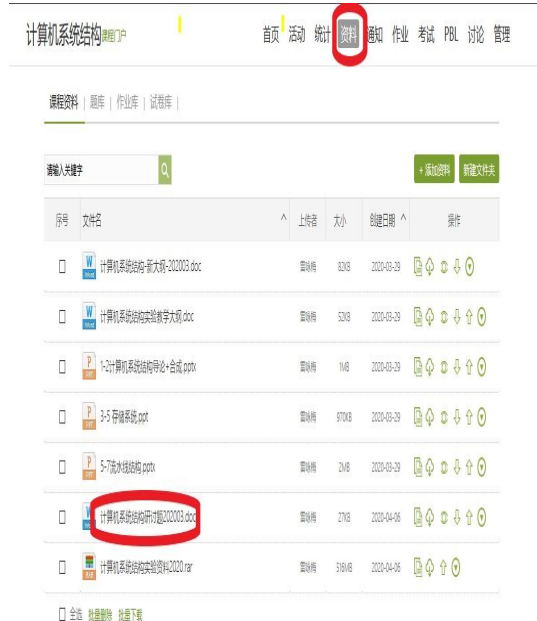
□ 课程组织

## ■ 课程计划与要求

- 课程目标
- 课程计划
- 研讨课要求
- 实验教学要求
- 课程教材与参考书

➔ 为什么要学习研究系统结构-思考和建议

- 课程支撑平台：
- 课堂教学 研讨 上机实践
- 课堂不下线 数据不流失 请各位同学及时查看超星学习通的通知！！！！
- 超星学习通 全体同学签到 学习资料上传和下载 讨论区 项目



## 学习通资料区

超星学习通APP

## 构建互动：平等 高效 有趣



- **授课方式** 探索 交流 研发 **知识再发现** 发现新知识
  - **课堂讲授** 分班研讨 实验实践 ……
  - **积极学习** 欢迎加入研究团队
- **课程背景:**
  - 目前并行计算也就是高性能计算正在进入体系架构的大变革时代
  - 大变革是“硬件、软件和算法同步演进”
  - **大数据** 与**智能** 云计算时代计算机系统尤为重要
  - **自强系列高性能计算机** **大数据编程环境** **科研成果** **在研项目**
- **时间安排** 一 3-4上课 三 5-6 研讨（单） **五**3-4上机
- **研讨方案** 课代表
- **成绩评定** **笔试成绩** **平时成绩**
- **平时成绩**  
=学习态度（出勤，作业）+上机实验成绩+研讨课成绩



“计算机系统结构”是一门非常注重教授**思想**和**方法**的课程。

- 计算机系统结构以**功能**和**性能**为目标，紧密围绕**计算**、**存储**、**网络**三方面来提高系统性能，以满足社会不断增长的需求。
- **注重理论联系实践**，通过学习 OpenMP, CUDA, MPI 等并行编程及性能调优方法将有助于理解和掌握系统结构的基础理论和方法，并建立**系统级**的思想。
- **教学安排**
  - Ch1 计算机系统结构导论 1周
  - Ch2 处理器及其相关技术 2周
  - Ch3 存储系统结构 2周
  - Ch4 流水线结构 2周
  - Ch5 并行处理机与多处理机系统 2周
  - Ch6 集群、网格和云计算等 1周
- **典型并行计算系统实例**
  - 自强系列 高性能计算机研发成果
  - 实验环境
  - 虚拟机构建
  - .....

# 研讨课要求



## 计算机系统结构

计算机系统结构课程门户

首页 活动 统计 **资料** 通知 作业 考试 PBL 讨论 管理

课程资料 | 题库 | 作业库 | 试卷库 |

请输入关键字



+ 添加资料

新建文件夹

序号	文件名	上传者	大小	创建日期	操作
<input type="checkbox"/>	计算机系统结构-新大纲-202003.doc	雷咏梅	82KB	2020-03-29	
<input type="checkbox"/>	计算机系统结构实验教学大纲.doc	雷咏梅	52KB	2020-03-29	
<input type="checkbox"/>	1-2计算机系统结构导论+合成.pptx	雷咏梅	1MB	2020-03-29	
<input type="checkbox"/>	3-5 存储系统.ppt	雷咏梅	970KB	2020-03-29	
<input type="checkbox"/>	5-7流水线结构.pptx	雷咏梅	2MB	2020-03-29	
<input type="checkbox"/>	<b>计算机系统结构研讨题202003.doc</b>	雷咏梅	27KB	2020-04-06	
<input type="checkbox"/>	计算机系统结构实验资料2020.rar	雷咏梅	516MB	2020-04-06	

☐ 全选 批量删除 批量下载

- 成绩评定
  - 考试成绩
  - 平时成绩
- 平时成绩:
  - 学习态度
  - 研讨

□ 形式: 分班 分组

□ 报告 **超星讨论区**

□ 展示 讨论 交流 点评

**每人必须报告 准备PPT及运行展示等**

□ **研讨题目** 根据课程理论学习布置 发布在超星资料区

□ 研讨方式 表述 逻辑 难度 创新

- 知识学习

- 研究归纳

- 实践创新

□ 报告和展示  
提问与回答 点评与评述

### → 综合性层次化实验课程设计

- 综合性课程设计1: Vmware虚拟机的安装, 创建和管理
  - 结合知识主题计算机系统概念的理论讲授
  - 运行、调试、测试和应用的基本软件工具
- 综合性课程设计2: 多核环境下OPenMP并行编程
  - 结合知识主题存储系统和流水线结构的理论讲授
  - 设计实现 (含调试、测试)
- 综合性课程设计3: MPI的安装和运行
  - 结合知识主题并行处理机与互连网络的理论讲授
  - 设计实现 (含调试、测试)
- 综合性课程设计4: HPL安装和测试
  - 结合知识主题多处理机系统及现代技术的理论讲授
  - 综合应用, 个性化作品设计

### → 指导和检查

- 加强实践
- 鼓励创新
- 独立完成
- 质量保证
- 注意规范

### → 实验 实验完成情况 创新 实验报告规范

复杂度增加,  
综合性增加





























计算机系统结构

首页 活动 统计 资料 通知 作业 考试 PBL 讨论 管理

课程资料 | 题库 | 作业库 | 试卷库 |

请输入关键字

+ 添加资料 新建文件夹

序号	文件名	上传者	大小	创建日期	操作
1	计算机系统结构-新大纲-202003.docx	雷明梅	82KB	2020-03-29	   
2	计算机系统结构实验教学大纲.doc	雷明梅	53KB	2020-03-29	   
3	1-2计算机系统结构绪论-合成.pptx	雷明梅	10KB	2020-03-29	   
4	3-5 存储系统.ppt	雷明梅	970KB	2020-03-29	   
5	5-7流水线结构.pptx	雷明梅	24KB	2020-03-29	   
6	计算机系统结构研讨题202003.docx	雷明梅	27KB	2020-04-06	   
7	计算机系统结构实验资料2020.rar	雷明梅	516KB	2020-04-06	   

全选 批量删除 批量下载



## ■ 教材

→ **教材**: 计算机系统结构(第4版)

沈文枫 徐炜民

→ 计算机系统结构(第3版)

徐炜民 严允中 电子工业出版社

## ■ 参考书

→ 清华大学计算机系列教材: **计算机系统结构**, 清华大学出版社

→ David A. Patterson and John L. Hennessy, Computer Architecture: A Quantitative Approach 3 Ed. San Francisco: Morgan Kaufmann Publishers

中文: 计算机系统结构: 一种定量的方法, 清华大学出版社

→ Kai Hwang, Advanced Computer Architecture Parallelism Scalability Programmability

中文: 高等计算机系统结构: 并行性 可扩展性 可编程性, 清华大学出版社、广西科学技术出版社

→ Randal E. Bryant David R. O' Hallaron Computer Systems A programmer's Perspective

中文: 深入理解计算机系统, 机械工业出版社





## ■ 第1章 计算机系统结构导论

→ 引言 为什么要研究系统结构

→ 1.1 计算机系统的基本概念

→ 1.2 计算机系统的发展- 冯·诺依曼体系结构的特点

→ 1.3 计算机系统的功能和结构

- 计算机系统的层次结构

- 计算机系统结构定义

→ 1.4 计算机系统设计方法

→ 1.5 现代计算机系统结构研究领域



- 引言 计算机系统性能提升
- 计算机系统发展
- 计算机系统的基本概念
- 计算机系统的发展- 冯·诺依曼体系结构的特点
- 计算机系统的功能和结构
- 计算机系统的层次结构
- 计算机系统结构定义



衡量计算机的指标：性能 价格 功耗

计算机的性能：

完成一个任务所需要的时间 指令数 完成每条指令需要的拍数 每拍需要的时间

提高处理机运算速度： $MIPS = Fz \times IPC$

**MIPS**(Million Instructions Per Second), GIPS, TIPS

其中：**Fz**为处理机的工作主频

**IPC**(Instruction Per Cycle)为

每个时钟周期平均执行的指令条数

**提高IPC：依靠先进系统结构**

过去，几个或几十个周期完成一条指令

现在，一个周期完成几条指令 单发射 多发射 转移猜测策略 存储层次结构

**提高Fz：微结构设计 缩短门电路延迟，依靠技术进步**

流水线技术等，依靠先进系统结构 从指令串行执行，到P4的**20级流水线**



# 计算机性能提高的几个阶段

20世纪70年代末之前，大型机和小型机计算机性能**每年提高25%~30%**。

20世纪80年代初，出现微处理机，依靠集成电路技术，性能**每年提高35%**

20世纪80年中期，出现RISC技术、高级语言和操作系统等，

性能**每年提高50%**

**1995年**，**先进系统结构**对微处理器性能的贡献是单纯技术进步的**5倍**。

近几年及将来，计算机性能按摩尔定理发展。

计算机系统的设计者们要不断创新。

计算机系统结构是最活跃和最具革命性的，社会需求发展会推动计算机相关的技术发展，而这些技术发展又会促进计算机系统结构的发展。



# 计算机性能提高的几个阶段

- 20世纪70年代末之前，大型机和小型机计算机性能**每年提高25%~30%**。
  - 20世纪80年代初，出现微处理机，依靠集成电路技术，性能**每年提高35%**
  - 20世纪80年中期，出现RISC技术、高级语言和操作系统等，性能**每年提高50%**
  - 1995年**，**先进系统结构**对微处理器性能的贡献是单纯技术进步的**5倍**。
- 计算机性能按摩尔定理发展。
- 计算机系统的设计者们不断创新。
- **微处理器结构**发展 即使在**串行计算机**中并行结构也起着重要作用
- 在**多个层级**上采用了并行技术
  - 指令级并行
  - 多处理机服务器
  - 大规模多处理机



## 1. 虚拟计算机

**定义：**从不同角度所看到的计算机系统的属性是不同的。

主要观察角度包括：

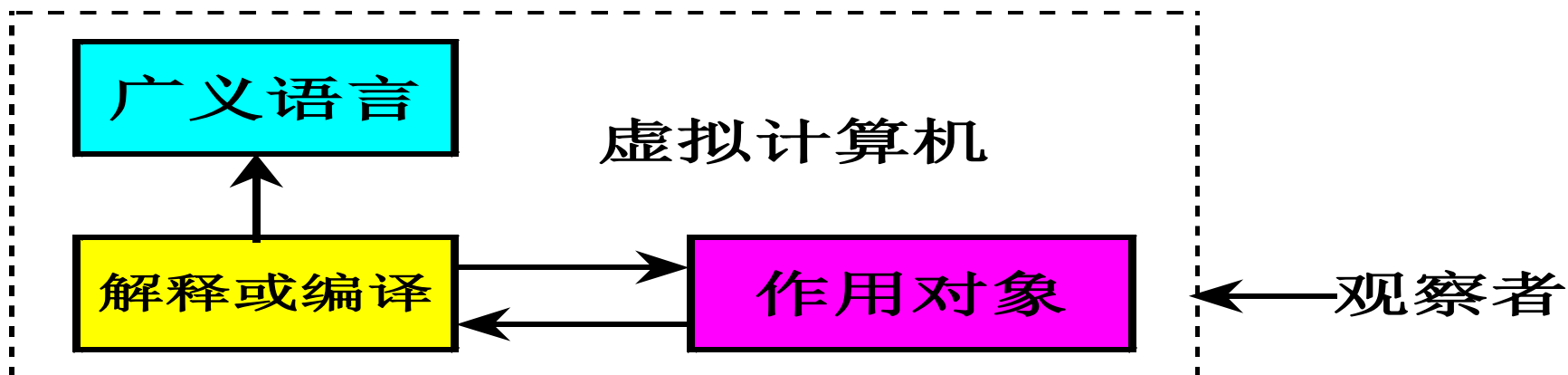
应用程序员

系统程序员

硬件设计人员

对计算机系统的认识通常只需要在某一个层次上

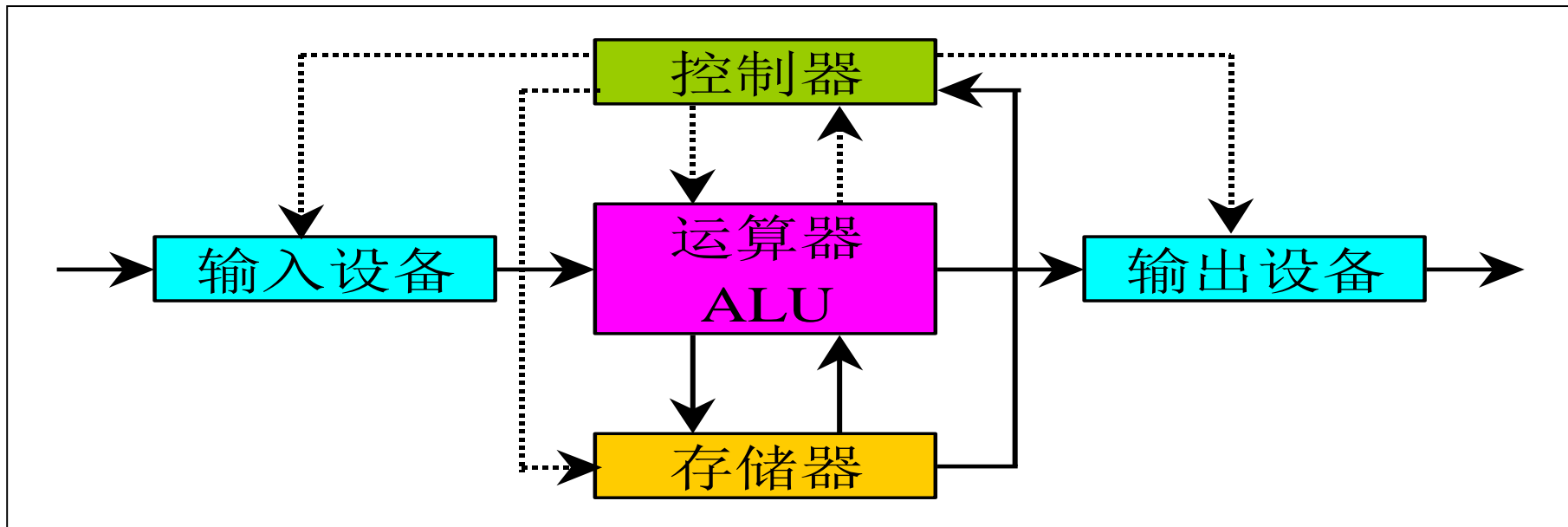
虚拟计算机系统



# 冯·诺依曼结构



Van Nennann基本思想于1936年~1946年期间形成，由冯·诺依曼等人于1946年提出



## 特点：存储程序、运算器为中心、集中控制

存储器是字长固定的、顺序线性编址的一维结构，每个地址是唯一定义的。 4096个字、40位。

由指令形式的低级机器语言驱动

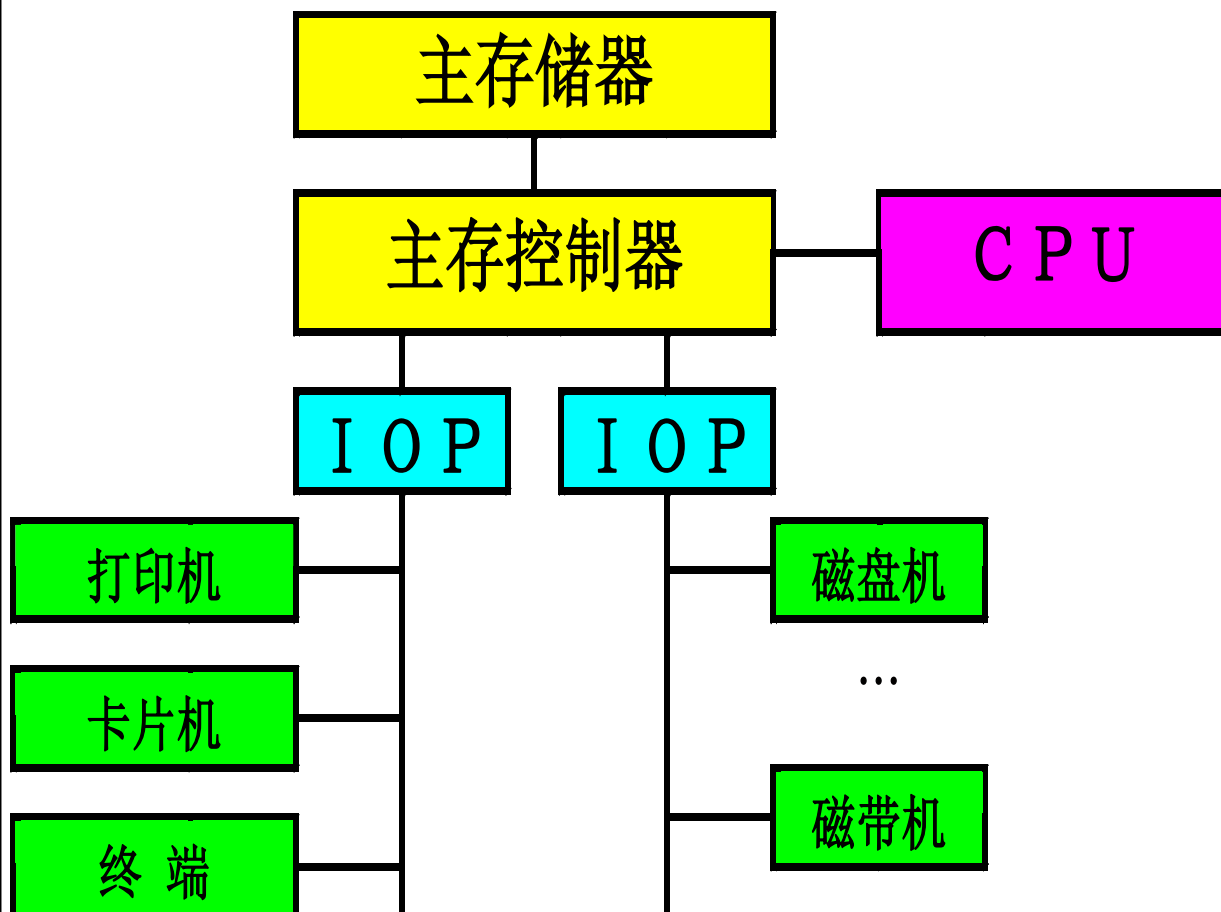
**指令顺序执行**，即一般按照指令在存储器中存放的顺序执行，程序分支由转移指令实现。

**运算器为中心**，输入输出设备与存储器之间的数据传送都途经运算器。

运算器、存储器、输入输出设备的操作以及它们之间的联系都由控制器集中控制。



## IBM360系列计算机的概念性结构



Amdahl于1964年在推出IBM360系列计算机时提出:

程序员所看到的计算机系统的属性,即概念性结构和功能特性

程序员: 系统程序员  
(包括: 汇编语言、机器语言、编译程序、操作系统)

看到的: 编写出能在机器上正确运行的程序所必须了解到的



# 1.2 计算机系统的发展



## 现代处理机对冯·诺依曼结构的改进

不变的：存储程序

改变的：存储器为中心，总线结构，分散控制

从基于串行算法变为适应并行算法，出现了向量计算机，并行计算机、多处理机等  
流水线处理机，超标量处理机，超流水线处理机，超标量超流水线处理机

数据库计算机和知识库计算机

专用计算机，如FFT变换机、过程控制计算机

为获得高可靠性而研制容错计算机

功能分散化、专业化，出现了各种分布计算机、外围处理机、通信处理机等

社会需求发展会推动计算机相关的技术发展，而这些技术发展又会促进计算机系统结构的发展

### 思考：

器件发展对系统结构的影响：

应用对系统结构的影响

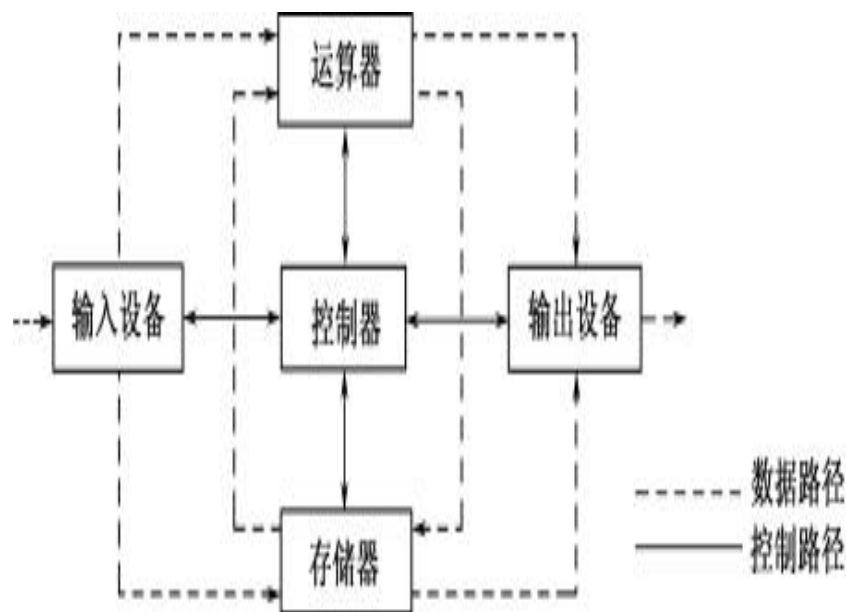
算法对系统结构的影响

价格对系统结构的影响



## 冯·诺依曼型计算机系统结构的基本特点

- ① 存储器是按地址访问的顺序线性编址的一维结构，每个单元的位数是固定的
- ② 指令由操作码和地址码组成
- ③ 指令在存储器中是按其执行顺序存储的，由程序计数器指明每条指令所在单元的地址。
- ④ 在存储器中指令和数据被同等对待
- ⑤ 计算机系统结构以运算器、控制器为中心
- ⑥ 指令、数据均以二进制编码表示，采用二进制运算。



### 1-4 冯·诺依曼结构

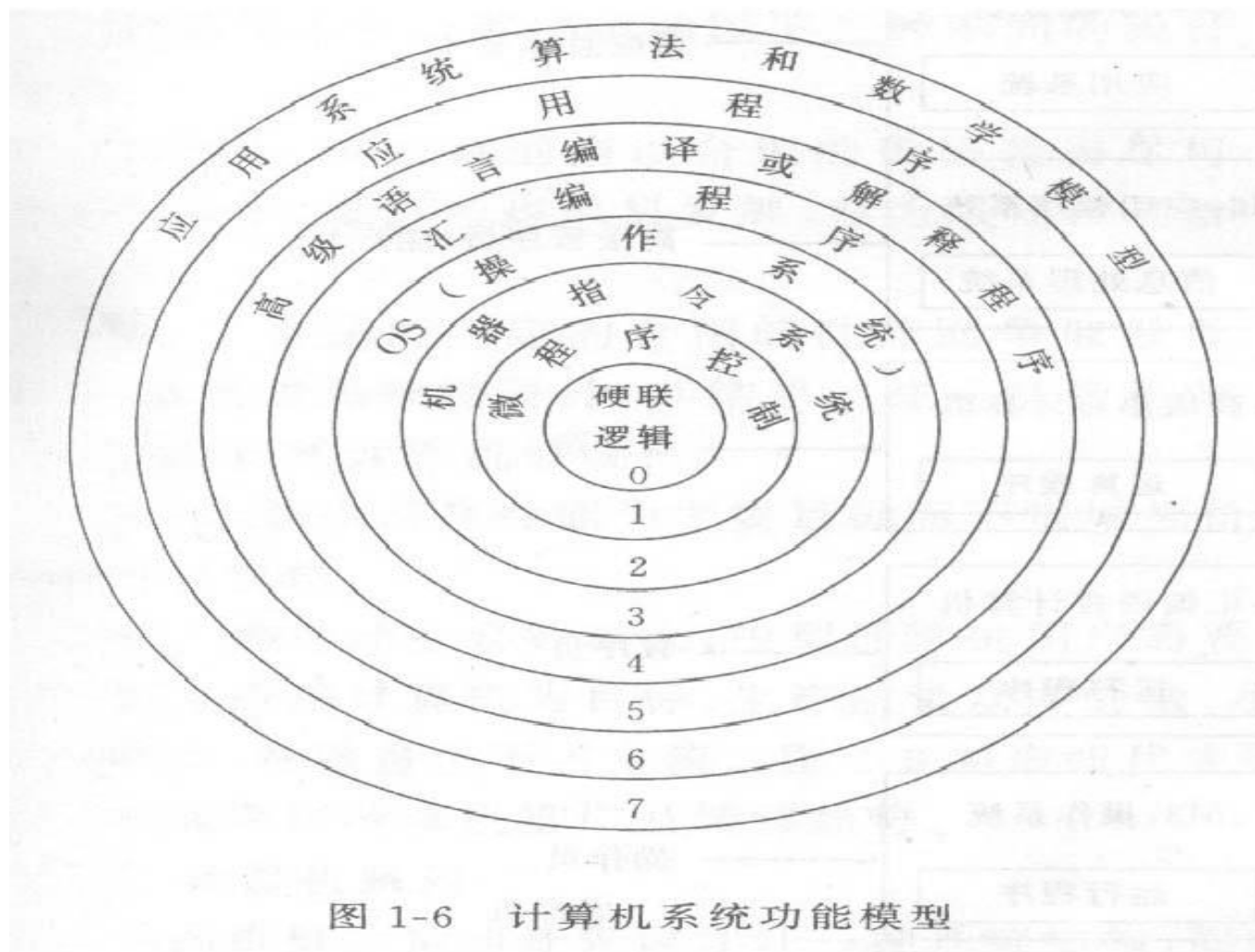


## 冯·诺依曼型结构的局限性:

- ① 由于冯·诺依曼结构以数值计算为主，因而对自然语言、图像、图形和符号处理的能力较差，不能满足上述领域的应用需求。
- ② 由于程序算法从整体上来说顺序型的，从而限制了并行操作的发挥，使计算机的运算速度不能在现有基础上取得根本性的突破。
- ③ 在冯·诺依曼结构上发展起来的软件系统越来越复杂，正确性无法保证，软件生产率低下。
- ④ 该结构的硬件投资较大，可靠性差，在体系结构发展上受到限制。
- ⑤ 使用该结构的计算机应用人员需要既懂专业知识，又具备编程技

# 计算机系统功能模型

计算机系统结构





- 软件相对硬件成本越来越贵。
- 软件产量和可靠性提高越来越困难

## 软件危机

计算机系统发展和应用扩大，积累了大量成熟的系统软件和应用软件

如何保护用户投资？

要求软件具有兼容性，即可移植性

## 价格对系统结构的影响

除非用户对性能的渴望很大，否则都希望获得好的性价比

## 应用对系统结构的影响

专门为某个领域设计专用计算机，计算效率高

例如：高结构化的数值计算：气象模型、流体流动、有限元、蒙特卡洛

## VLSI对系统结构的影响

## 技术的发展对价格的影响

采用不同技术的计算机获得性能价格比不同。某一系统结构性能超过某种器件技术的极限值时，就要采用新器件了。

## 六、算法对系统结构的影响

## 模拟和仿真

采用统一的高级语言方法

标准化很重要，但难以短期内解决



## 1.2.2 器件发展对系统结构的影响

1. 器件的功能和使用方法对系统结构的影响
2. 器件的发展对系统结构的影响

## 1.2.3 应用对系统结构的影响

应用对系统结构的发展有着重要的影响。不同的应用会对计算机系统结构的设计提出不同的要求。例如：AI应用

用户总希望计算机应用范围越广越好，能同时支持科学计算、事务处理和实时控制。多功能通用机概念源于功能强大的大、中型机，后来随着技术发展下移至小、微型机。从系统结构观点来看，各档（型）计算机的性能随时间下移，实质上就是在低档机上引用或照搬高档机的系统结构和组成。

## 1.2.4 算法对系统结构的影响

- 好的算法需要充分利用系统结构的高性能；一些重要的算法会影响计算器件设计或者系统结构的设计。
- 算法与系统结构之间总是会有不协调的地方，实际应用时要解决两者之间的矛盾，充分发挥计算、存储和通信性能，提高系统速度或者效率。





## 1.2.5 价格对系统结构的影响

- 性能价格比能比较好地反映系统之间的相对性能
- 从用户的角度分析有两个特点：
  - ① 用户总是希望以相同价格获得性能更高的系统；
  - ② 大多数用户更愿意获得更高性能的系统。
- 小型机-->集群

## 1.2.6 功耗对系统结构的影响

- 计算机的功耗和电压有关，降低电压可以减少功耗。CPU的延迟与供电电压成反比关系，即供电电压越高，单元的延迟越低。
- 人们对计算能力的需求不断增长，计算开始大量汇聚在云端，系统运营的能耗费用也高。功耗问题越来越引起了人们的关注。减少功耗对于提高系统稳定性、降低能耗有好处。这就对系统结构产生了影响。
- 嵌入式系统、便携式设备的大量涌现和广泛应用，对功耗也提出了特殊要求，功耗又反过来影响着这些设备的设计和使用。



## 1.3 计算机系统的功能和结构

- 描述控制流程的，有一定规则的字符集合的“计算机语言”。
- 计算机语言并不专属软件范畴，它可以分属计算机系统的各个层次，分别对该层次的控制流程进行描述。
- 基于对**语言广义的理解**，可以把计算机系统看成由多级“虚拟”计算机所组成。从内向外，层层相套，形成“洋葱”式结构的**功能模型**。

例：用户--建模--应用程序--高级语言--汇编语言--操作系统--机器语言--微程序--硬布线逻辑



# 计算机系统的层次结构

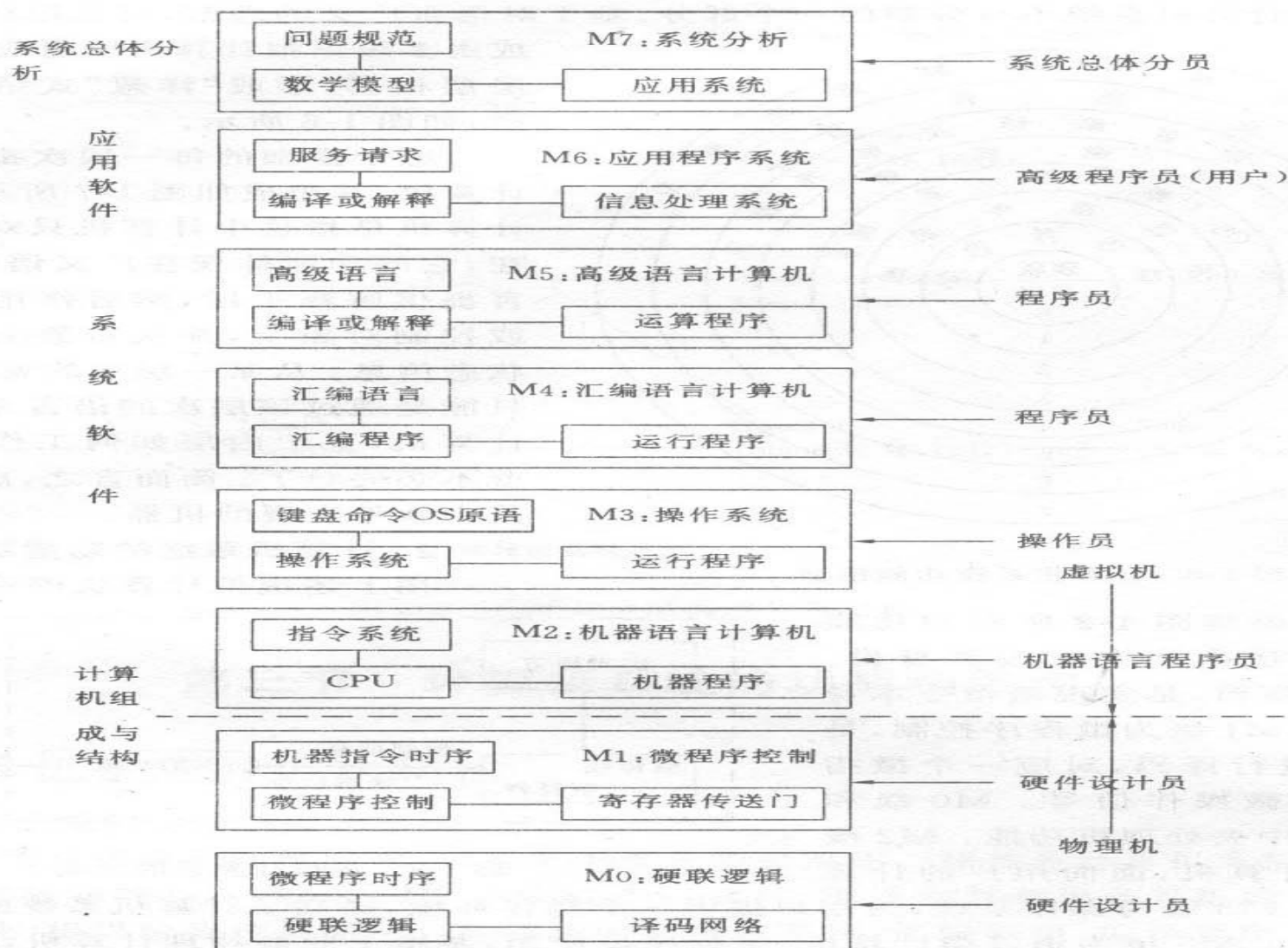


图 1-8 用虚拟计算机观点定义计算机系统的功能层次

# 计算机系统的层次结构



- 计算机系统可分为7个层次
- 第3级至第6级由**软件实现**, 称为**虚拟机**
- 从学科领域来划分:
  - 第0级和第1级属于**计算机组成原理**
  - 第2级属于**计算机系统结构**
  - 第3至第5级属于**系统软件**
  - 第6级属于**应用软件**
- 它们之间有交叉  
例如: 第3级必须依赖第4级和第5级来实现

# 计算机系统的层次结构

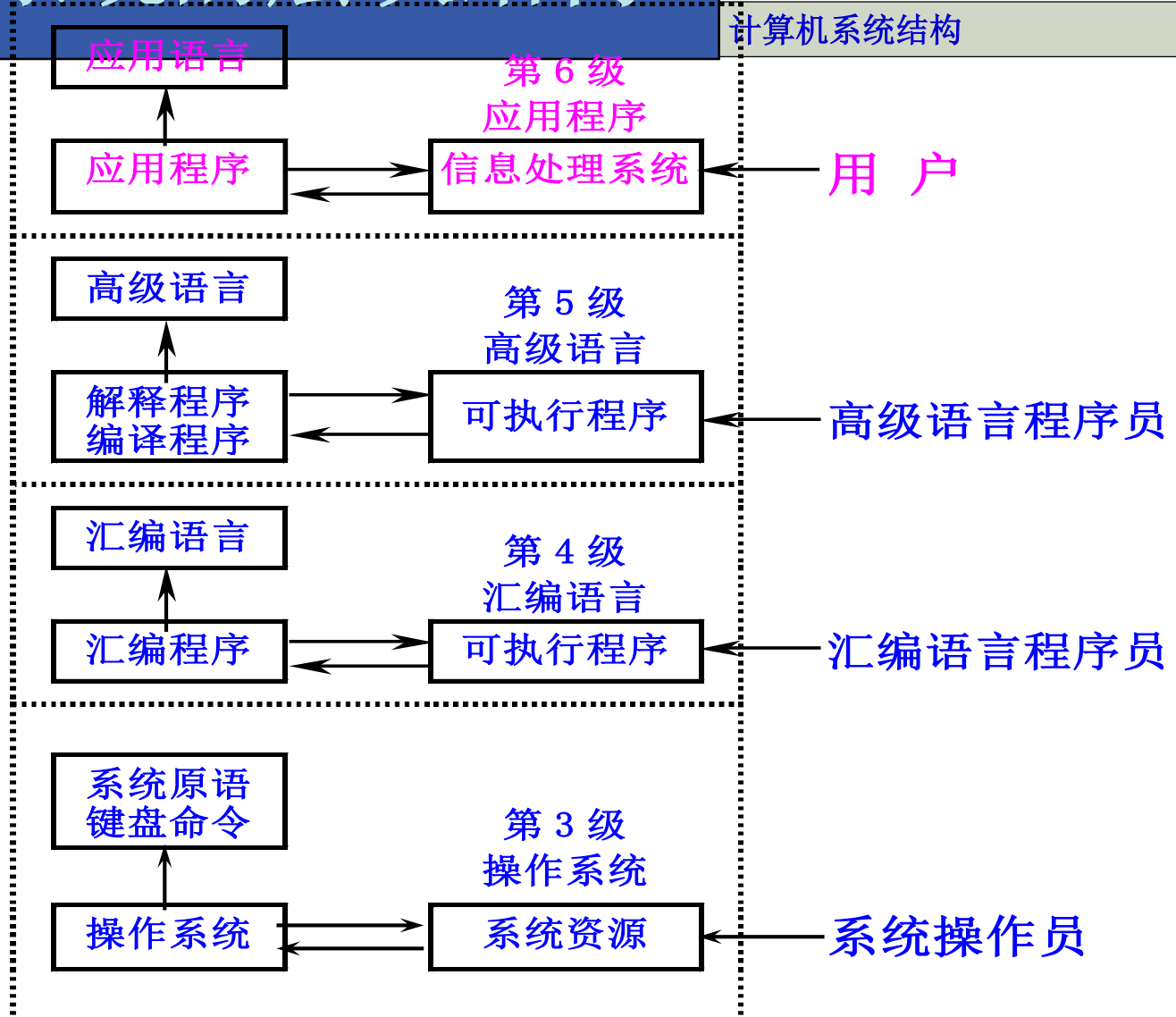
计算机系统结构





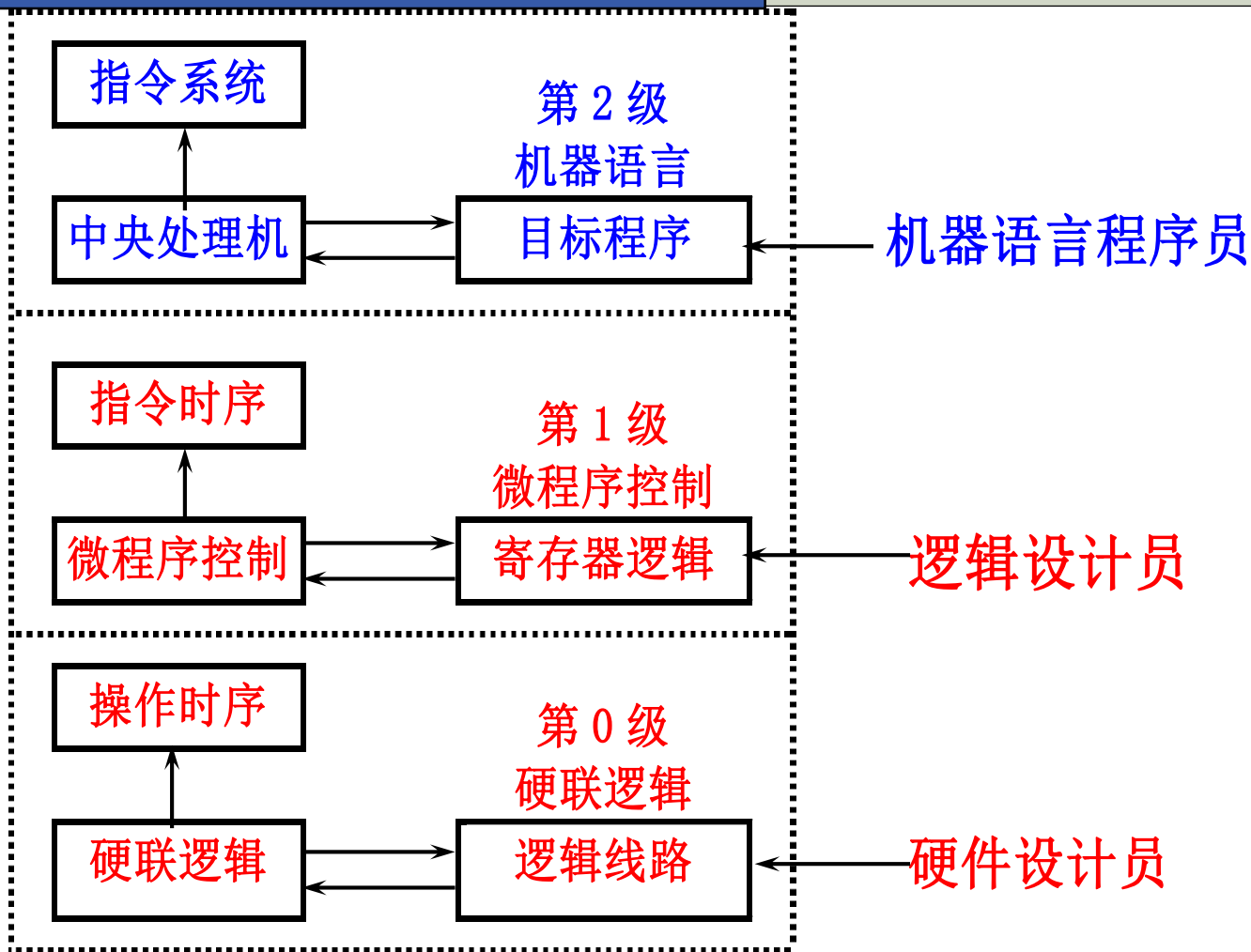
# 计算机系统的层次结构

计算机系统结构



# 计算机系统的层次结构

计算机系统结构



# 计算机系统结构的定义



- Amdahl提出：**计算机系统结构**是从程序设计者所看到的计算机的属性，即概念性结构和功能特性。这实际上是计算机系统的外特性。
- 从计算机系统的层次结构概念出发，不同级的程序设计者所看到的计算机属性显然是不一样的，“系统结构”就是指计算机系统中对各级之间界面的定义及其上、下的功能分配。
  - ➔ 例：图中M2级：机器语言级计算机。其界面之上是所有软件功能，界面之下是所有硬件和固件的功能。
- **研究软硬件功能分配和对软硬件界面的确定**
  - 计算机系统由软件、硬件和固件组成，它们在功能上是同等的。
  - 同一种功能可以用硬件实现，也可以用软件或固件实现。
  - 不同的组成只是性能和价格不同，他们的系统结构是相同的。

**系列计算机概念：**相同系统结构,不同组成和实现的一系列计算机系统。

**透明性概念：**

确定存在 用户无法控制和设置对

计算机系统不同位置的观察者所看到的计算机属性是不一样的,低层次看到的,对于高层次往往是透明的。



计算机系统结构与组成差别：

例1：确定指令系统功能--系统结构。

**指令的实现**：取指，取操作数，运算，送结果--组成。

例2：确定有乘，除法指令--系统结构。

用乘法器还是用加法器右移加实现--组成。

例3：确定MEM容量，编址方式（按bit，byte，word访问）--系统结构。

MEM速度，逻辑结构，性能价格比--组成。



## 计算机系统结构、组成和实现三者的关系

系统结构是计算机系统的软硬件界面

组成是计算机系统结构的逻辑实现

计算机实现是计算机组成的物理实现

\*.计算机实现:计算机组成的物理实现

\*.计算机实现包括:

处理器、主存等部件的物理结构, 芯片的集成度和速度, 芯片、模块、插件、底板的划分与连接, 专用芯片的设计, 微组装技术, 总线驱动, 电源、通风降温、整机装配技术等。

它着眼于芯片技术和组装技术, 其中, 芯片技术起着主导作用。

组成与实现差别:

例1: 乘法指令用乘法器还是用全加器右移加--组成。

乘法器或加法器用什么器件线路及工艺技术--实现。



# 1.4 计算机系统设计



- **软、硬件平衡**
- 用软件实现优点是设计容易，修改方便；用硬件实现优点是速度快，有较好性能。但并非硬件实现一定比软件实现速度快，算法在其中起重要作用符合今后发展的方向
- 一个成功的系统结构设计应能承受软、硬件发展和应用的变化。注意计算机技术和计算机应用的发展趋势
- 在硬件设计中必须考虑易扩性、兼容性，便于今后机器的升级换代
- **存储系统（尤其是主存）必须要有可扩性**
- 要求系统结构对编译器支持进一步提升



### ■ 由上往下

做法：确定用户需求，逐级向下设计、优化

优点：很好的应用效能，适于专用机设计

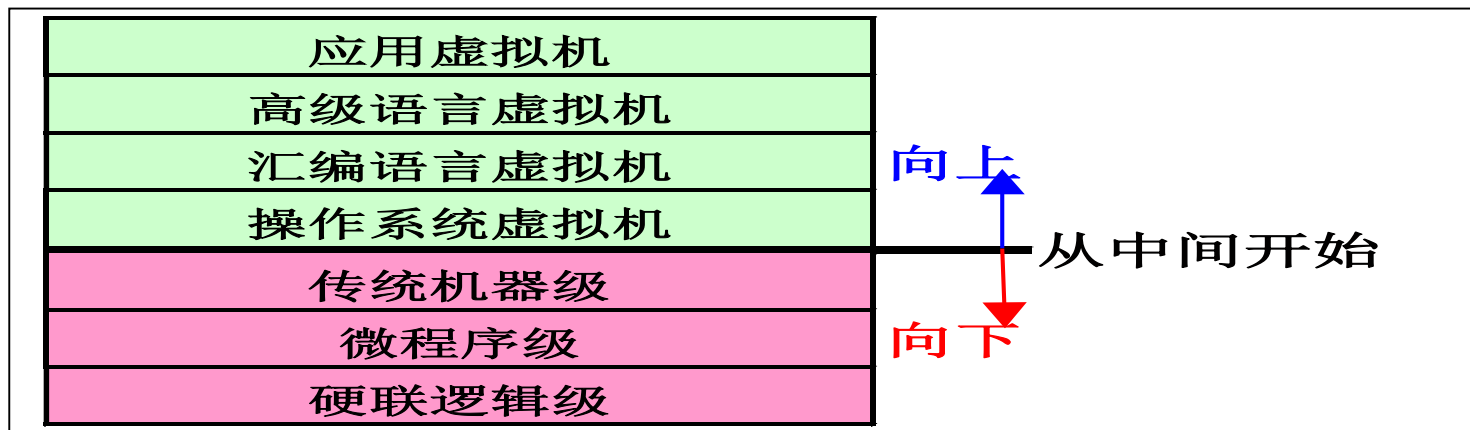
缺点：应用变化会导致系统效率急剧下降

### 由下往上

做法：根据现有器件和参照现有机器，从下往上设计

缺点：容易软、硬件脱节，软件无法优化。

不适合现在芯片技术飞速发展的今天



交界面开始。合理分配软、硬件功能。

优点：同时进行，缩短设计周期，利于软、硬件交互协调。



## \*. 设计步骤:

### 1. 需求分析

应用环境（实现，分时网络，远程，数据处理，计算等），语言种类和特性，外设特性，对OS要求，技术经济和市场分析。



## 2. 需求说明:

设计准则（造价，可靠性，可行性，可扩展性，兼容性，速度，安全性）功能说明，器件性能，易于程序设计。

.....

3. 概念设计：软、硬功能分配。

4. 具体设计：指令系统，数据表示，寻址方式。

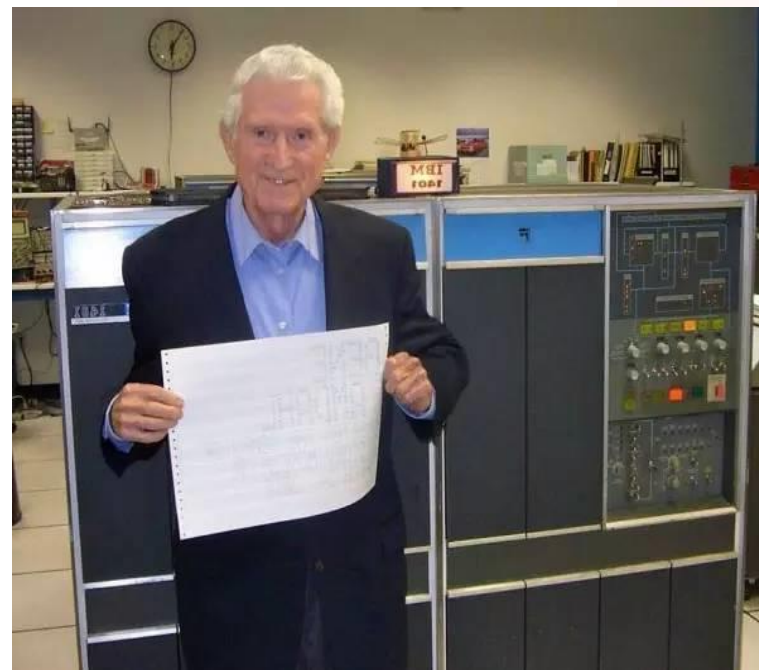
5. 设计优化和评价。



## Amdahl's Law

- Amdahl's Law states that the **performance** improvement to be gained from using some faster mode of execution is **limited by** the fraction of the time the faster mode can be used.

- 系统中某部件因采用某种更快执行方法后整个系统性能的提高与这种执行方式使用频率或占总执行时间的比例有关



Gene Amdahl曾是IBM大型机System/360的关键架构师



Amdahl 定律定义了加速比:

$$\text{加速比} = \frac{\text{采用改进措施后的性能}}{\text{采用改进措施前的性能}} = \frac{\text{采用改进措施前执行某任务的时间}}{\text{采用改进措施后执行某任务的时间}}$$

加速比与两个因素有关:

$$F_e = \frac{\text{可改进部分占用时间}}{\text{改进前整个任务执行时间}} (F_e < 1)$$

$$S_e = \frac{\text{改进前改进部分执行时间}}{\text{改进后改进部分执行时间}} (S_e > 1)$$





## 1.4.2 计算机系统设计的定量原则

由此得到下列结论。

① 改进后整个任务的执行时间：

$$T_n = T_o \left[ (1 - F_e) + \frac{F_e}{S_e} \right]$$

式中， $T_o$ 为改进前整个任务的执行时间。

② 改进前后整个系统的加速比：

$$S_n = \frac{T_o}{T_n} = \frac{1}{(1 - F_e) + \frac{F_e}{S_e}}$$



## 例 1:



- Suppose that we are considering an enhancement to the processor of a server system used for Web serving. The new **CPU is 10 times** faster on computation in the Web serving application than the original processor. Assuming that the original CPU is busy with **computation 40%** of the time and is waiting for **I/O 60%** of the time, what is the overall speedup gained by incorporating the enhancement?
- *Answer*    **Fractionenhanced = 0.4**    (I/O时cpu不工作)

$$\text{Speedupenhanced} = 10$$

$$\text{Speedup}_{\text{overall}} = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} \approx 1.56$$

# 例2:



- 设求浮点数平方根FPSQR操作占整个测试程序执行时间的**20%**。一种实现方法是采用FPSQR硬件，使其速度加快到**10倍**。
- 另一种实现方法是使所有浮点数指令FP速度加快到**2倍**，同时，设FP指令占整个程序执行时间**50%**。
- 请比较两种方案优劣。

# 例2



硬件方案  $F_e=0.2$   $S_e=10$ , 则

$$S_n = \frac{1}{(1-0.2) + \frac{0.2}{10}} = \frac{1}{0.82} = 1.22$$

FP 加速方案  $F_e=0.5$   $S_e=2$ , 则

$$S_n = \frac{1}{(1-0.5) + \frac{0.5}{2}} = \frac{1}{0.75} = 1.33$$



## The CPU Performance Equation (CPU性能公式)

- CPU性能取决三个要素；时钟频率 $f$ ；每条指令时钟周期数；指令条数 $IC$ 。时钟周期 $T=1/f$ 。
- 一个程序执行时的CPU时间可表示为；
- CPU时间=CPU时钟周期总数\*时钟周期 $T$

$$\text{CPU 时间} = \frac{\text{CPU时钟周期总数}}{\text{时钟频率}f}$$



如果一个程序的指令条数为 IC，  
则每条指令平均时钟周期数 CPI 为

$$CPI = \frac{\text{CPU时钟周期总数}}{IC}$$

经代换，可得：

$$\begin{aligned} \text{CPU 时间} &= IC * CPI * T \\ &= IC * CPI * \frac{1}{f} \end{aligned}$$

# CPU时钟周期的计算



一个程序的 CPU 时钟周期总数也可用下列方法计算;

$$\text{CPU 时钟周期总数} = \sum_{i=1}^n (\text{CPI}_i * I_i) \quad \leftarrow$$

$$\text{CPU clock cycles} = \sum_{i=1}^n \text{IC}_i \times \text{CPI}_i$$

$$\text{CPU time} = \left( \sum_{i=1}^n \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time}$$

$$\text{CPI} = \frac{\sum_{i=1}^n \text{IC}_i \times \text{CPI}_i}{\text{Instruction count}} = \sum_{i=1}^n \frac{\text{IC}_i}{\text{Instruction count}} \times \text{CPI}_i$$



- 程序访问的局部性原理
- 经统计：90%的时间去执行10%的程序代码，
- 即大部分时间是访问程序的局部空间。
- 程序访问的局部性是构建存储体系和建立Cache的理论基础。

# 1.5 计算机系统结构分类



- Flynn分类法
- 冯氏分类法
- H 分类法



# 1.5.1 计算机系统结构分类



- **Flynn分类法**

1966年 M.J.Flynn提出如下定义

(1)指令流(Instruction Stream):

机器执行指令序列。

(2)数据流(Data Stream):

指令流调用的数据序列。

(3)多倍性 (Multiplicity):

在系统最受限制的部件上，同时处于同一执行阶段的指令或数据的最大的个数

# 1.5.1 计算机系统结构分类



Flynn将系统结构分成：

SISD:单指令流单数据流。

SIMD:单指令流多数据流。

MISD:多指令流单数据流。

MIMD:多指令流多数据流。

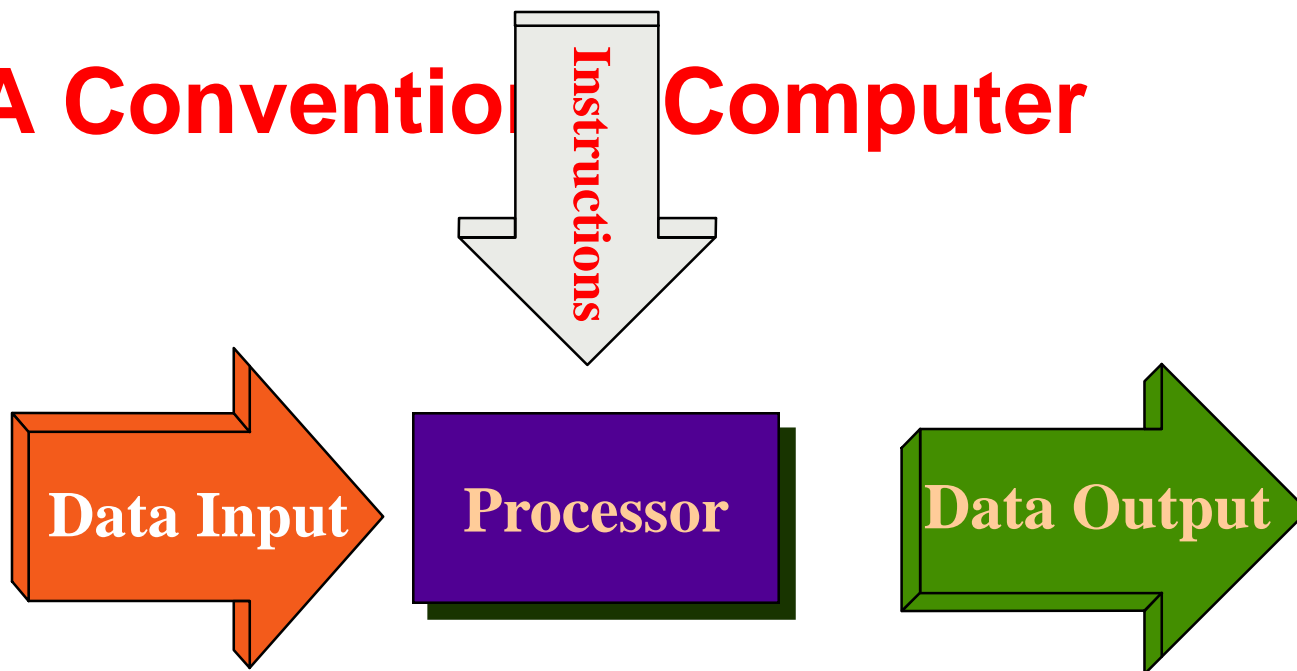
发展：

SPMD:单程序多数据流。

SIMT:单线程多数据流。



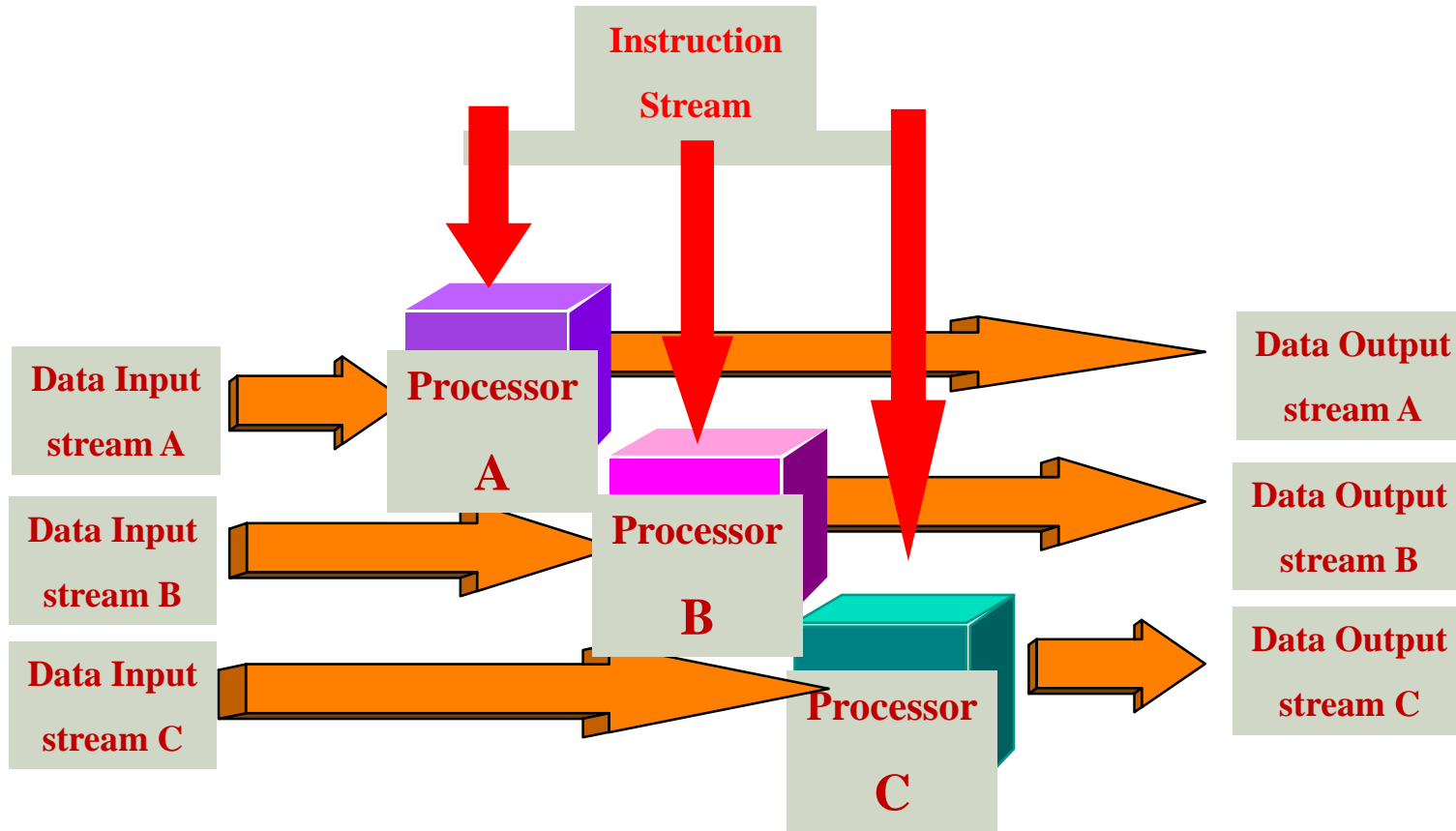
## SISD : A Conventional Computer



→ Speed is limited by the rate at which computer can transfer information internally.

# SIMD Architecture

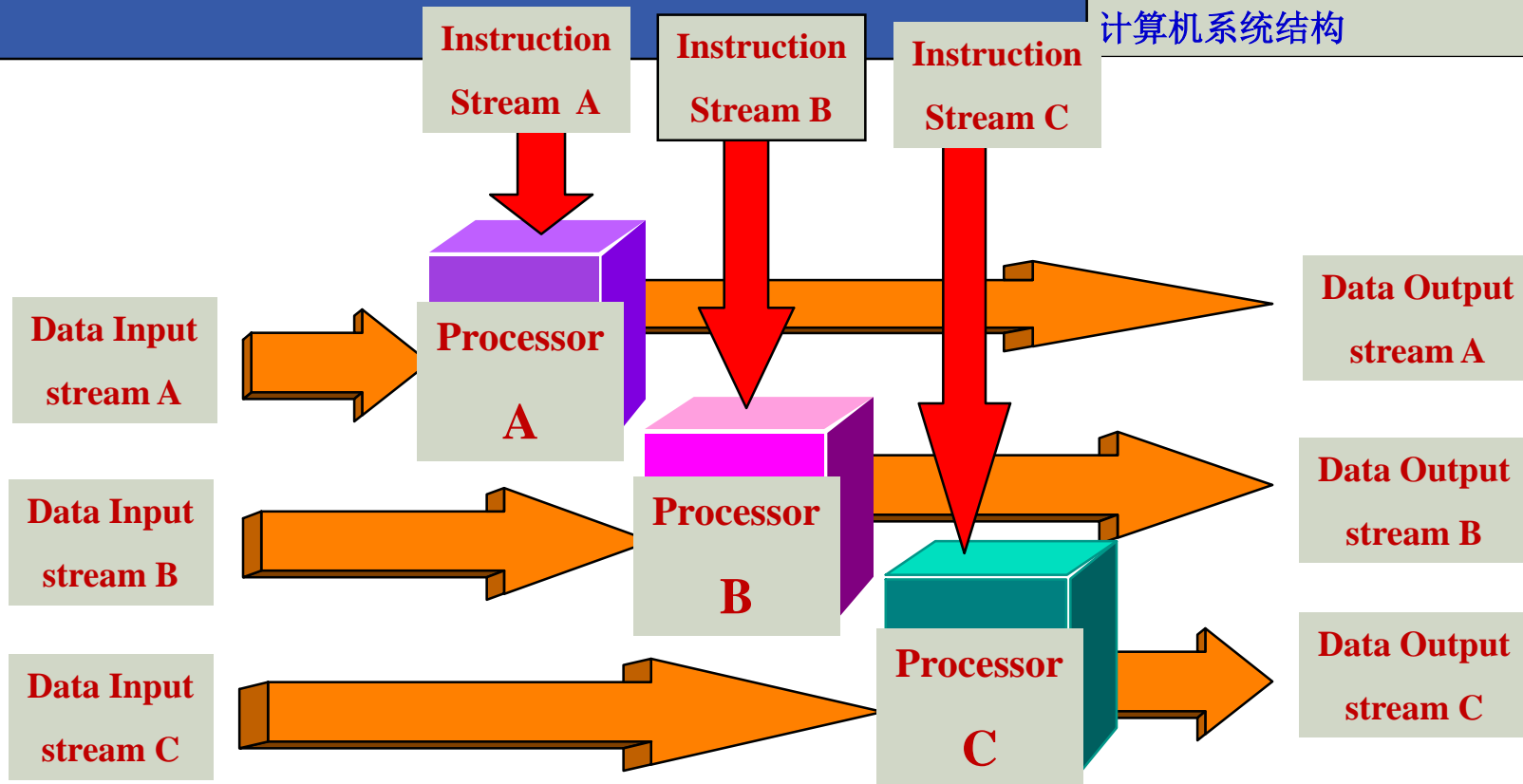
计算机系统结构



Ex: CRAY machine vector processing, Thinking machine  $cm^*$

# MIMD Architecture

计算机系统结构



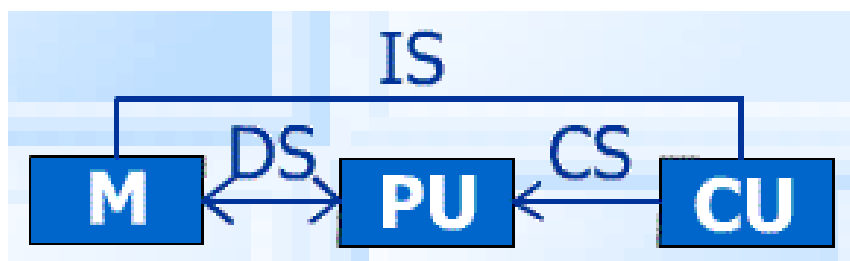
Unlike SISD, MISD, MIMD computer works asynchronously.

- Shared memory (tightly coupled) MIMD
- Distributed memory (loosely coupled) MIMD

# 1.5.1 计算机系统结构分类



CU--控制部件 M--主存 IS--指令流  
PU--处理部件 CS--控制流 DS--数据流

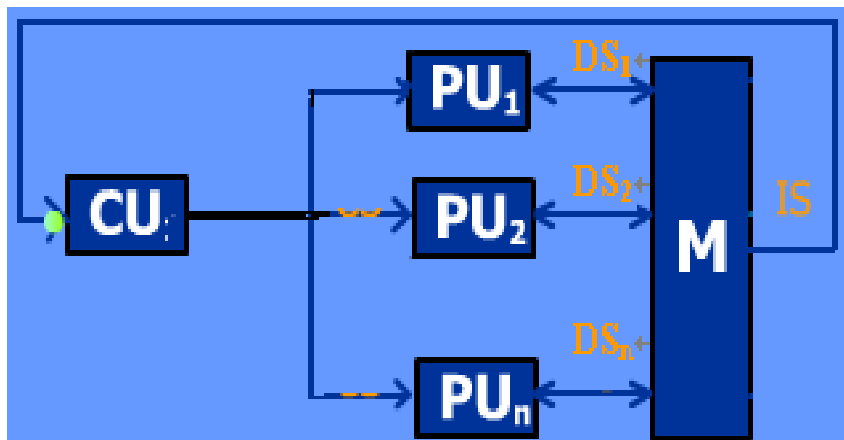


\*.串行计算机

由一个控制部件管理



## 1.5.1 计算机系统结构分类



### \*.并行处理机（阵列机）

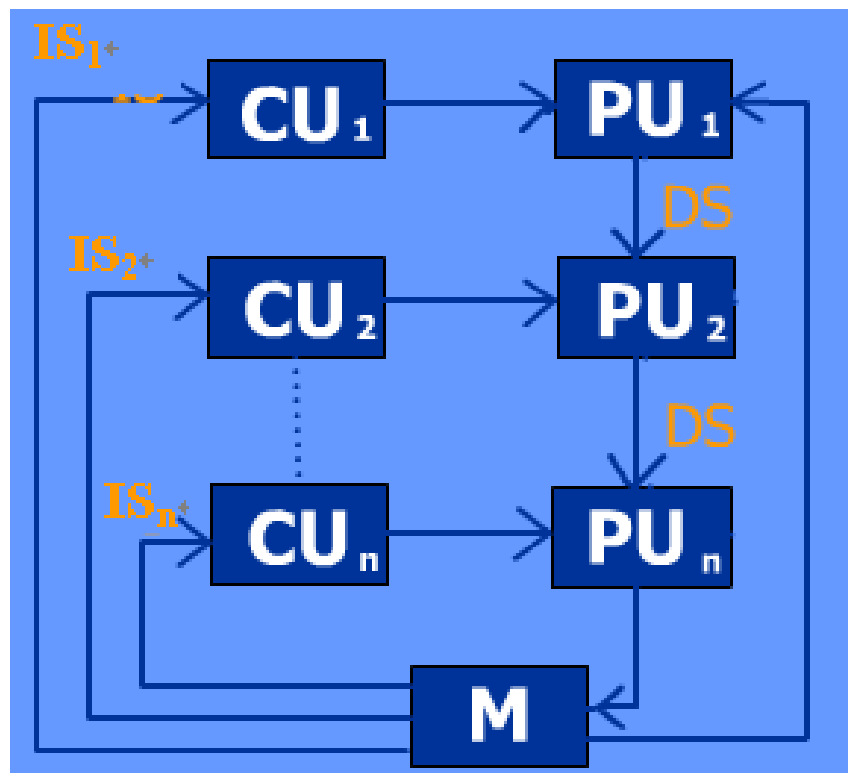
由一个控制部件管理，多个PU接受CU来的同一条指令

SIMD的技术应用：

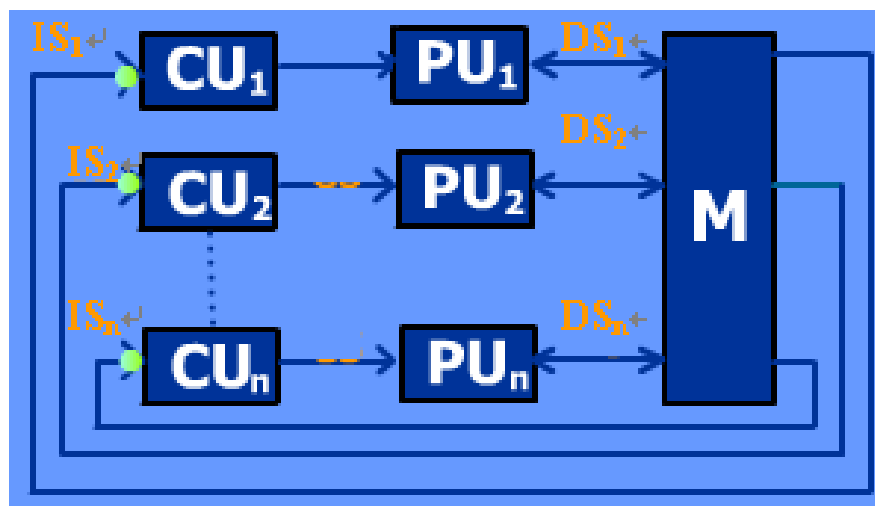
在CPU、GPU等加速部件中广泛应用

- MMX (MultiMedia eXtensions)
- SSE (Streaming
- SIMD Extensions)
- AVX (Advanced Vector eXtensions)
- ARM的SVE

# 1.5.1 计算机系统结构分类



MIMD:多指令流多数据流



\*.多处理机和多计算机

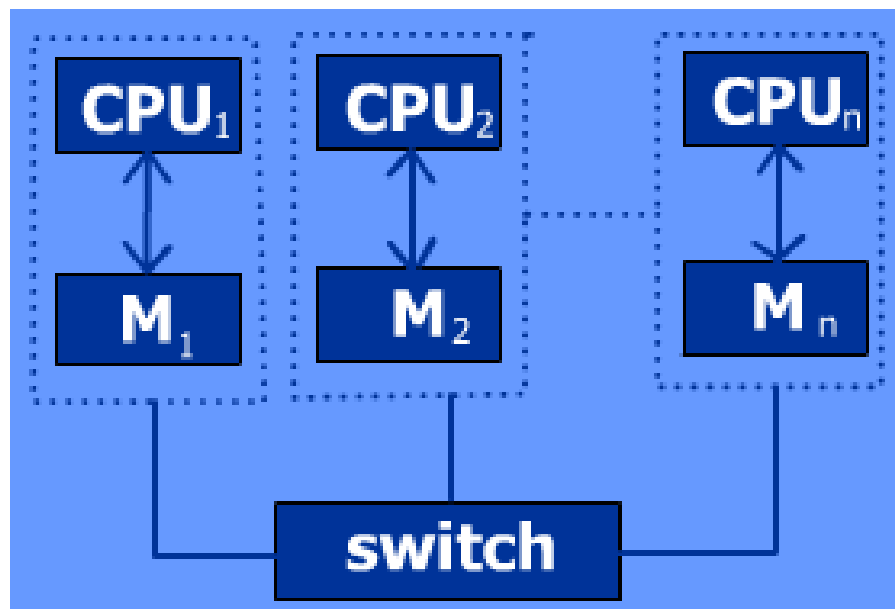


# 1.5.1 计算机系统结构分类



SPMD: 单程序多数据流

\*. 基于Cluster



# 1.5.1 计算机系统结构分类



**SIMT:单线程多数据流**

Single Instruction Multiple Threads

Nvidia 公司提出



Kepler GK110 Full chip block diagram



## 1.5.2 现代计算机系统结构研究方向

计算机系统结构的研究工作正向着多种高性能方向发展。

当前的研究主要集中在如下9方面：

- ① RISC结构；
- ② Client/Server结构；
- ③ Mpp矢量模式（Vector Mode）；
- ④ 并行处理（Massively Parallel Processing）；
- ⑤ 集群计算结构；
- ⑥ Network结构；
- ⑦ 数据流计算机结构；
- ⑧ 分布计算环境结构，如云计算；
- ⑨ AI计算。



## 1.5.3 计算机系统结构发展趋势

- **并行计算**的发展不断推动云计算和高性能计算的发展。
- AI的大规模应用加速推动市场对性能的渴求，传统通用CPU已无法满足需求，大量专用高性能的加速卡应用在云平台 and 智能系统中，
- 这些系统中的主要算力是加速卡提供的。
- 系统的性能不再由CPU的核数来决定。
- 随着AI的发展，甚至出现了针对某些计算领域的通用指令集、通用智能芯片和专用计算系统，随之性能评测指标也不再以双精度浮点运算为主，而和应用

## 1.5.3 计算机系统结构发展趋势

从计算机系统结构的观点看，可以分为：

- 单处理系统
- 并行与多处理系统
- 分布式处理系统

