



计算机硬件系统

北京大学计算机学院

董豪

<https://zsdonghao.github.io>

hao.dong@pku.edu.cn



- 基础原理
- 硬件系统 ◀
- 软件基础
- 信息表示
- 网络与存储
- 程序设计语言

从计算机外观出发（由外而内）看热闹

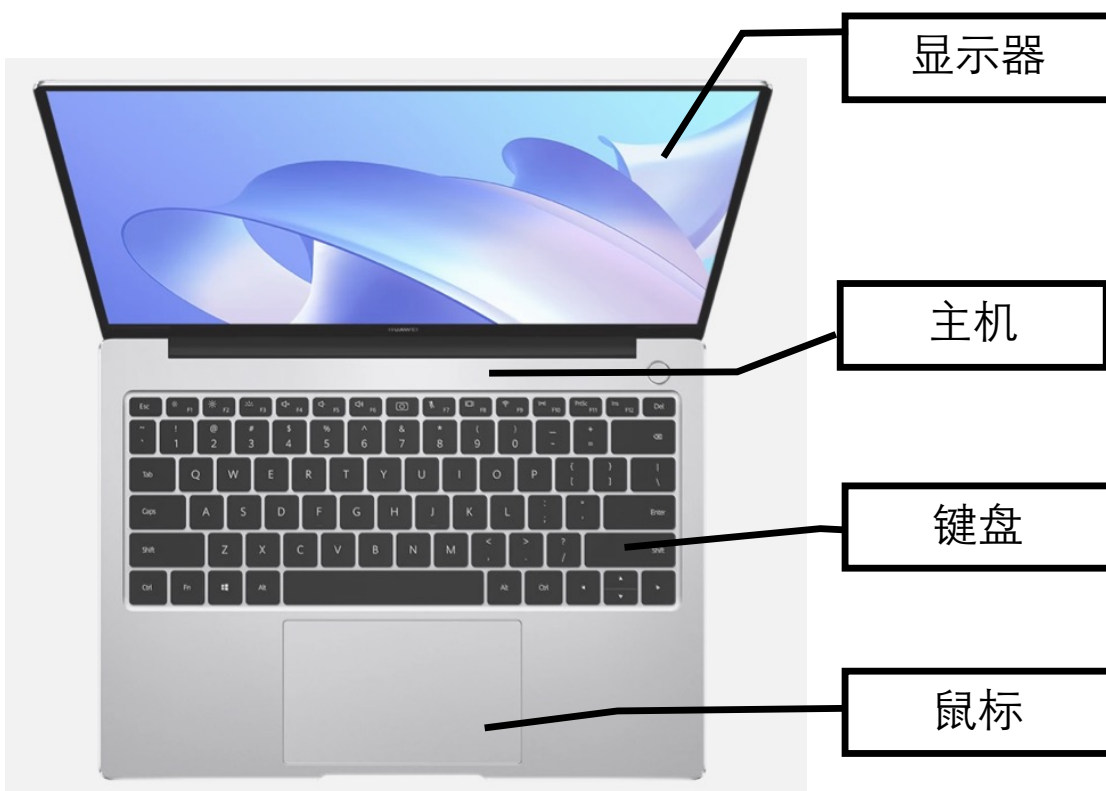
从“冯·诺依曼”结构出发（由内而外）看门道

从计算机外观出发（由外而内）



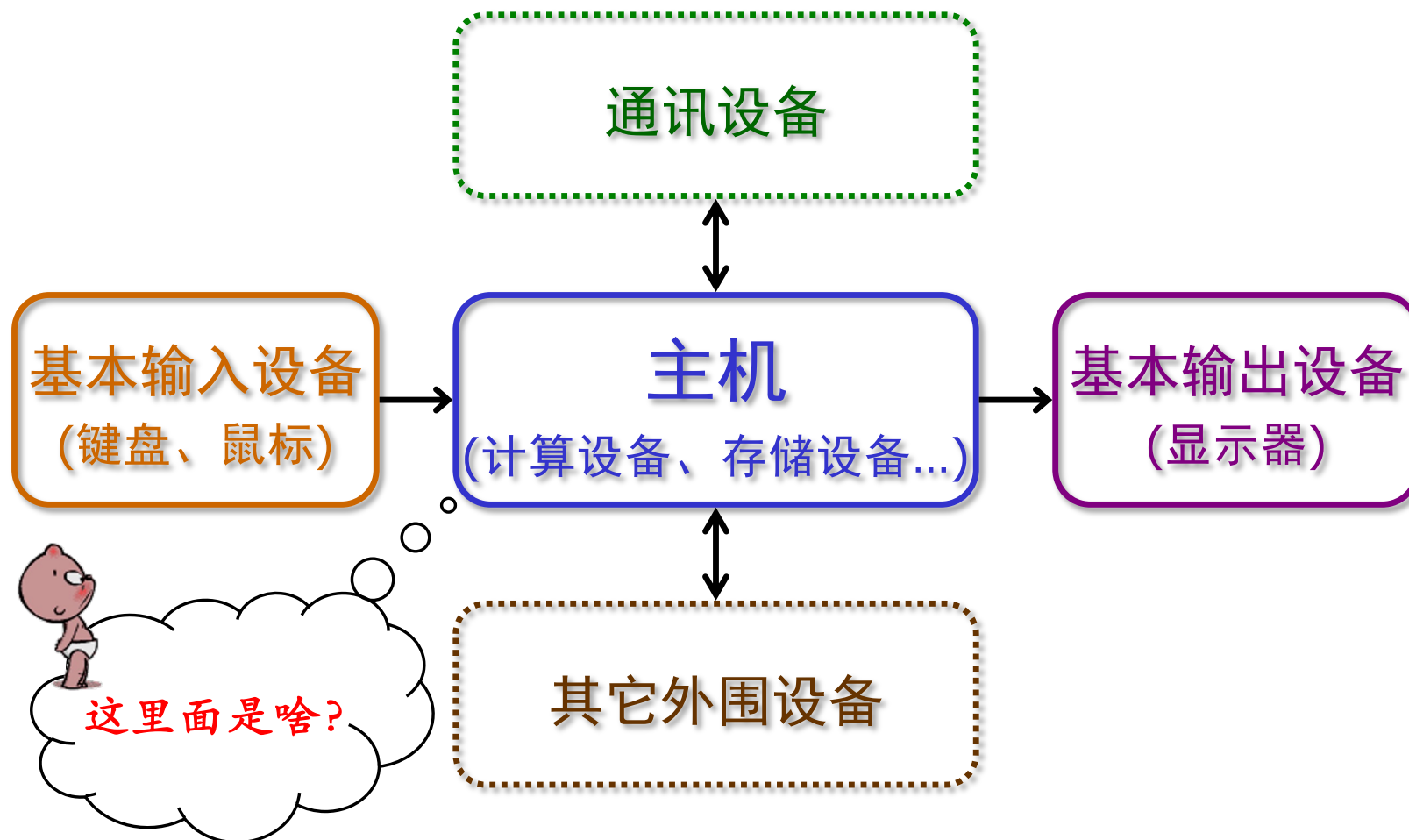
从计算机外观出发（由外而内）

HUAWEI MateBook 14 2021



CPU : Intel®酷睿™ i5-1135G7
显卡 : NVIDIA® GeForce® MX450
内存 : 16 GB DDR4
硬盘 : 512 GB NVMe PCIe SSD

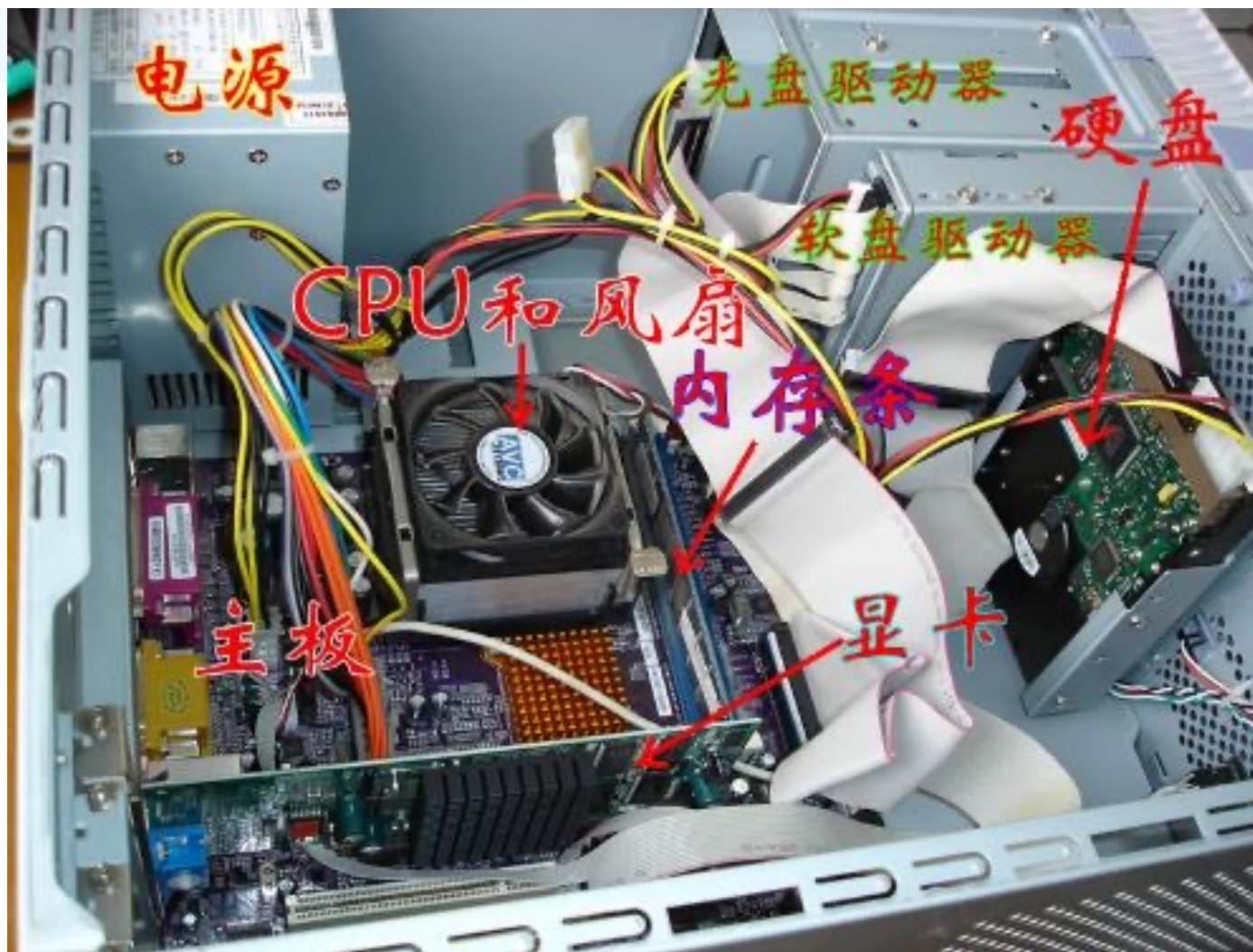
从计算机外观出发（由外而内）



从计算机外观出发（由外而内）



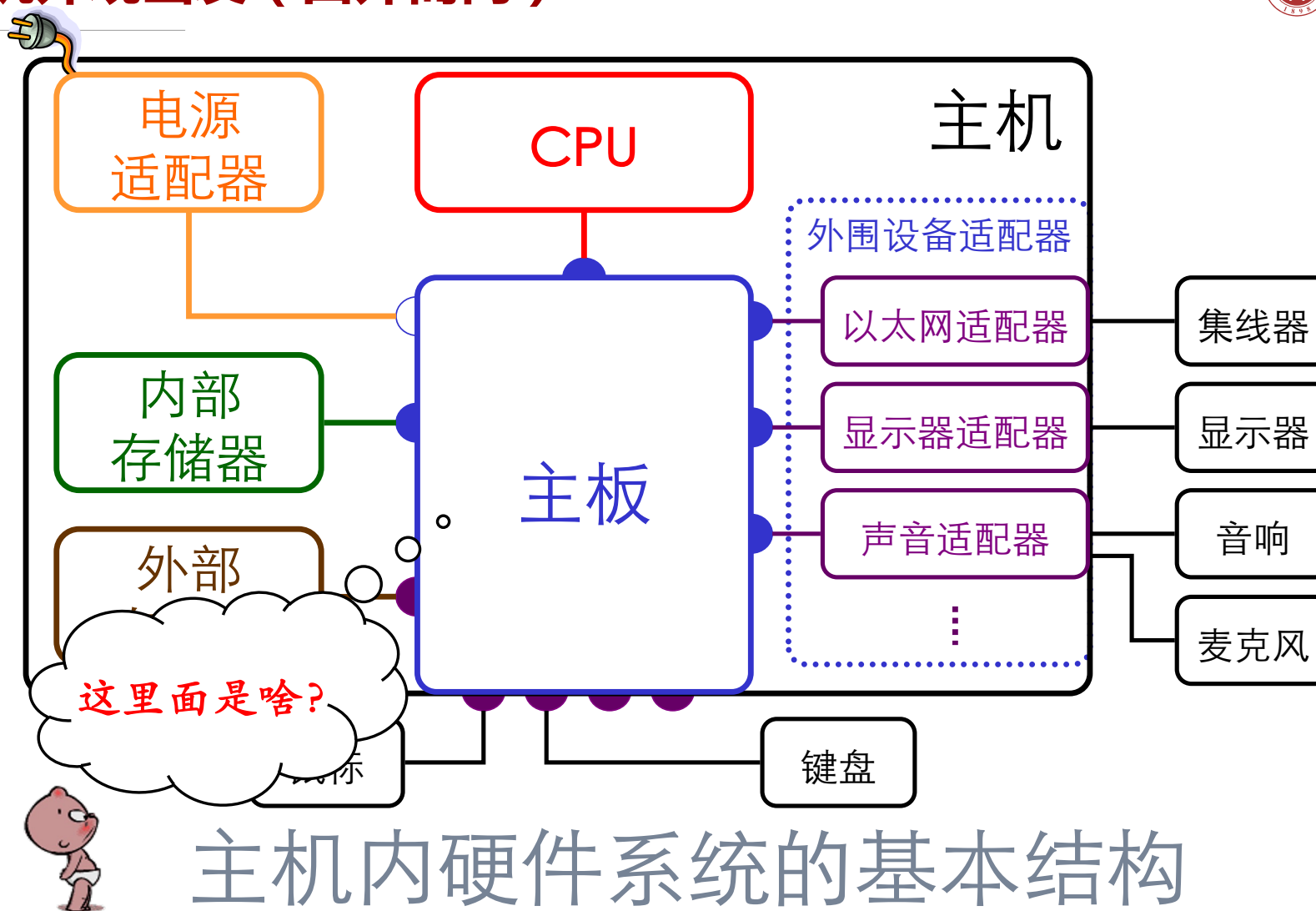
从计算机外观出发（由外而内）



从计算机外观出发（由外而内）



从计算机外观出发（由外而内）



主机内硬件系统的基本结构

从计算机外观出发（由外而内）

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来
- 芯片组（南桥、北桥）控制链接
- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

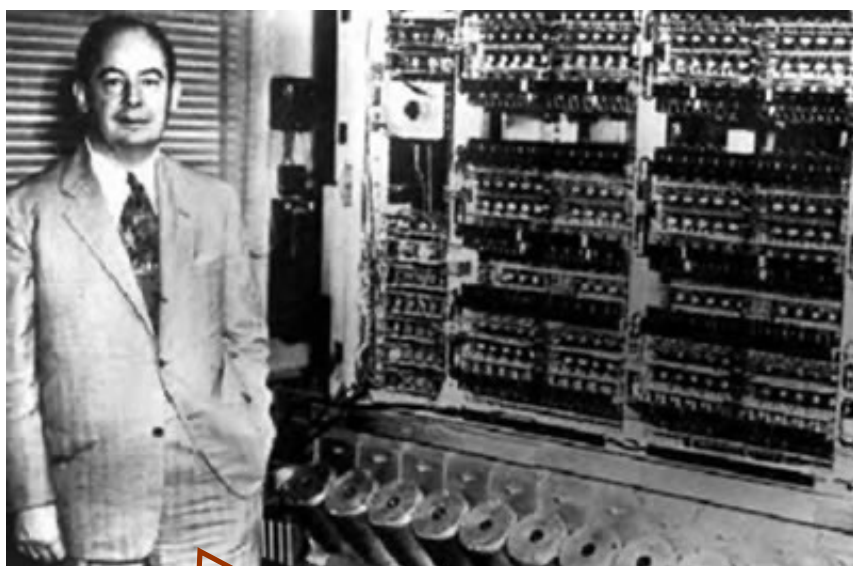


从计算机外观出发（由外而内）看热闹

从“冯·诺依曼”结构出发（由内而外）看门道

- 运算器
- 控制器
- 存储器
- 输入设备
- 输出设备

第一台 现代意义 上的计算机：EDVAC



冯·诺依曼（主要设计者）

■ EDVAC (Electronic Discrete Variable Automatic Computer)

- 1945年提出设计方案，1951年完成
- 占地面积和消耗电力只有ENIAC的1/3
- 每秒 2000 次乘法运算，
或 10000 次加法运算
- 特点：
 - 采用二进制表示数据和程序
 - 使用“存储程序”的方式对计算机进行控制

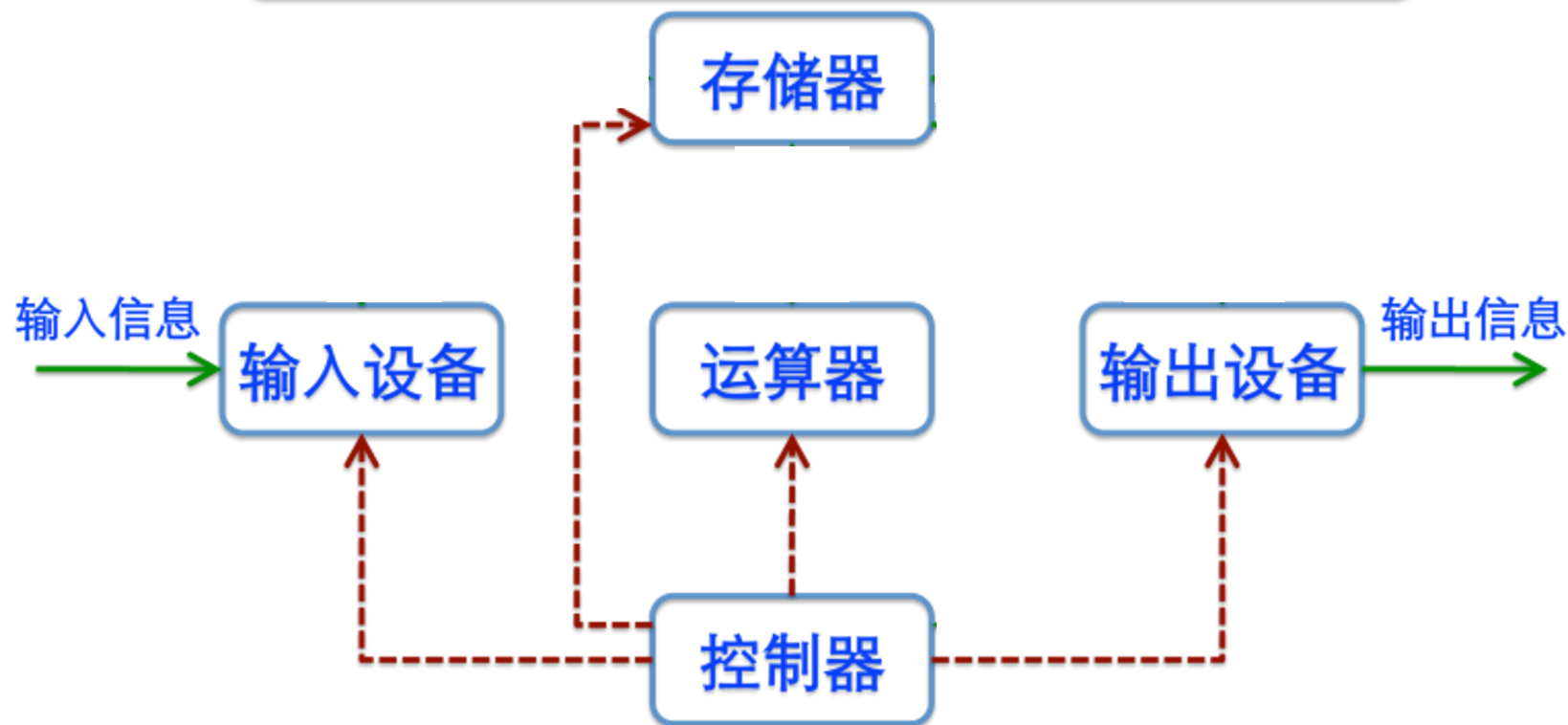
冯·诺依曼结构的基本思想

- “冯·诺依曼”结构计算机体系结构的核心思想是：**存储程序**
- 可以概况为：
 - 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
 - 采用**二进制**来表示指令和数据
 - **自动高效**地从存储器中取出指令加以执行

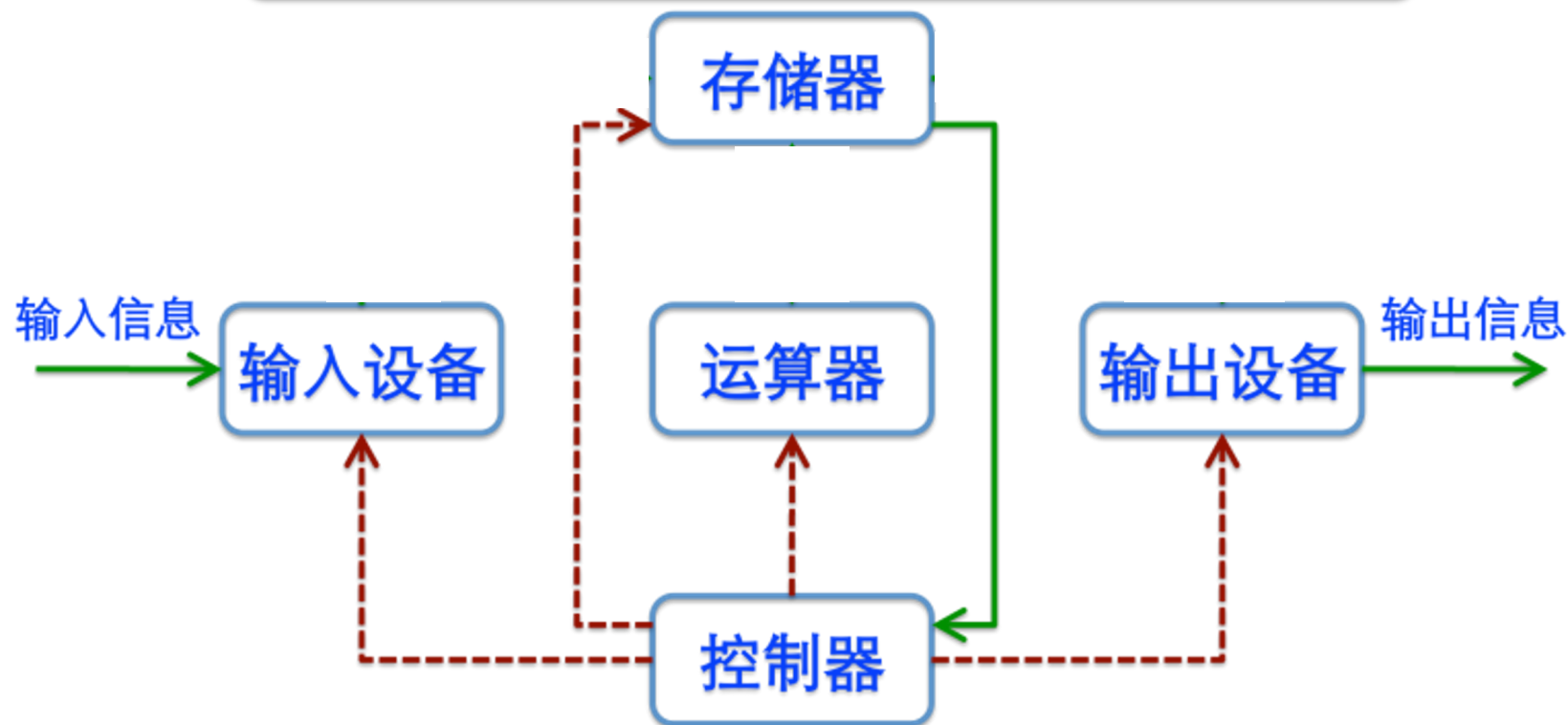
冯·诺依曼结构的五大部件



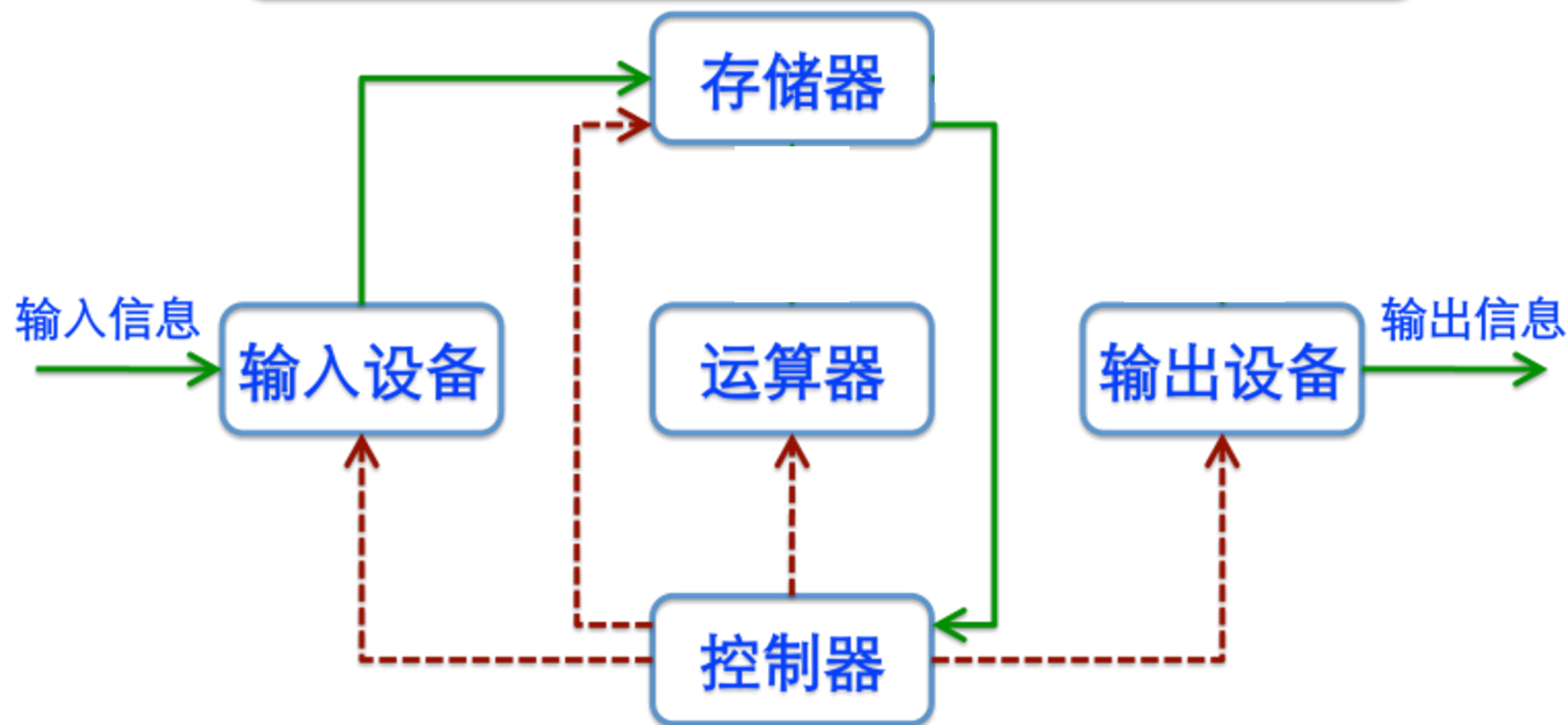
冯·诺依曼结构的五大部件



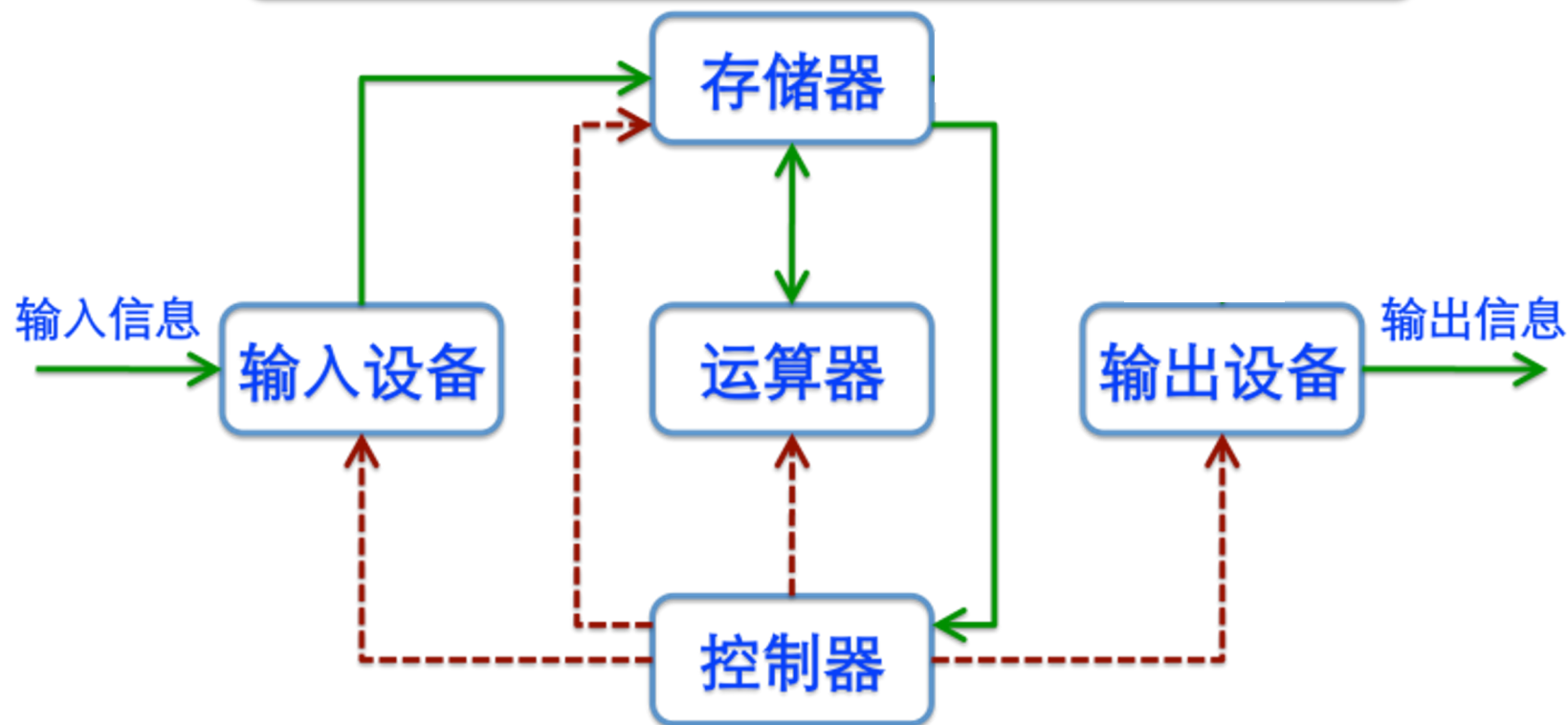
冯·诺依曼结构的五大部件



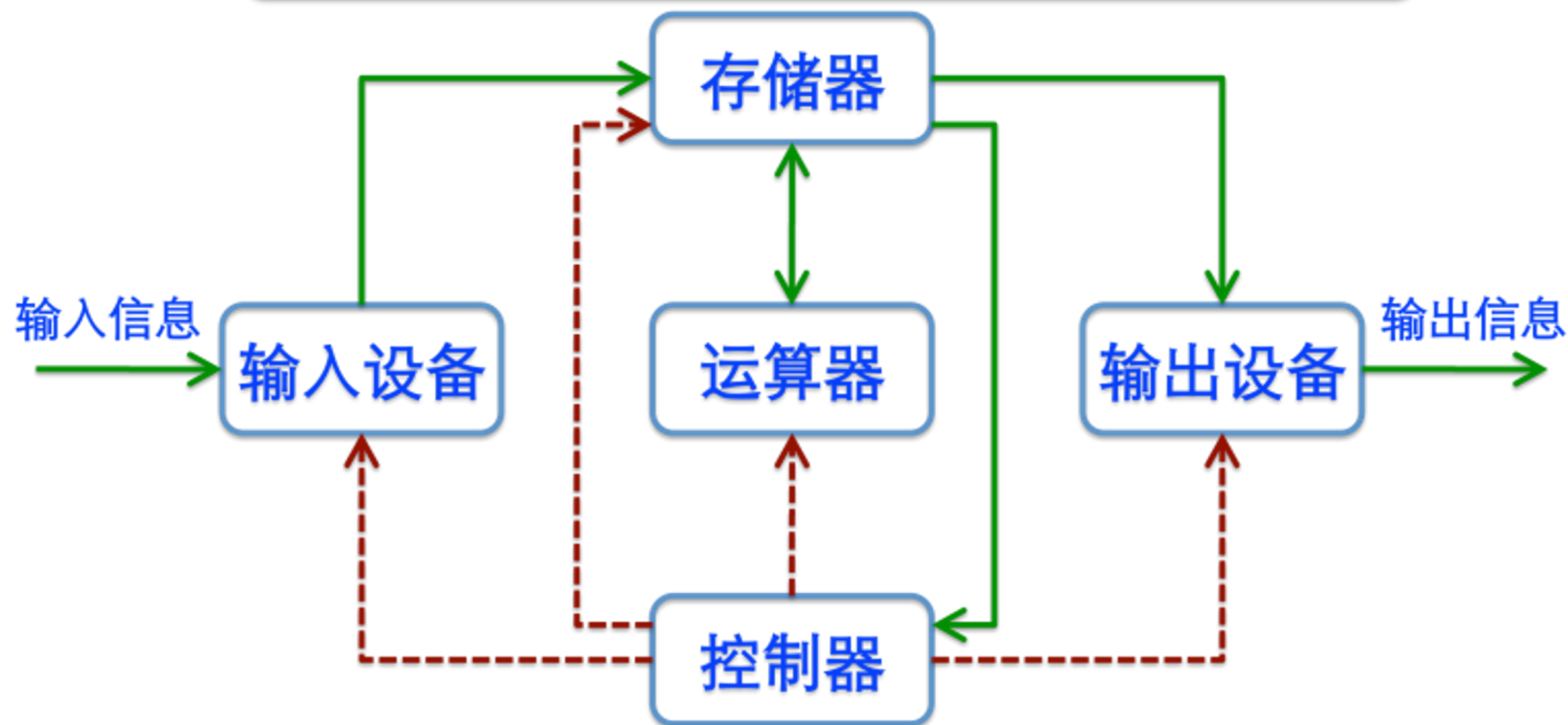
冯·诺依曼结构的五大部件



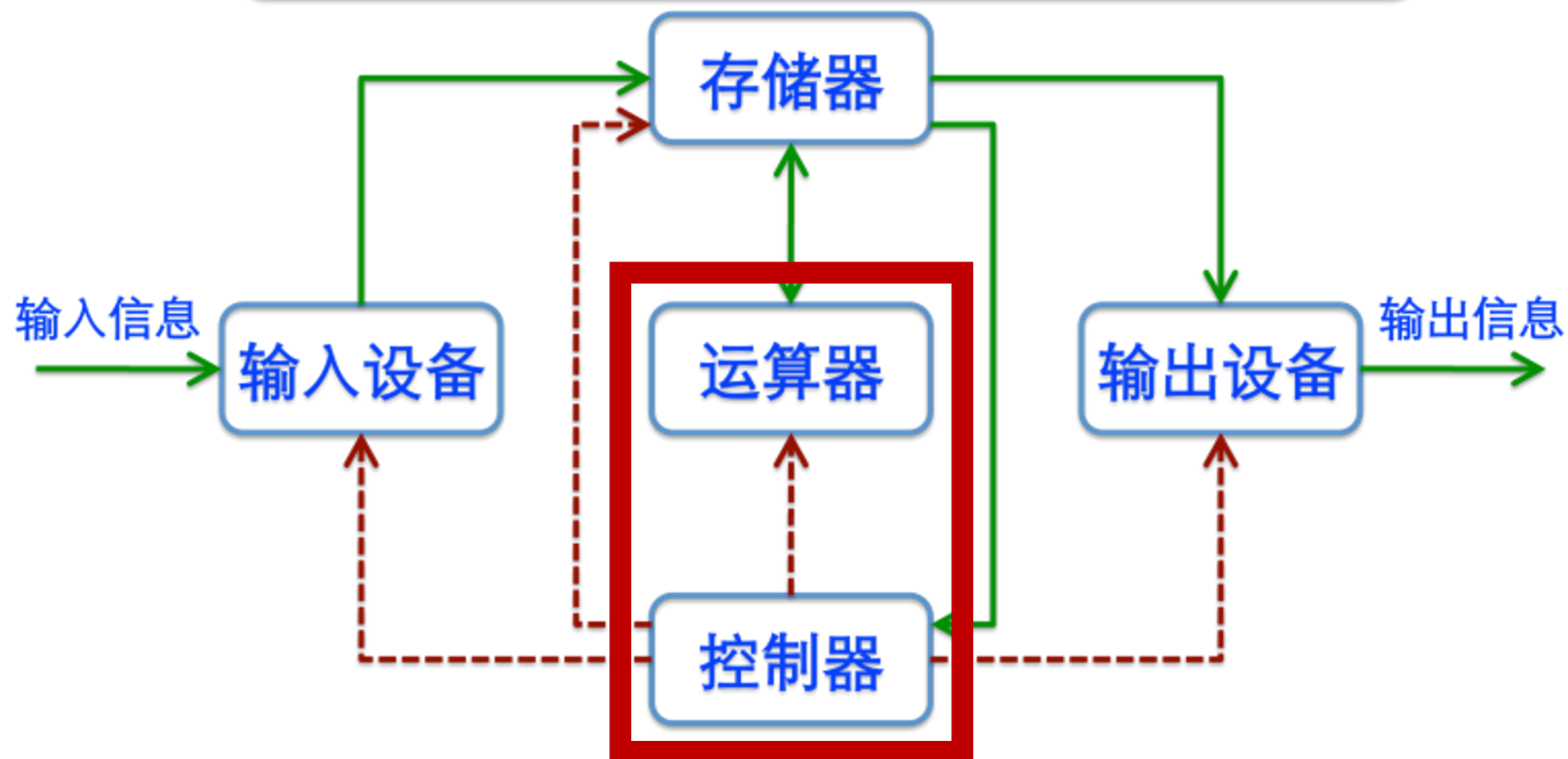
冯·诺依曼结构的五大部件



冯·诺依曼结构的五大部件



冯·诺依曼结构的五大部件



中央处理器 CPU

- 中央处理器（微处理器） Central Processing Unit（CPU）

- CPU由**运算器**和**控制器**组成



- **控制器**：协调指挥计算机各个部件自动、连续地执行命令

- **运算器**（Arithmetic Logical Unit, **ALU**）：

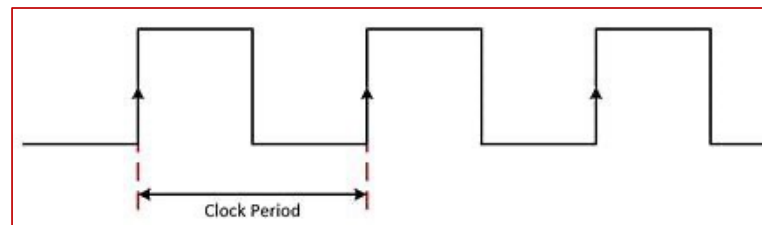
- 负责对数据进行加工运算，功能：**算术运算**和**逻辑运算**

- 包含：**寄存器**（暂时存储等待处理的数据）

- **累加器**（暂时存储计算结果）

■ 中央处理器（微处理器） Central Processing Unit（CPU）

- **字长：** CPU一次处理的二进制位数
 - 越长则处理速度越快、精度越高
 - 如32位，64位
- **主频率：** 时钟频率（工作频率）单位是赫兹（Hz）
 - 决定计算机的运行速度，代表每秒运行的“周期数”
 - 如1.6GHz，3.4GHz
 - 越高则越快



■ 低级语言 之 机器语言

00000001000000001000 数据装入寄存器0

00000001000100001010 数据装入寄存器1

00000101000000000001 寄存器0与1的数据乘

00000001000100001100 数据装入寄存器1

00000100000000000001 寄存器0与1的数据加

00000010000000001110 保存寄存器0里的数据

计算机能执行哪些指令是由硬件决定，并且决定了指令的集合。

比如：00000001是数据装入命令，它是给硬件部件加电信号的一种命令

无论多么复杂的程序，到了硬件一层就是输入驱动硬件逻辑电路动作

■ 低级语言 之 汇编语言

load 0 a	数据a装入寄存器0
load 1 b	数据b装入寄存器1
mult 0 1	寄存器0与1的数据乘 (结果存在寄存器0中)
load 1 c	数据c装入寄存器1
add 0 1	寄存器0与1的数据加 (结果存在寄存器0中)
save 0 d	保存寄存器0里的数据 (放到数据d中)

■ 机器语言 与 汇编语言

00000001000000001000	load 0 a
00000001000100001010	load 1 b
00000101000000000001	mult 0 1
00000001000100001100	load 1 c
00000100000000000001	add 0 1
000000100000000001110	save 0 d

■ C++语言写同样的程序：

d = a * b + c;

中央处理器 CPU

- 指令集合(指令系统):
 - CPU所能够处理的全部指令的集合,是一个CPU的根本属性(计算机硬件能够直接执行的操作命令。)
 - 不同结构的计算机有不同的指令集合
- CISC---复杂指令集计算机
 - Complex Instruction Set Computer
 - 包含大量的指令以及相应的硬件逻辑
- RISC---精简指令集计算机
 - Reduced Instruction Set Computer
 - 只包含最基本的功能, 其它复杂的功能由简单的执行组合而成

CISC:

昂贵, 指令集大
电路复杂, 功耗大

RISC:

便宜, 指令集小
电路简单, 功耗低

中央处理器 CPU

- **X86**是**CISC**指令系统
 - Intel和AMD的产品是X86系统



- **ARM**和**POWER**是**RISC**指令系统
 - ARM公司设计指令系统和处理器ip核，不出产芯片
比如：APPLE M1也是ARM指令系统
 - IBM的POWER，基于开源的RISC-V



中央处理器 CPU

编译程序（翻译）

人理解的高级语言符号化表示

```
int a = 5, b = 4, c;  
c = a + b;  
cout << c << endl;
```

可执行程序机器语言 01代码
10001000000110000100000000
10001000000100000010000000
1010001000111000

ARM系统的机器语言。

AMD系统的机器语言。

Intel系统的机器语言。

苹果系统的机器语言。

可执行程序机器语言01代码，在不同的系统下是不同的

程序是顺序执行的，但很多问题可以并行执行

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

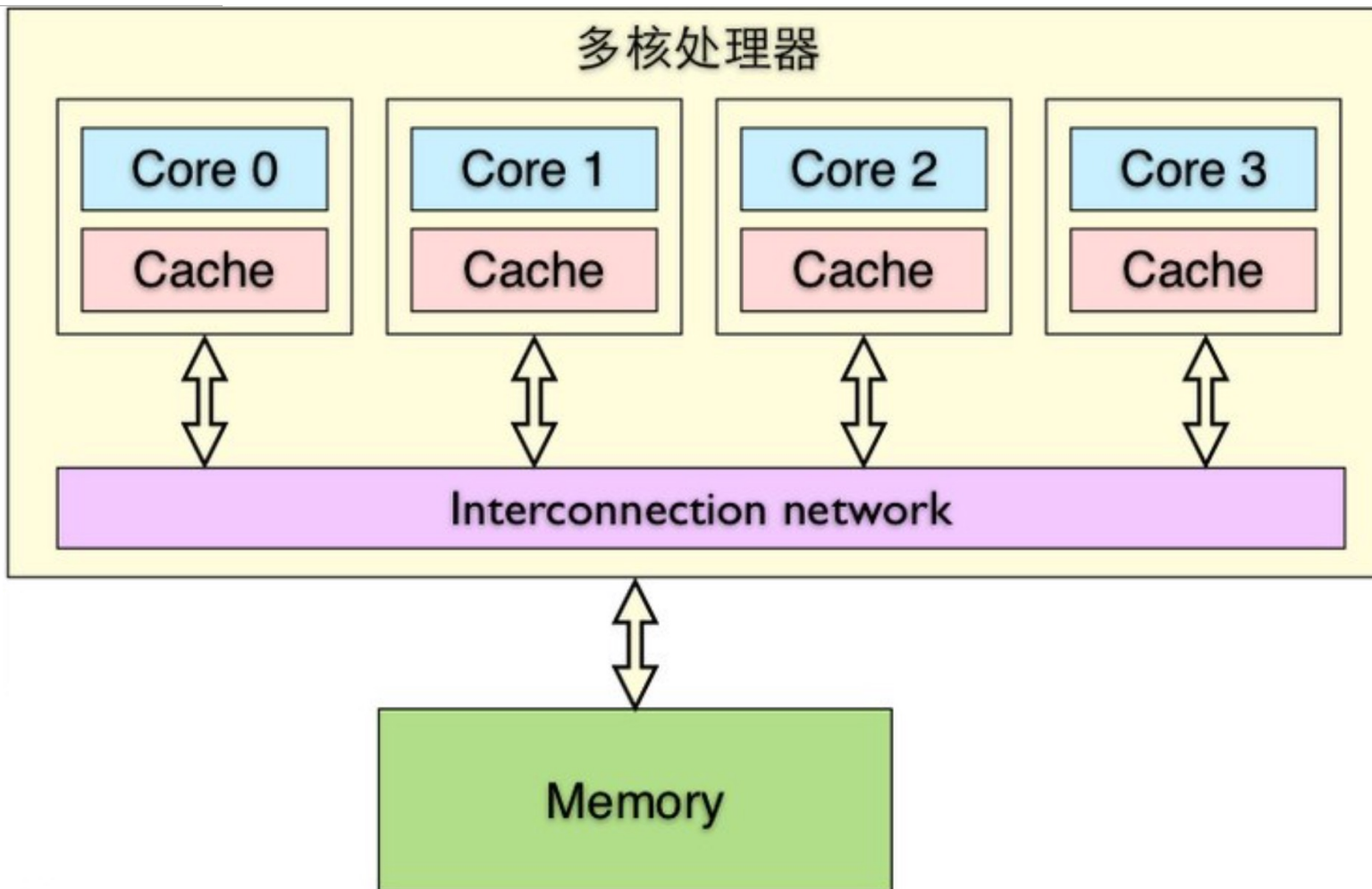
$$1 \times 5 + 2 \times 7 = 19$$

$$1 \times 6 + 2 \times 8 = 22$$

$$3 \times 5 + 4 \times 7 = 43$$

$$3 \times 6 + 4 \times 8 = 50$$

多核处理器



■ 多核处理器带来的好处:

- 横向扩展，通过划分任务，线程应用能够充分利用多个执行内核，并可在主频不变的情况下、特定的时间内执行更多任务。

1. 程序 (Program) : 计算机程序是指以某些程序设计语言编写的指令序列
2. 进程 (Process) : 是一个应用程序在计算机上的一次执行过程
3. 线程 (Thread) : 是进程中的一部分，一个进程往往包含多个线程在运行


```
void one_function()
{
    int array[100];
    for(int i=0; i<100; i++)
        array[i] = i*10;
}
```

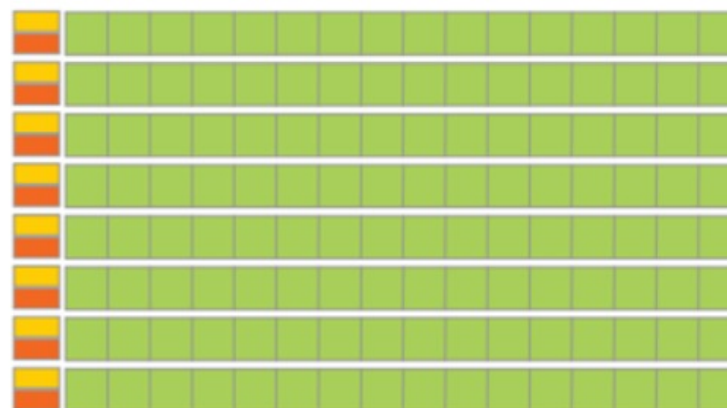
- 多核情况下的执行方式
- 每个处理器核心并行计算，每个核心负责处理一部分数据
- 每个核心都有独立的一级缓存和二级缓存
- 三级缓存被所有核心共享，可用于核心之间的同步，通信

多核处理器 – GPU

- 图形处理器（显卡、graphics processing unit, GPU）



CPU架构示意



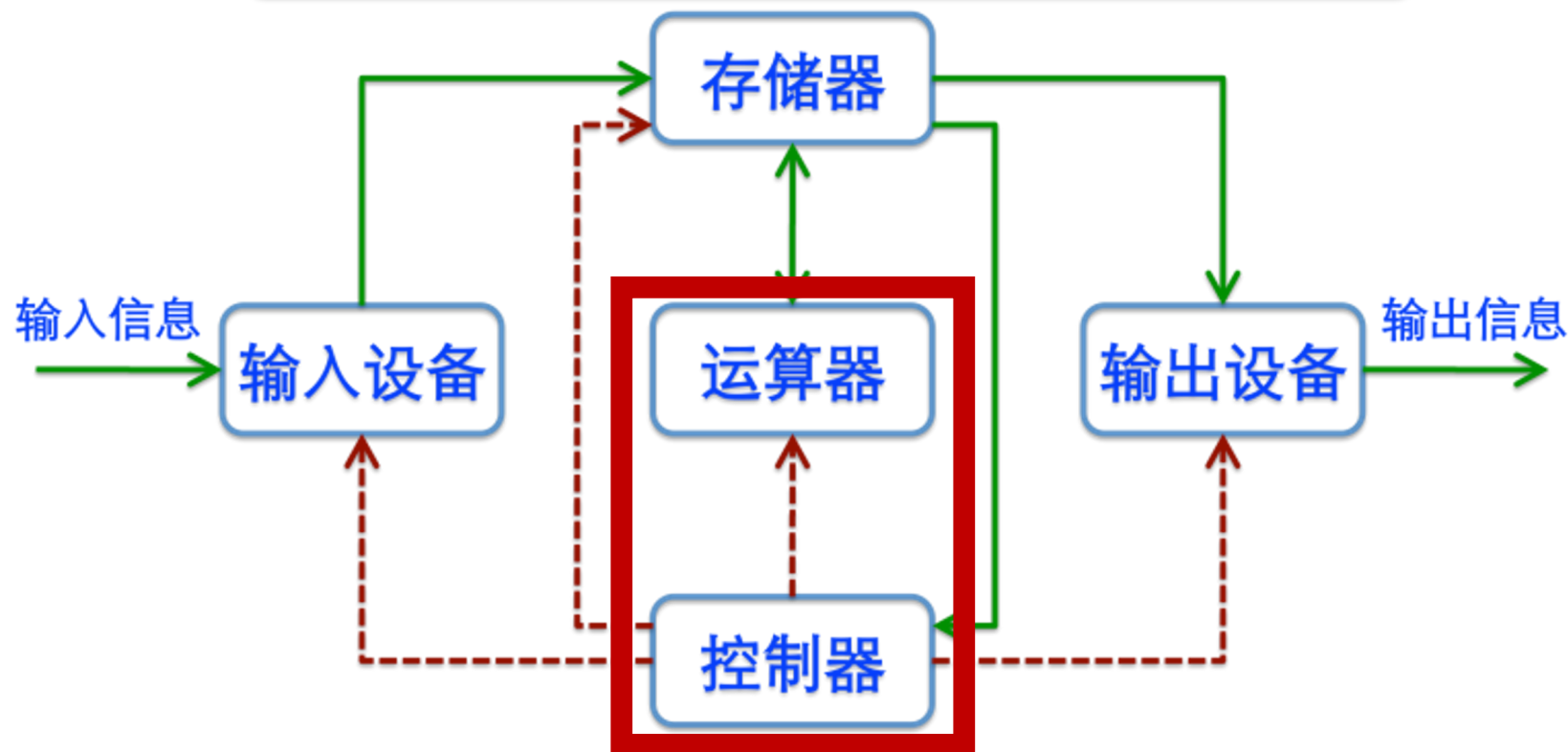
GPU架构示意

- CPU芯片空间的5%是ALU，而GPU空间的40%是ALU

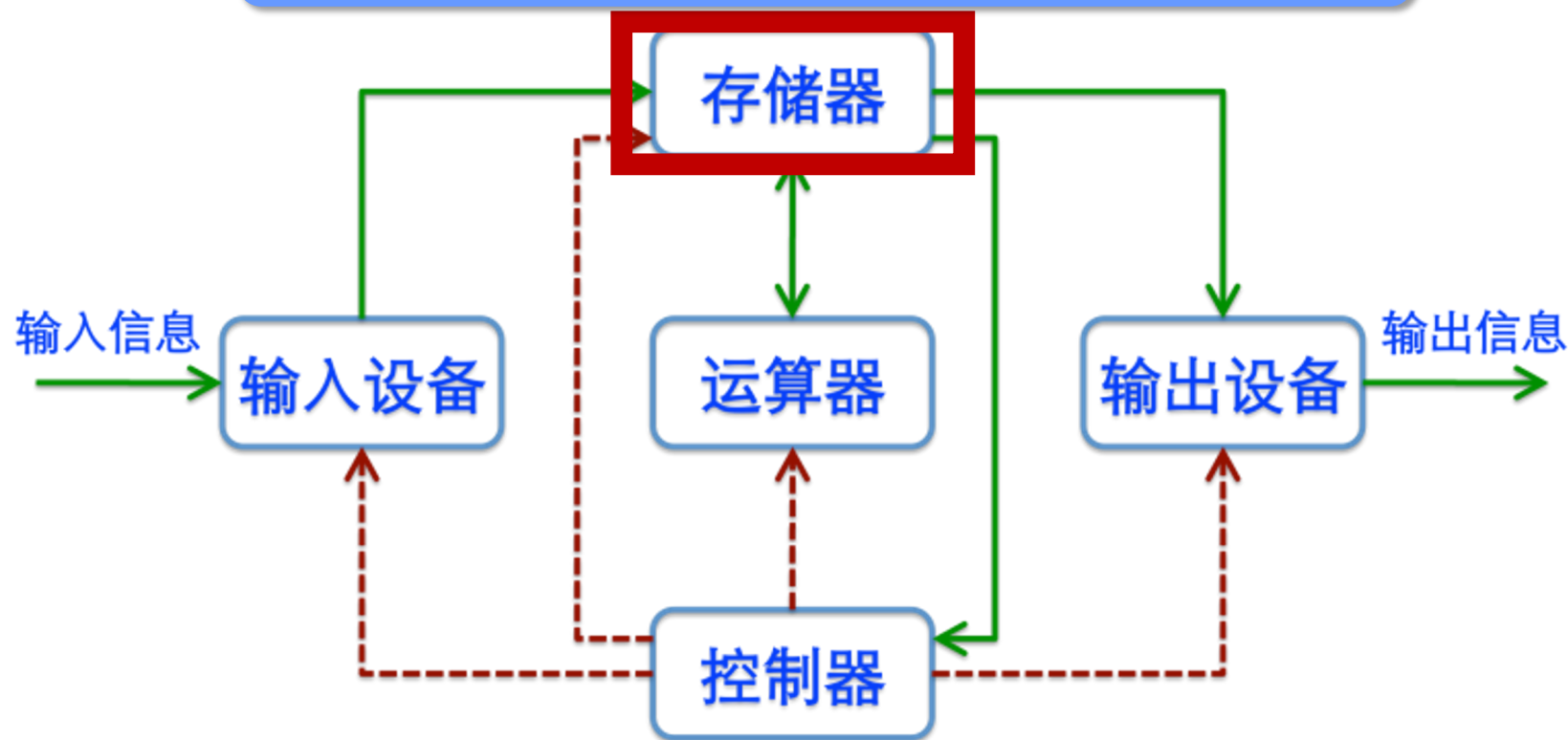
分布式计算



冯·诺依曼结构的五大部件



冯·诺依曼结构的五大部件



存储器

- 计算机的记忆装置，用于存放程序和数据 的器件
- 分为两大类：**内部存储器**和**外部存储器**

存储器

- 计算机的记忆装置，用于存放程序 and 数据的器件
- 分为两大类：**内部存储器**和**外部存储器**
- **存储容量**的单位
 - 1) 位 (bit) 简写b，也称为比特
是计算机**表示数据的最小单位**
 - 2) 字节 (Byte) 简写B，也称为字节
是计算机**存储信息的基本单位**

$$1\text{B} = 8\text{b}$$

小

大

B KB MB GB TB (小到大, $2^{10}=1024$)

存储器

- 计算机的记忆装置，用于存放程序和数据器件
- 分为两大类：**内部存储器**和**外部存储器**
- **存储容量**的单位
 - 1) 位 (bit) 简写b，也称为比特
是计算机**表示数据的最小单位**
 - 2) 字节 (Byte) 简写B，也称为字节
是计算机**存储信息的基本单位**
 - 3) 字
CPU一次存储、加工、传输的数据称为一个字
跟CPU字长有关

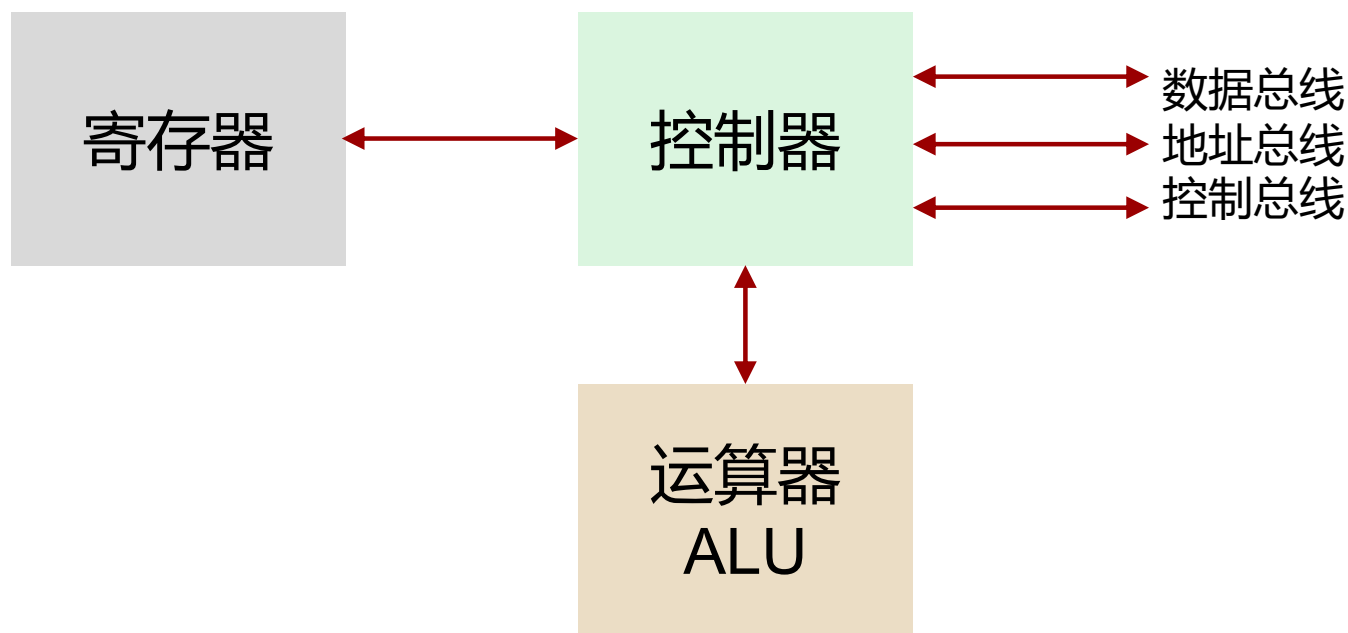
存储器

- CPU寄存器（特例，极小）
- 内部存储器（也称为主存、内存）
 - ROM、RAM、Cache
- 外部存储器（也称为辅存、外存）
 - 软盘、硬盘、U盘、光盘

存储器

■ CPU寄存器

- 在CPU内部的小型存储区域，暂时存放参与运算的数据和运算结果
- 其实就是常用的时序逻辑电路，直接链接到运算电路



内部存储器

- 内部存储器
 - ROM、RAM、Cache
 - CPU可直接访问的存储器（为了速度）

内部存储器

- ROM（只读存储器，Read Only Memory）
 - 其信息只能读出、不能修改
 - 断电后信息不消失，可靠性高
 - 主要存储BIOS（Basic Input Output System基本输入输出系统）
 - ROM在出厂时，就将中断服务程序、启动程序固化在其中，因此也称为固件，其内容是永久保存的、不能写信息

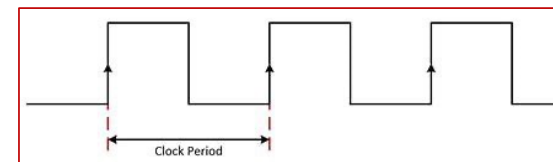
内部存储器

- ROM（只读存储器，Read Only Memory）
 - 为什么需要永久保留不变的指令？
 - 计算机要在操作系统指挥下才能工作，在操作系统进入内存之前谁指挥工作？
 - 在刚刚接通电源的时候，整个系统由BIOS控制
 - 操作系统存储在外存储器中，BIOS引导操作系统加载器到硬盘（等会介绍硬盘）指定的位置来加载操作系统

内部存储器

- RAM（随机存储器，Random Access Memory）
 - 其信息可读可写
 - 断电后信息消失
 - 通常所说的内存一般指的就是RAM
- 存储运行中的程序和变量，程序从硬盘（等下会讲）中读取
- RAM又分为
 - 静态随机存储器（SRAM）和动态随机存储器（DRAM）

- **静态随机存储器 (SRAM) vs 动态随机存储器 (DRAM)**
 - DRAM中的信息隔段时间会进行刷新，而SRAM不刷新
 - DRAM: 主要器件是电容，周期性刷新充电，**否则数据会消失**
 - SRAM: 不刷新电路即能保护数据，具有**较高的性能**
 - SRAM功耗比DRAM低，价格比DRAM高，集成度低
- **DDR SDRAM (Double Data Rate 双倍速率同步动态)**
 - SDRAM在一个时钟周期内只传输一次数据，它是在时钟的上升期进行数据传输；而DDR SDRAM内存则是一个时钟周期内传输两次数据，它能够在时钟的上升期和下降期各传输一次数据



- Cache（高速缓冲存储器）
 - 高速：速度很快
 - 缓冲：缓和谁和谁之间的冲突？
- 内存的存取速度比CPU慢，制约了计算机整体的速度，为了协调两者之间的速度差，在它们之间引入了一个与CPU速度接近、高速的、容量相对较小的缓冲存储器Cache




- Cache（高速缓冲存储器）
 - 为什么有用？
 - 程序的局部性原理
 - 程序在一定的时间段、通常只会访问一个局部内存空间
 - 时间局部性：如果一个内存地址正在被使用，那么在近期他很可能还会再次被访问
 - 空间局部性：在最近的将来可能用到的信息可能是与当前使用的信息相邻的


Cache (高速缓冲存储器)

```
for(j = 0; j < 100; j = j++)  
  for(i = 0; i < 5000; i = i++)  
    x[i][j] = 2 * x[i][j]
```

```
for(i = 0; i < 5000; i++)  
  for(j = 0; j < 100; j++)  
    x[i][j] = 2 * x[i][j]
```

假设取数据要10个周期，数据在Cache中，则3个周期能够完成乘法


$$5000 \times 100 \times (10+3)$$
$$= 6,500,000 \text{ 个周期}$$


$$5000 \times 10 + 5000 \times 100 \times 3$$
$$= 1,500,000 \text{ 个周期}$$

每次读入一个横着的一维数组

- Cache（高速缓冲存储器）
 - 缓存命中率
 - CPU向Cache读数据时，如果所需要的数据刚好在Cache中，则称为命中
 - 如果没有命中，则需要从内存中读取新的数据，替换掉Cache中的一部分内容
 - 替换算法
 - 读取新的数据，需要一定的策略，替换掉老的数据

- Cache（高速缓冲存储器）
 - 替换算法
 - FIFO（First Input First Output）先进先出
 - LFU（Least Frequently Used）最不经常使用
看一定时间段内被使用的频率
 - LRU（Least Recently Used）最近最少使用算法
看最后一次被使用到发生调度的时间长短

内部存储器

- 存储器的两个指标：容量和访问速度
 - 一般个人电脑是 8~32G，服务器通常更大
 - 访问速度在1800MHz、2500MHz不等
 - 问：容量会影响处理速度吗？
 - 答：会
 - 软件越来越大，对内存要求越来越高

■ 小结

- 速度关系 : 寄存器 > Cache > 内存 > 外存
- ROM : 只读不写、断电不丢、存BIOS
- RAM : 可读可写、断电丢失、通常所说的微机内存
- Cache : 可读可写、断电丢失

外部存储器

- 相对于内部存储器，**外部存储器**的容量要大得多，但存取速度却低很多。
- 外部存储器的主要种类包括：
 - **硬盘**、光盘/光盘驱动器、软盘/驱动器、磁带/磁带机等
- 具有“**非易失性**”：掉电后，其内部存储的信息仍然会长时间或永久保留
- 从物理原理上讲，它们是利用物质的光、电、磁等物理特性，来表示0/1数据



外部存储器

- 机械硬盘（HDD）使用磁碟进行数据存储
 - 速度慢、体积大；成本低
- 固态硬盘（SSD）使用闪存颗粒进行数据存储
 - 速度快、体积小；成本高



机械硬盘



固态硬盘

内部存储器 vs 外部存储器

■ 内部存储器

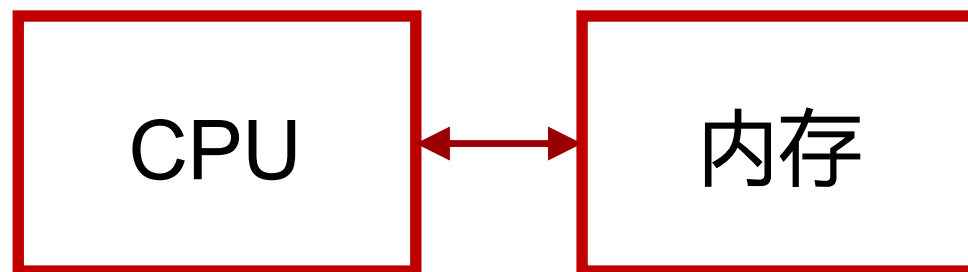
- 断电丢失 (ROM除外)
- 存储运行时的程序和变量
- 速度快
- 容量小

■ 外部存储器

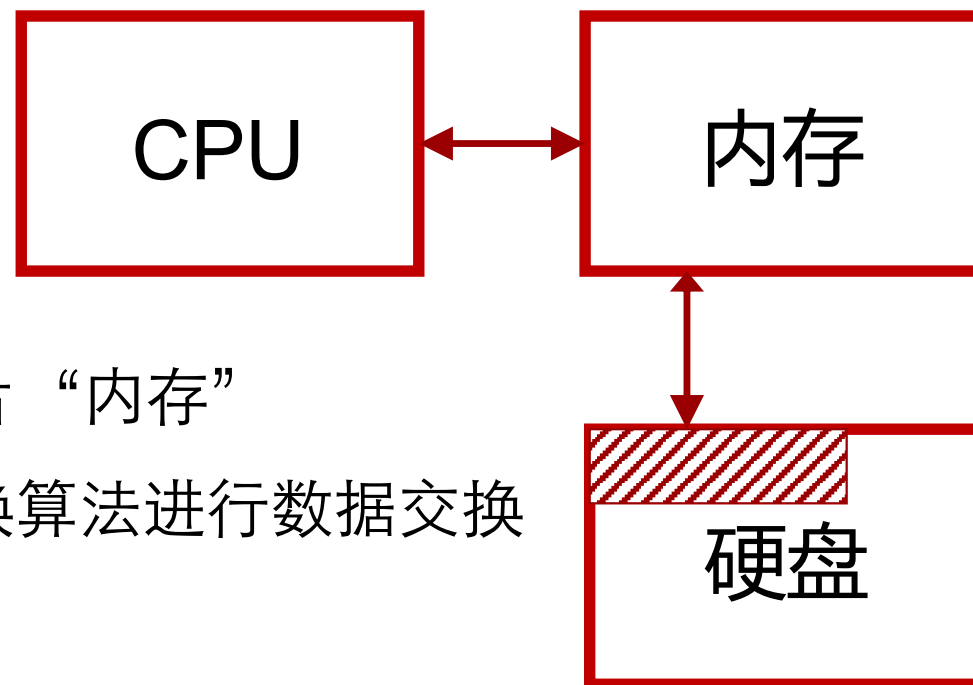
- 断电不丢失
- 存储程序和数据
- 速度慢
- 容量大

CPU寻址空间

- CPU对于内存寻址的能力
- 32位CPU的物理地址
 - $2^{32} = 4096\text{MB} = 4\text{GB}$ 个地址
- 目前主流的CPU是64位



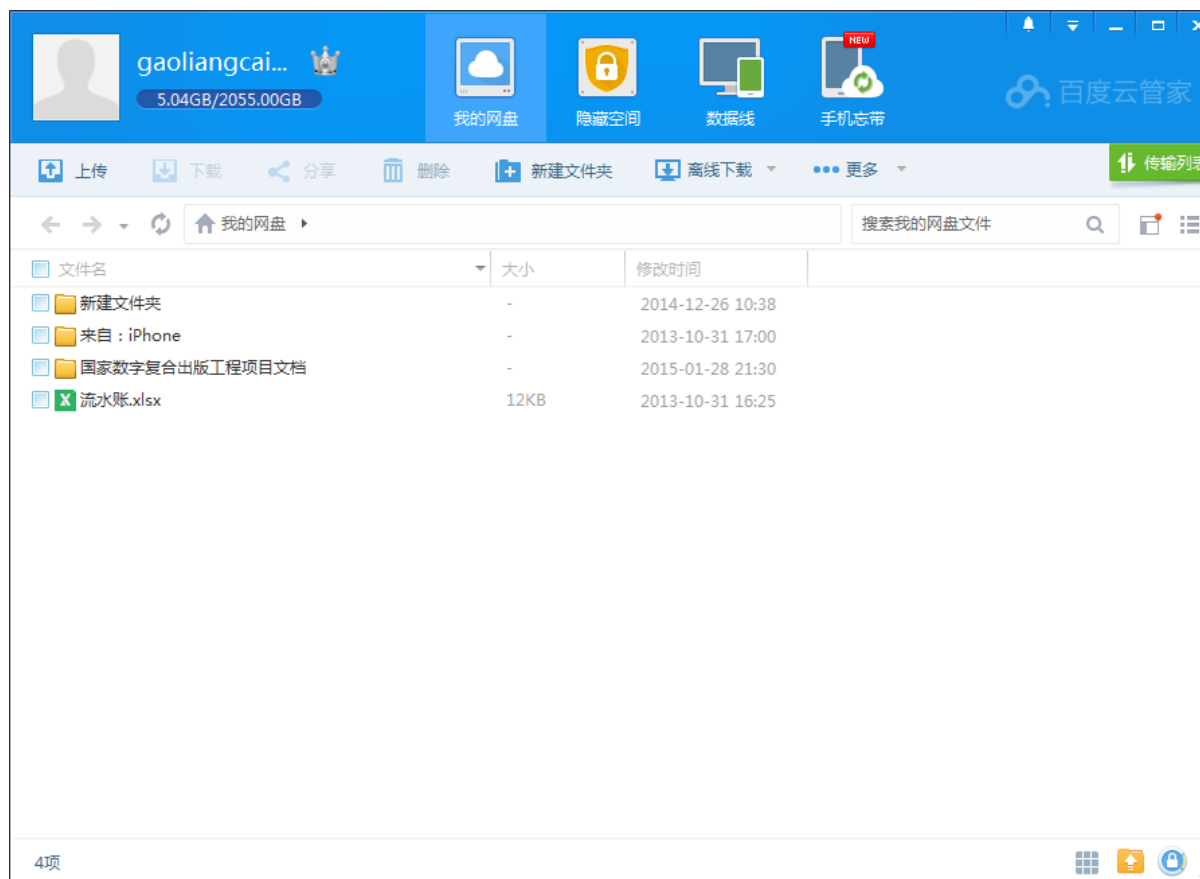
- 如果内存不足，大的程序可能运行不了，怎么办？



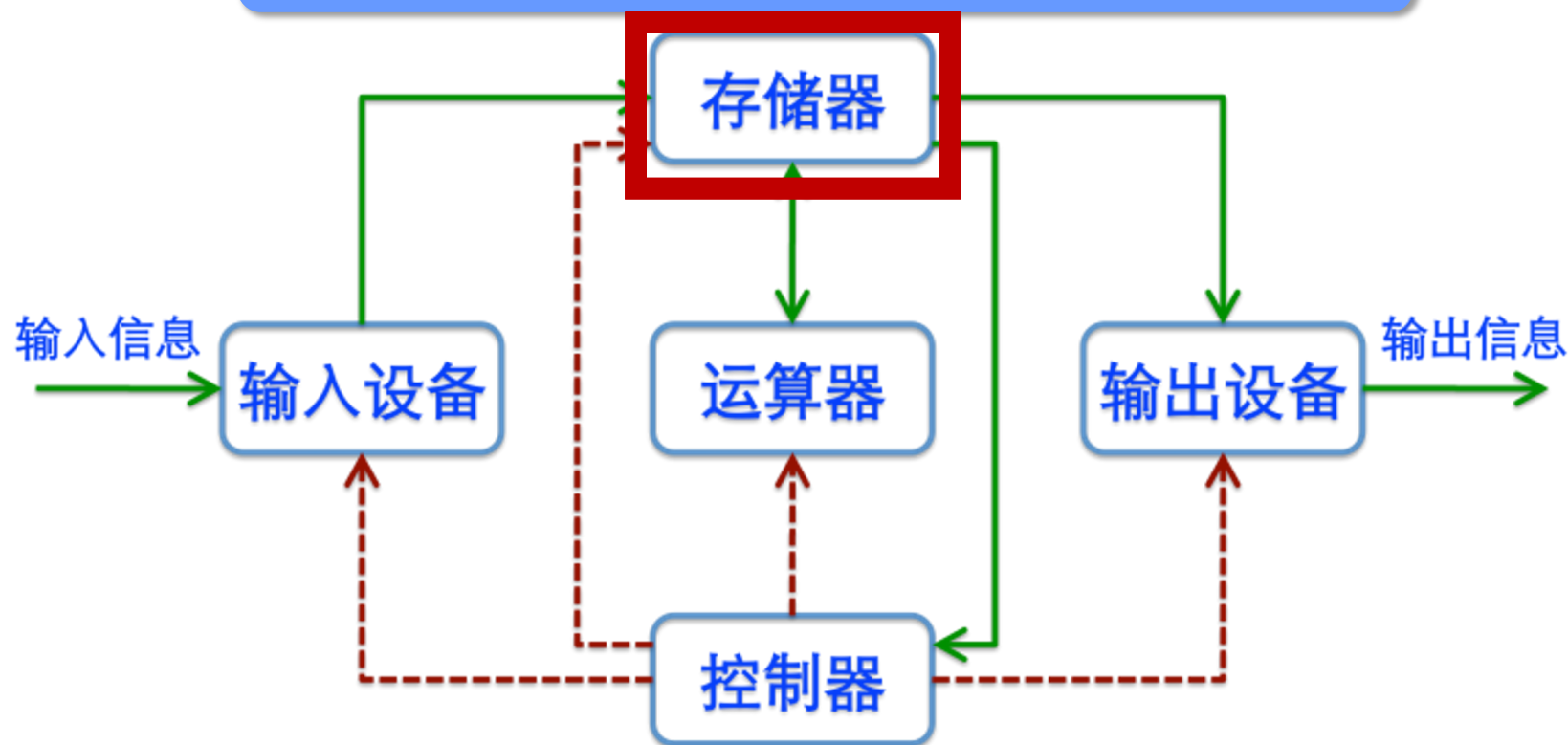
- 当程序启动时，在硬盘上虚拟出一片“内存”
- CPU访问内存，内存和硬盘通过置换算法进行数据交换

云存储

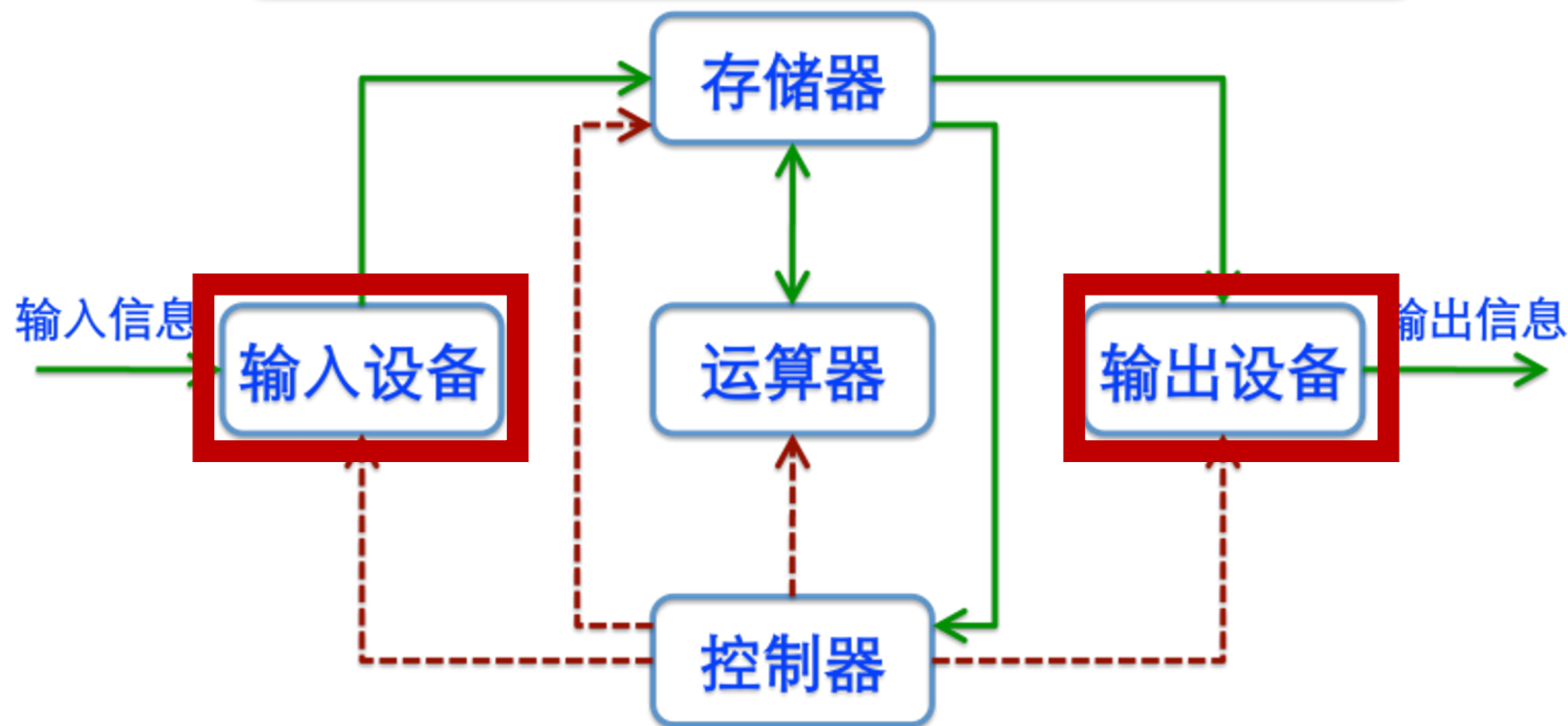
- 云存储
 - 北大网盘、百度云、阿里云、腾讯云、坚果云
- 速度受网络影响
- 容量不受限制
- 可共享



冯·诺依曼结构的五大部件



冯·诺依曼结构的五大部件



I/O 输入/输出设备

- 基本输入设备
 - 键盘
 - 鼠标
- 基本输出设备
 - 显示器



I/O 输入/输出设备

- 1. 早期的计算机采用**等待方式**向外部设备发出请求，在外部设备准备期间，一直等待
- 2. **中断方式**向外部设备发出请求，在外部设备准备期间，干别的事情
- 3. **DMA方式**把数据发给一个直接访问控制器，CPU解放出来

CPU需要等待外部设备执行完后，才能继续执行自己接下来的任务

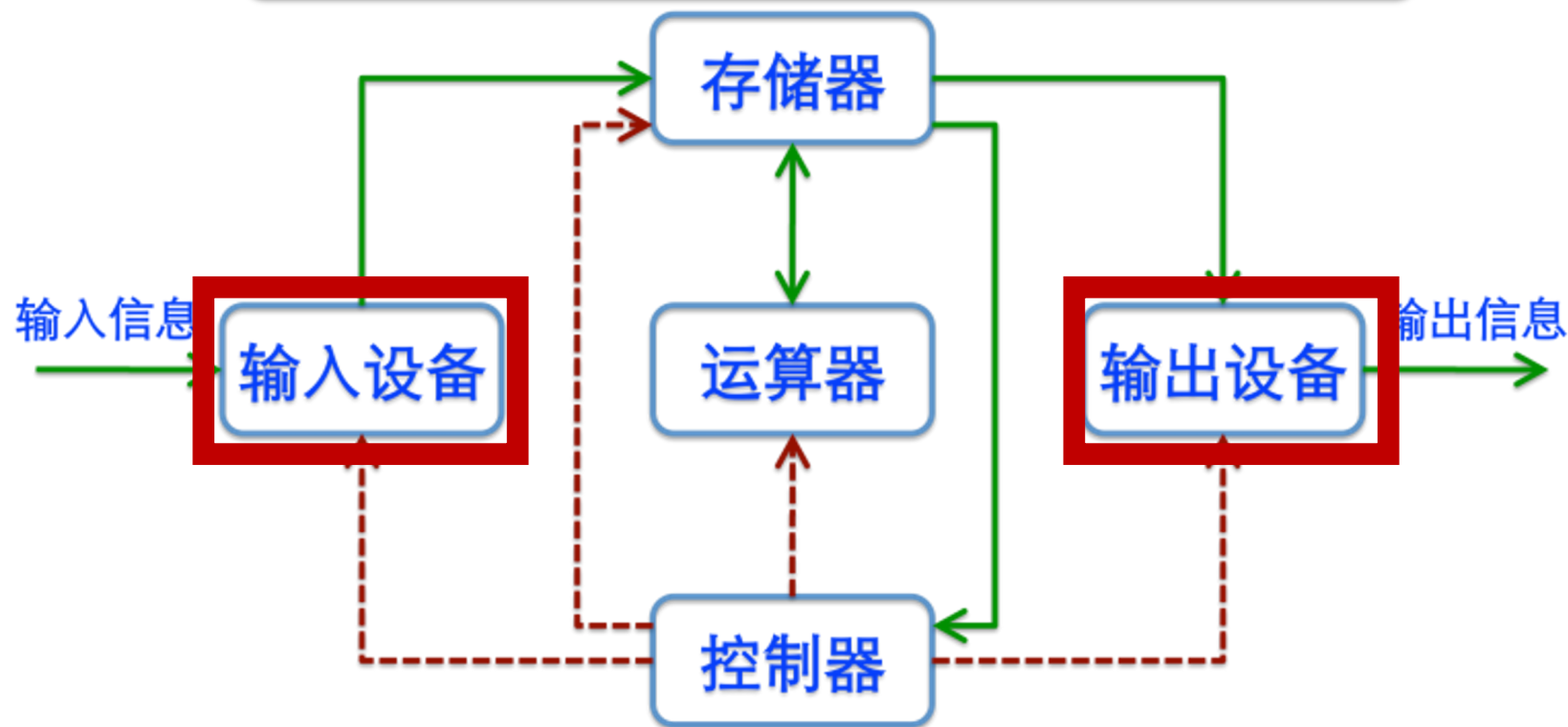
CPU发送请求以后，不需要等待，可以继续执行自己接下来的任务

Direct Memory Access 直接内存操作，通过硬件线路直接实现。CPU根本就不参与传送操作，省去了CPU取指令、取数、送数等操作

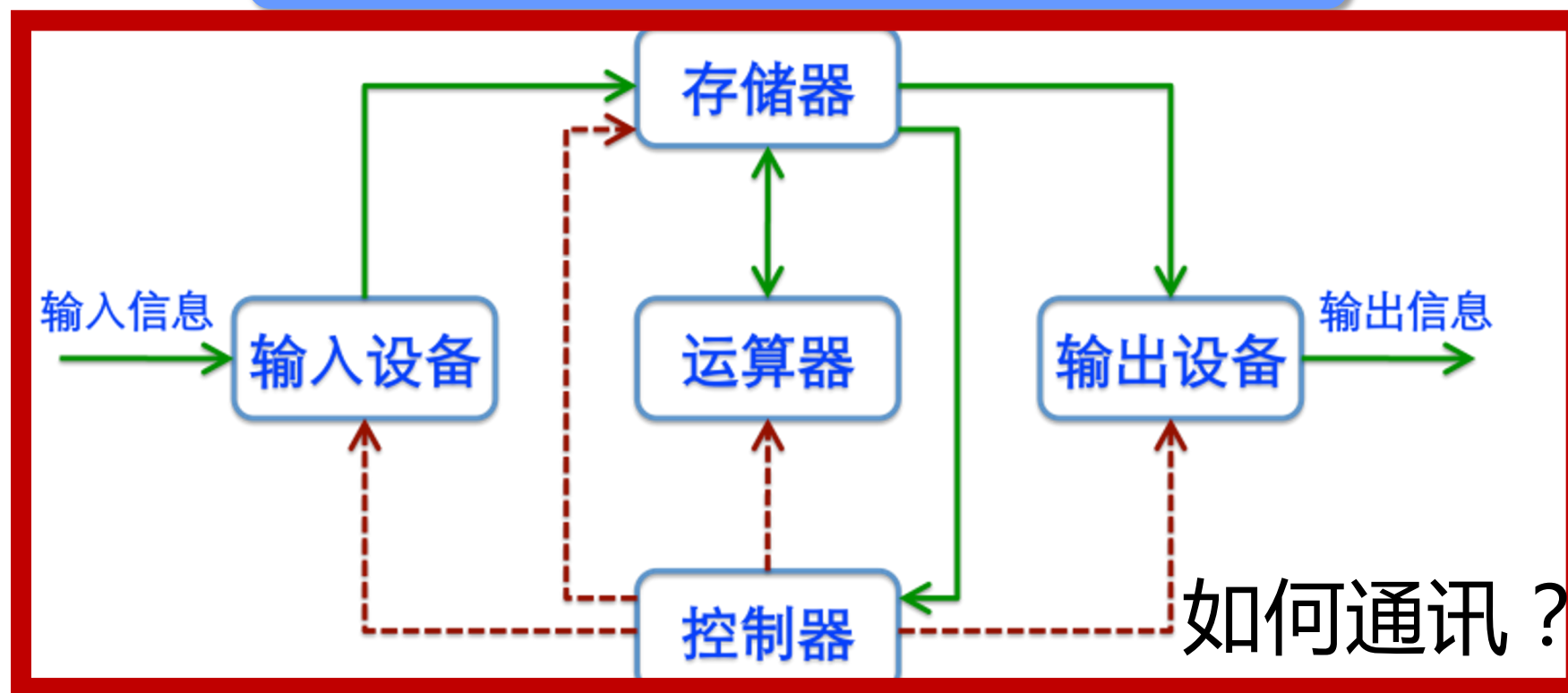
慢

快

冯·诺依曼结构的五大部件



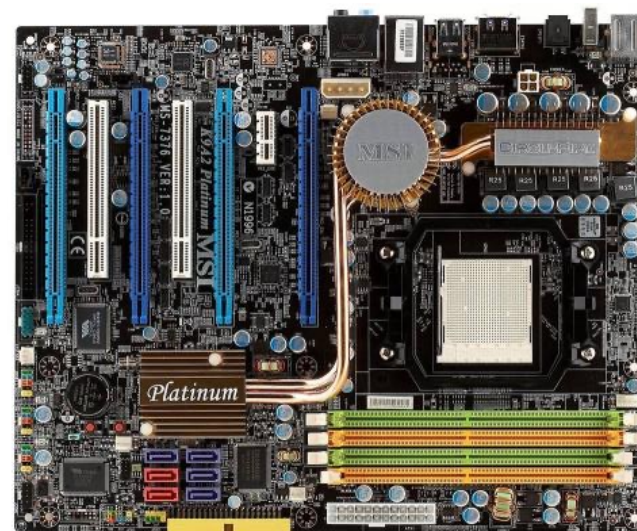
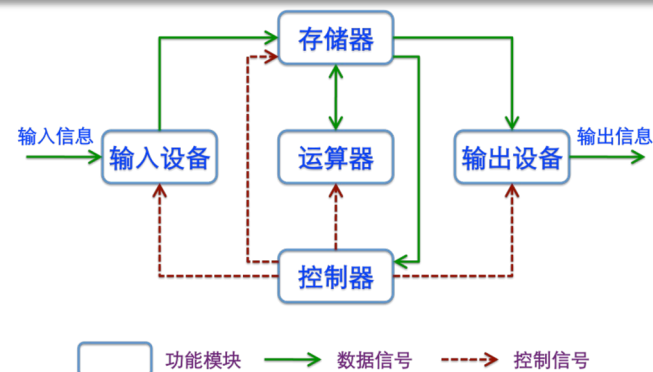
冯·诺依曼结构的五大部件



回到主板

- 个人计算机的**电路主体**
- 通过**总线**将CPU、存储设备以及各种输入/输出设备连接起来
- 芯片组（南桥、北桥）**控制链接**
- 还提供**多种接口**，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

冯·诺依曼结构的五大部件



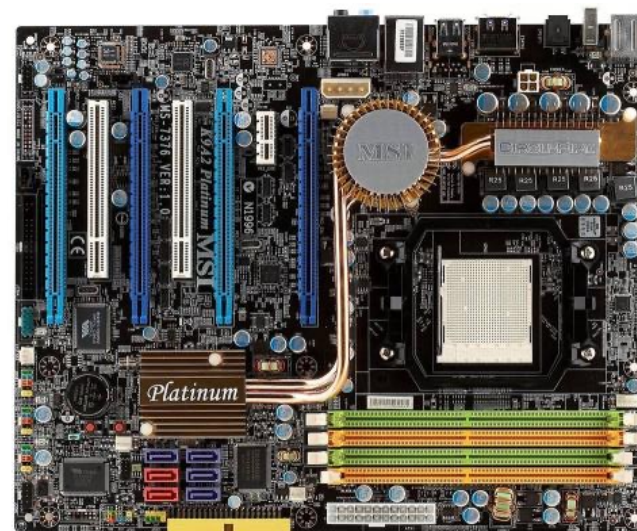
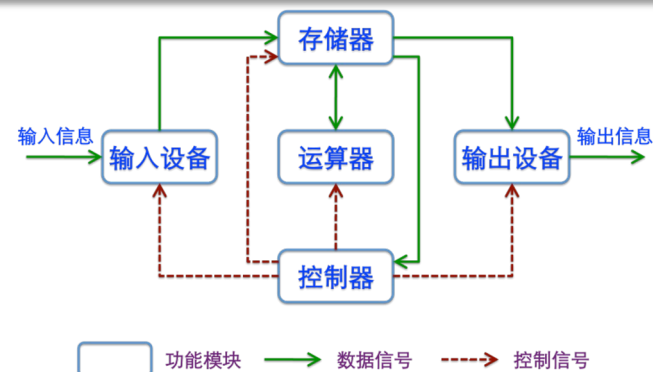
回到主板

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来

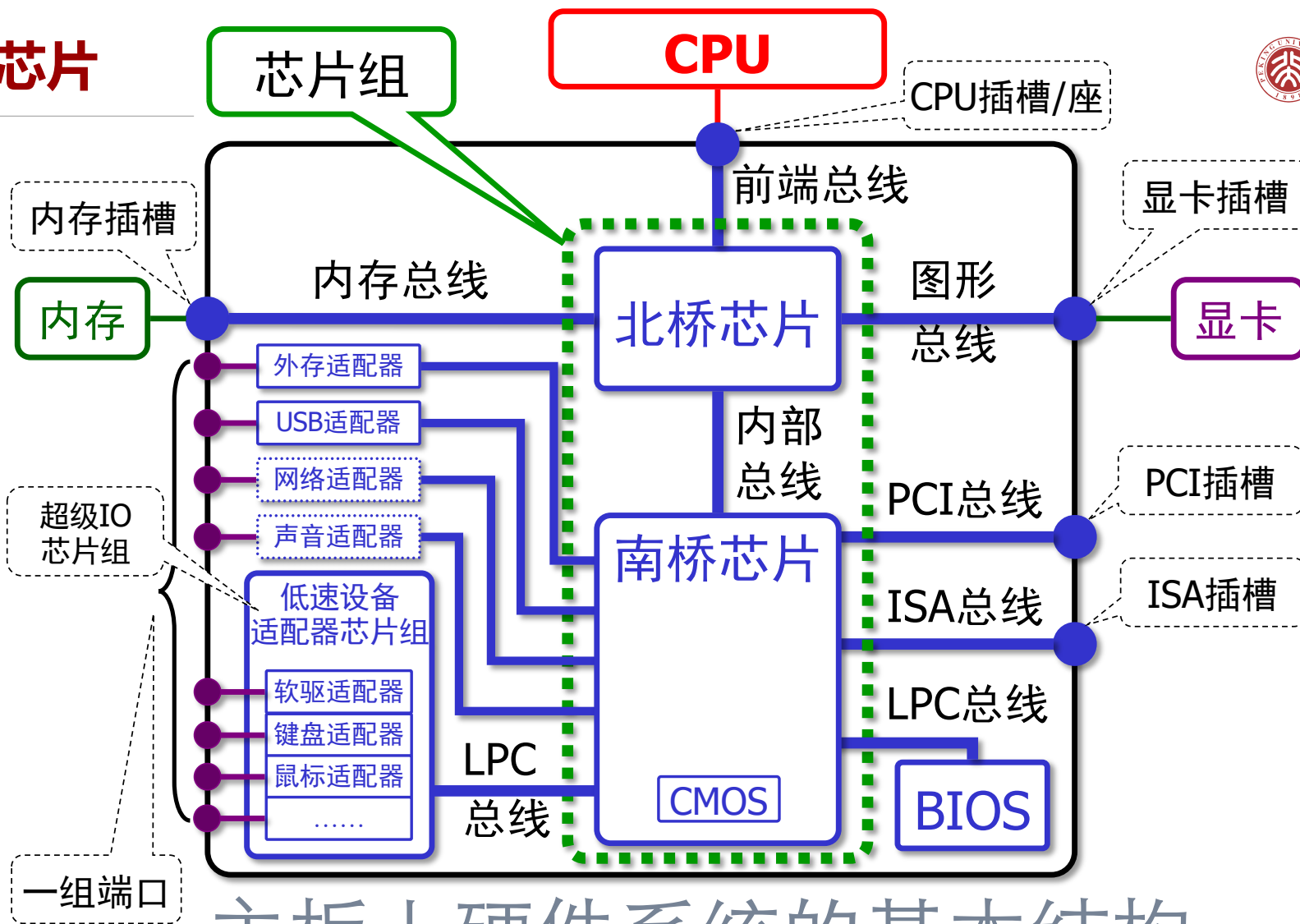
■ 芯片组（南桥、北桥）控制链接

- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

冯·诺依曼结构的五大部件

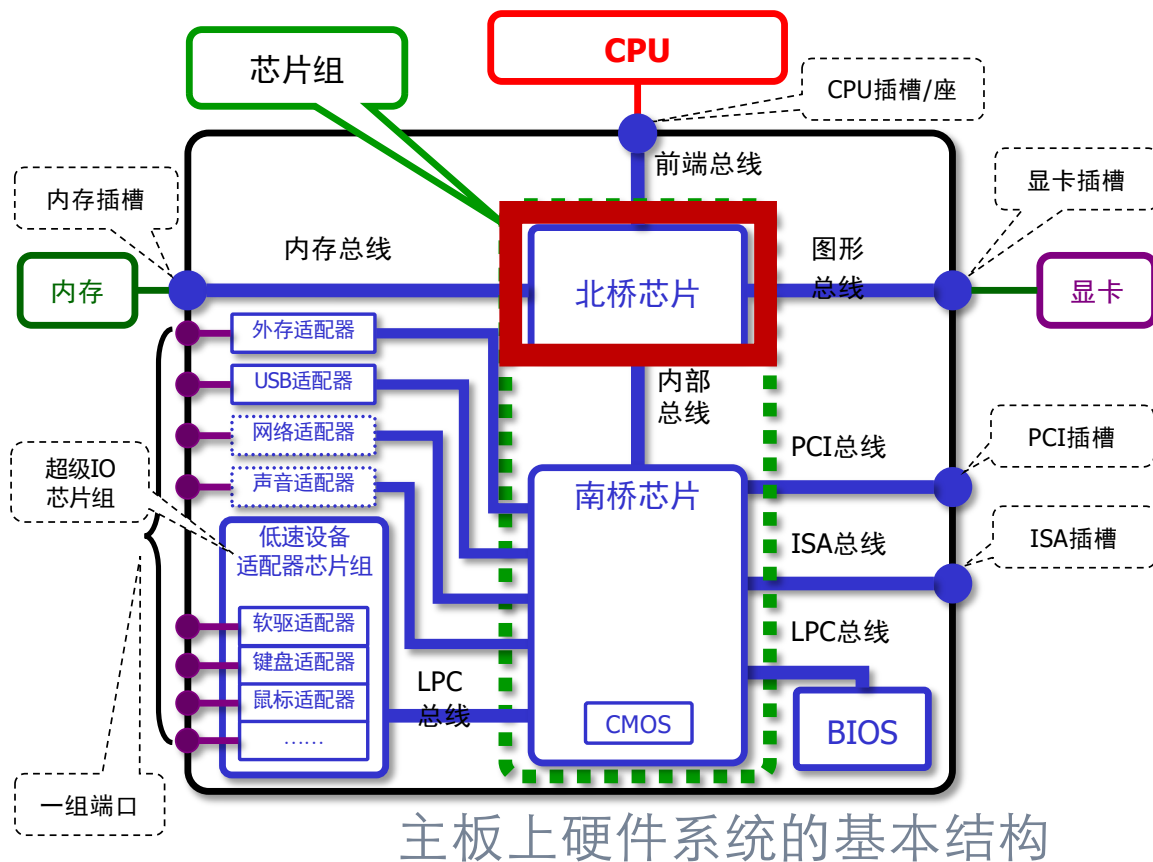


南北桥芯片



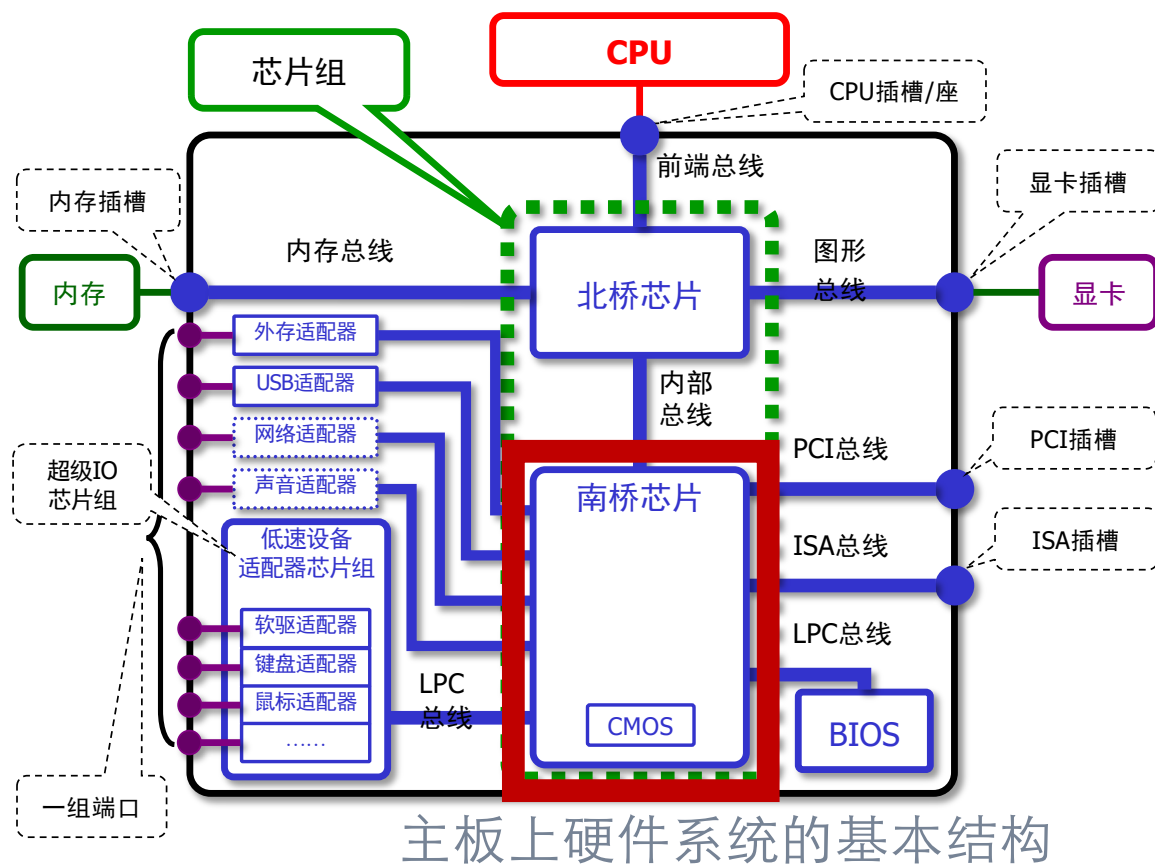
主板上硬件系统的基本结构

南北桥芯片



北桥芯片

- 主板芯片组中起主导作用
- 离CPU最近的新品，负责与CPU联系并控制内存，对建立通信起到很大作用



南桥芯片

- 负责I/O总线之间的通信
- 如USB、LAN、键盘控制器、音频控制器、电源管理等

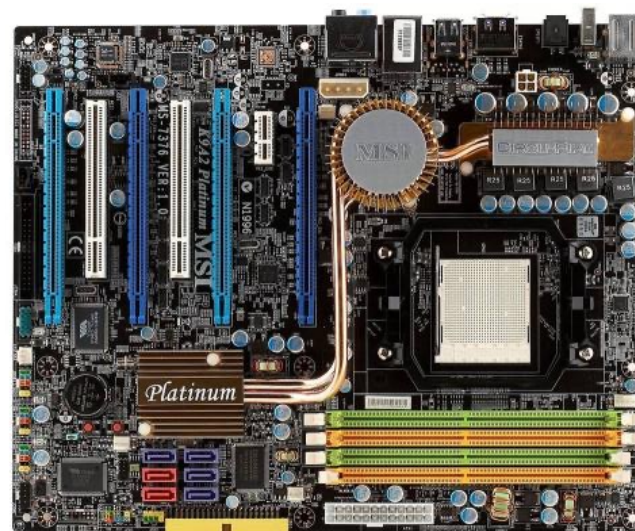
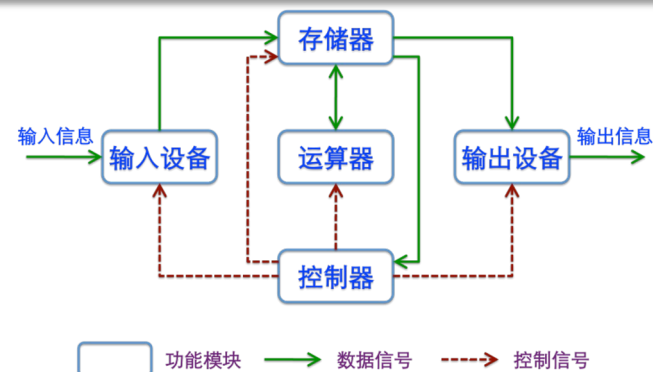
回到主板

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来

■ 芯片组（南桥、北桥）控制链接

- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

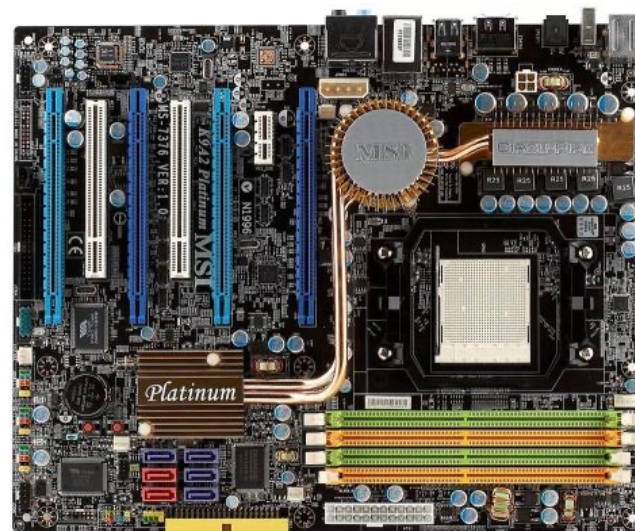
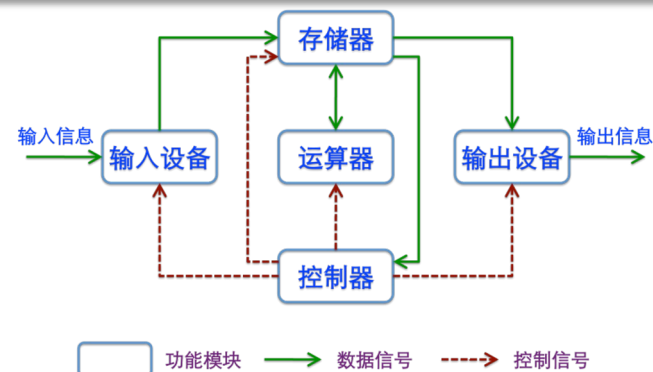
冯·诺依曼结构的五大部件



回到主板

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来
- 芯片组（南桥、北桥）控制链接
- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

冯·诺依曼结构的五大部件

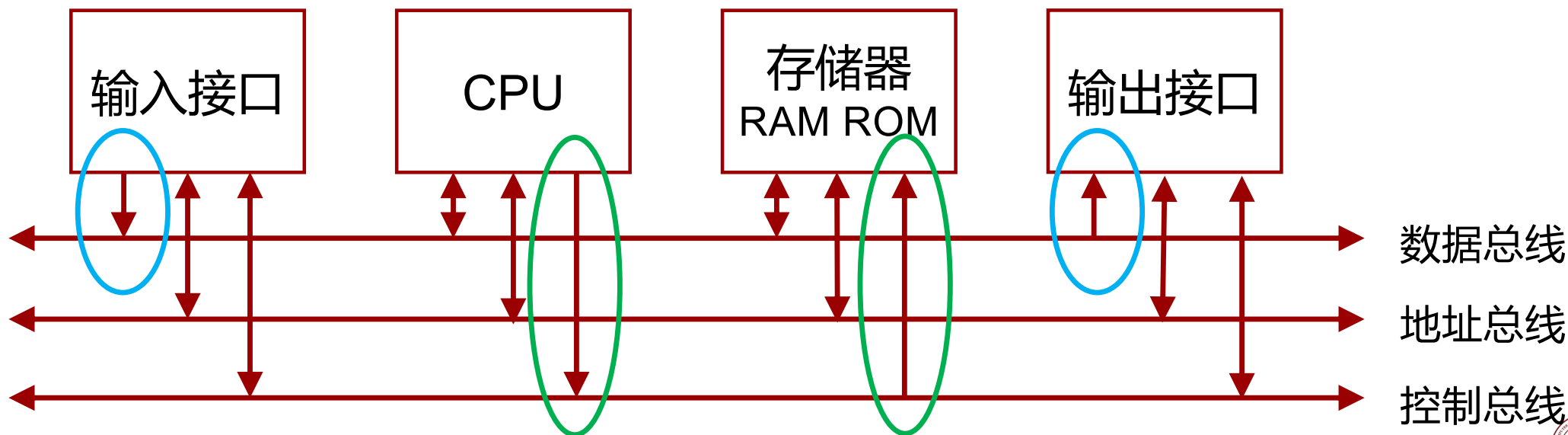


总线 BUS

- **数据总线**： 传送数据
- **地址总线**： 传送地址
- **控制总线**： 传送指令

总线宽度 == 字长

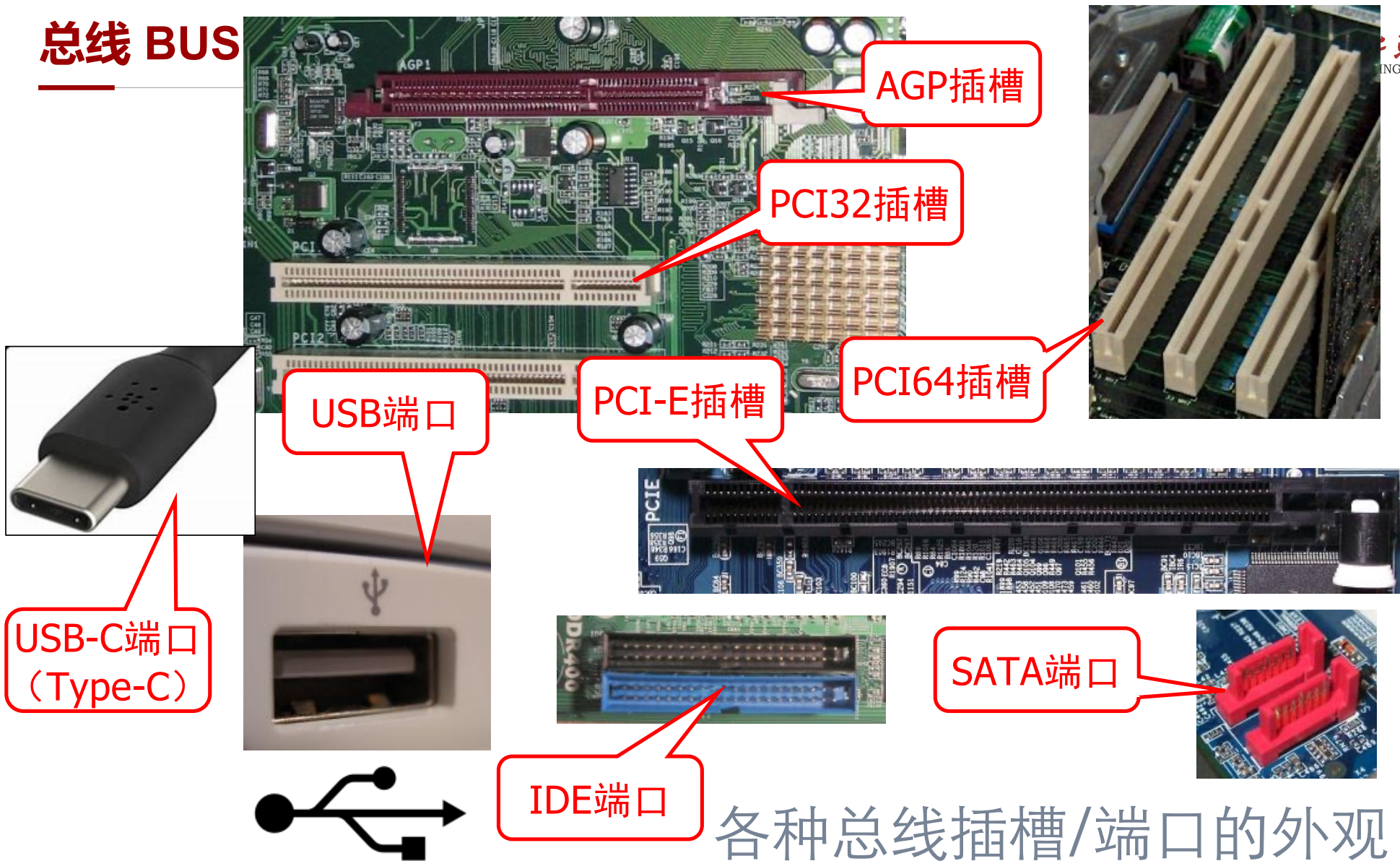
越长则表示更多的存储单元地址，能计算更大的数据



总线 BUS

- 常见的总线标准：
- ISA (Industrial Standard Architecture Bus) : 8/16位
- PCI (Peripheral Component Interconnection Bus) : 32/64位
- AGP (Accelerated Graphics Port) : 显卡专用总线
- PCI-E (PCI Express) : PCI的升级版；替代PCI和AGP
- IDE (Integrated Drive Electronics) : 用于连接硬盘、光驱
- SATA (Serial Advanced Technology Attachment) : IDE的升级版
- USB (Universal Serial Bus) : 由IBM、Intel及Microsoft等多家公司共同开发的新型外设连接技术
 - USB-C (也称为Type C) 最新的USB标准

总线 BUS

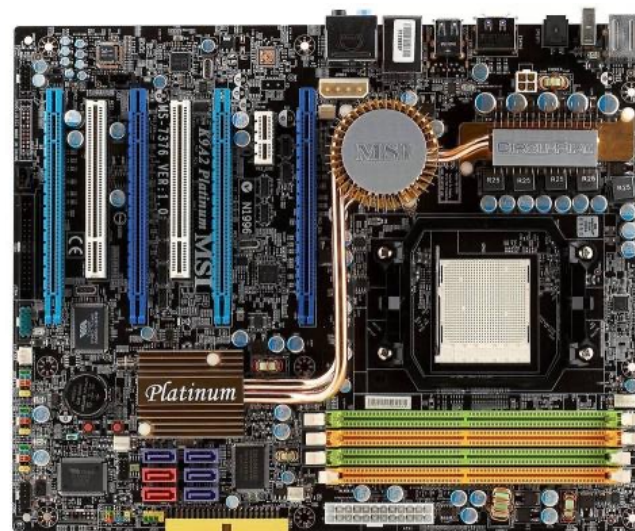
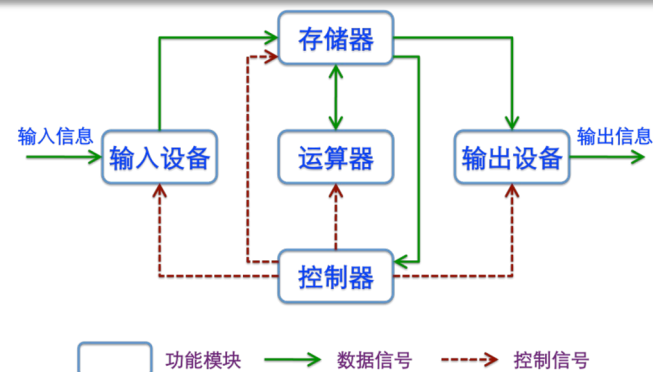


各种总线插槽/端口的外观

回到主板

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来
- 芯片组（南桥、北桥）控制链接
- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

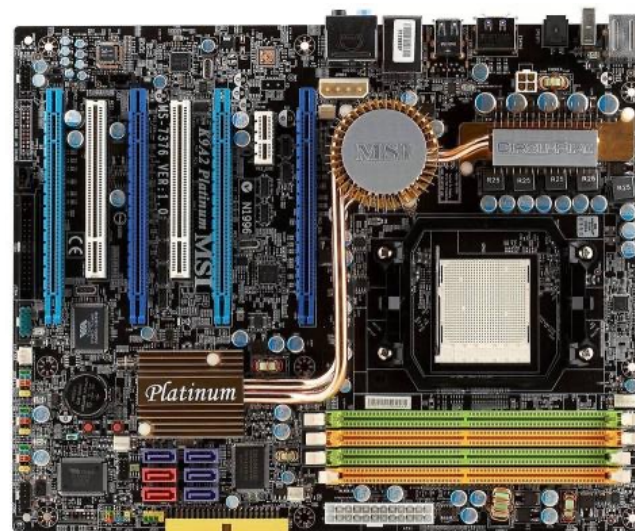
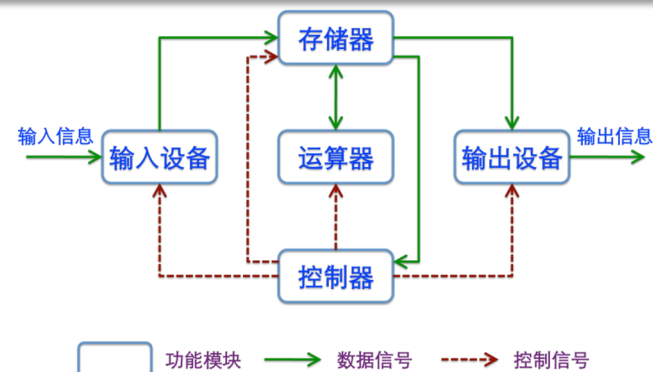
冯·诺依曼结构的五大部件



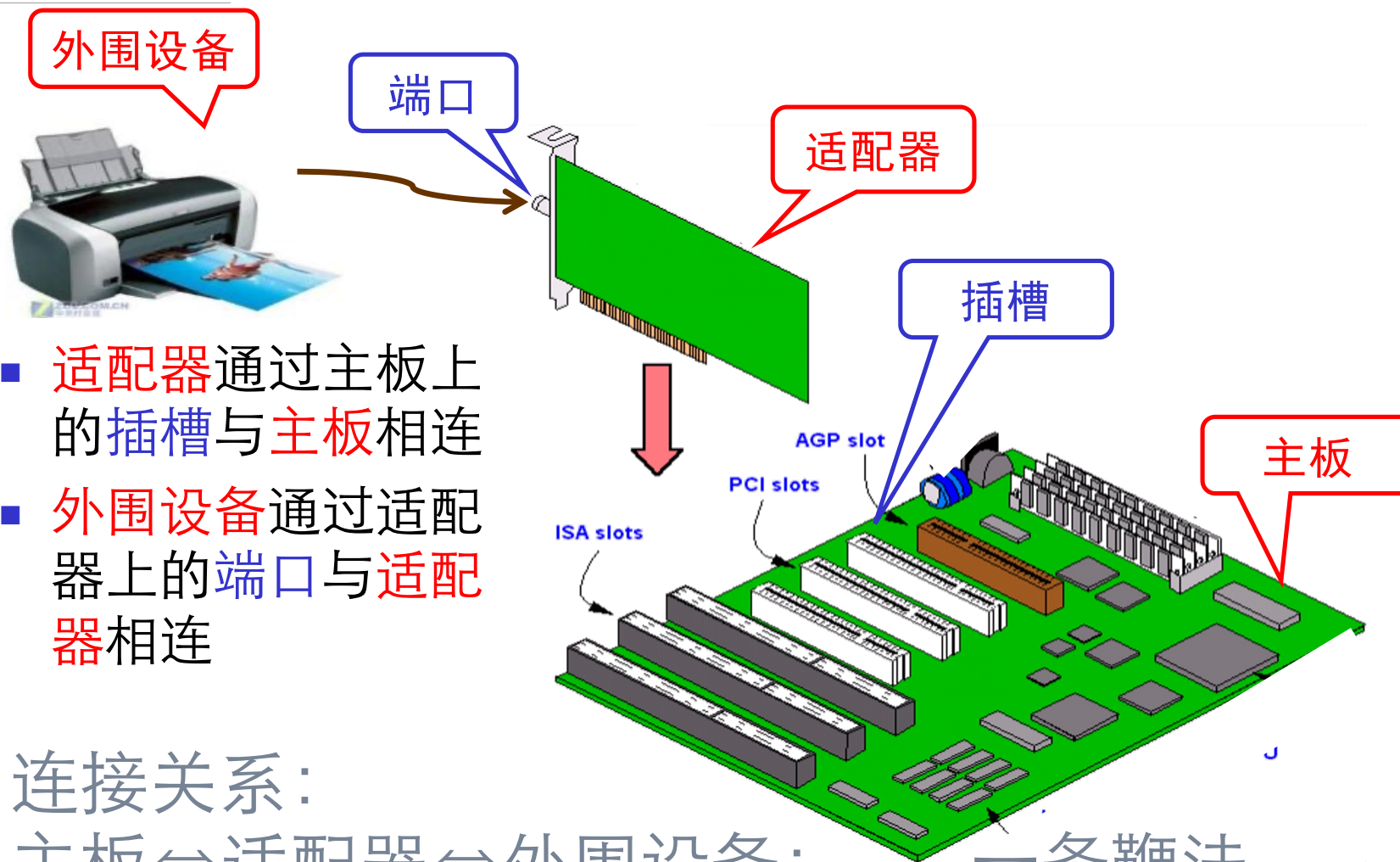
回到主板

- 个人计算机的电路主体
- 通过总线将CPU、存储设备以及各种输入/输出设备连接起来
- 芯片组（南桥、北桥）控制链接
- 还提供多种接口，以便将更多的存储设备和输入/输出设备（外围设备）连接到主机中来

冯·诺依曼结构的五大部件



外围设备



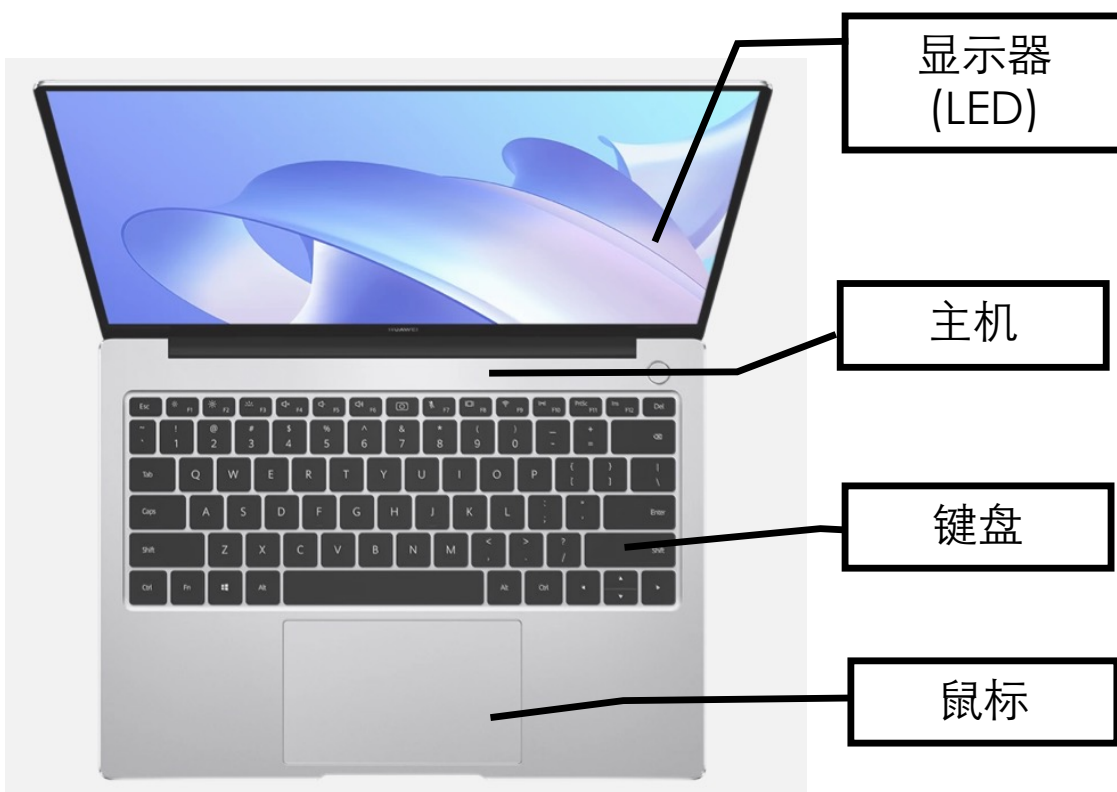
- 适配器通过主板上的插槽与主板相连
- 外围设备通过适配器上的端口与适配器相连

连接关系：

主板 ↔ 适配器 ↔ 外围设备：一条鞭法

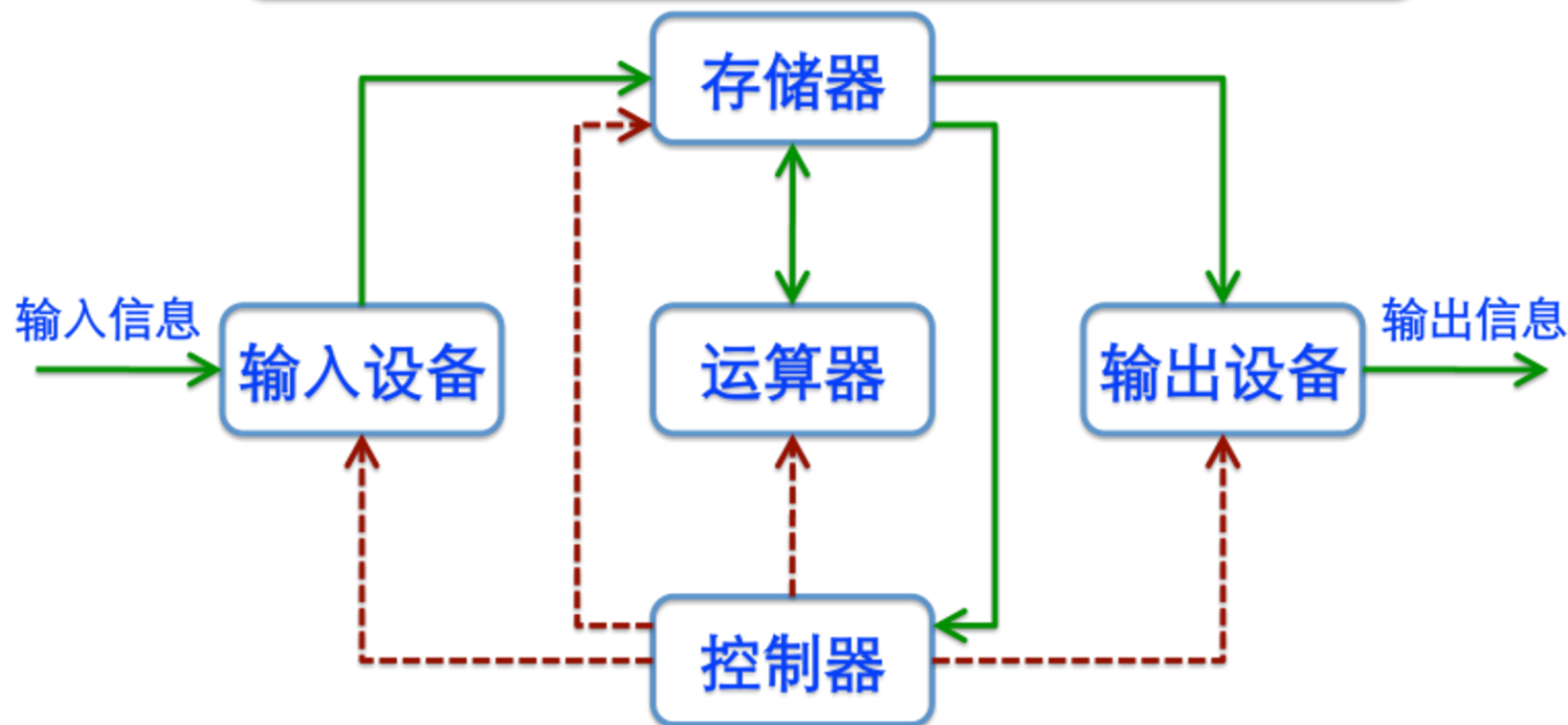
回到主机

HUAWEI MateBook 14 2021



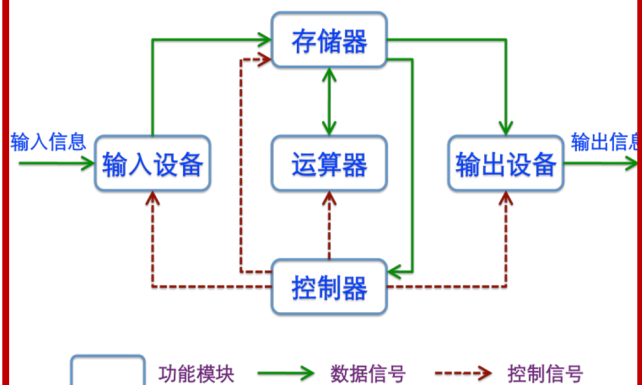
CPU : Intel®酷睿™ i5-1135G7
显卡 : NVIDIA® GeForce® MX450
内存 : 16 GB DDR4
硬盘 : 512 GB NVMe PCIe SSD

冯·诺依曼结构的五大部件



操作系统 OS

计算机硬件



- 一个简单的程序中会涉及很多部件和设备的操作。这些操作的控制不是我们在程序中完成的，看起来像是自动的，这就是操作系统（Operation System, OS）帮我们做的

屏蔽复杂，提供简单

计算机系统结构

Algorithm and Application

算法与应用

Programming Language and Compiler

编程语言和编译器

Operating System/Virtual Machines

操作系统

Instruction Set Architecture (ISA)

} 计算机系统结构

Micro Architecture

Gates/Register-Transfer Level (RTL)

数字系统

Circuits

集成电路

Devices

制造工艺

Physics

材料的物理特性

从计算机外观出发（由外而内）看热闹

从“冯·诺依曼”结构出发（由内而外）看门道

总结

■ 构成计算的一些基本要素

- 没有计算模型，就不可能有通用的计算机
- 计算机更新换代，但是冯·诺依曼结构没有变化
- 存储程序原理是现代通用计算机的核心思想
- 二进制是电子数字计算机电子元件当前的最佳选择（苏联曾经有三进制）
- 布尔代数奠定了数字逻辑的理论基础
- 硬件逻辑与指令系统是互为决定的
- 机器语言--> 汇编语言--> 高级语言--> 更高级 ..是计算机发展的必然产物
- 算法决定一个问题的能行问题
- 好的软件取决于好的程序设计方法和新的软件工程方法在构成计算机的一些基本的要素
- 好的系统结构是支撑算法和应用的基本保障

谢谢！

- 基础原理
- 硬件系统 ◀
- 软件基础
- 信息表示
- 网络与存储
- 程序设计语言

随堂测试



课程网站 <https://jigai.tk>

个人网站 <https://zsdonghao.github.io>

hao.dong@pku.edu.cn