

# 第一章 计算机系统概论



计算机系统不同于一般的电子设备，它是由硬件、软件组成的复杂的自动化设备。

本章先说明计算机的分类，然后自上而下、从粗到细的方法，概括地介绍计算机系统硬件、软件的概念和组成，目的在于建立一个较粗的计算机系统整机概念，以便详细展开后续各章内容。

# 内容简介



- 计算机的分类
- 计算机的发展简史
- 计算机的硬件
- 计算机的软件
- 计算机系统的层次结构
- 冯.诺依曼计算机





## ☞ 基本概念

硬件、软件、固件

存储程序控制方式

计算机系统层次结构、虚拟计算机

## ☞ 计算机的性能指标

☞ 计算机硬件系统的概念性结构，各个部分的作用。

☞ 指令流、数据流？计算机如何区分指令和数据？

☞ 冯·诺依曼计算机的技术特点



# 课后作业



P15: 4、7



# 1.1 计算机的分类

## —— 模拟计算机与数字计算机



模拟计算机：

数值由连续量来表示，运算过程也是连续的。

如电表用角度来反映电量大小、时钟用指针在表盘上转动来表示时间。



# 1.1 计算机的分类

## —— 模拟计算机与数字计算机



模拟化与数字化：

**模拟化：**利用装置转换为连续变化的电压/电流信号。

**数字化：**通过模/数转换，转换为离散的数字量。



# 1.1 计算机的分类

## —— 模拟计算机与数字计算机



数字计算机：

用数字来表示数量的大小。

按位运算，不连续地跳动计算。

基本数字：0、1，用电平高低、正负跳变、脉冲信号有无来表示。





# 1.1 计算机的分类

## — 专用计算机与通用计算机

专用计算机：

针对特定应用领域设计的，是最有效、或最经济、或最快速的。

如用于科学计算的超级计算机；用于工业控制的工控机；用于智能仪器仪表的单片机。

# 1.1 计算机的分类

## — 专用计算机与通用计算机

通用计算机：

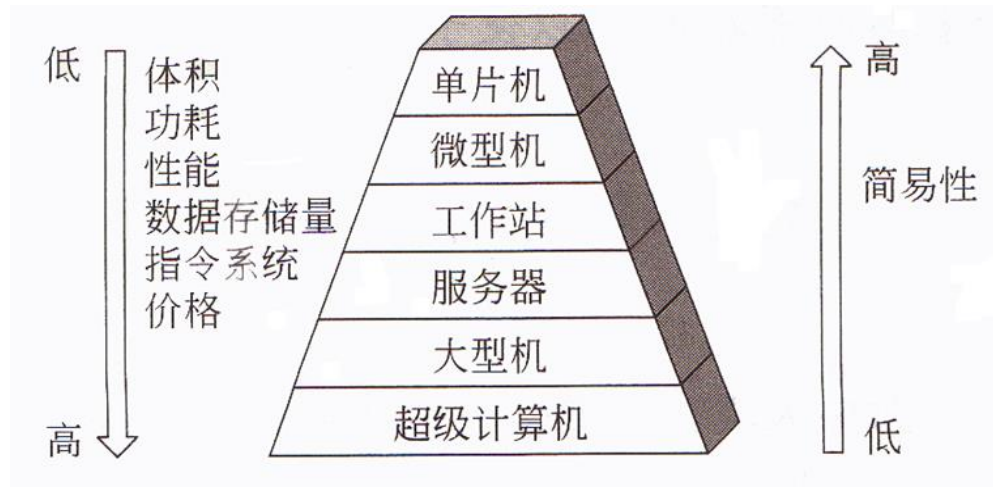
通用计算机适应性很强，但是往往牺牲了效率、速度和经济性。

如通用微机、工作组级工作站、服务器（PC服务器、刀片集群系统、通用高性能小型机）。

# 1.1 计算机的分类

## — 专用计算机与通用计算机

### 通用计算机的分类：



**相对性：**分类应该说是相对的，随着计算机技术和微电子技术的发展，计算机的结构规模、体积在逐步减小，而集成度、性能指标越来越强。

未来，一台与微机规模相同的**量子计算机**，可达到现在超级计算机的计算性能。

# 1.1 计算机的分类

## — 专用计算机与通用计算机

单片机:

单片机 = 中央处理器CPU + 外围接口 (串口、并口、定时器计数器、中断系统、高速I/O接口、A/D+D/A等) 集成在一块硅片上

= 一块芯片 (需要开发装置和运行配套装置)

# 1.1 计算机的分类

## —— 专用计算机与通用计算机



多核机：

一个CPU芯片中有多套运算器+控制器；

多核协同工作机制。



## 1.2 计算机的发展简史



- 计算机的五代变化
- 半导体存储器的发展
- 微处理器的发展
- 计算机的性能指标



## 1.2.1 计算机的五代变化



第一代 电子管计算机, 1946-1957

第二代 晶体管计算机, 1958-1964, 工控机

第三代 中小规模集成电路计算机, 1965-1971, 小型机

第四代 大规模/超大规模集成电路计算机, 1972-1990, 微机

第五代 巨大规模集成电路计算机, 1991-, 单片机



## 1.2.2 半导体存储器的发展



从1970年起，半导体存储器单个芯片1KB、4KB、16KB、64KB、256KB、1MB、4MB、16MB、256MB、1GB，存储密度不断增加，位成本持续下跌。

RAM、ROM、PROM、EPROM、E<sup>2</sup>PROM、FLASH。





## 1.2.3 微处理器的发展



Intel公司微处理器的演化：4004（4位）、8008（8位）、8080、8086、8088、80286（16位）、80386（32位）、80486、奔腾Pentium、Pentium Pro（64位）、Pentium II、Pentium III、Pentium 4、安腾Itanium、Itanium 2。



## 1.2.4 计算机的性能指标



**吞吐量：**表征一台计算机在某一时间间隔内能够处理的信息量，用bps度量。

**响应时间：**表征从输入有效到系统产生响应之间的时间度量，用时间单位来度量。

**利用率：**在给定的时间间隔内，系统/部件被实际使用的时间所在的比率，用百分比表示。如CPU利用率。



## 1.2.4 计算机的性能指标



**处理机字长：**常称**机器字长**，指处理机运算中一次能够完成二进制运算的位数，如32位机、64位机。

**总线宽度：**一般指CPU中的运算器与内存储器之间互连总线完成一次操作可传输的二进制位数。

**存储器容量：**存储器中所有存储单元（通常是字节）的总数目，通常用KB、MB、GB、TB来表示。



## 1.2.4 计算机的性能指标



**存储器带宽：**单位时间内从存储器读出的二进制数信息量，一般用B/s（字节/秒）表示。

**主频/时钟周期：**CPU的工作节拍受主时钟控制（控制器功能之一时间控制），按照规定在某个时间段做什么、从什么时候开始、多长时间必须完成。主时钟不断产生固定频率的时钟信号，可通过分频形成多级时序。

主时钟的频率叫主频（ $f$ ），度量单位是MHz、GHz。主频的倒数称为时钟周期（ $T$ ）， $T=1/f$ ，度量单位是 $\mu s$ 、 $ns$ 。



## 1.2.4 计算机的性能指标



**CPU执行时间：**表示CPU执行一段程序所占用的CPU时间。

$\text{CPU时间} = \text{CPU时钟周期数} \times \text{CPU时钟周期长}。$

**CPI：**执行一条指令所需要的平均时钟周期数。

$\text{CPI} = \text{执行某段程序所需的CPU时钟周期数} / \text{该程序包含的指令条数}。$



## 1.2.4 计算机的性能指标



**MIPS:** 平均每秒执行多少百万条定点指令数。

$$\text{MIPS} = \text{指令条数} / (\text{程序执行时间} \times 10^6)$$

**FLOPS:** 平均每秒执行浮点操作的次数，用来衡量机器浮点操作的性能。

$$\text{FLOPS} = \text{程序中的浮点操作次数} / \text{程序执行时间 (秒)}$$



## 1.2.4 计算机的性能指标



【例1】对于一个给定的程序， $I_N$ 表示执行程序中的指令条数， $t_{CPU}$ 表示执行该程序所需的CPU时间， $T$ 为时钟周期， $f$ 为时钟频率（ $T$ 的倒数）， $N_c$ 为CPU时钟周期数。设CPI表示每条指令的平均时钟周期数，MIPS表示每秒钟执行的百万条指令数，请写出如下四种参数的表达式。

(1)  $t_{CPU}$     (2) CPI    (3) MIPS    (4)  $N_c$

【解】 (1)  $t_{CPU} = N_c \times T = \frac{N_c}{f} = I_N \times CPI \times T = \left( \sum_{i=1}^n CPI_i \times I_i \right) \times T$

$$(2) CPI = \frac{N_c}{I_N} = \sum_{i=1}^n \left( CPI_i \times \frac{I_i}{I_N} \right)$$

$$(3) MIPS = \frac{I_N}{t_{CPU} \times 10^6} = \frac{I_N}{N_c \times T \times 10^6} = \frac{f}{CPI \times 10^6}$$

$$(4) N_c = \sum_{i=1}^n (CPI_i \times I_i)$$

$I_i$ :  $i$ 种指令在程序中执行的次数

$CPI_i$ :  $i$ 种指令所需平均时钟周期数

$n$ : 指令种类

$I_i/I_N$ :  $i$ 种指令在程序中所占比例

## 1.2.4 计算机的性能指标

【例2】用一台50MHz处理机执行标准测试程序，它包含的混合指令数和相应所需的时钟周期数如下表所示。

【解】

指令类型	指令数目	CPI <sub>i</sub>
整数运算	45000	1
数据传送	32000	2
浮点运算	15000	2
控制传送	8000	2

$$CPI = \frac{N_C}{I_N} = \sum_{i=1}^n (CPI_i \times \frac{I_i}{I_N}) \text{ 其中 } \frac{I_i}{I_N} \text{ 表示 } i \text{ 指令在程序中的比例}$$
$$= \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{45000 + 32000 + 15000 + 8000} = 1.55 (\text{周期/指令})$$

$$MIPS = \frac{f}{CPI \times 10^6} = \frac{50 \times 10^6}{1.55 \times 10^6} = 32.26 (\text{百万条指令/秒})$$

$$t_{CPU} = \frac{N_C}{f} = \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{50 \times 10^6} = 31 \times 10^{-4} (\text{秒})$$



# 1.3 计算机的硬件



☞ 硬件组成要素

☞ 运算器

☞ 存储器

☞ 控制器

☞ 适配器与输入输出设备

☞ 组织结构（总线）



# 1.3 计算机的硬件



软件

应用软件

支撑软件

系统软件

操作系统OS

硬件

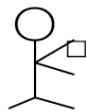
硬件（裸机）

网络（网卡、协议）

计算机系统=硬件子系统+软件子系统



# 1.3 计算机的硬件



软件

编辑软件office

输入法软件

汉字系统

操作系统OS

硬件

硬件（裸机）

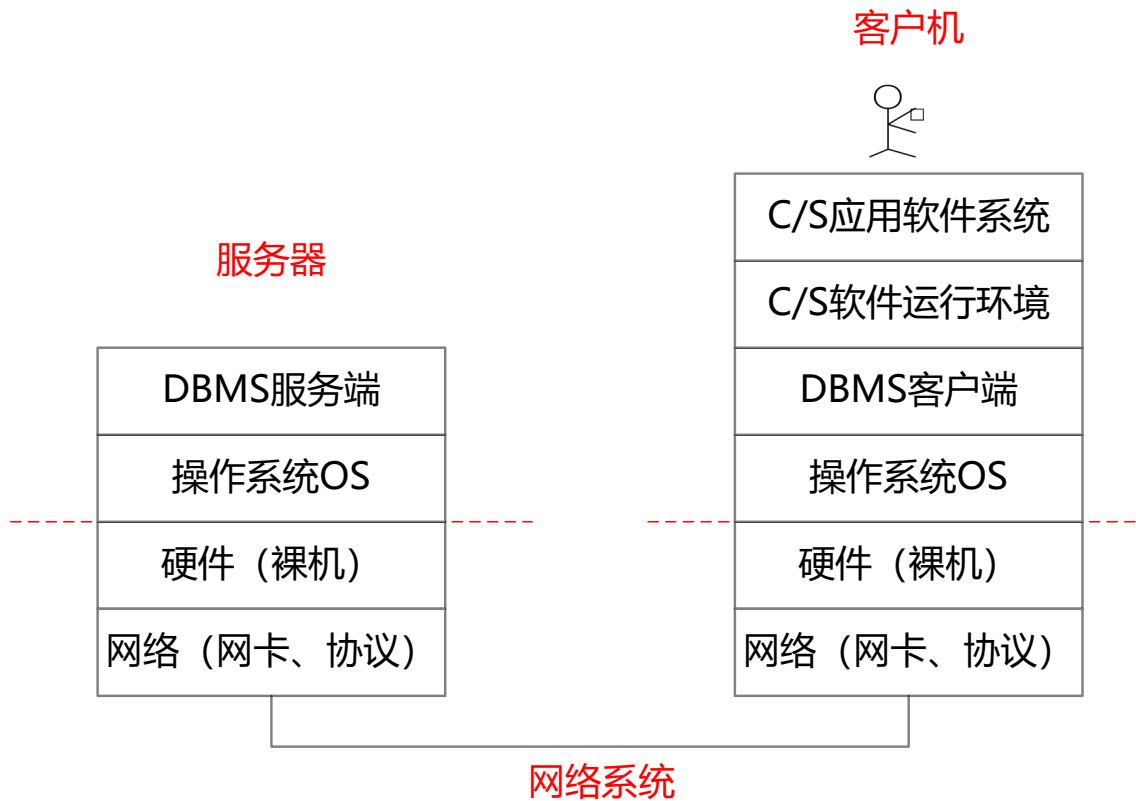
计算机系统：文字处理



# 1.3 计算机的硬件



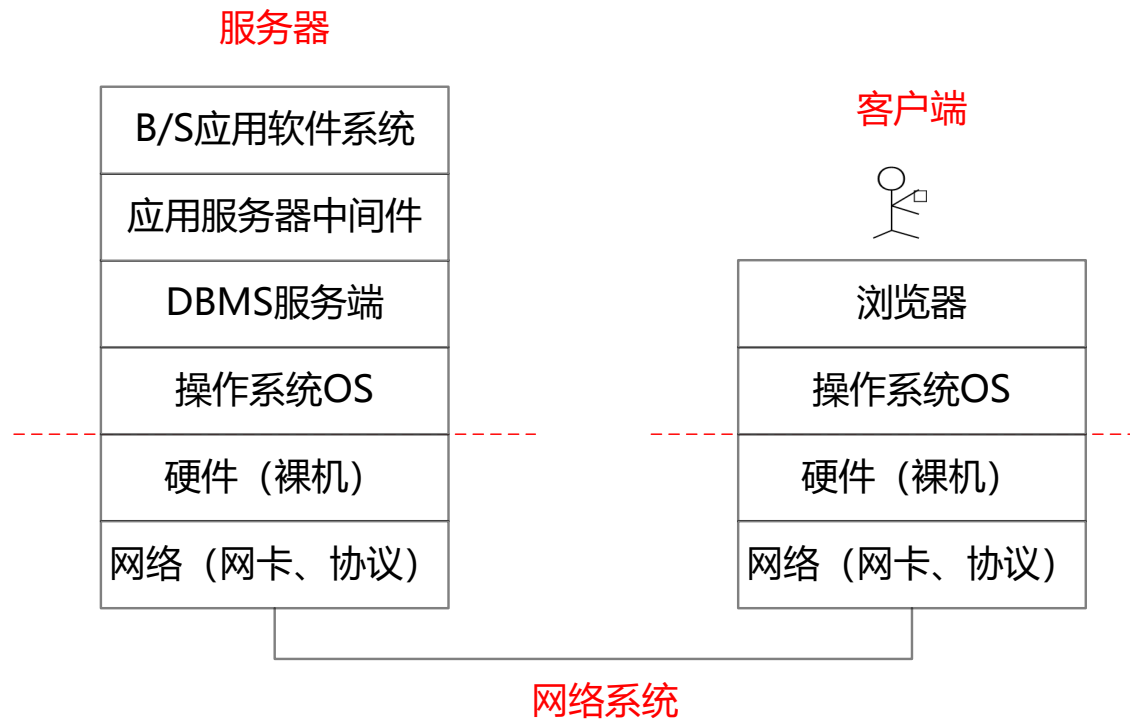
## C/S软件结构



# 1.3 计算机的硬件



## B/S软件结构



## 1.3 计算机的硬件



**硬件：**是指可以看得见、摸得着的物理设备（部件）实体，一般讲硬件还应包括将各种硬件设备有机组织起来的体系结构。

**软件：**程序（代码）+ 数据 + 文档 = 算法 + 数据结构 + 数据 + 文档。软件由两部分组成，一是使计算机硬件能完成计算和控制功能的有关计算机指令和数据定义的组合，即机器可执行的程序及有关数据；二是机器不可执行的，与软件开发、过程管理、运行、维护、使用 and 培训等有关的文档资料。



## 1.3 计算机的硬件



**文 档：** 技术文档 + 管理文档。

**技术文档：** 包括设计阶段的需求报告、需求分析报告、总体设计报告、详细设计报告、测试计划书；测试阶段的单元、集成、系统测试报告；操作与维护手册、安装维护手册、运行记录、维护记录等。

**管理文档：** 包括软件过程定义、规范、进度计划、评审报告、变更管理、不符合项跟踪表等。



## 1.3 计算机的硬件



**计算机系统：**硬件、软件的有机结合、缺一不可。计算机应用系统的建设应该两手都要抓，既重视硬件建设、又重视软件建设。一个软件项目成功的保证应是硬件、软件、配套制度建设的有机融合，而不是手工模式的模仿。





# 1.3.1 硬件组成要素



利用一张纸、一个  
算盘、一支笔计算  
 $y=ax+b-c$ 的演算过程  
，从而描绘出数字计算  
机硬件组成的轮廓。

行数	解题步骤和数据	说 明
1	取数 (9)→算盘	(9)表示第9行的数a,下同
2	乘法 (12)→算盘	完成 $a*x$ , 结果在算盘上
3	加法 (10)→算盘	完成 $ax+b$ , 结果在算盘上
4	减法 (11)→算盘	完成 $ax+b-c$ , 结果在算盘上
5	存数 $y \rightarrow 13$	算盘上的y值记到第13行
6	输出	把算盘上的y值写出给人看
7	停止	运算完毕, 暂停
8		
9	a	数据
10	b	数据
11	c	数据
12	x	数据
13	y	数据



## 1.3.1 硬件组成要素



### 算盘计算

**纸：**用于存储解题的原始信息（数据、步骤）。

**算盘：**用于数据的加、减、乘、除等算术运算。

**笔：**用于记录原始数据、解题步骤、运算结果。

**人本身(主要是大脑)：**编写并按解题步骤控制。

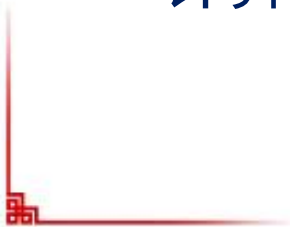
**运算器：**相当于算盘功能的部件。

**存储器：**相当于纸那样具有“记忆”功能的部件。

**输入输出设备：**相当于笔，把原始解题信息输入到计算机或把运算结果显示出来的设备。

**控制器：**相当于人的大脑，自动控制整个计算过程。

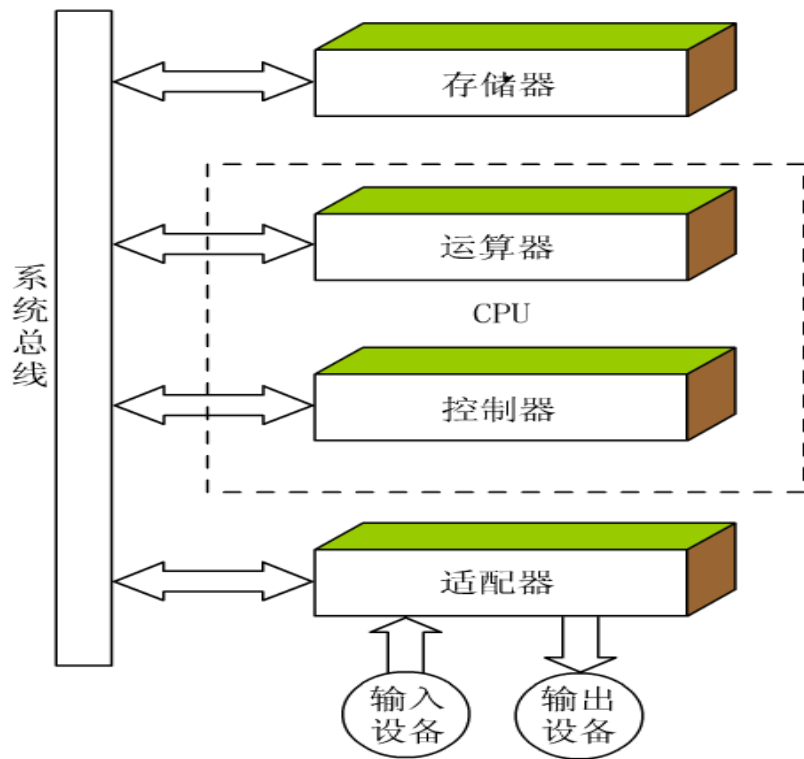
### 计算机



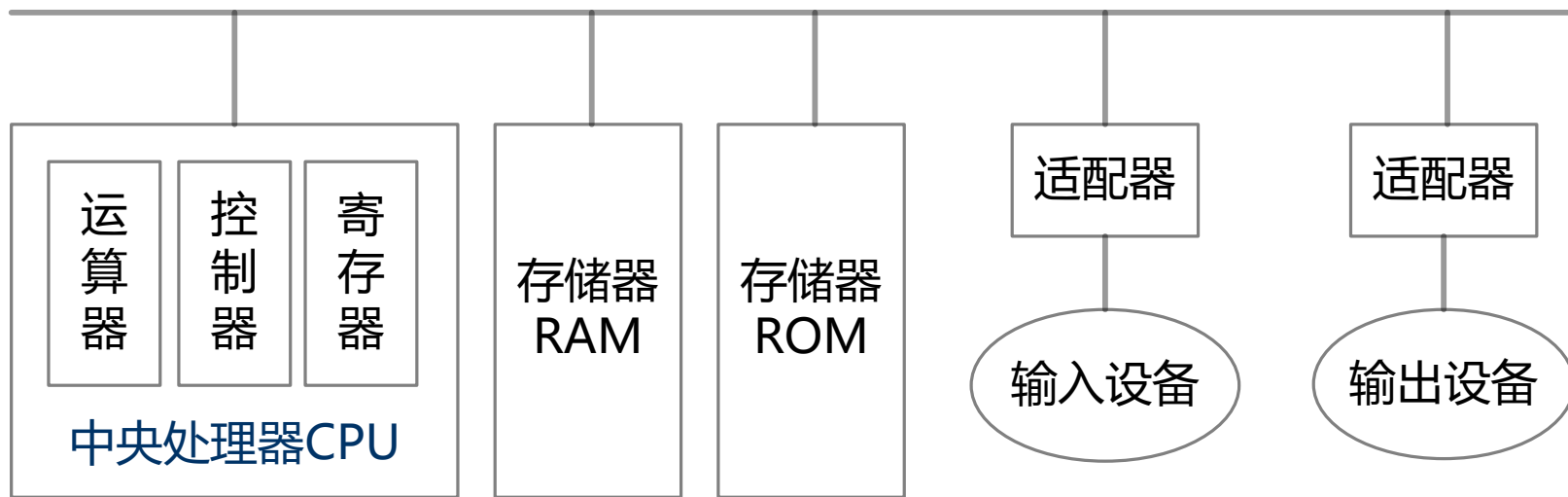
# 1.3.1 硬件组成要素

硬件组成要素 =

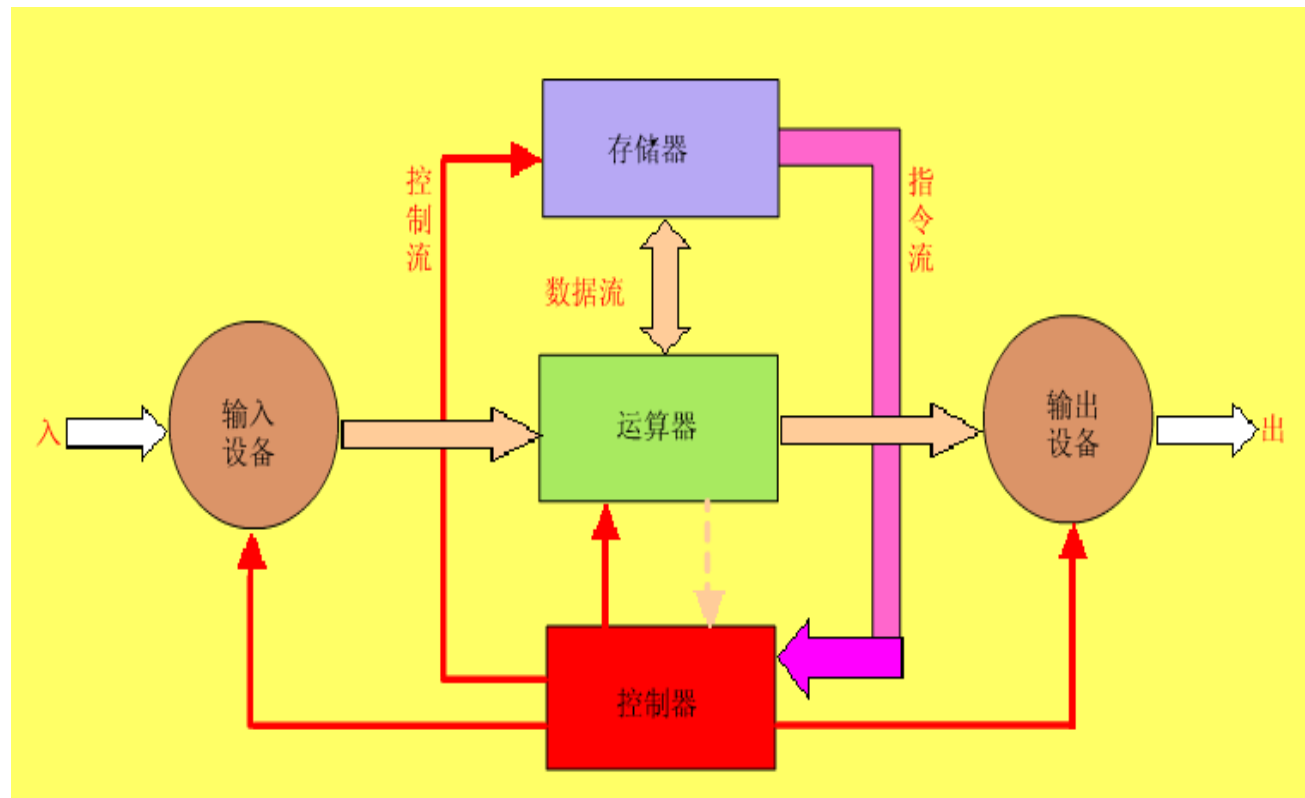
运算器 + 控制器 + 存储器 + 输入设备 + 输出设备 + 连接结构 (总线)



# 1.3.1 硬件组成要素



# 1.3.1 硬件组成要素



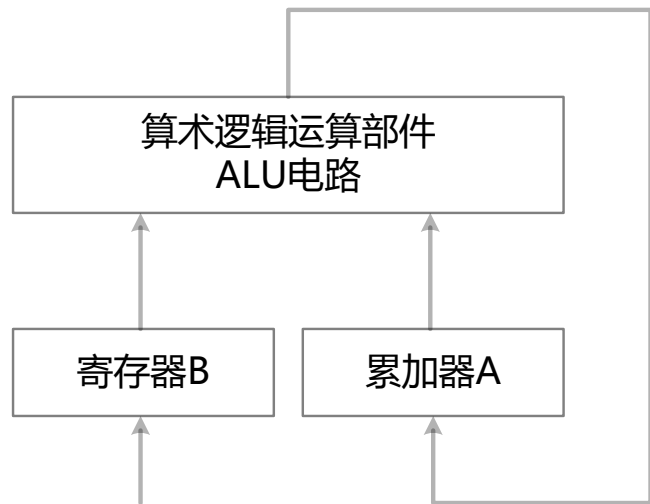
冯.诺依曼  
计算机



## 1.3.2 运算器



运算器 = 算术逻辑运算部件ALU + 寄存器 + 数据通路



操作:  $A \text{ op } B \rightarrow A$

$A + B \rightarrow A$

A、B均是寄存器，由于A保存结果、赋予其他更多功能，称为**累加器**。

## 1.3.2 运算器



### 1、计算机中采用二进制运算

人们习惯于十进制数的运算，但是由于二进制数的运算规律简单，电子线路容易实现，实现设备最省，因此，计算机中采用二进制运算。

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=\textcolor{red}{1}0$$



## 1.3.2 运算器



### 2、机器字长

运算器进行一次二进制数运算的位数。

位数越长、相应地计算精度越高。

机器字长通常有8位、16位、32位、64位、128位，一般是 $N \times 8$ 位。相应地称为8位机、16位机、32位机、64位机、...



## 1.3.2 运算器



### 3、广义运算

算术运算、逻辑运算、移位运算、关系（比较）运算、位运算、地址运算



# 1.3.3 存储器



## 1、功能

保存或“记忆”解题的数据和解题步骤，即程序（代码）和数据。程序和数据都是**二进制代码化**的。



## 1.3.3 存储器



### 2、内部存储器、内存、主存

容量小、存取速度快；

CPU可直接访问；

半导体存储器，掉电后信息丢失；

基本存储元是触发器，一个触发器可以存储一个二进制位。



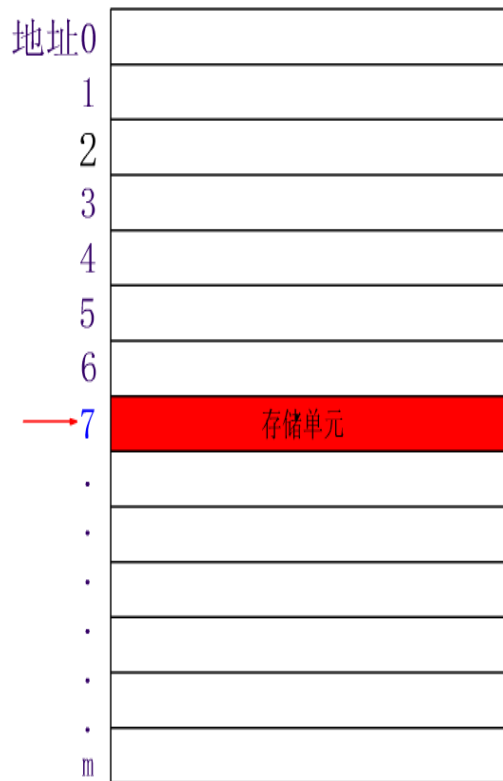
# 1.3.3 存储器



## 2、内部存储器、内存、主存

若干触发器并列组成一个存储单元，  
存储若干二进制位（ $N \times 8$ ），典型是1个  
字节（8个二进制位）。

内部存储器由若干的存储单元构成，  
每个内存单元都有唯一的地址编号（物理  
地址、十六进制）。



# 1.3.3 存储器



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0																	00000H
1																	00001H
2																	00002H
3																	00003H
4	.....																00004H
$2^{20}-3$																	FFFFDH
$2^{20}-2$																	FFFFEH
$2^{20}-1$																	FFFFFH



## 1.3.3 存储器



### 2、内部存储器、内存、主存

位 (bit) : 一个二进制位

存储单元: 若干二进制位、 $N \times 8$ 位、典型8位

字节 (byte) : 8个二进制位

字 (word) : 与机器字长相同, 通常 $N \times 8$ 位

物理地址: 唯一的地址编号, 通常十六进制表示

地址空间范围: 如00000H-FFFFFFH

存储容量: 存储单元数 $\times$ 存储单元位数, KB MB GB TB

## 1.3.3 存储器



### 3、外部存储器、外存、辅存

容量大、存取速度慢；

磁性材料、可永久保存；

CPU不能直接访问，通过操作系统将外存的内容加载到内存；

无直接的地址空间概念（软盘：磁道、扇区；硬盘：柱面、磁道、扇区）；

虚拟存储器技术。

## 1.3.3 存储器



### 4、存储器的操作

读操作（取操作）

写操作（取操作，写后读验证、错误校验）

保持操作

刷新操作（DRAM）





## 1.3.4 控制器



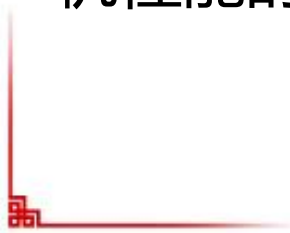
### 1、机器指令

CPU能够直接执行;

二进制代码化;

CPU可执行的所有机器指令的集合称为**指令系统**;

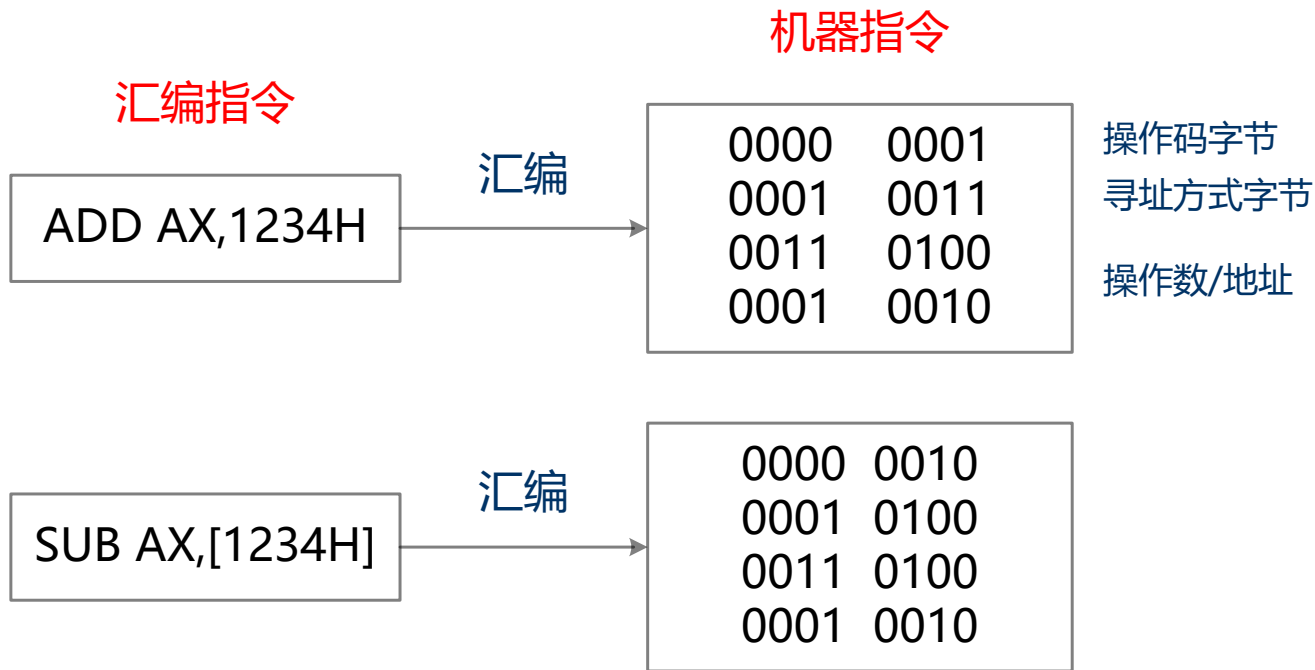
指令系统是硬件设计的依据、软件设计的基础,是衡量计算机性能的重要指标。



# 1.3.4 控制器



## 1、机器指令



## 1.3.4 控制器



### 2、汇编指令

**格 式：** 操作码 操作数 {, 操作数}

**操作码：** 告诉计算机完成什么操作

**操作数：** 告诉计算机如何找到指定运算所需要的操作数（可能直接给出操作数、也可能给出操作数存放的内存单元地址、I/O端口地址）

## 1.3.4 控制器



### 2、汇编指令

ADD AX, 1234H

SUB AX, [1234H] ; 内存地址

ADD AX, BX ; 内部寄存器

IN AL, 80H ; I/O端口地址

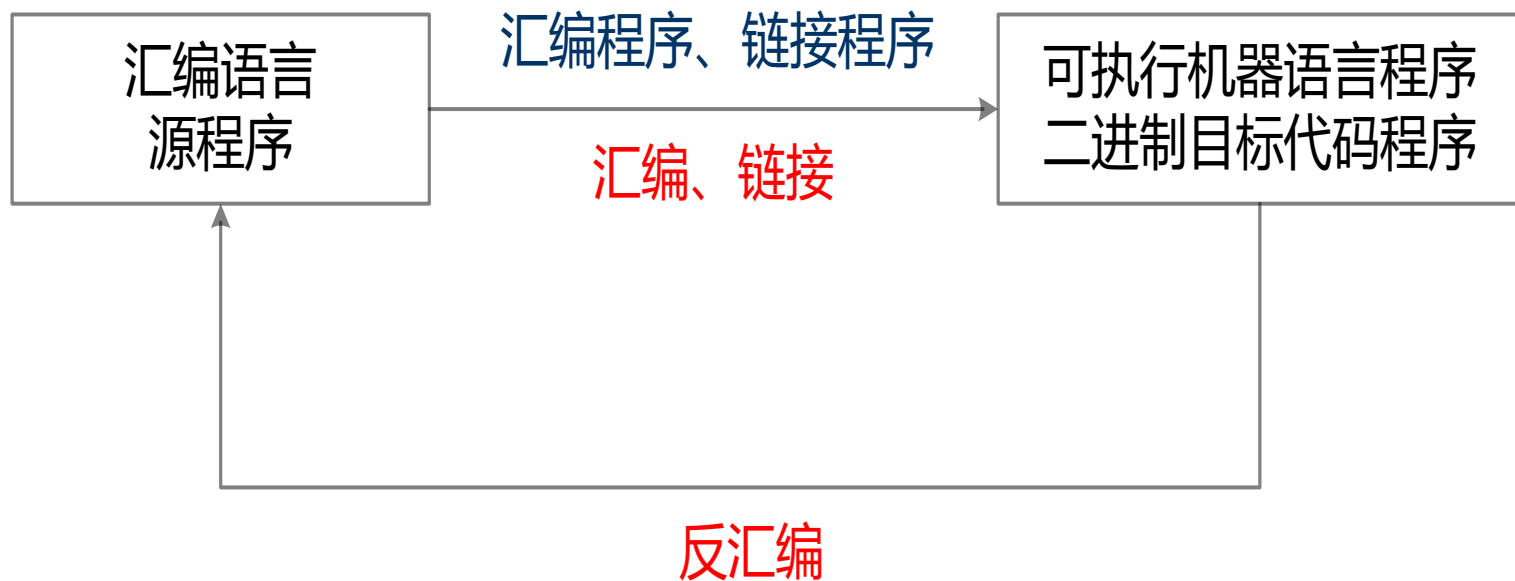
OUT 80H, AL ; I/O端口地址



## 1.3.4 控制器



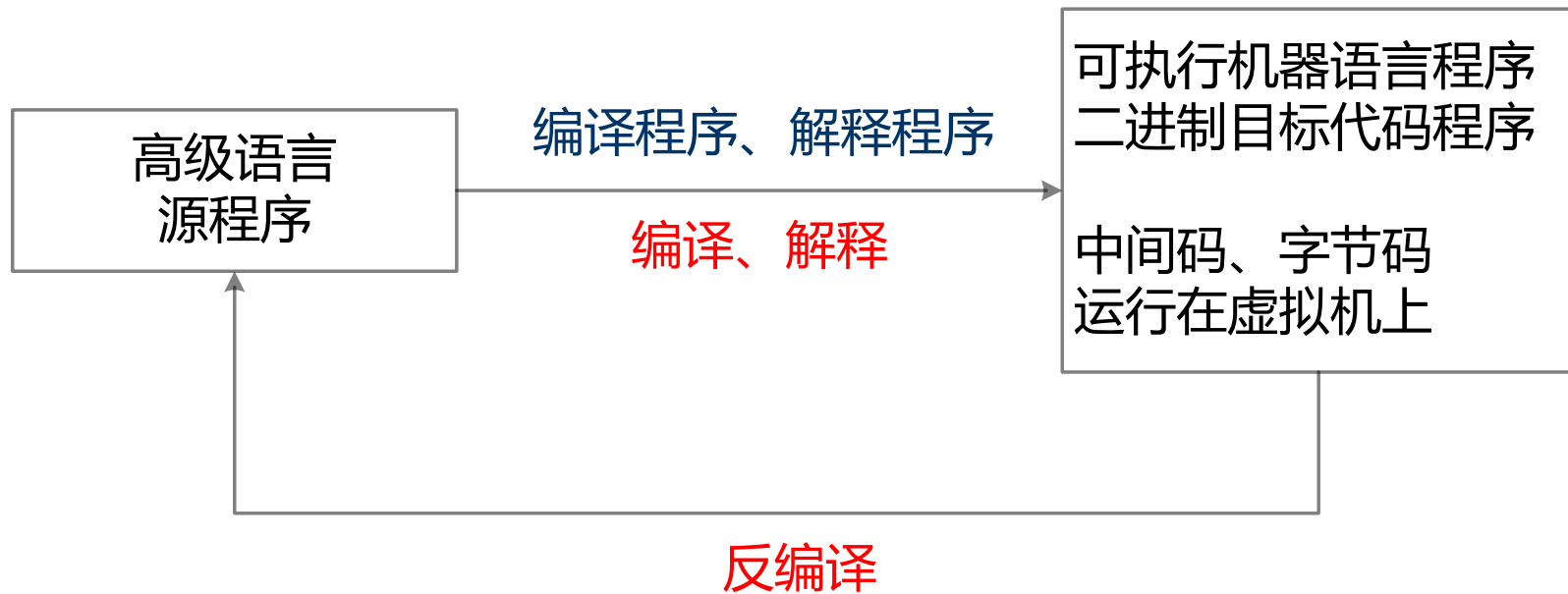
### 2、汇编指令



## 1.3.4 控制器



### 3、高级语言语句：近似自然语言书写



## 1.3.4 控制器



### 4、程序（计算程序）

解决某一问题的一串有序的指令/语句序列，称为该问题的计算程序，简称为程序。



# 1.3.4 控制器

## 5、程序的二进制代码化

行数	解题步骤和数据	说明
1	取数(9)→算盘	(9)表示第9行的数a,下同
2	乘法(12)→算盘	完成 $a \times x$ , 结果在算盘上
3	加法(10)→算盘	完成 $ax+b$ , 结果在算盘上
4	减法(11)→算盘	完成 $y=ax+b-c$ , 结果在算盘上
5	存数 $y \rightarrow 13$	把算盘上的y值记到第13行
6	输出	把算盘上的y值写出给人看
7	停止	运算完毕, 暂停
9	a	数据
10	b	数据
11	c	数据
12	x	数据
13	y	数据

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
⋮	

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



## 1.3.4 控制器



### 6、存储程序

将程序和数据存放到内存储器中，称为存储程序。一般将程序和数据分开存放。

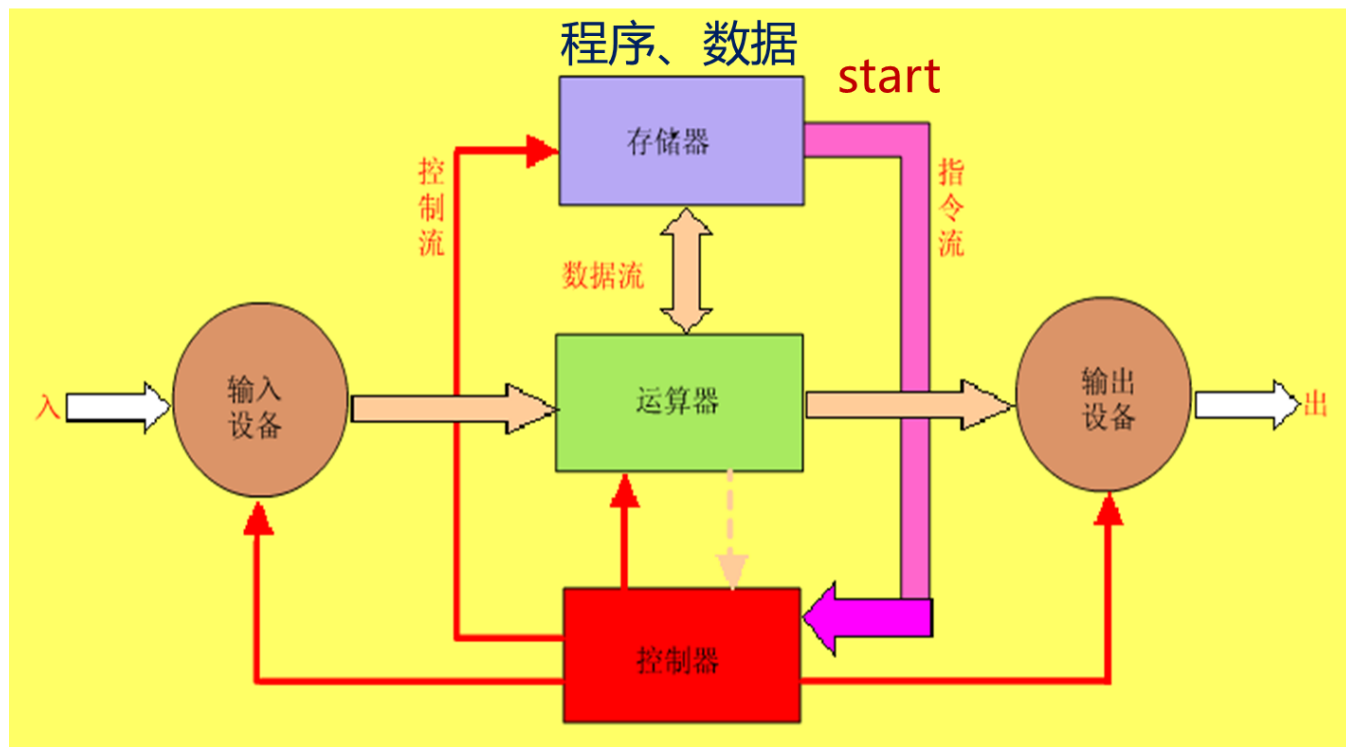
### 7、存储程序控制方式

控制器按地址从内存中依次逐条取指令，翻译，并产生相应的取操作数、运算、保存结果、取下条指令等操作的控制信号，以便控制计算机各个部分有机地协调的完成指令规定的操作。



# 1.3.4 控制器

## 7、存储程序控制方式



# 1.3.4 控制器

## 8、控制器的基本任务

从内存储器中按一定顺序取指令

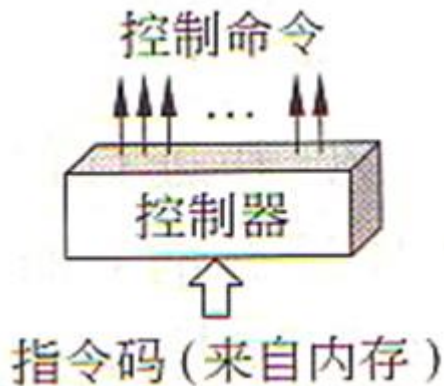
译码（翻译）、产生控制信号

控制取操作数（源操作数、目的操作数）

控制执行（运算）

控制保存结果

形成下条指令地址（顺序、转移）

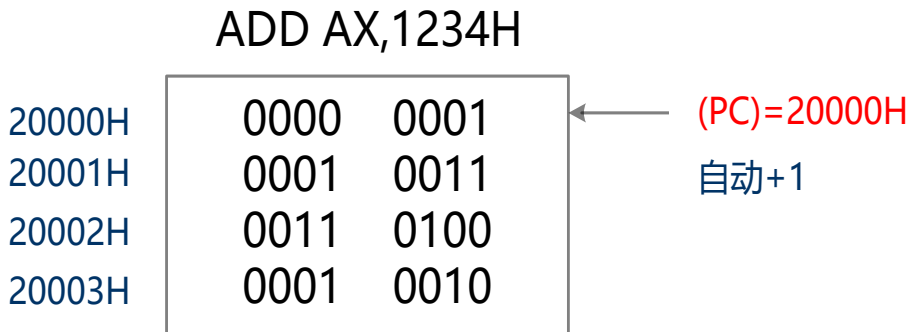


## 1.3.4 控制器

9、程序计数器PC：中央处理器中的专用寄存器，控制器每次取指令，从PC中获取指令的存放地址。

**顺序程序：**PC自动加1功能、控制程序顺序执行；

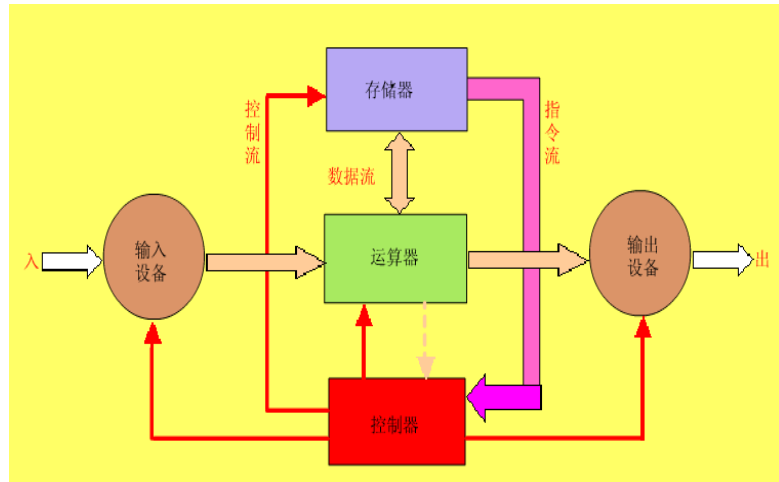
**分支程序（转移、循环、子程序、中断）：**通过地址变换策略，生成新的地址保存到PC。



## 1.3.4 控制器

### 10、取指周期、执行周期

一条指令的执行过程分为取指阶段、执行阶段。相应地，从时间概念上可以划分为取指周期、执行周期。

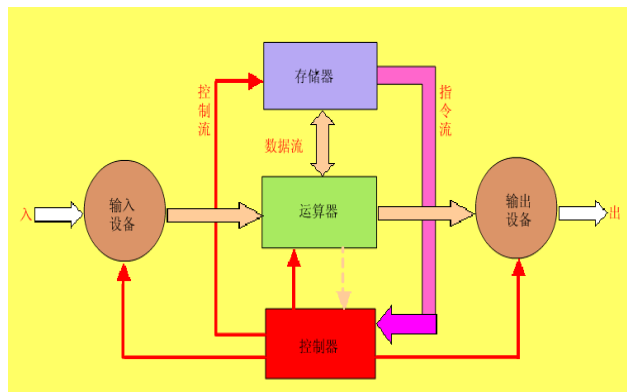


# 1.3.4 控制器

## 10、取指周期、执行周期

**取指阶段：**从存储器中取指令，控制器翻译、产生相应的控制信号。主要是**控制器工作**。

**执行阶段：**取操作数、运算、保存结果。主要是**运算器工作**。

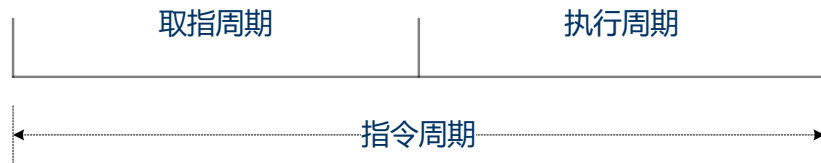


## 1.3.4 控制器



### 10、取指周期、执行周期

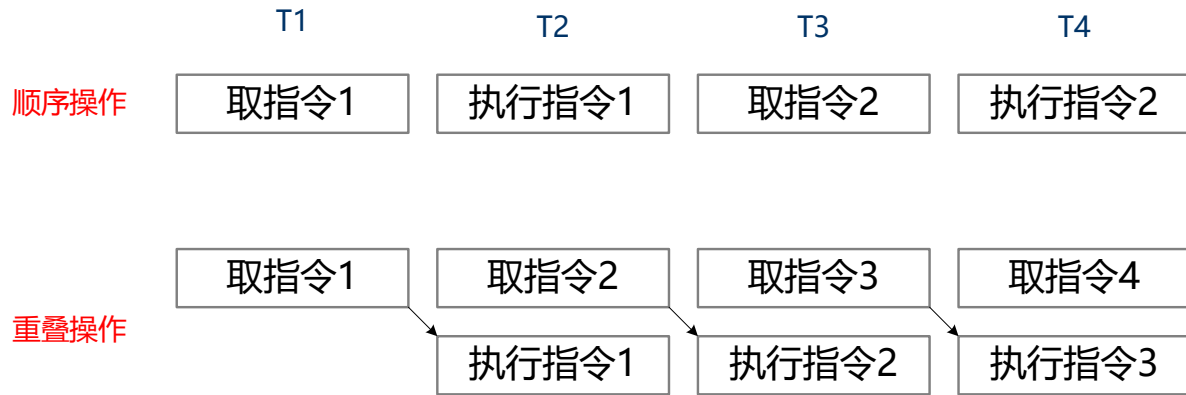
将指令的执行过程按阶段进行划分，其目的是有利于实现指令的重叠操作，从而提高指令的执行速度。也是流水线处理机的基本思想。



# 1.3.4 控制器



## 10、取指周期、执行周期



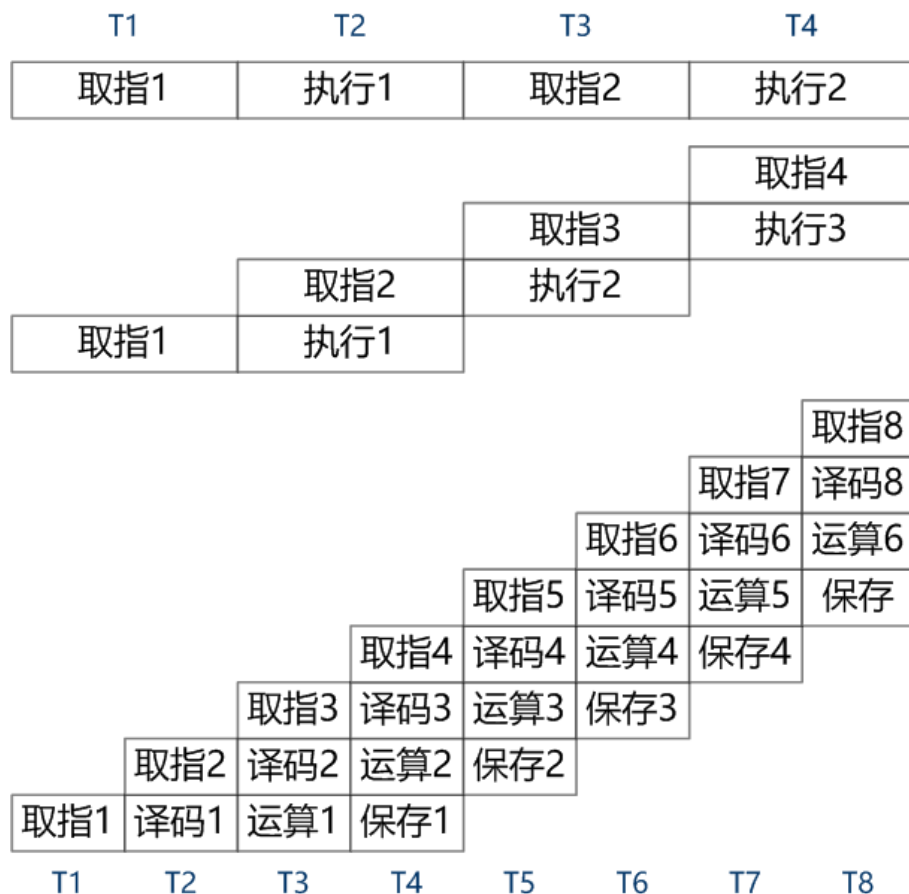
4个单位时间内，顺序操作执行了2条指令，而重叠操作进行到取第4条指令。



## 1.3.4 控制器

### 10、取指周期、执行周期

显然，划分的段数越多，重叠效率越高。



## 1.3.4 控制器



### 11、中央处理器CPU

中央处理器 = 运算器 + 控制器

+ 寄存器组 (如PC、累加器A、寄存器B)

+ 片内高速缓冲存储器cache

+ ..... (逐步扩充)



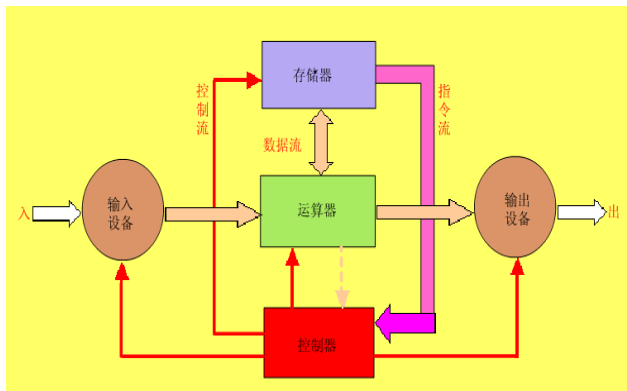
# 1.3.4 控制器

## 12、指令字、数据字

计算机中存储、传输、处理的信息包括程序（指令）和数据，相应地有指令字和数据字两个概念。

**指令字**一定交由控制器翻译、处理；

**数据字**一定交由运算器加工、运算。



## 1.3.4 控制器



### 12、指令字、数据字

**bit流：**在连续的指令读取、执行过程中，就好象bit在排着队列行进一样。网络上也有bit流的概念。

**指令流：**在取指周期中从内存中读出的信息流称为指令流，它通过总线、CPU内部数据通路流向控制器。

**数据流：**在执行周期中从内存中读出的信息流称为数据流，它通过总线、CPU内部数据通路流向运算器。



## 1.3.5 适配器与输入输出设备



### 1、理想的I/O设备

理想的输入设备：会看、会听。

理想的输出设备：会写、会讲。

### 2、通用的I/O设备

输入设备：键盘、鼠标、数字扫描仪OCR等。

输出设备：打印机、显示器、绘图仪等。

既是输入、又是输出设备：主要是指软盘、硬盘、光盘、磁带、阵列等外存。



## 1.3.5 适配器与输入输出设备



### 3、适配器 (I/O接口部件)

**作用：**由于主机与I/O设备之间的速度差异，适配器主要在两者之间起数据缓冲、地址识别、信号转换等，确保输入输出设备按照计算机所要求的数据形式和格式发送或接收信息。

**通用适配器：**显卡、多功能卡、磁盘通道卡、网卡、MODEM、触摸屏控制卡等。



## 1.3.6 组织结构与总线



### 1、计算机硬件组成要素

运算器、控制器、内存储器、输入设备、输出设备、适配器。

### 2、组织结构

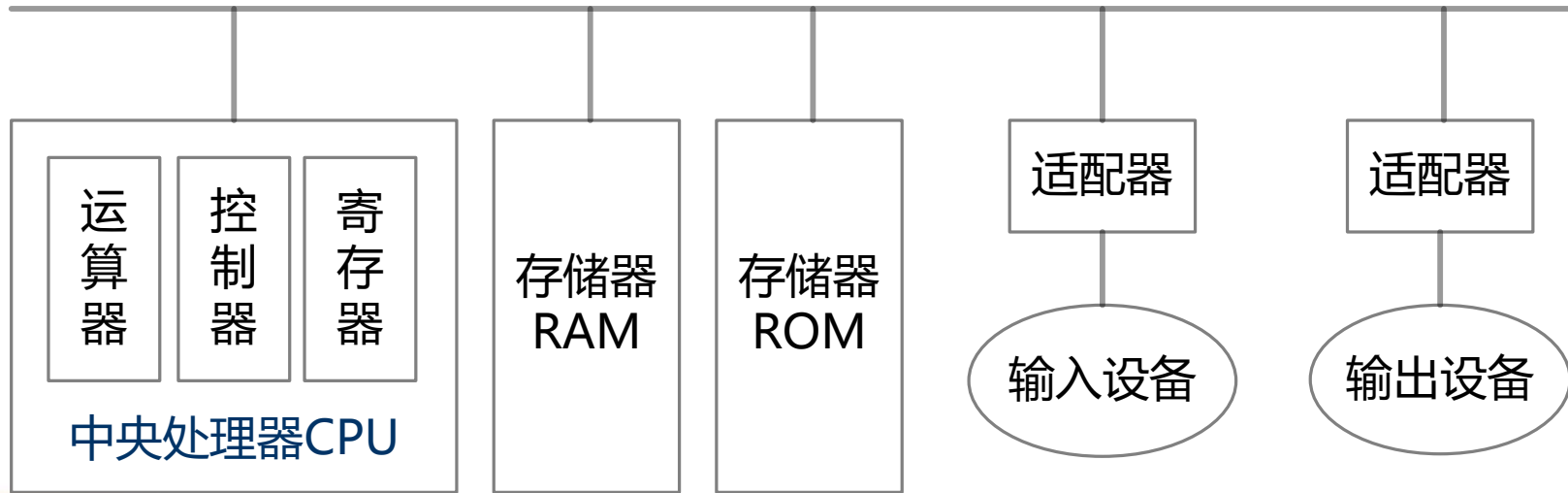
各设备（部件）之间相互有机联系的连接结构，为设备（部件）间信息交换提供通路和控制机制。

典型的组织结构是**总线结构**。

## 1.3.6 组织结构与总线

### 3、总线

总线是一组能为多个部件分时共享的信息传送通路，用来连接多个部件并为之提供信息传输交换服务。

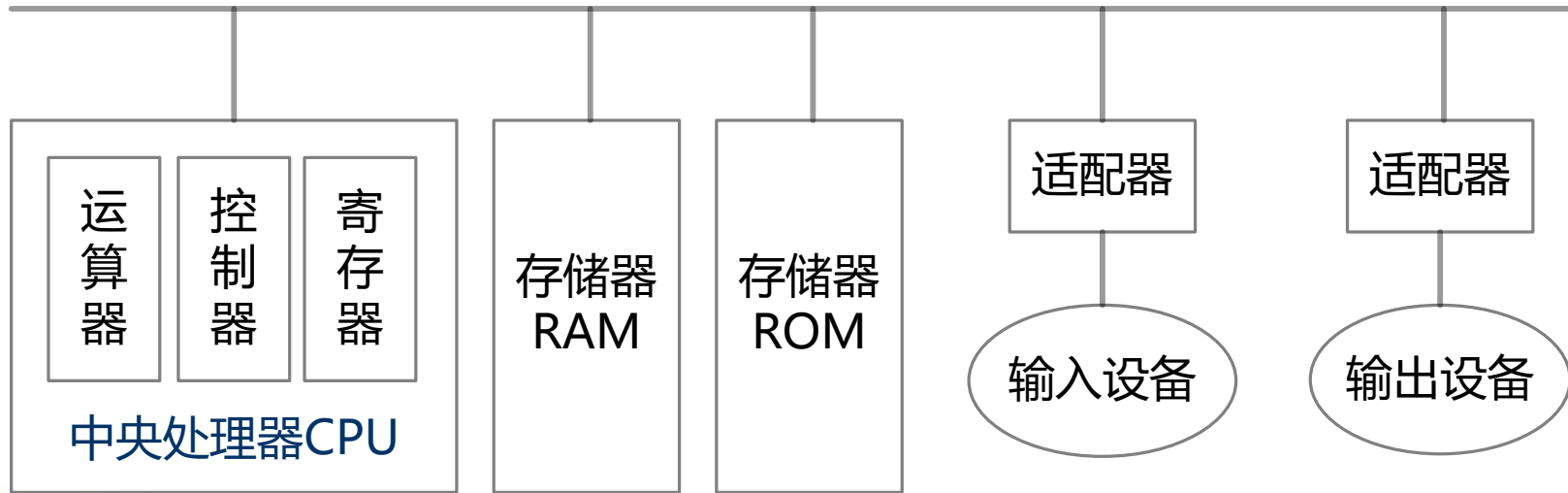




## 1.3.6 组织结构与总线

### 3、总线

**定义：**总线是一组能为多个部件分时共享的信息传送通路，用来连接多个部件并为之提供信息传输交换服务。



## 1.3.6 组织结构与总线



### 3、总线

**共享性：**总线所连接的所有部件都可以通过它传递信息。

**分时性：**在某一个时刻或者时间范围内总线只允许一个部件发送信息到总线上。显然，共享是分时实现的。

**总线协议：**总线不仅是一组传输线路，同时连接到总线上的所有部件都必须共同遵守一组规则和约定，称为总线协议 (Protocol) 。它一般包括信号线定义、数据格式、时序关系、信号电平、控制逻辑等。



# 1.4 计算机的软件



 软件的组成与分类

 编程语言的发展

 应用软件体系结构



# 1.4.1 软件的组成与分类



## 1、软件定义

程序是用程序设计语言描述的、适合于计算机处理的、为完成特定任务的语句/指令的有序序列（集合）。

程序 = 算法 + 数据结构（数据的数据）

软件 = 程序（代码） + 数据 + 文档



## 1.4.1 软件的组成与分类



### 2、软件分类

一般划分：系统软件、应用软件

业界划分：系统软件、支撑软件、应用软件



# 1.4.1 软件的组成与分类



## 2、软件分类

应用 软件	信息管理系统MIS
支撑 软件	语言处理程序Java/JSP、应用服务器WebLogic、 数据库管理系统DBMS、地理信息系统GIS、 工作流workflow、门户系统Portal、消息服务MQ
系统 软件	操作系统、磁盘阵列管理软件、双机软件HA、 集群软件cluster、备份软件veritas
基础 设施	主机系统、网络系统、存储系统、安全系统、备份系统



## 1.4.2 编程语言的发展

### 1、 机器语言

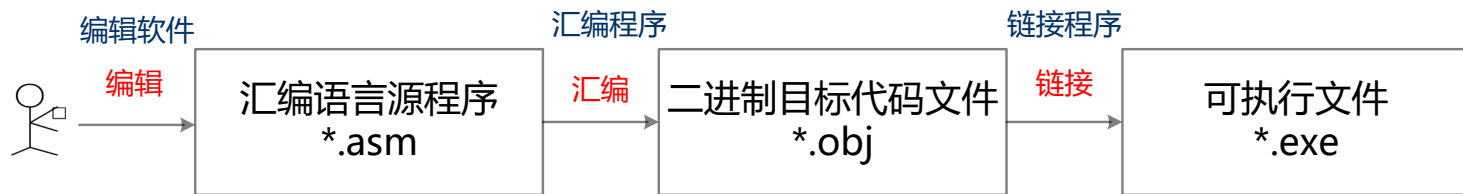
直接用机器指令编写程序，计算机可以直接执行。

面向具体机器，编程是非常麻烦的。

### 2、 汇编语言

采用汇编指令（指令助记符）编写程序。

汇编指令基本上与机器指令一一对应，也是面向具体机器。



## 1.4.2 编程语言的发展



### 3、高级语言

接近于数学（自然）语言的算法语言，使不熟悉具体计算机的人也能方便地使用计算机、编写程序。高级语言源程序必须翻译为目的程序，通常有编译、解释两种方式。

**编译：**首先利用编译系统，把源程序翻译成目的程序，然后执行目的程序。

**解释：**逐条翻译、执行源程序中的所有语句，即执行时逐条翻译。





## 1.4.2 编程语言的发展



### 4、第四代语言4GL

可视化工具PB、VB、DELPHI、VC。

**程序半自动生成：**界面展示部分的程序自动生成，而事件处理程序、业务处理逻辑需要人工编写。

面向对象程序设计（对象、类、继承、消息通信）。

编译：可执行程序、动态连接库DLL。



## 1.4.2 编程语言的发展



### 5、 JAVA/JSP

Java是一种简单、跨平台、面向对象、分布式的、多线程、动态的编程语言。

HTML、JSP：页面构造；

JAVA：业务处理逻辑；

JAVAscript：动态功能。



## 1.4.2 编程语言的发展



### 6、Python

Python是一种解释性、交互式、面向对象、跨平台的编程语言。

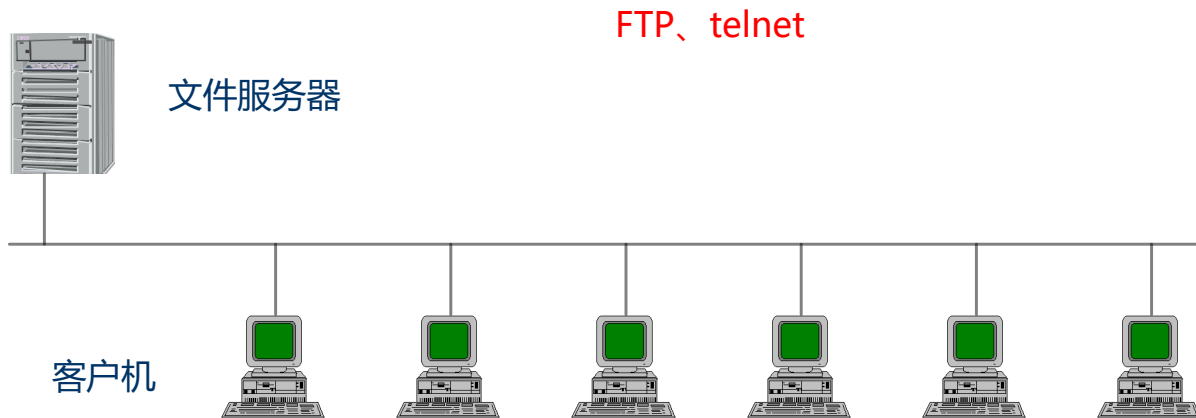
Python是人工智能、数据分析等领域的核心编程语言。



## 1.4.3 应用软件体系结构



### 1、文件服务结构



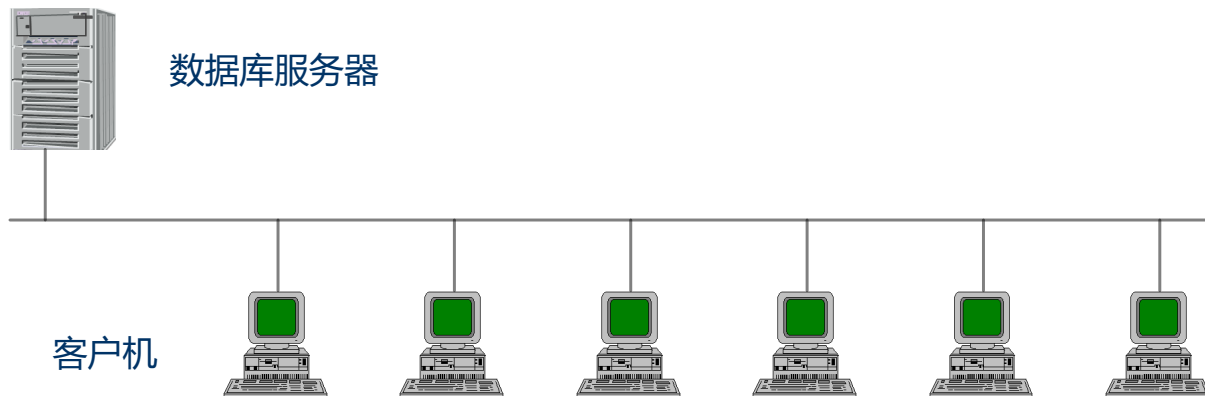
put/mput: 从客户机上传文件到服务器  
get/mget: 从服务器下载文件到客户机



## 1.4.3 应用软件体系结构



### 2、客户/服务器结构C/S

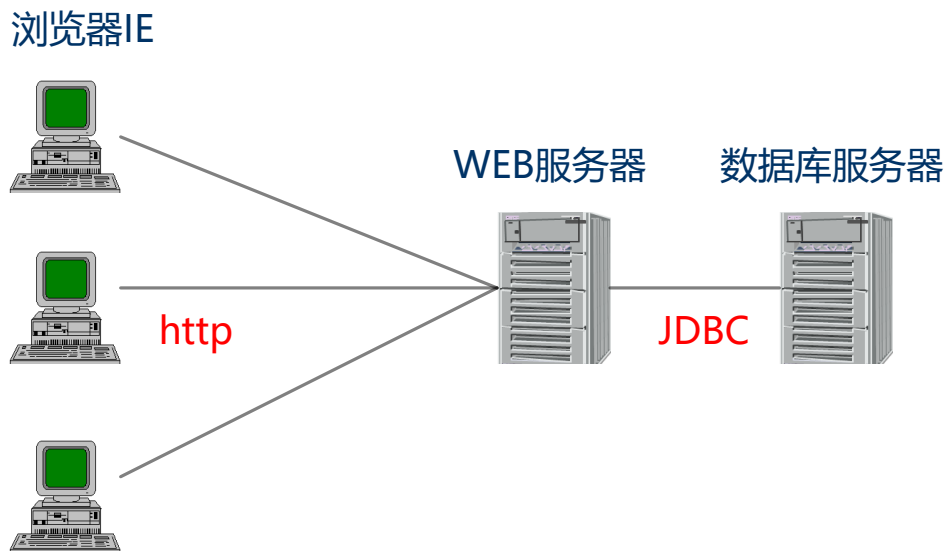


- 1、执行客户端应用程序时，其中SQL通过网络提交给服务器；
- 2、服务器执行SQL，对数据库进行操纵；
- 3、执行结果返回客户端。

## 1.4.3 应用软件体系结构



### 3、 浏览器/服务器结构B/S



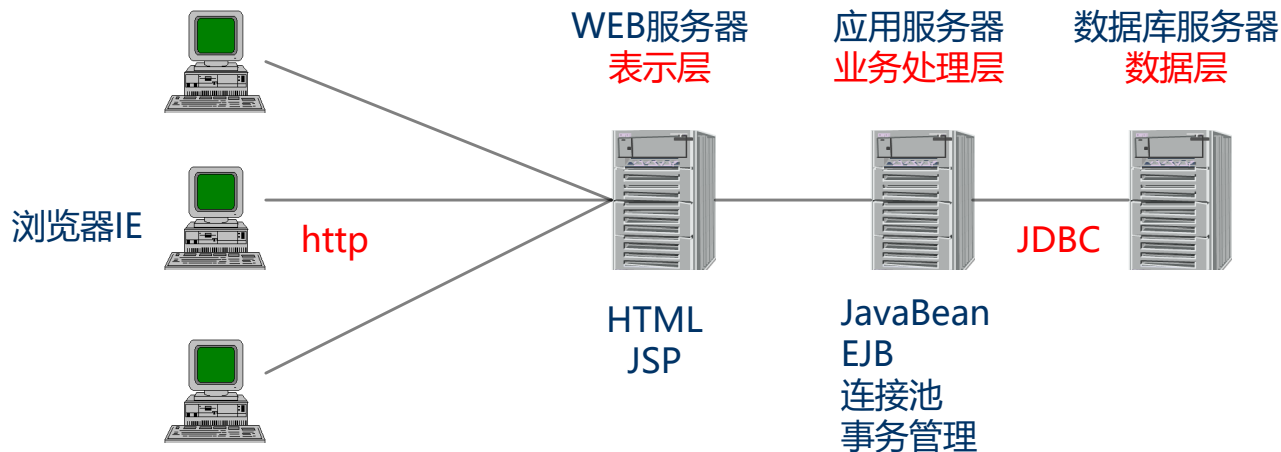
WEB/JSP: HTML、JSP、JAVA、JAVAscript



## 1.4.3 应用软件体系结构

### 4、 三层/多层结构

表示层、业务逻辑层（理论上可设置多个）、数据层，典型结构有SUN J2EE、MS .NET。



## 1.4.3 应用软件体系结构



5、 云计算与大数据

6、 边缘计算



图 2-1. 云计算的架构



## 1.4.3 应用软件体系结构



关于软件开发能力学习：

- 1、关系数据库基本原理、SQL、一种数据库的安装配置。
- 2、一个4GL语言开发工具PB、VB、DELPHI、VC，C/S结构的应用程序开发技术。
- 3、三层/多层结构的应用程序开发技术，编程语言主要包括html、JSP、java、javascript，可以从开发一个动态网站入手。
- 4、软件工程、软件项目管理、软件过程改进、技术文档。



# 1.5 计算机系统的层次结构



- ❏ 多级组成的计算机系统
- ❏ 软件与硬件的逻辑等价性
- ❏ 冯.诺依曼计算机



## 1.5.1 多级组成的计算机系统



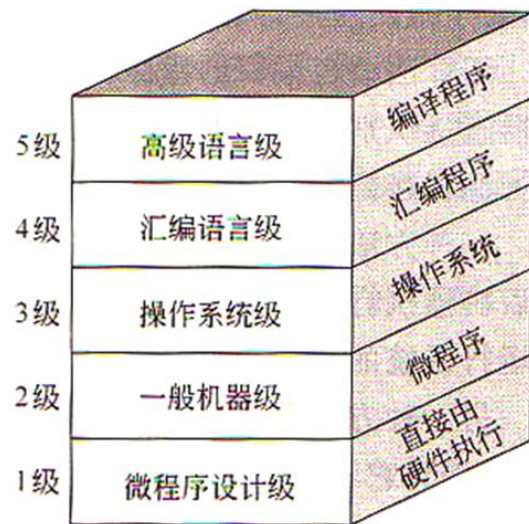
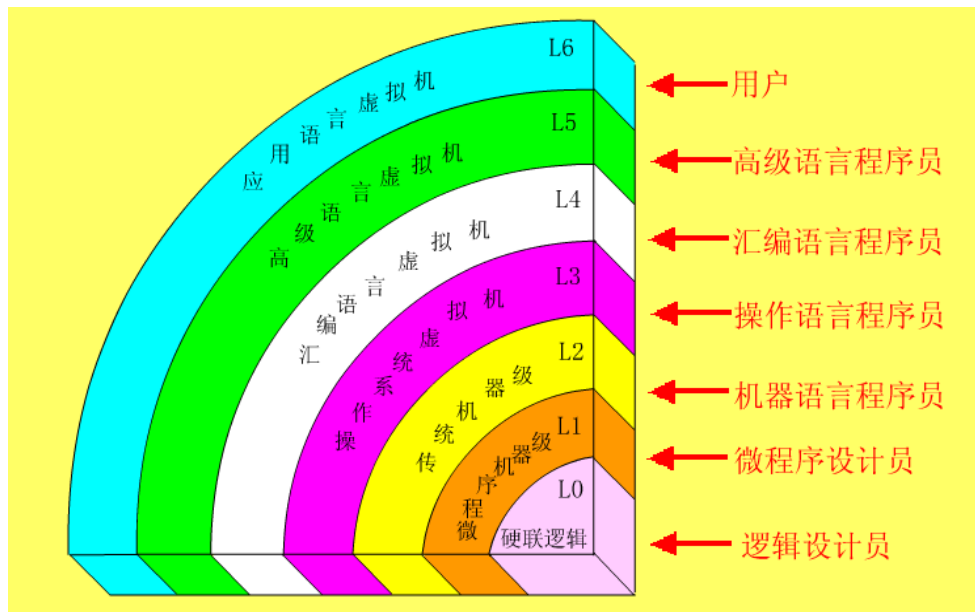
### 1、计算机系统层次结构

现代计算机系统是由硬件、软件有机结合的十分复杂的整体。

在了解、分析、设计计算机系统时，人们往往采用**分层（分级）的方法**，即将一个复杂的系统划分为若干个层次，即计算机系统的层次结构。最常见的是从计算机编程语言的角度划分的计算机系统层次结构。

# 1.5.1 多级组成的计算机系统

## 2、 计算机编程语言角度划分的计算机系统层次结构



不同层次上的用户，所看到的计算机的“功能”是不同的。

## 1.5.1 多级组成的计算机系统



### 3、虚拟机

虚拟机、物理机。

是指通过**配置软件**扩充物理机（硬件/固件实现）功能以后所形成的一台计算机，而物理机并不具备这种功能。

虚拟机概念是计算机分析设计中的一个重要策略，它将提供给用户的功能抽象出来，使用户摆脱具体物理机细节的束缚。

## 1.5.2 软件与硬件的逻辑等价性



### 1、 软件与硬件的逻辑等价性

任何操作功能，既可以由软件来实现，也可以由硬件/固件来实现；任何指令的执行可以由硬件/固件完成，也可以由软件来完成。

对于某一机器操作功能采用硬件/固件实现、软件实现方案，取决于器件价格、速度、可靠性、存储容量、变更周期等因素。

## 1.5.2 软件与硬件的逻辑等价性



### 2、软件硬化

一些计算机的特点是，把原来明显地在一般机器级通过编制程序实现的操作，如整数乘法指令、浮点运算指令、字符串操作指令等等，改为直接由硬件/固件完成。

随着大规模集成电路技术和计算机体系结构的发展，实体硬件机的功能范围不断地扩大。



## 1.5.2 软件与硬件的逻辑等价性



### 3、 固化、固件

将传统的部分软件写入只读存储器ROM中，称为**固化**。

只读存储器及其写入的软件称为**固件**，固件是介于硬件和软件之间的一种形态，从物理形态上看是硬件，从运行机制上看是软件。

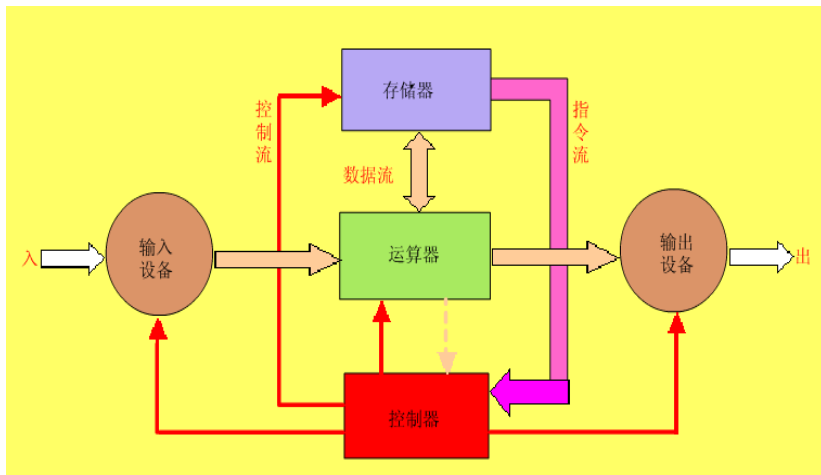




# 1.5.3 冯·诺依曼计算机

## 1、概念结构

冯·诺依曼等人于1946年提出了计算机的雏形，简称为冯·诺依曼计算机，也称为冯·诺依曼体制。



## 1.5.3 冯·诺依曼计算机



### 2、技术特点

- 【1】 由运算器、控制器、存储器、输入设备、输出设备五大部分构成计算机硬件系统概念结构；
- 【2】 采用二进制代码表示数据和指令；
- 【3】 采用存储程序控制方式（指令驱动）。

