

# 计算机组成原理

## Principle of Computer Organization

### ✓第一章 计算机系统概述

北京邮电大学  
计算机学院

戴志涛



北京邮电大学

计算机学院

2021/3/1

1

# 本章内容

- 从存储程序的概念入手，说明计算机的基本组成
- 从信息处理过程出发，说明一个完整的计算机系统是由哪些硬件和软件构成的，以及软硬件之间的关系
- 对各部分有一些粗浅的、整体的认识



# 计算机的分类



## ➤ 按照输入输出信号的形式分类

- ❑ 电子模拟计算机

- ❑ 电子数字计算机

计算机 - >  
电子计算机 - >  
电子数字计算机



# 计算机的分类

## ➤ 按照计算机的体系结构、运算速度、结构规模、适用领域分类：

### 1989年IEEE科学巨型机委员会：

- ❑ 个人计算机（Personal Computer）：PC
- ❑ 工作站（Workstation）：SUN、HP工作站
- ❑ 小型计算机（Mini-computer）：VAX-11
- ❑ 主机（Mainframe）：IBM4381
- ❑ 小巨型计算机（Mini-supercomputer）：  
ELXSII6400
- ❑ 巨型计算机（Super-computer）：Cray-I



# 计算机的分类



## ➤ 以应用为中心的分类方法

### □ 通用计算机:

- ✉ 具有计算机的标准形态，通过装配不同的应用软件，以类同面目出现并应用在社会各个方面

### □ 专用计算机/嵌入式（Embedded）计算机:

- ✉ 被安装、固定、嵌入到交通工具、仪器仪表、控制系统、通信设备、家用电器等部件中的模块化数据处理机
- ✉ 以嵌入式系统的形式隐藏在各种装置、产品和系统中



# 哪些属于嵌入式计算机?



A. 台式计算机



B. 笔记本电脑



C. 工业控制计算机



D. 平板计算机



E. 移动电话  
智能手机



F. 智能眼镜  
可穿戴计算机



北京邮电大学

# 计算机的发展历史——计算机体系结构创始人

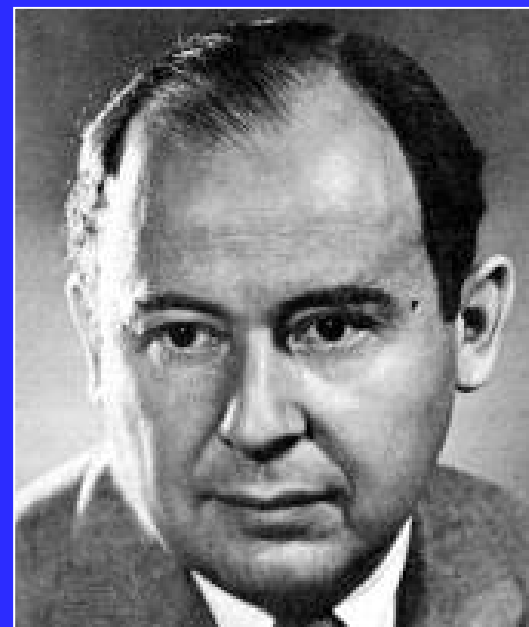
## ➤ 冯·诺依曼

□ John von Neumann

□ 1903~1957

## ➤ 1945年6月，发表“101页报告”

□ First Draft of a Report on the EDVAC



现代电子计算机之父  
——冯·诺伊曼



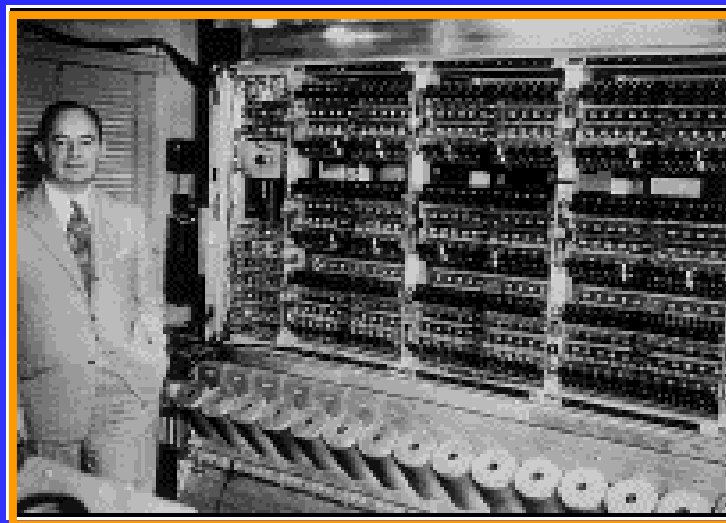
北京邮电大学

# 计算机的发展历史——计算机体系结构创始人

- 1946年7月，论文《电子计算工具逻辑设计初探》（Preliminary discussion of the the logical design of an electronic computing instrument）

- 提出了“存储程序”的概念，完整的计算机模型

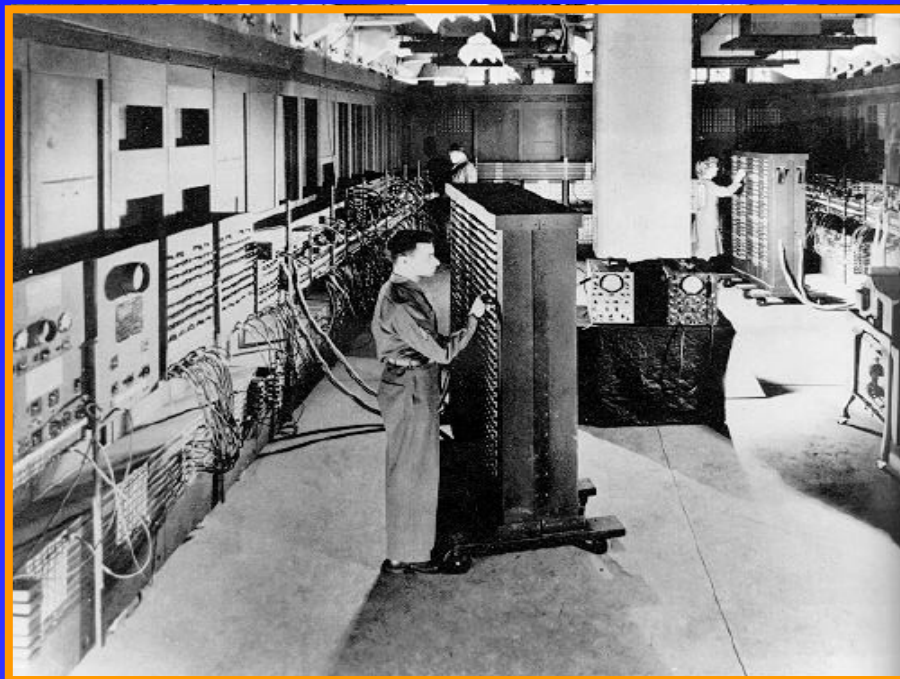
- 冯·诺依曼体系结构（von Neumann architecture）





# 世界上第一台电子数字计算机

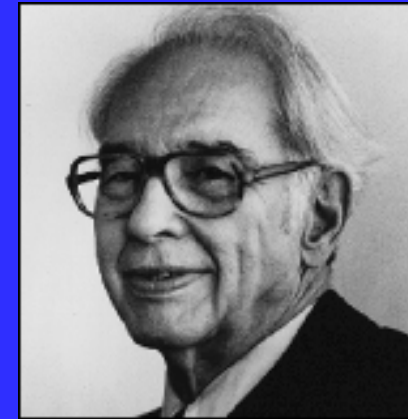
- 1946年，宾西法尼亚大学Moore 学院的约翰. 毛克利（John Mauchley）与普莱思伯. 艾克特（Presper Eckert）设计与制造
- ENIAC（“电子数字积分计算机”，Electronic Numerical Integrator and Computer）



# 世界上第一台电子数字计算机（版本二）

➤ 第一台电子计算机的试验样机于1939年10月开始运转

➤ 约翰·文森特·阿塔那索夫（John V. Atanasoff，1903-1995），爱荷华州立大学（Iowa State University）物理学教授



John Atanasoff教授

□ 研究生克利福特·贝瑞（Clifford E. Berry，1918-1963）

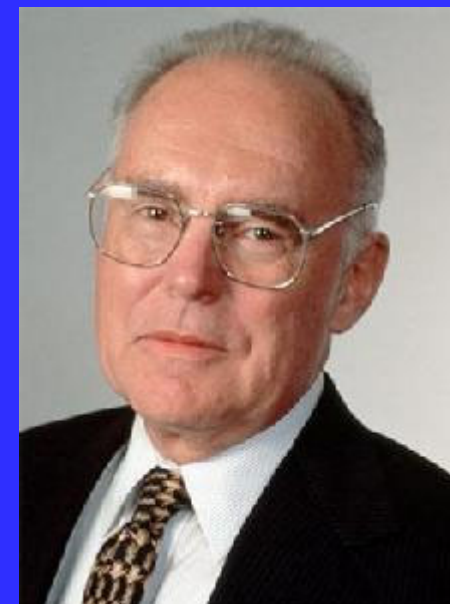
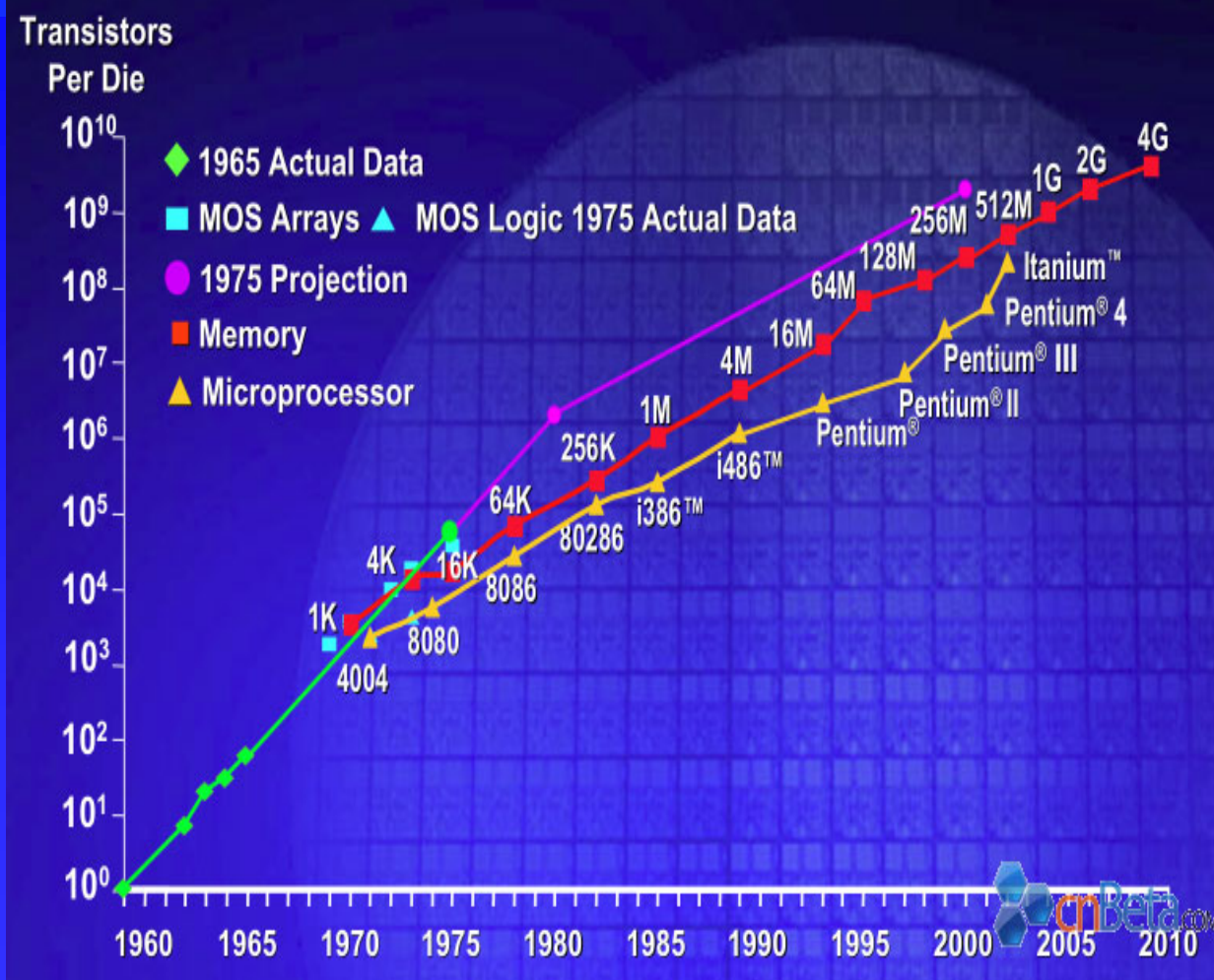
➤ 阿塔那索夫把这台机器命名为ABC（Atanasoff- Berry-Computer）



ABC



# 摩尔定律



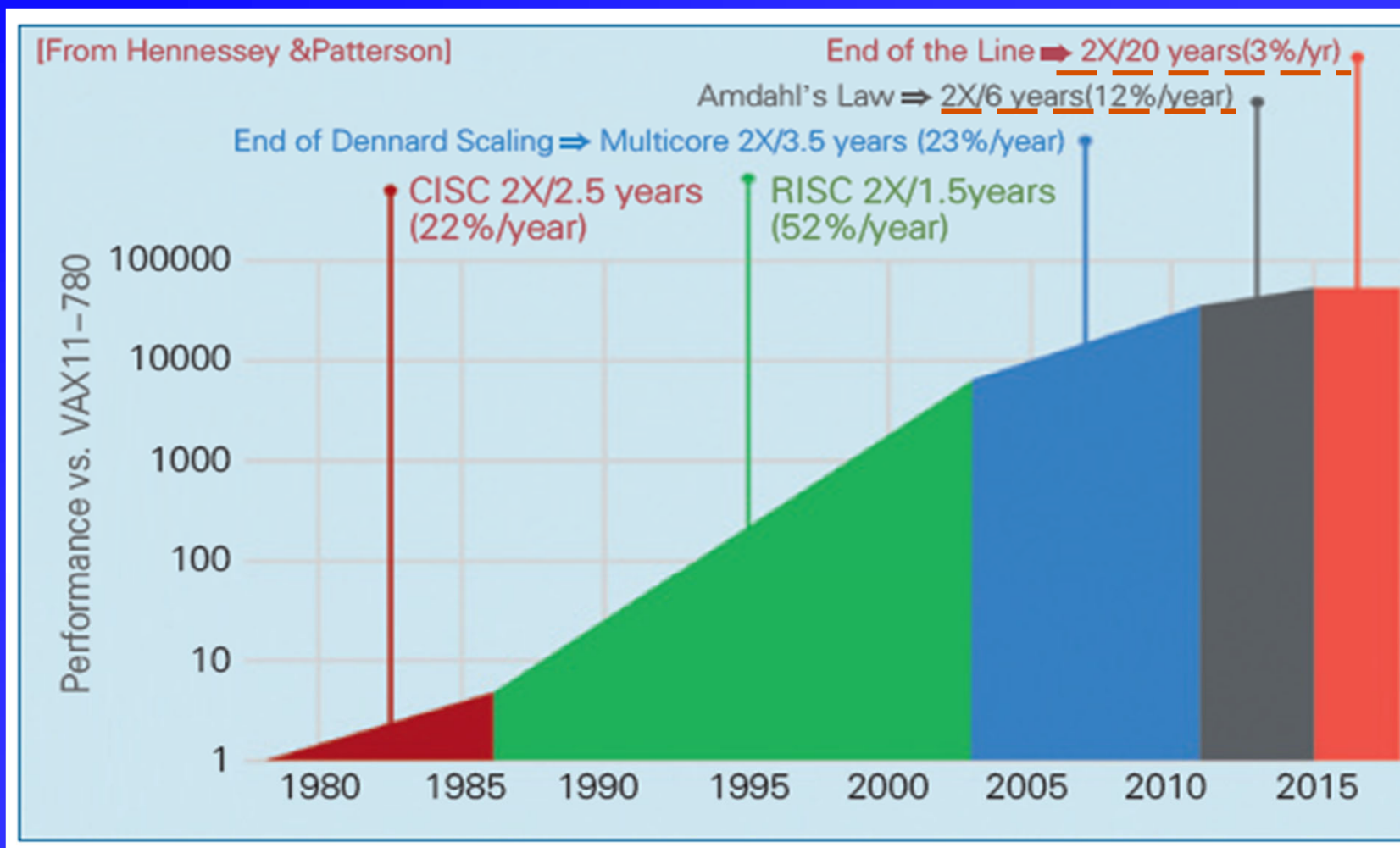
高登 摩尔  
(Gordon Moore)  
Intel创始人之一

集成电路大体上每18个月翻一番，何时终结？





# 摩尔定律的终结



图片来源：后摩尔时代高性能软件开发的挑战，中国计算机学会通讯第 15 卷 第10 期，2019 年 10 月



# 计算机技术的发展

软件工程一改长期以来面向机器、语言、中间件和实现等面向主机的形态，转为面向需求、网络和服务。

中国工程院李德毅院士：

1970s



面向过程

1980s



面向对象

1990s



面向构件

2000s



面向领域

2010s



面向服务

SOC

面向服务的计算

软件工程

交互方式

从人围着计算机转到计算机围着人转，交互、分享、群体智能等，都超出早先图灵机范畴。



键盘



鼠标



触摸



手势



语音

计算设施

大型机



1960s

小型机



1970s

个人电脑和局域网



1980s

桌面互联网



1990s

移动互联网



2000s



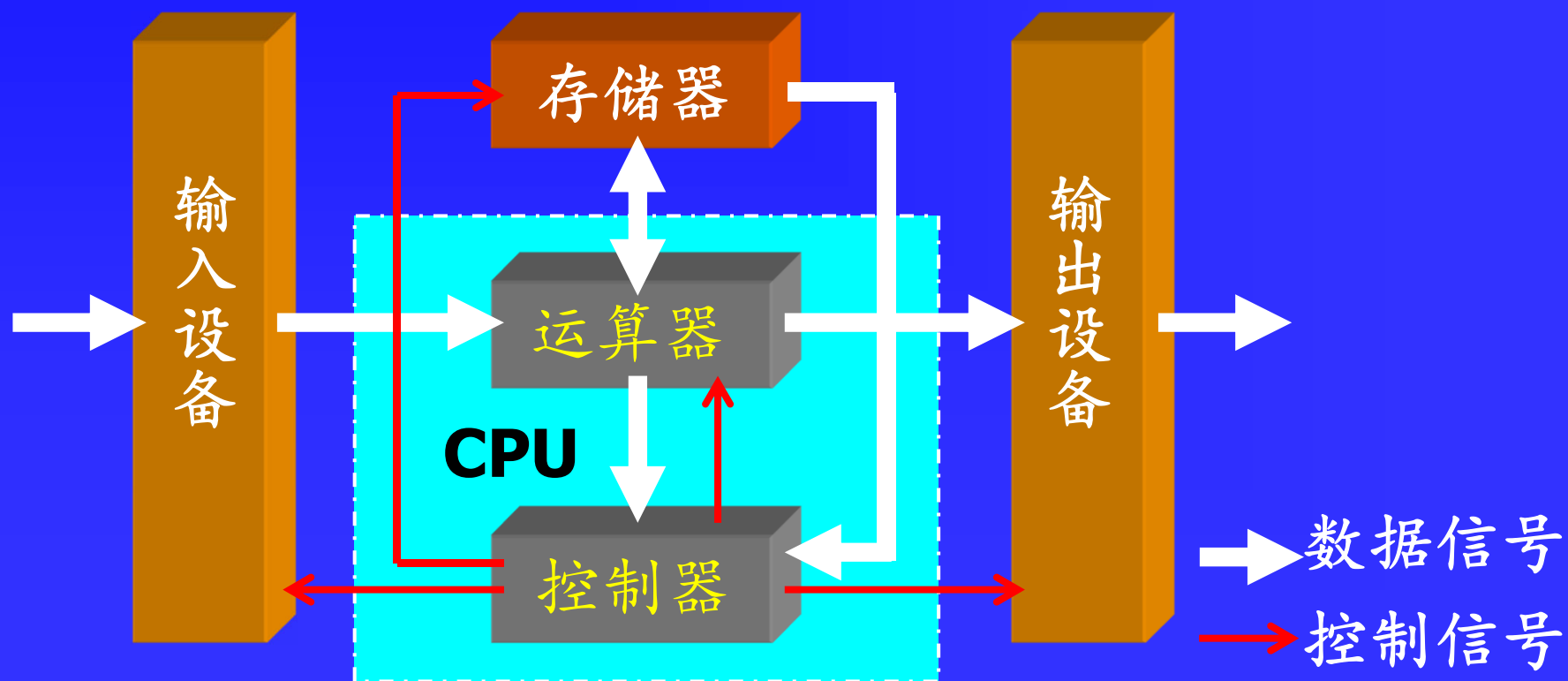
北京邮电大学

从以计算机为中心，到以网络为中心，再到以人为中心

# 计算机的硬件组成

## ➤ 计算机系统硬件由五大部件组成

□ 控制器、运算器、存储器、输入设备、输出设备



# 计算机的硬件组成



## ➤ 运算器

- ❑ 又称为算术/逻辑运算单元ALU  
(Arithmetical/Logical Unit)
- ❑ 信息加工部件，完成数据的算术和逻辑运算，以及移位操作等
- ❑ 运算器的字长：运算器每次可以完成多少位二进制数的运算
  - ✉ 一般为8位、16位、32位或64位（比特，bit，一般用小写字母b表示）
  - ✉ 运算器的位数越多，计算的精度就越高，但所消耗的电子器件也越多，成本就越高



# 计算机的硬件组成



## ➤ 存储器 (Memory)

□ 存储控制计算机操作的命令信息（指令）和被处理（加工）的信息（数据），也存储加工的中间结果和最终结果

✉ 命令：能被计算机所执行，指挥计算机系统工作，以完成所要求的任务

✉ 数据：是被处理的对象或者结果

□ 所有的数据和指令均以二进制数的形式存放在存储器中





# 计算机的硬件组成



## ➤ 存储器 (Memory)

□ 为解决容量与成本、速度的矛盾：将存储器分为内存储器和外存储器

✉ 内存储器：由半导体存储器构成

□ 主存 + 高速缓冲存储器 (cache, [kæʃ])

✉ 外存储器：多由磁盘、磁带、光盘等构成

□ 狭义的存储器仅指内存储器



# 计算机的硬件组成



## ➤ 存储器 (Memory)

□ 存储器的容量：存储器能保存的二进制信息的数量

✉ 存储单元数  $\times$  位数 / 单元

✉ 如：1M  $\times$  8 比特 = 8Mb

□ 8 个比特的二进制序列称为一个字节 (byte, 通常用大写字母 B 表示)



# 计算机的硬件组成



## ➤ 控制器 (Control Unit)

- ❑ 从内存中取出指令加以分析，然后发出控制信号执行某种操作
- ❑ 每条指令可以完成一次算术或逻辑运算，或是存取数据等操作
- ❑ 程序：能完成某种功能的一串指令的序列
- ❑ 控制器根据程序指挥系统工作，以完成程序所规定的功能



# 计算机的硬件组成



## ➤ 指令的功能

- ❑ 每条指令应能表示该指令完成哪一种操作，该操作需要的数据在内存中的存放位置

## ➤ 指令的组成

操作码	地址码
-----	-----

- ❑ 操作码：指出指令所进行的操作的种类，如加、减、乘、除、与、或、非、存数、取数、移位、转移等。
- ❑ 地址码：表示参加运算的数据应从存储器的哪个单元中取，或运算结果应该存放到哪个单元中。一条指令可以有一个、两个、三个或无地址码。
- ❑ 操作码和地址码均用二进制数表示



# 指令和程序实例



指令	操作码
加法	001
减法	010
乘法	011
除法	100
取数	101
存数	110
打印	111
停机	000

图1.5 指令和数据在  
存储器中用二进制码存储

存储器	
地址1	101 1001
2	011 1100
3	001 1010
4	010 1011
5	110 1101
6	111 xxxx
7	000 xxxx
8	
9	a(二进制数)
10	b
11	c
12	x
13	y
⋮	



# 控制器的基本任务



- 按照程序所排的指令序列，先从存储器中取出一条指令放到控制器中
  - 由指令译码器对该指令的操作码进行分析判别
  - 根据指令性质，向计算机各部件发出相应的控制信号，有序地控制各部件完成规定的操作
  - 从存储器中取出下一条指令，再执行，依此类推
  - 每取出一条指令，控制器中的程序计数器就加1，为取下一条指令做准备
- 
- 取指周期：取指令的一段时间
  - 执行周期：执行指令的一段时间



# 指令流和数据流



## ➤ 信息的单位

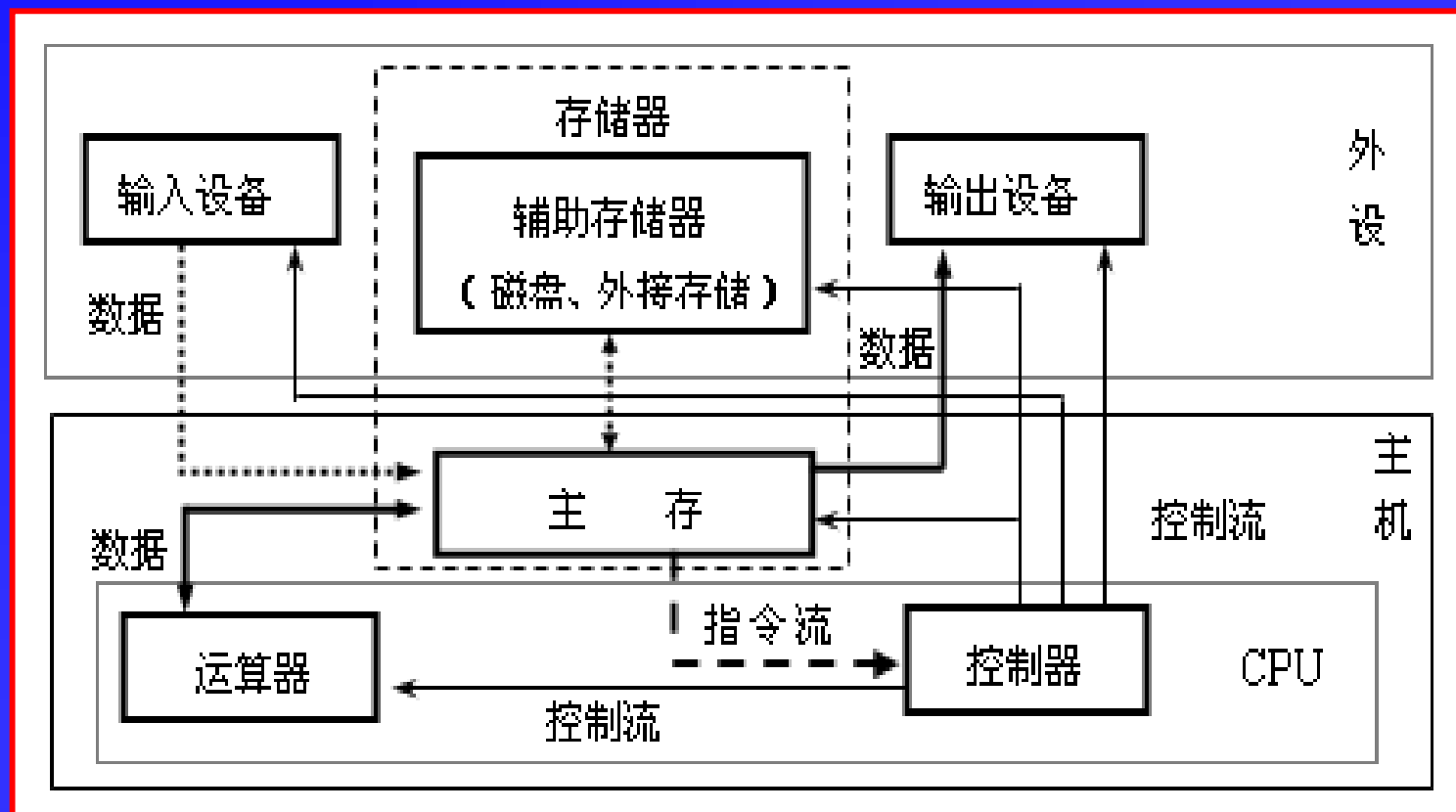
- 比特 (Bit) : 二进制位
- 字节 (Byte) : 由8位二进制数组成的信息单位
- 字 (Word)
  - ✉ 数据字和指令字
  - ✉ 字长: 组成一个字的二进制位数



# 指令流和数据流

## ➤ 区分指令和数据

- ❑ **指令流**：取指周期中从内存读出的信息流，流向控制器
- ❑ **数据流**：执行周期中从内存读出的信息流，流向运算器





# 计算机的硬件组成



## ➤ 输入输出设备

- ❑ 输入/输出设备（I/O设备），也称外围设备
- ❑ 作用：进行信息形式的转换，也即把外界的语言文字、声音、机械动作等信息形式转换成计算机能识别的电信号表示的二进制数的形式，或是进行相反方向的转换。
- ❑ 适配器（I/O接口）：
  - ✉ 由于种类繁多且速度各异，外围设备不是直接地同高速工作的主机相连接，而是通过适配器部件与主机相连接
  - ✉ 作用：相当与一个转换器，保证外围设备用计算机所要求的形式发送或接收信息



# 计算机的硬件组成

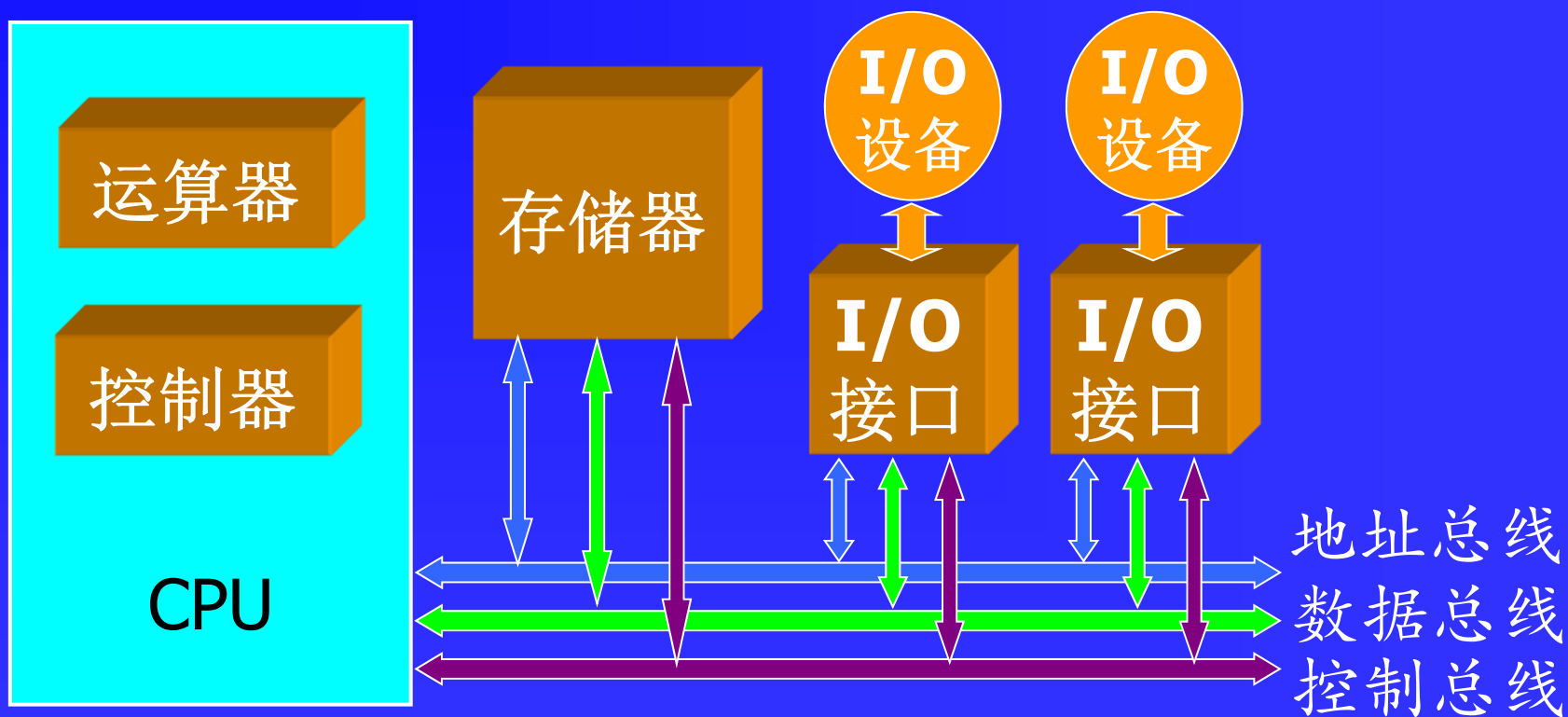


## ➤ 总线 (bus)

- ❑ 计算机各部件间传送信息的公共通路
- ❑ 各部件分时复用总线，以保证数据、地址、指令和控制信息在各部件之间的传送
- ❑ 基本总线：
  - ✉ 数据总线DB (Data Bus)
  - ✉ 地址总线AB (Address Bus)
  - ✉ 控制 (命令) 总线CB  
(Command/Control Bus)



# 计算机的硬件组成



# 鲲鹏920处理器片上系统的组成部件

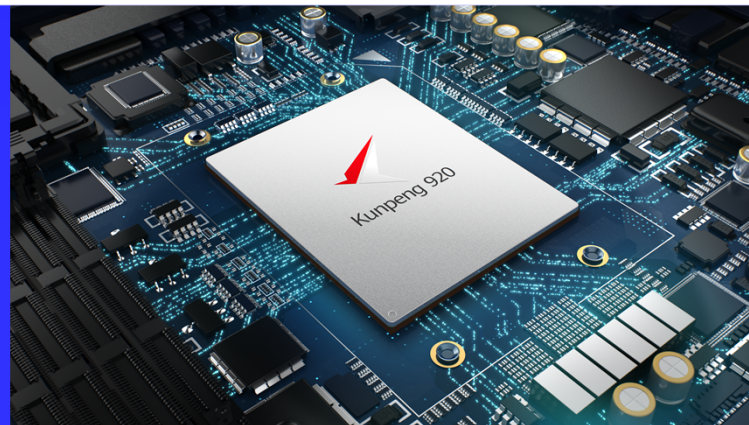
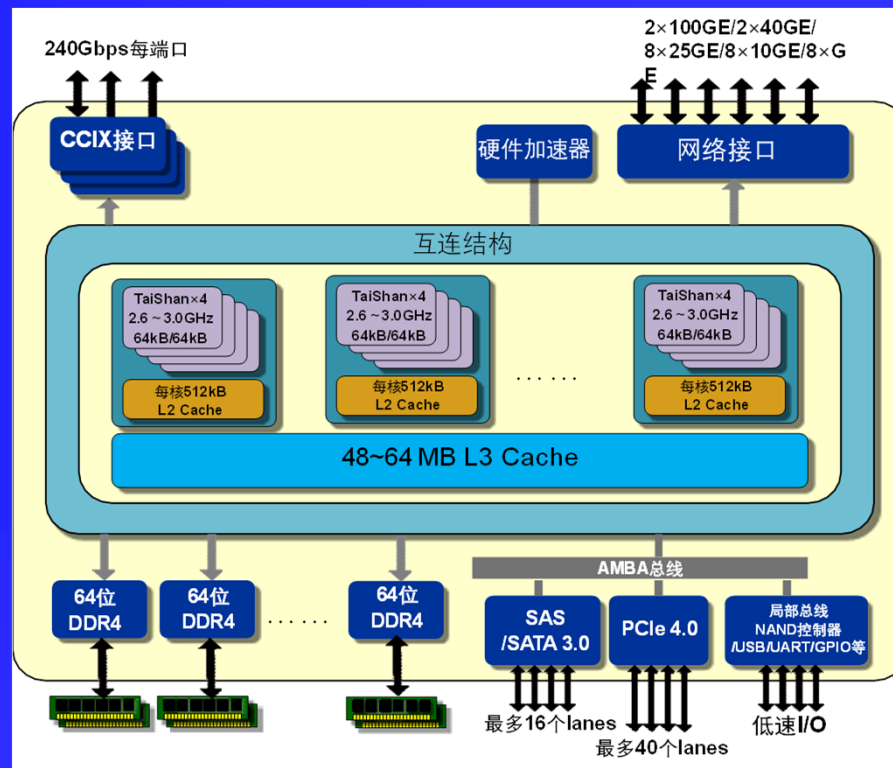
## ➤ 由一致性互连总线包围起来的处理器内核

❑ 最多配置64个TaiShan V110处理器内核

✉ 典型主频为2.6GHz

## ➤ 三级Cache结构

- ❑ 64KB L1 ICache、64KB L1 DCache
- ❑ 独享512KB L2 Cache
- ❑ 系统级L3 Cache

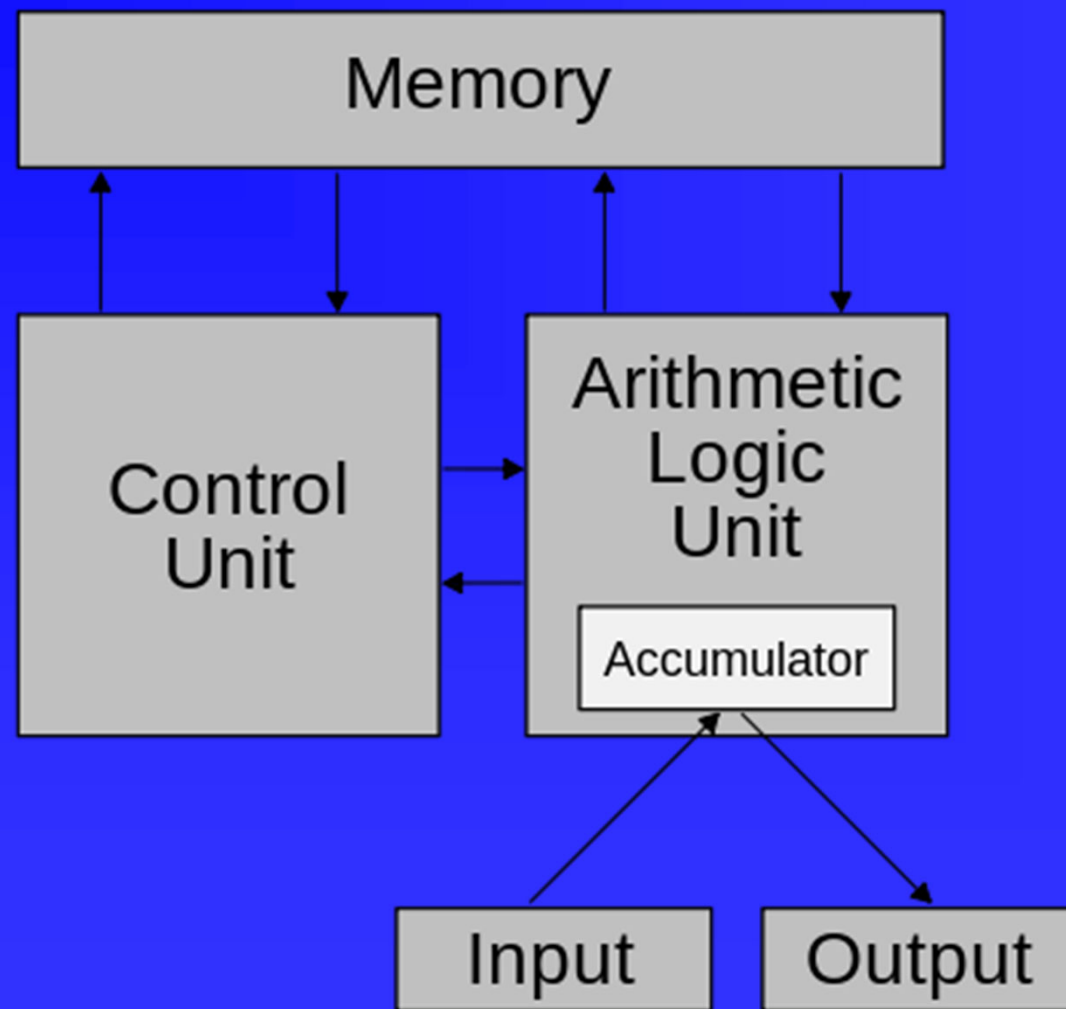


# 冯·诺依曼结构（普林斯顿结构）

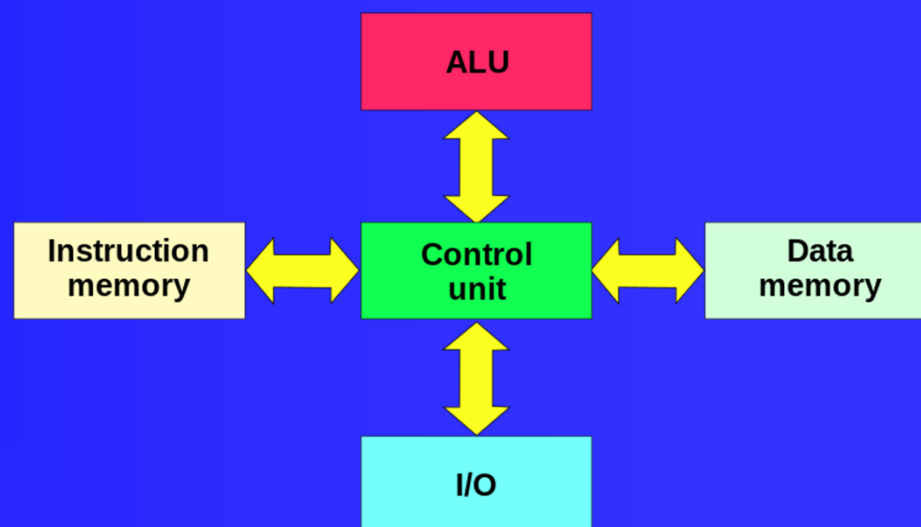
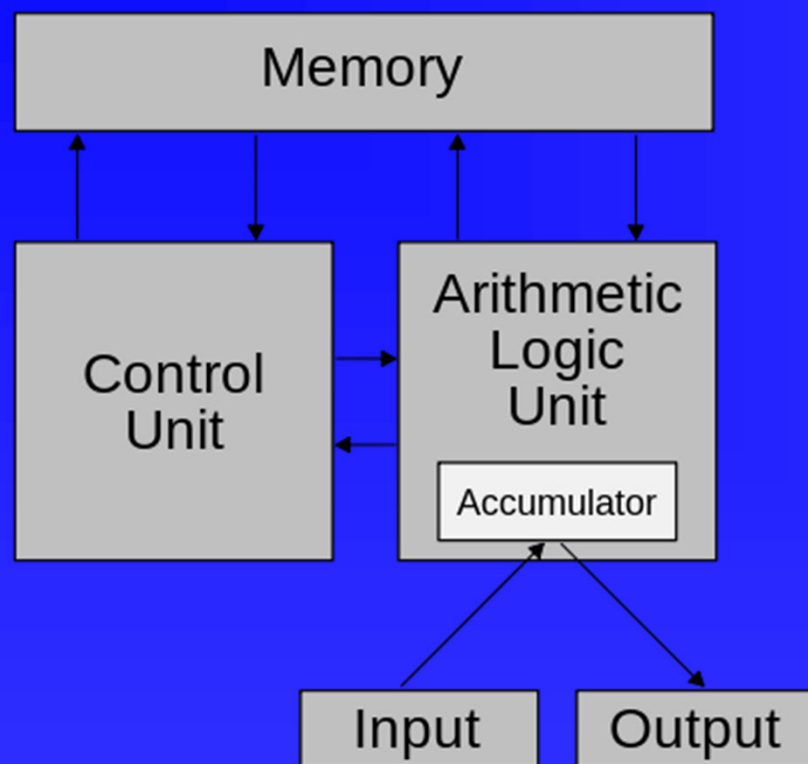
- 计算机由运算器、控制器、存储器、输入设备和输出设备五大部件组成
  - ❑ 存储器（内存）：不仅能存放数据，也能存放指令
  - ❑ 控制器：能自动执行指令
  - ❑ 运算器：能进行算术和逻辑运算以及附加操作
  - ❑ 输入/输出设备：保证人与计算机的相互通信
- 在机器内部，指令和数据均以二进制码表示
  - ❑ 指令由操作码和地址码组成
- 机器以运算器为中心，数据传送都经过运算器
- 采用存储程序（stored program）的方式——编制好的程序和数据存放在同一存储器中，计算机可以在无人干预的情况下自动完成逐条取出指令和执行指令的任务



# 冯·诺依曼体系结构模型



# 哈佛体系结构 (Harvard architecture)

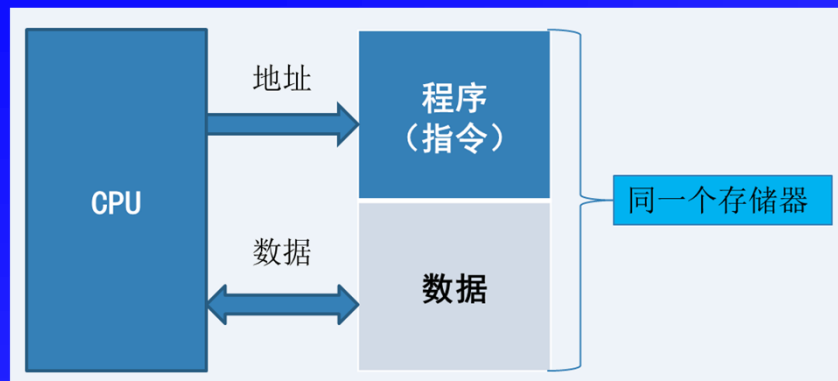


## 哈佛体系结构模型

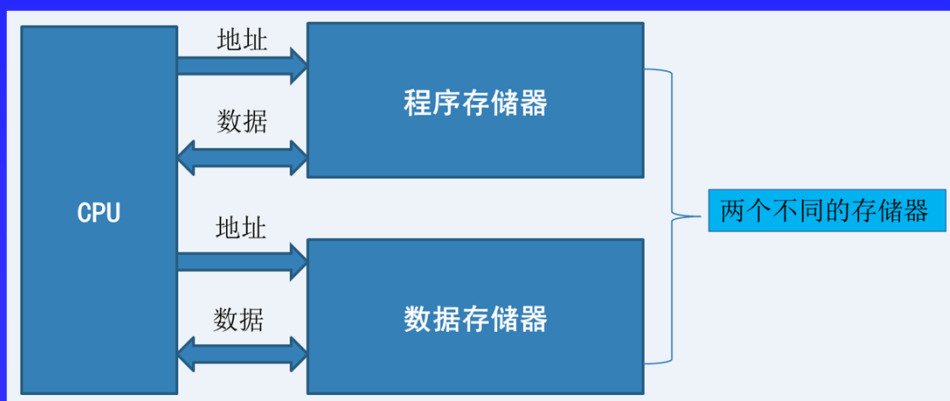
- 冯·诺依曼体系结构模型
  - 程序指令存储器和数据存储器合并到同一存储器中
- 两个独立的存储器模块分别存储指令和数据
- 独立的两条总线



# 冯·诺依曼结构与哈佛体系结构



冯·诺依曼体系结构模型



哈佛体系结构模型

➤ 冯诺依曼体系结构计算机：**Intel 80X86、ARM7、MIPS.....**

➤ 哈佛结构计算机：**Motorola MC68、Zilog Z8、ARM9、ARM10、ARM11.....**

➤ 现代高性能计算机典型结构：

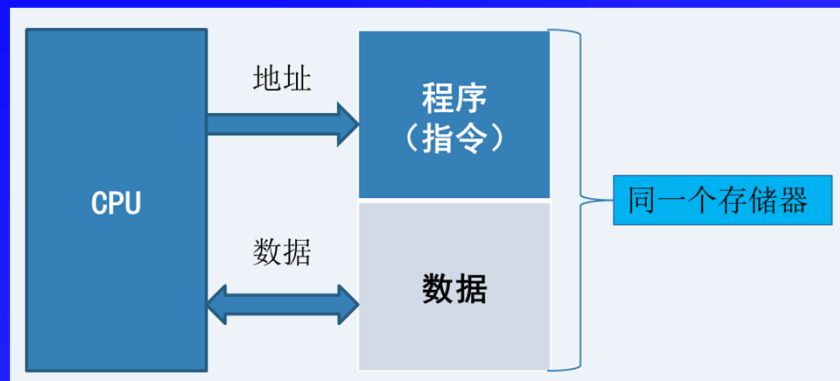
❑ **Cache**采用哈佛结构

❑ 主存采用冯诺依曼结构

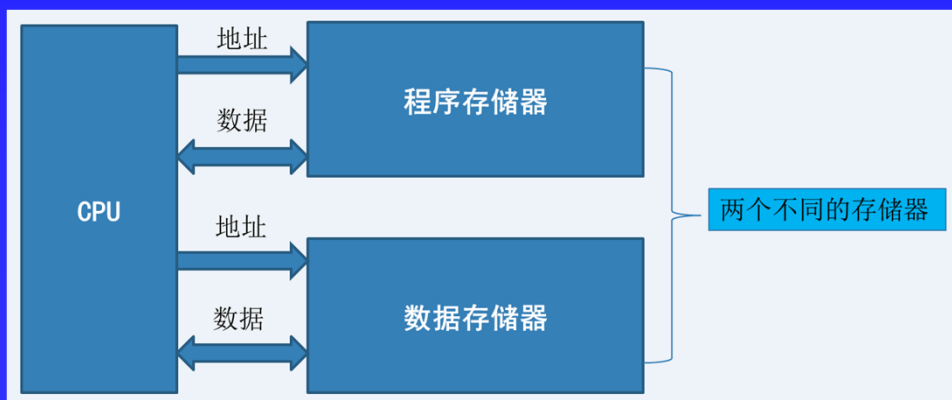




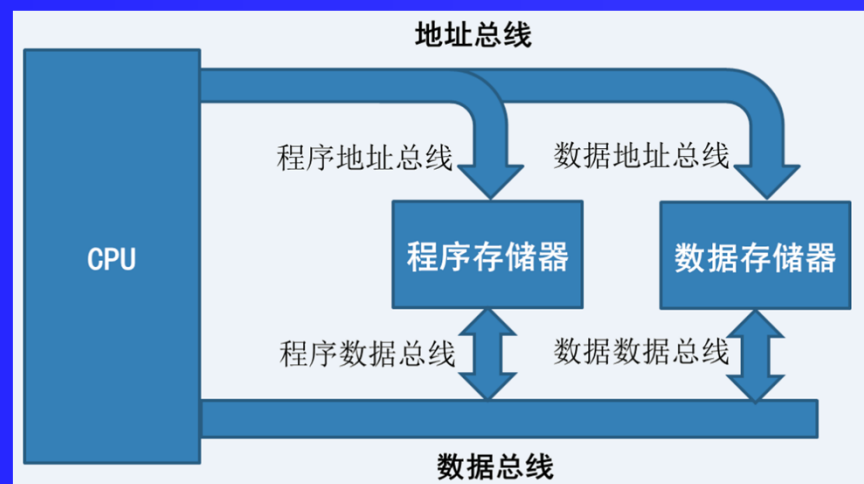
# 冯·诺依曼结构、哈佛体系结构、改进型哈佛结构



## 冯·诺依曼体系结构模型



## 哈佛体系结构模型



## 改进哈佛体系结构模型

### ➤ 两条总线由程序存储器和数据存储器分时共用

- ❑ 公用地址总线访问程序存储模块和数据存储模块
- ❑ 公用数据总线完成程序存储模块或数据存储模块与CPU之间的数据传输



# 非诺依曼化



## ➤ 传统的冯·诺依曼型计算机:

- ❑ 指令流（控制流）驱动
- ❑ 串行顺序处理的工作机制：即使有关数据已经准备好，也必须逐条执行指令序列

## ➤ 提高计算机性能的根本方向之一：并行处理

- ❑ 突破传统冯·诺依曼体制的束缚——非诺依曼化

## ➤ 非诺依曼化的主要表现

- ❑ 在冯·诺依曼体制范畴内进行改造：流水、阵列机
- ❑ 用多个冯·诺依曼机组成多机系统，支持并行算法结构
- ❑ 从根本上改变冯·诺依曼机的控制流驱动方式：如数据流计算机



# 计算机的性能指标



- 处理能力和可用性
- 吞吐量：一台计算机在某一时间间隔内能够处理的信息量
- 响应时间：从输入有效到系统产生响应之间的时间度量
- 利用率：在给定时间间隔内，系统被实际使用的时间所占的比率
- 处理机字长：处理机一次能够完成的二进制运算的位数
- 主频：在单位时间内发出的最基本时间单位的时钟脉冲的数目
  - 单脉冲的时间长度即CPU时钟周期



# 计算机的性能指标

- CPU执行时间 = CPU时钟周期数 × CPU时钟周期长
  - ❑ CPU时钟周期数：程序中所有指令的时钟周期数之和
- CPU执行时间  $T_{CPU} = I_n \times CPI \times T_C$ 
  - ❑  $I_n$ ：程序中的指令总条数
  - ❑  $T_C$ ：CPU时钟周期长
  - ❑ CPI (Cycle Per Instruction)：某程序中执行一条指令所需的平均时钟周期数



# 计算机的性能指标



- CPI: 某程序中执行一条指令所需的平均时钟周期数
- 某程序中所有指令的平均CPI:

□ CPI = 程序中所有指令的时钟周期数之和 / 程序中指令总条数



$$CPI = \frac{\sum_{i=1}^n (CPI_i \times IC_i)}{I_n}$$

$$CPI = \sum_{i=1}^n (CPI_i \times \frac{IC_i}{I_n})$$

⊗ n: 程序中的指令种类数

⊗  $CPI_i$ : 第i类指令执行时一条指令所需的时钟周期数

⊗  $IC_i$ : 第i类指令的条数

⊗  $I_n$ : 程序中的指令总条数

□  $CPI = \sum ( \text{程序中各种指令的CPI} \times \text{程序中该种指令所占的比例} )$



# 计算机的性能指标

➤ CPU执行时间  $T_{\text{CPU}} = I_n \times \text{CPI} \times T_c$

□  $I_n$  : 程序中的指令总条数

□  $T_c$  : CPU时钟周期长

□ CPI (Cycle Per Instruction) : 某程序中执行一条指令所需的平均时钟周期数

□ CPU执行时间  $T_{\text{CPU}} = \sum_{i=1}^n (\text{CPI}_i \times \frac{IC_i}{I_n}) \times I_n \times T_c$



# 计算机的性能指标



## ➤ 平均运算速度:

- 根据不同类型的指令在典型计算过程中出现的比率，对每种类型的指令的执行时间进行加权并求和，得到运算速度的统计平均值
- 单位时间内可执行的指令条数
  - ✉ 定点运算能力：
    - MIPS: Million Instructions Per Second, 百万指令每秒



# 计算机的性能指标

## ➤ 平均运算速度:

### □ 单位时间内可执行的指令条数

✉ 浮点运算能力:

□ **FLOPS**: Floating point number

Operations Per Second, 浮点操作每秒

1 MFLOPS (megaFLOPS) = 每秒1百万 ( $10^6$ ) 次浮点运算

1 GFLOPS (gigaFLOPS) = 每秒10亿 ( $10^9$ ) 次浮点运算

1 TFLOPS (teraFLOPS) = 每秒1兆 ( $10^{12}$ ) 次浮点运算

1 PFLOPS (petaFLOPS) = 每秒1千兆 ( $10^{15}$ ) 次浮点运算





# 计算机的性能指标：例题

设一台计算机主频为1GHz，在其上运行的一段程序共有200000条指令，该程序由4类指令组成，其所占的比例和各自的CPI如下表所示。求平均CPI和MIPS。

指令类型	CPI	指令所占比例
算术和逻辑运算	1	60%
Load/Store	2	18%
转移	4	12%
访问主存（cache失效时）	8	10%

解：

$$\text{平均CPI} = 1 \times 60\% + 2 \times 18\% + 4 \times 12\% + 8 \times 10\% = 2.24$$

$$\begin{aligned} \text{MIPS} &= 1 / (1/f \times \text{CPI}) / 10^6 = 1 \times 10^9 / (2.24 \times 10^6) \\ &= 446.4 \end{aligned}$$



# SPEC测试基准



- 测试基准（Benchmark）程序针对特定的应用环境，通过一组有代表性的程序评估系统的性能
  - ❑ 以程序运行时间长度为度量，或是以单位时间内完成的操作数量为度量
  - ❑ 对性能做出接近真实状况的评估
  - ❑ TPC、SPEC、SAP SD、Linpack、HPCC.....
- SPEC: Standard Performance Evaluation Corporation
  - ❑ 非盈利性第三方应用性能基准测试组织
  - ❑ 根据应用领域的不同设置不同的测试基准
    - ✉ SPEC Cloud IaaS: 针对云计算场景的测试基准
    - ✉ SPECviewperf: 针对图形和工作站性能
    - ✉ .....



# SPEC测试基准



## ➤ SPEC CPU测试基准体系:

- ❑ 度量**运算密集**的计算机系统的性能，并支持在众多计算机系统之间进行比较
- ❑ 重点关注计算机系统的处理器、存储子系统和编译器，不涉及硬盘、输入/输出和网络等部件
- ❑ SPEC CPU 92、SPEC CPU 95、SPEC CPU 2000、SPEC CPU 2006和SPEC CPU 2017等多个版本

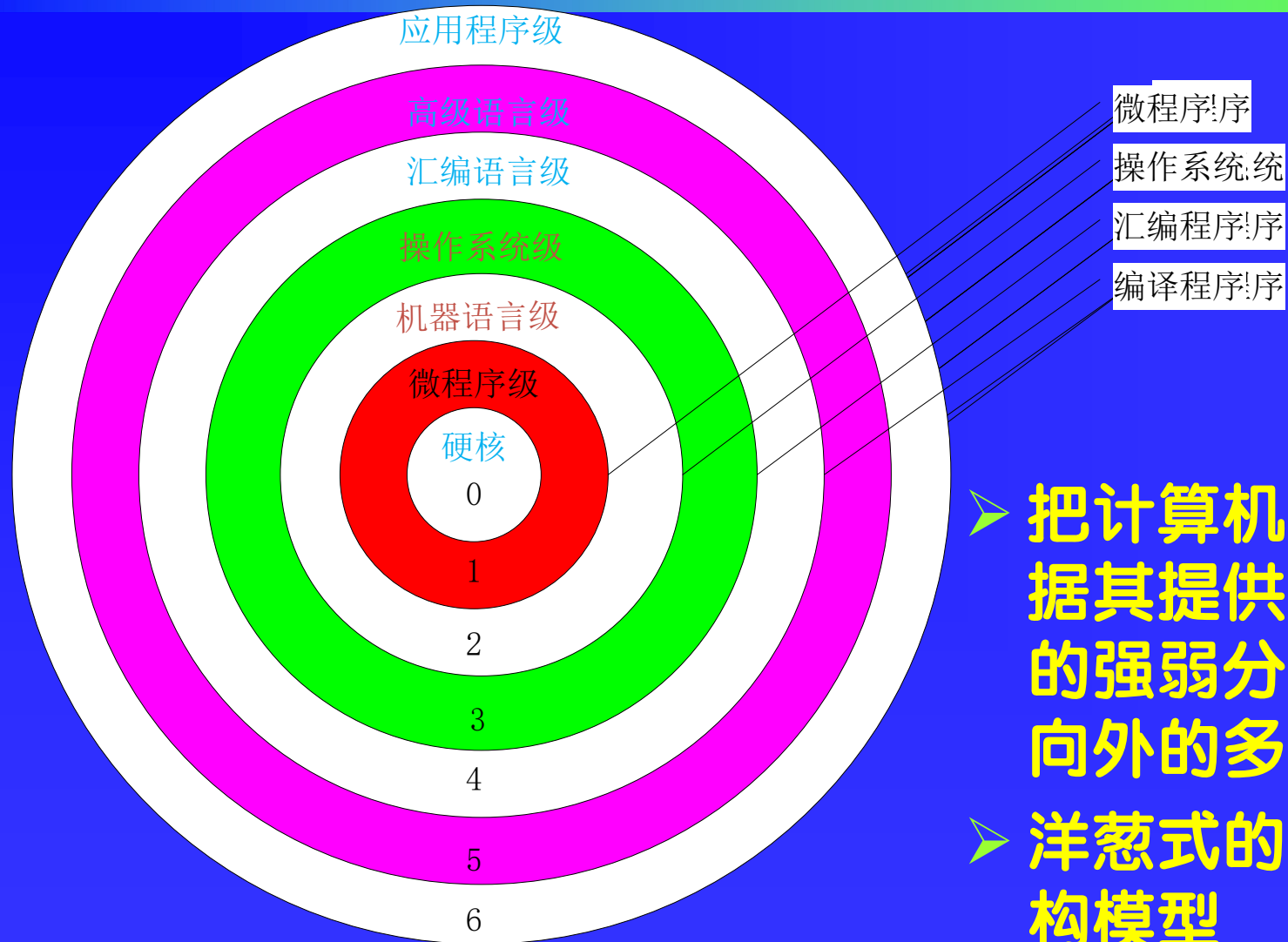
## ➤ <http://spec.org>

## ➤ 鲲鹏920-6426处理器（2.6GHz，64核）

- ❑ SPECint\_rate\_base2006 Benchmark得分超过930
- ❑ 比同档次的业界主流CPU性能高出25%



# 计算机系统的层次结构

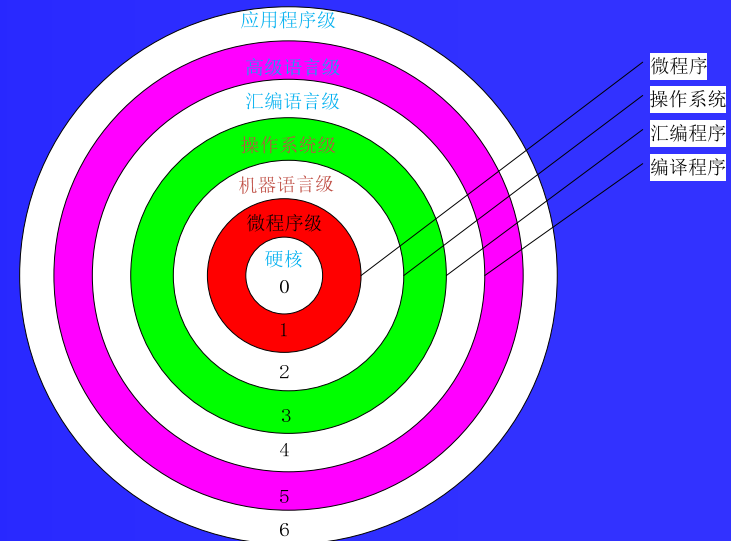


- 把计算机系统依据其提供的功能的强弱分成由内向外的多层
- 洋葱式的功能结构模型



# 计算机系统的层次结构

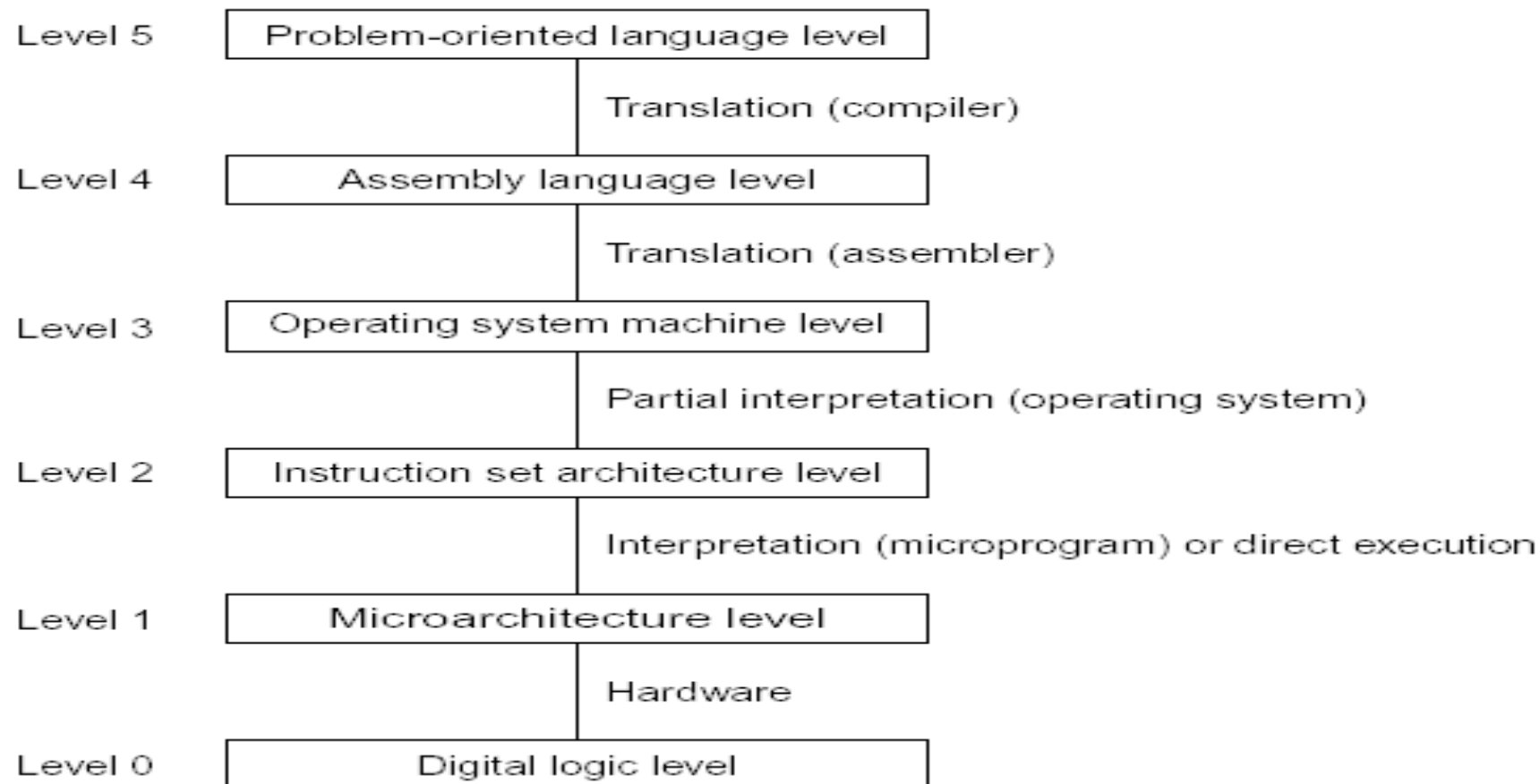
0—2级：物理机（裸机）  
3级以上：虚拟机



- 计算机系统由多级虚拟计算机组成
- 某一层的观察者通过该层的语言来了解和使用计算机，不必关心内部如何工作和实现



# A six-level computer



**Figure 1-2.** A six-level computer. The support method for each level is supported is indicated below it (along with the name of the supporting program).



# A multilevel machine

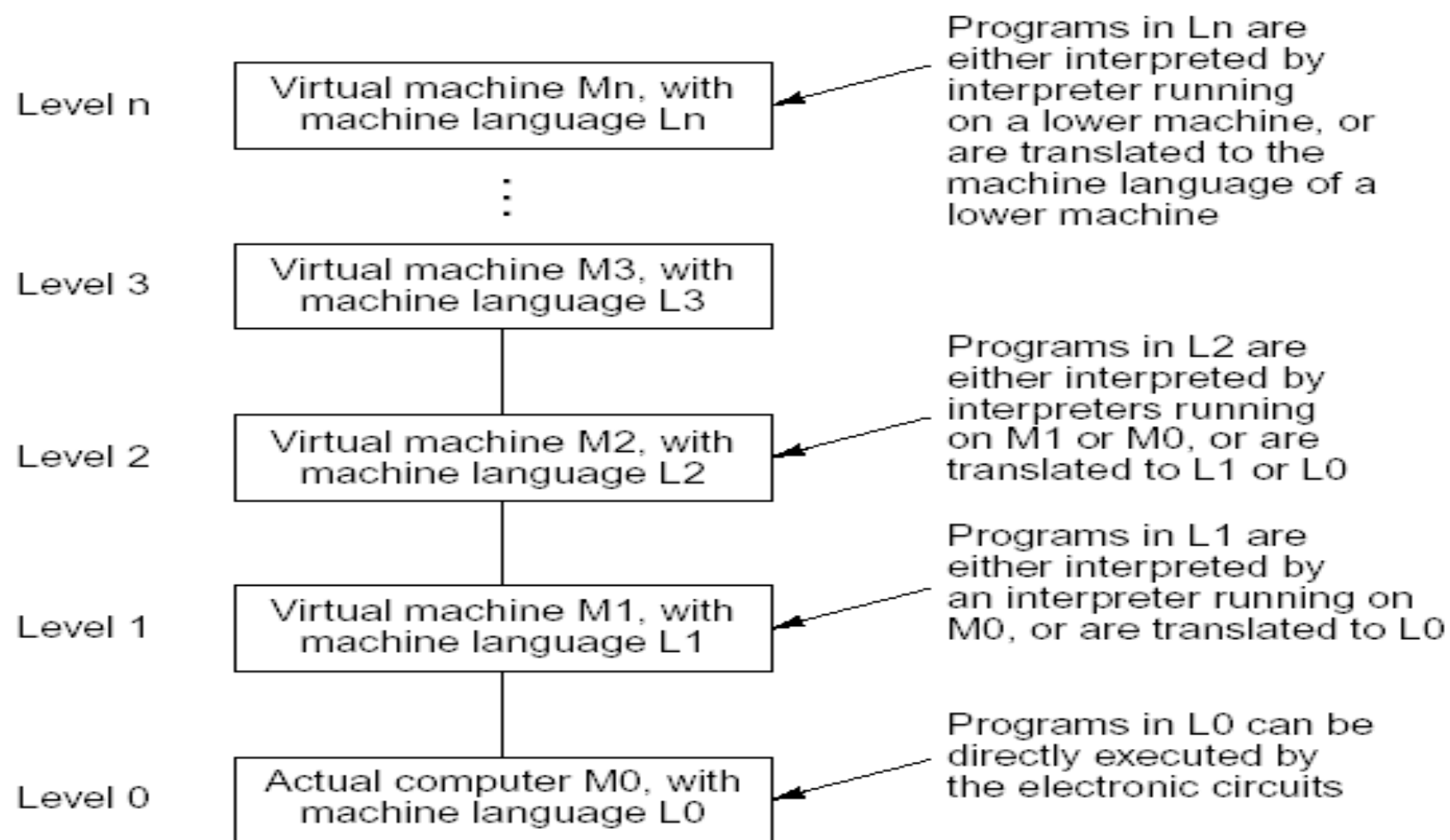


Figure 1-1. A multilevel machine.



# 软件与硬件的逻辑等价性

- 计算机系统与数字系统的区别：计算机系统有软件
- 从理论上说，任何软件算法都可以由硬件实现，反之亦然
- 对于某一具体功能，采用硬件方案还是软件方案，取决于器件价格、速度、可靠性、变更周期等因素

	成本	速度	变更周期
硬件	高	快	长
软件	低	慢	短





# 发展趋势



- 以前一般由软件实现的功能，现在更多地可以直接由硬件实现
- 实体硬件机的功能范围不断在扩大
- 固件（firmware，韧件）：形式上类似硬件，但功能上又象是软件，可以编程和修改
- 软件：“固化”、“硬化”
- 计算机系统的软、硬件界限越来越模糊



# 计算机组成的层次



## ➤ 计算机组成（组织）

□ Computer Organization

## ➤ 计算机结构（体系结构/系统结构）

□ Computer Architecture

## ➤ 计算机实现

□ Computer Implementation



# 计算机组成的层次



## ➤ 计算机组织

- 计算机硬件系统的具体实现
- 包括各操作部件、工作原理、逻辑实现、设计方法，以及将各部件按照“计算机结构”的特性要求连接成整机的方法
- 例如：控制信号、计算机和外设的接口



# 计算机组成的层次



## ➤ 计算机结构

### □ 计算机系统的功能特性

✉ 即硬件能实现哪些功能，能为软件提供哪些硬件支持

□ 是系统程序员与用汇编语言编写程序的用户所看到的硬件系统概念结构、功能特性和系统属性

□ 例如：指令系统、各种数据类型所占的比特数、I/O机制、内存的寻址技术



# 计算机组成的层次



## ➤ 计算机实现

- 计算机组成的物理实现

- 包括处理机、主存等部件的物理结构，器件的集成度、速度和信号，器件、模块、插件、底板的划分与连接，专用器件的设计，电源、冷却、装配等技术



# 计算机组成的层次



- 计算机（系统）结构
  - 指令系统及其执行模型
- 计算机组成
  - 计算机系统结构的逻辑实现
- 计算机实现
  - 计算机组成的物理实现
- 一种系统结构可以有多种组成
- 一种组成可以有多种物理实现



# 计算机组成的层次



- 寄存器级
- 微程序级
- 机器指令级（传统机器级）
- 操作系统级



# 教学安排



- 第一章 计算机系统概论
- 第二章 运算方法和运算器
- 第三章 存储系统
- 第四章 指令系统
- 第五章 中央处理器
- 第六章 总线系统
- 第七章 外围设备
- 第八章 输入输出系统
- 【第九章：并行组织与结构】
- 第十章：课程教学实验设计
- 第十一章：课程综合设计





# 第一章小结



- 【计算机发展历程】
- 计算机硬件的基本组成
  - 冯·诺依曼体系结构
- 【计算机软件的分类】
- 计算机的工作过程
- 计算机系统层次结构
- 计算机性能指标



# 本章重点



- 冯·诺依曼体系结构的特点
- 计算机系统层次结构
- 计算机性能指标



# 作业1



## 单项选择题

1. (2013年考研题) 某计算机主频为1.2GHz, 其指令分为4类, 它们在基准程序中所占比例及CPI如下表所示。

指令类型	所占比例	CPI
A	50%	2
B	20%	3
C	10%	4
D	20%	5

该机的MIPS数是( )。

A. 100                  B. 200                  C. 400                  D. 600

2. 下列说法中不正确的是( )。

(A) 任何可以由软件实现的操作也可以由硬件来实现。

(B) 固件就功能而言类似于软件, 而从形态来说又类似于硬件。

(C) 在计算机系统的层次结构中, 微程序级属于硬件级, 其它四级都是软件级。

(D) 面向高级语言的机器是完全可以实现的。



## 作业2

3. 某计算机主频为1GHz，共有A、B、C三类指令，每类指令的CPI分别为1、2、3。现使用两个不同的编译器分别生成一大段测试软件的二进制代码。编译器1生成的代码含5百万条A类指令、1百万条B类指令和1百万条C类指令；编译器2生成的代码含10百万条A类指令、1百万条B类指令和1百万条C类指令。问：

(1) 根据MIPS计算，哪个编译器生成的代码执行速度快？

(2) 根据CPU执行时间计算，哪个编译器生成的代码执行速度快？



# 计算机组成原理

## Principle of Computer Organization

### ✓第一章 计算机系统概述

本章结束

