

# ARM-嵌入式系统的总体结构

# ARM嵌入式系统

1. 嵌入式系统的总体结构
2. 硬件基础知识
3. 什么是ARM ?
4. ARM处理器功能特点介绍

# 嵌入式系统组成



被控对象



嵌入式计算机系统

# 1 嵌入式系统的总体结构

- 目前所提及的嵌入式系统一般指嵌入式计算机系统，主要包括：硬件层、中间层、系统软件层和应用层4个部分。
- 嵌入式硬件主要包括提供嵌入式计算机正常运行的最小系统（如电源、系统时钟、复位电路、存储器等）、通用I/O口和一些外设及其它设备。嵌入式系统中间层又称嵌入式硬件抽象层，如硬件驱动程序、系统启动软件等；嵌入式系统软件层为应用层提供系统服务，如操作系统、文件系统、图形用户接口等；而应用层主要是用户应用程序。

# 硬件层

嵌入式系统硬件通常指除被控对象之外的嵌入式系统要完成其功能所具备的各种设备，由嵌入式处理器、存储器系统、通用设备接口（A/D、D/A、I/O等）和一些扩展外设组成。

- 嵌入式系统的硬件层是以嵌入式处理器为核心的
- 嵌入式系统外设是指为了实现系统功能而设计或提供的接口或设备

# 1 嵌入式系统的总体结构

## 中间层

介于硬件层与系统软件层之间，将硬件的细节进行屏蔽，便于操作系统调用，因此称为为中间层，也称硬件抽象层（**Hardware Abstract Layer, HAL**）或板级支持包（**Board Support Package**）。具有硬件相关性和操作系统相关性特点。



# 主要包括系统初始化和设备驱动程序

## ● 系统初始化

- (1) 片级初始化：主要完成**CPU**的初始化，包括设置**CPU**的核心寄存器和控制寄存器，**CPU**核心工作模式以及**CPU**的局部总线模式等。
- (2) 板级初始化：完成**CPU**以外的其它硬件设备的初始化。除此之外，还要设置某些软件的数据结构和参数，为随后的系统级初始化和应用程序的运行建立硬件和软件环境。
- (3) 系统级初始化：这是一个以软件初始化为主的过程，主要是进行操作系统初始化。

## ● 硬件相关的设备驱动程序

- 与系统初始化过程相反，硬件相关的设备驱动程序的初始化和使用通常是一个从高层到底层的过程。
- 与操作系统中通用的设备驱动程序关联起来，在应用中由通用的设备驱动程序调用，实现对硬件设备的操作

# 系统软件层

系统软件由操作系统（**OS**）、文件系统（**FS**）、图形用户接口（**GUI**）、网络系统（**NM**）及通用组件模块（如**TCP/IP**协议包）等组成。

1. **RTOS**是嵌入式应用软件的基础和开发平台
2. **FS**是操作系统用于明确磁盘或分区上的文件的方法和数据结构；即在磁盘上组织文件的方法。
3. **GUI** 是 **Graphical User Interface** 的简称，即图形用户接口，准确来说 **GUI** 就是屏幕产品的视觉体验和互动操作部分。
4. **TCP/IP**协议包简称**Ip Pack**，**IP Pack**是**TCP/IP**网络协议软件，通常作为操作系统的一个重要组成部分。

## 功能层

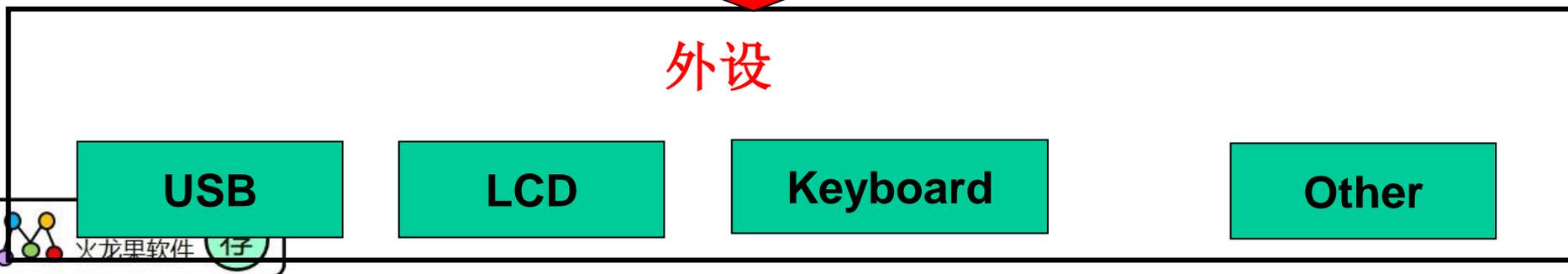
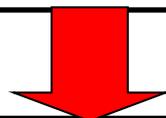
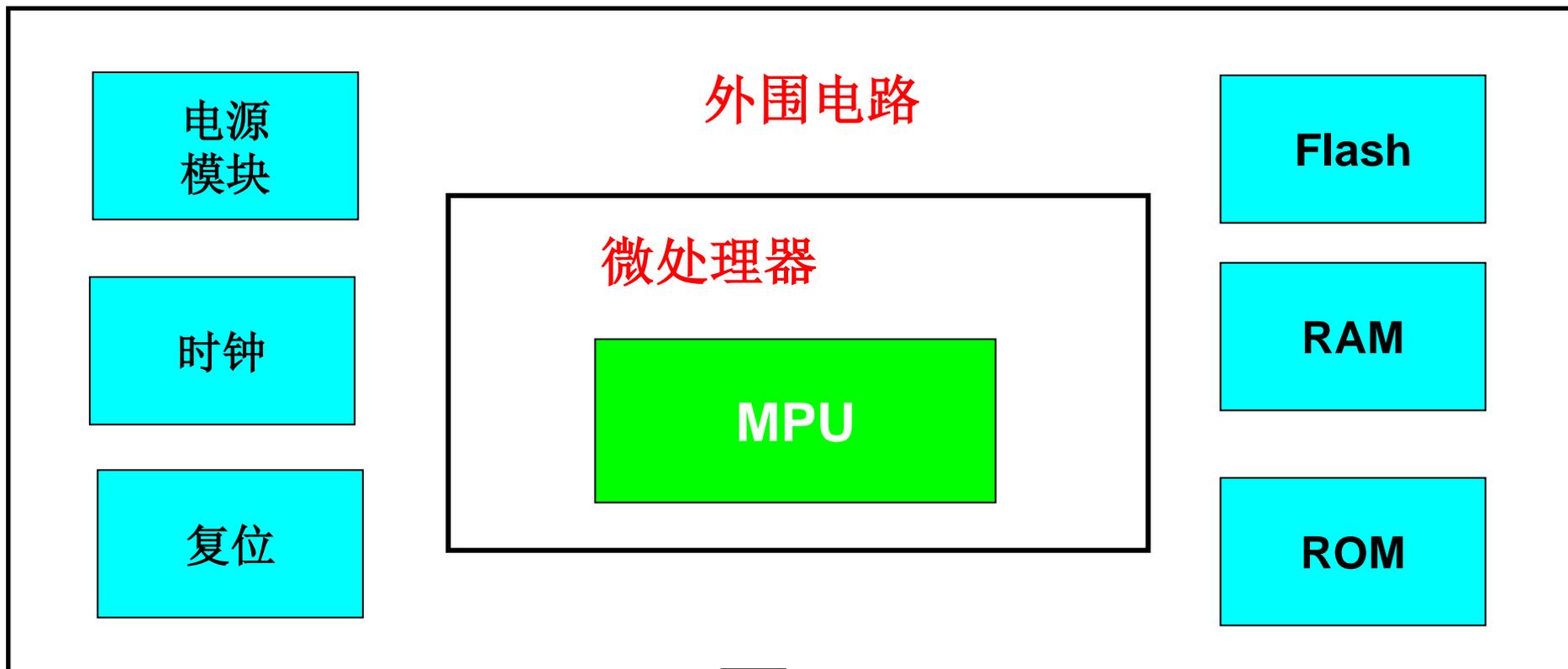
功能层又基于系统软件开发的应用软件程序组成，用来完成对被控对象的控制功能。直接与最终用户交互，**决定整个产品的成败**，是嵌入式软件的核心部分，根据用户需求定做的。

## 2 嵌入式系统硬件基础知识

主要从处理器和存储器进行介绍

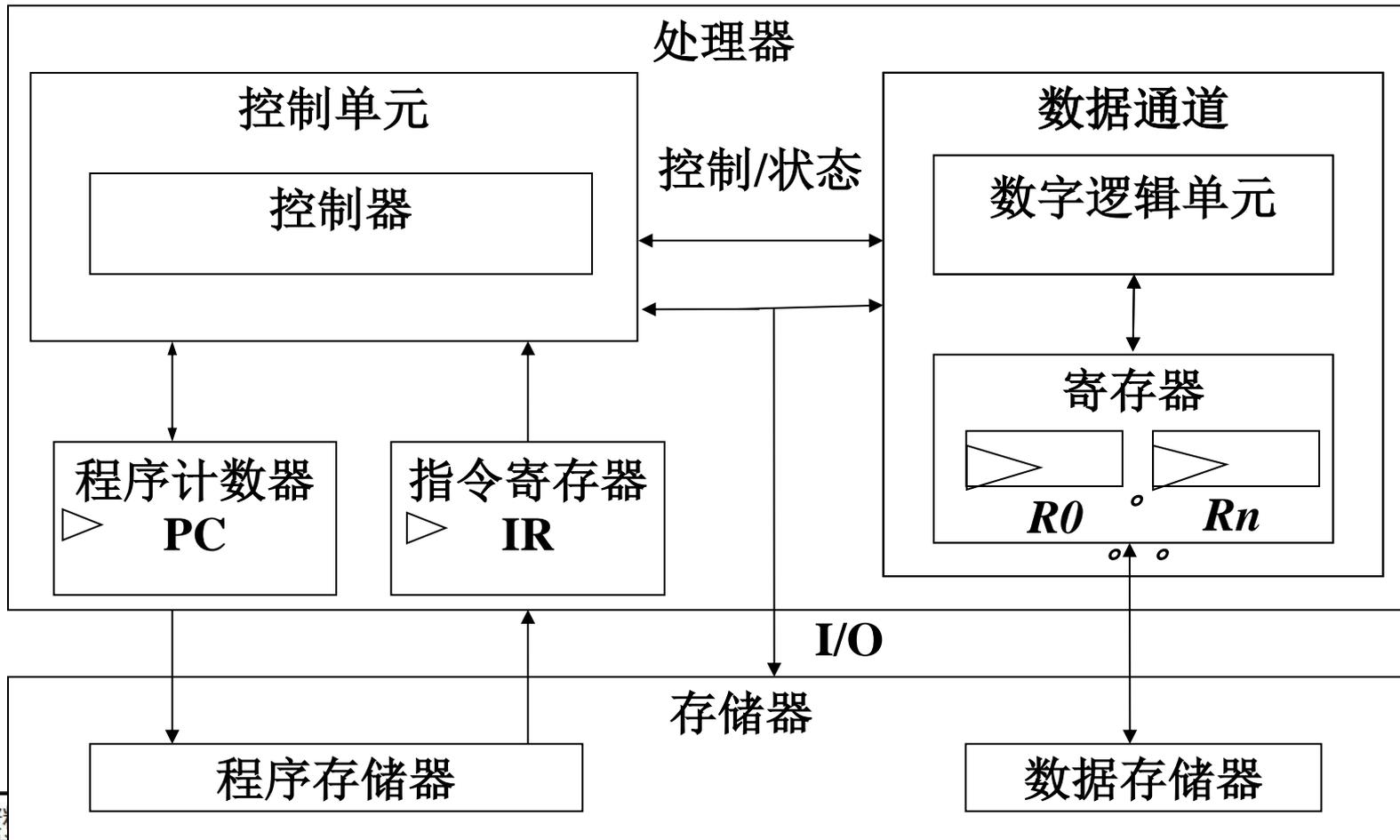
- 冯·诺依曼体系结构和哈佛体系结构
- **CISC与RISC**
- 影响**CPU**性能的因素
- 存储器系统

# 典型嵌入式系统基本组成—硬件



# 1. 微处理器结构

典型的微处理器由控制单元、程序计数器（PC）、指令寄存器（IR）、数据通道、存储器等组成



## 2. 处理器指令执行过程

指令执行过程一般分为：

- 取指

- 从存储器中获得下一条执行的指令读入指令寄存器
- PC**: 程序计数器, 总是指向下一条将要执行的指令
- IR**: 指令寄存器, 用于保持已取得指令

- 译码

- 解释指令, 决定指令的执行意义

- 执行

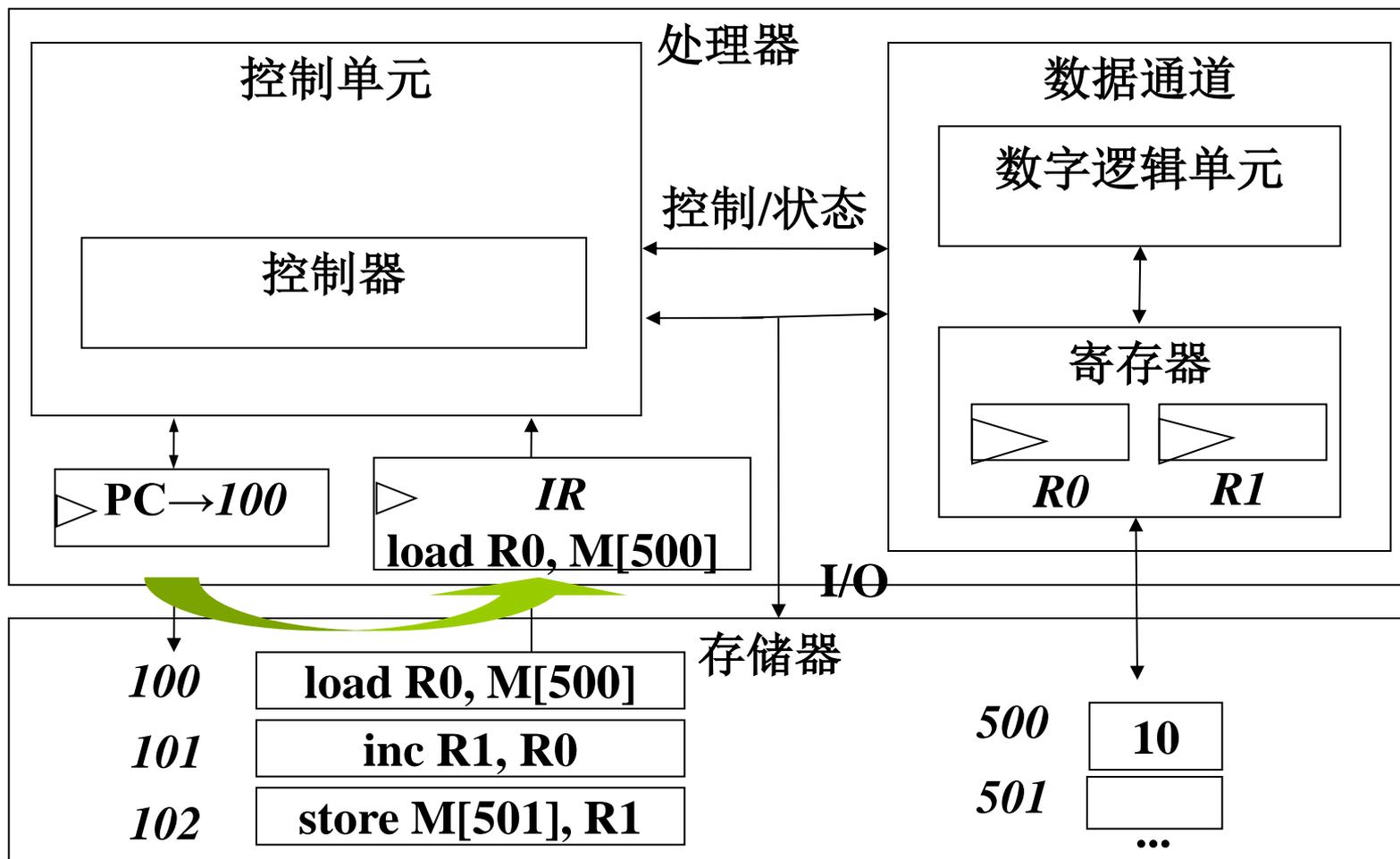
- 从存储器向数据通道寄存器移动数据
- 通过算术逻辑单元**ALU**进行数据操作

- 存储

从寄存器向存储器写数据

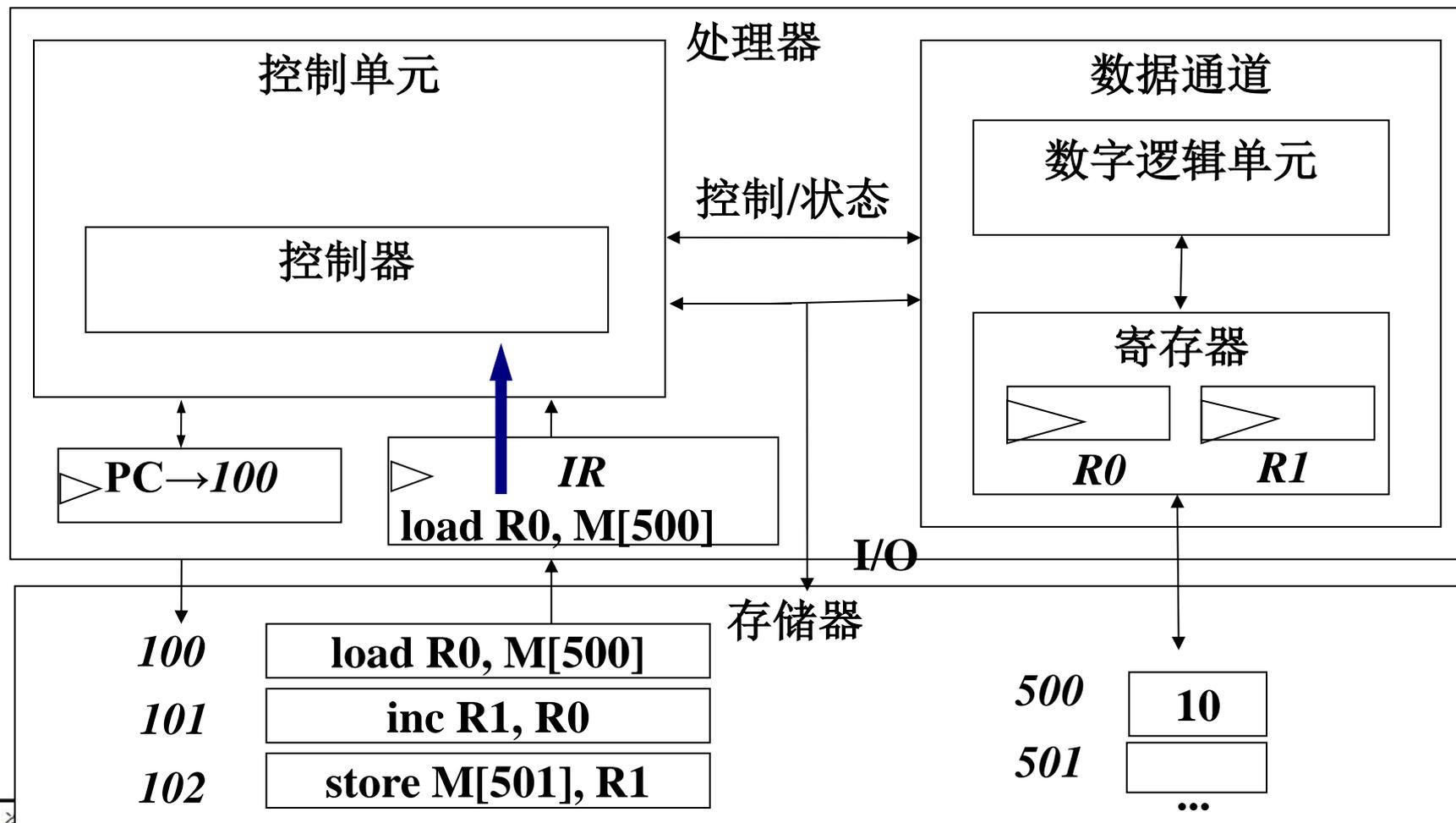
# (1) 取指

- 处理器从程序存储器中取出指令



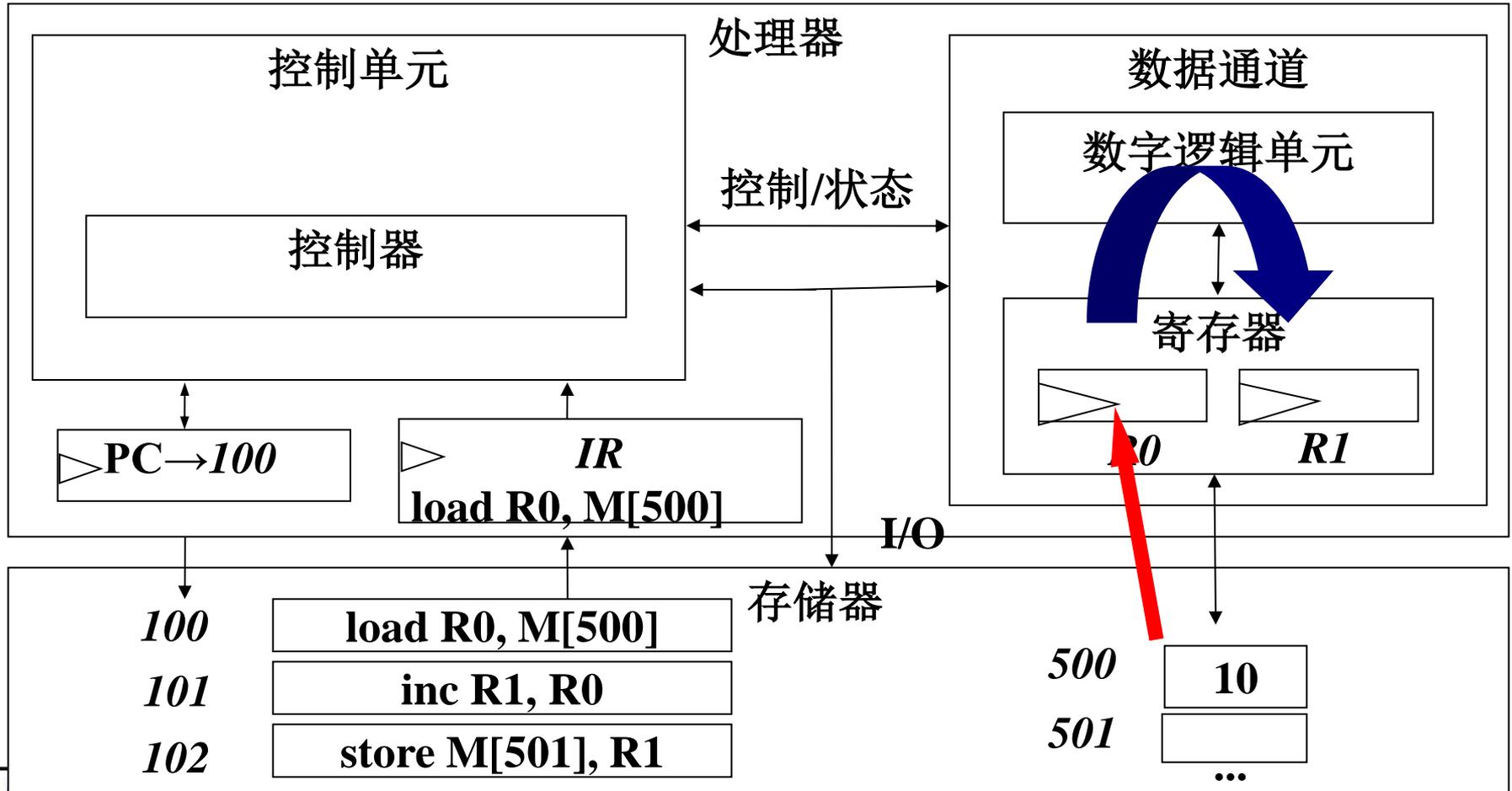
## (2) 译码

- 解释指令，决定指令的执行意义



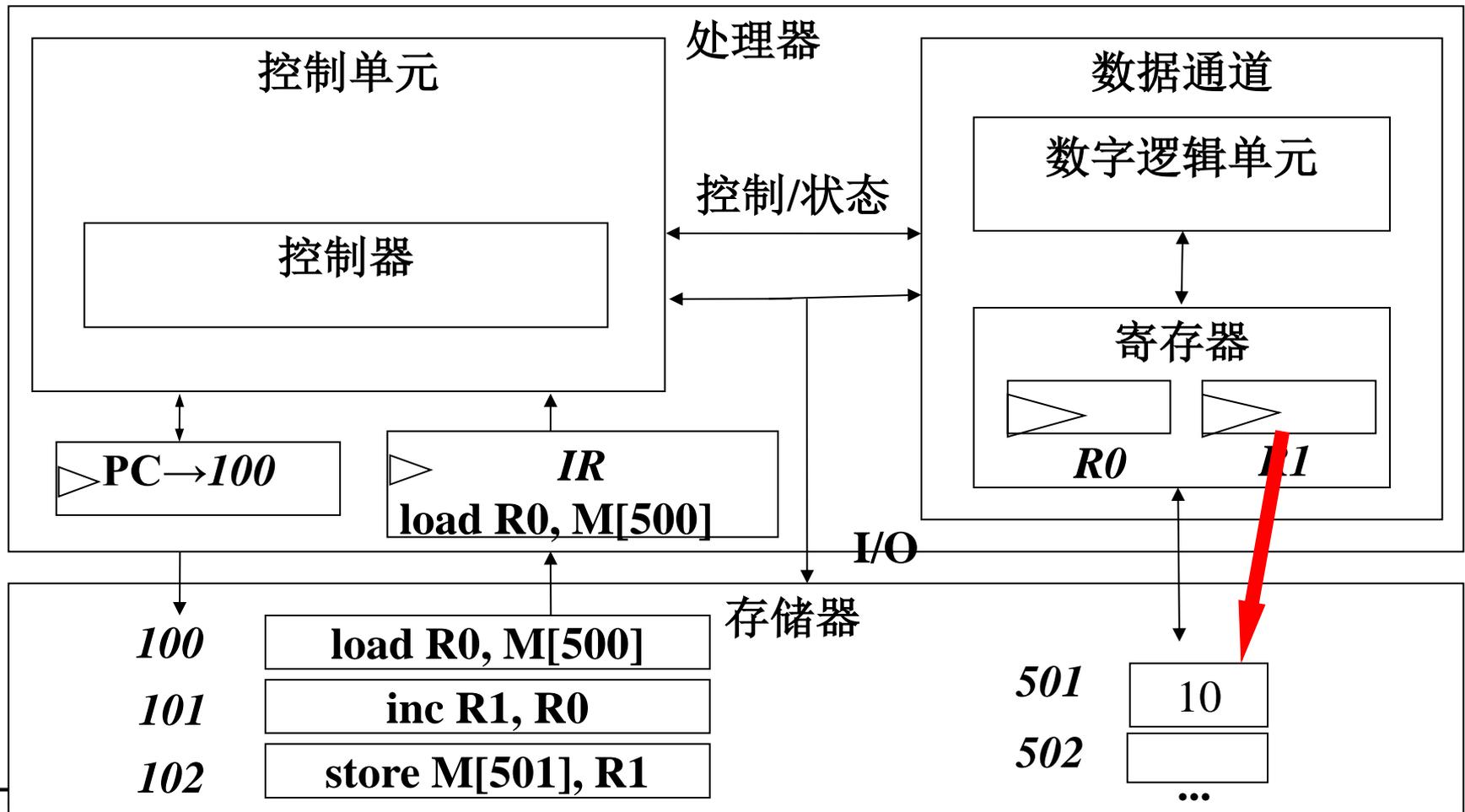
# (3) 执行

- 从存储器向数据通道寄存器移动数据



# (4) 存储

- 将指令执行结构保存到存储器中

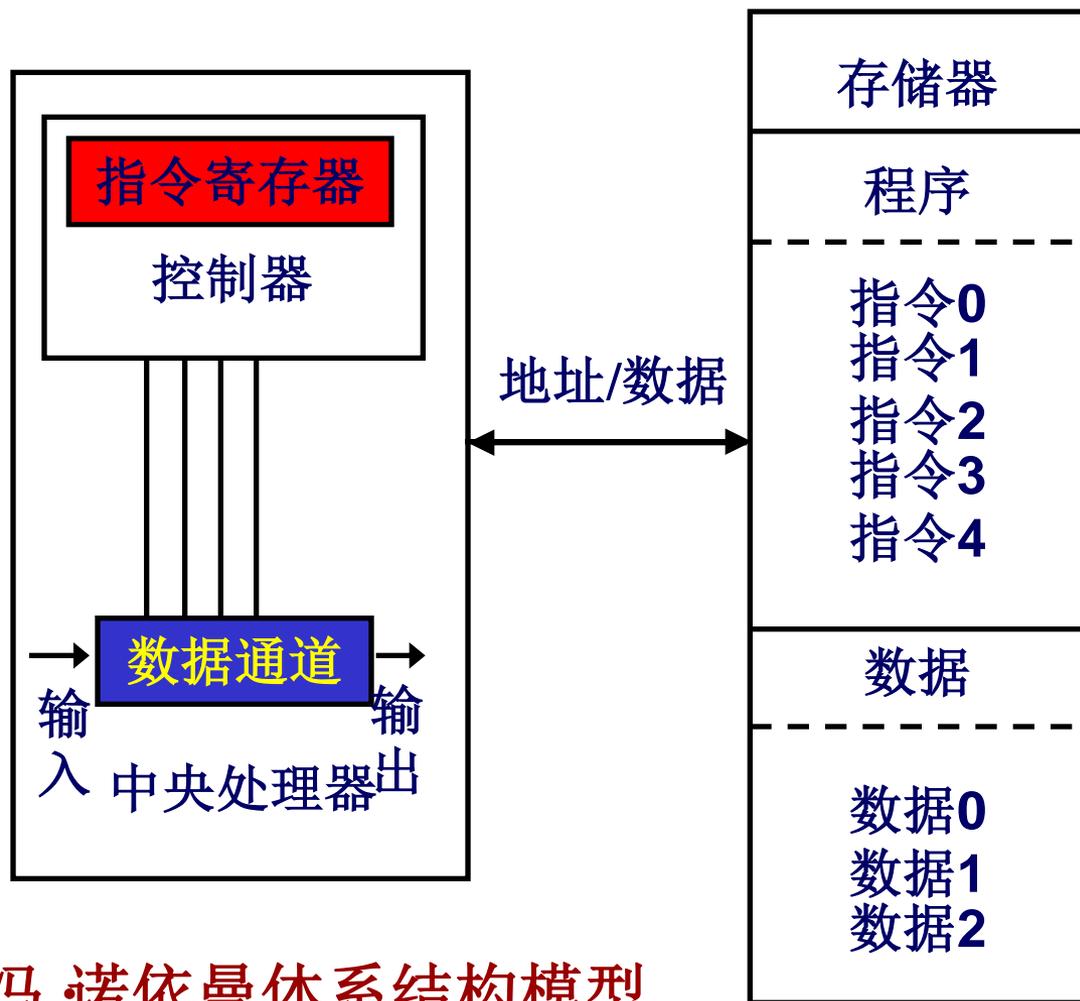


# 微处理器的结构体系

## (1) 按存储结构分：冯·诺依曼体系结构和哈佛体系结构

冯·诺伊曼结构也称普林斯顿结构，是一种将程序指令存储器和数据存储器合并在一起的存储器结构。处理器，经由同一个总线传输来访问程序和数据存储器，程序指令和数据的宽度相同。

如C51、X86系列、ARM7等

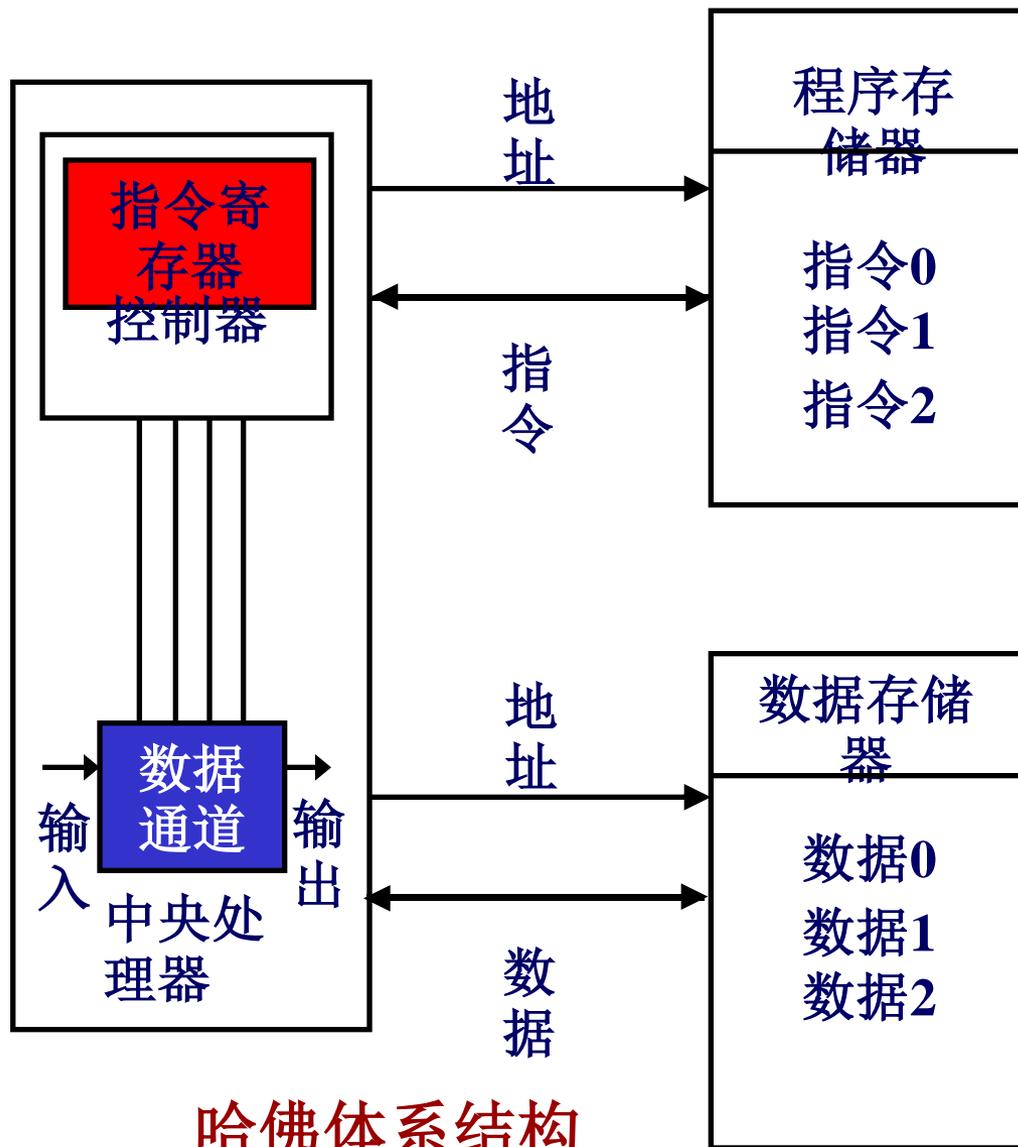


冯·诺依曼体系结构模型

# 微处理器的结构体系

哈佛结构是一种将程序指令存储和数据存储分开的存储器结构，目的是为了减轻程序运行时的访存瓶颈。哈佛结构的微处理器通常具有较高的执行效率。

如ARM9、TI的DSP等



哈佛体系结构

# 微处理器的结构体系

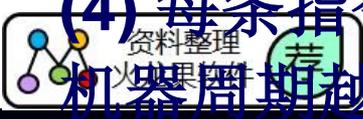
按指令类型可分为：复杂指令集（CISC）处理器和精简指令集（RISC）处理器

**CISC：复杂指令集（Complex Instruction Set Computer）**

- 具有大量的指令和寻址方式，那么就需要更多的解释器。
- 8/2原则：80%的程序只使用20%的指令
- 大多数程序只使用少量的指令就能够运行。

**CISC具有如下显著特点：**

- (1) 指令格式不固定，指令长度不一致，操作数可多可少；
- (2) 寻址方式复杂多样，以利于程序的编写；
- (3) 采用微程序结构，执行每条指令均需完成一个微指令序列；
- (4) 每条指令需要若干个机器周期才能完成，指令越复杂，花费的机器周期越多。



## ● RISC: 精简指令集 (Reduced Instruction Set Computer)

- 指令数目少，在通道中只包含最有用的指令
- 执行时间短，确保数据通道快速执行每一条指令
- 使CPU硬件结构设计变得更为简单
- 每条指令都采用标准字长

## CISC与RISC的区别

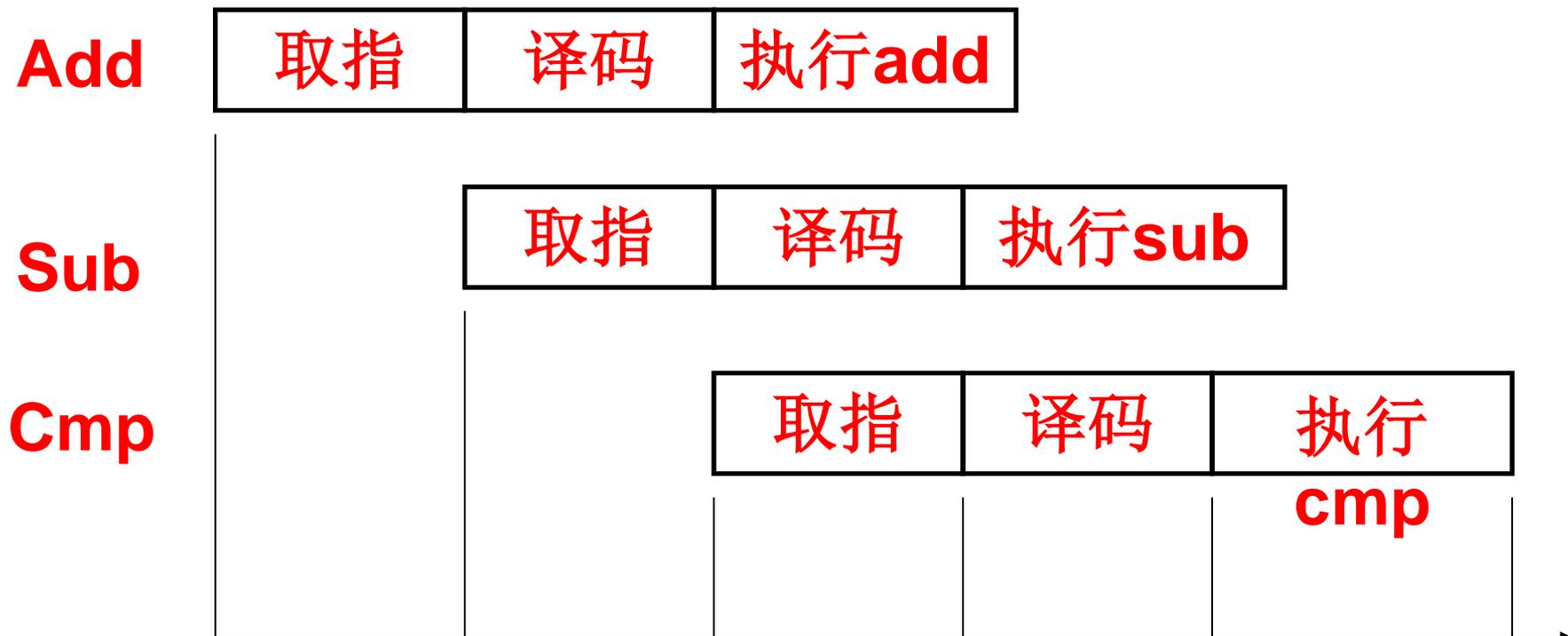
- 从硬件角度来看**CISC**处理的是不等长指令集，它必须对不等长指令进行分割，因此在执行单一指令的时候需要进行较多的处理工作。而**RISC**执行的是等长精简指令集，**CPU**在执行指令的时候速度较快且性能稳定。
- 从软件角度来看，大型操作系统较适合运行在支持**CISC**的处理器上。是我们所熟识的**DOS**、**Windows**操作系统。而实时操作系统大多运行在支持**RISC**的处理器上。

## 4 提高CPU性能的方法

影响CPU性能的因素：**流水线、超标量、缓存和总线**。对于任何处理器来说，要提高其效率，在设计上都是要减少数据的等待时间，并且努力减少处理单元的空闲时间。

# (1) 流水线技术

**流水线技术：**也就是将一个任务分解成为多个连续的子任务，在处理前一个子任务的同时就开始准备下一个子任务的数据并进行子处理器单元的初始化。

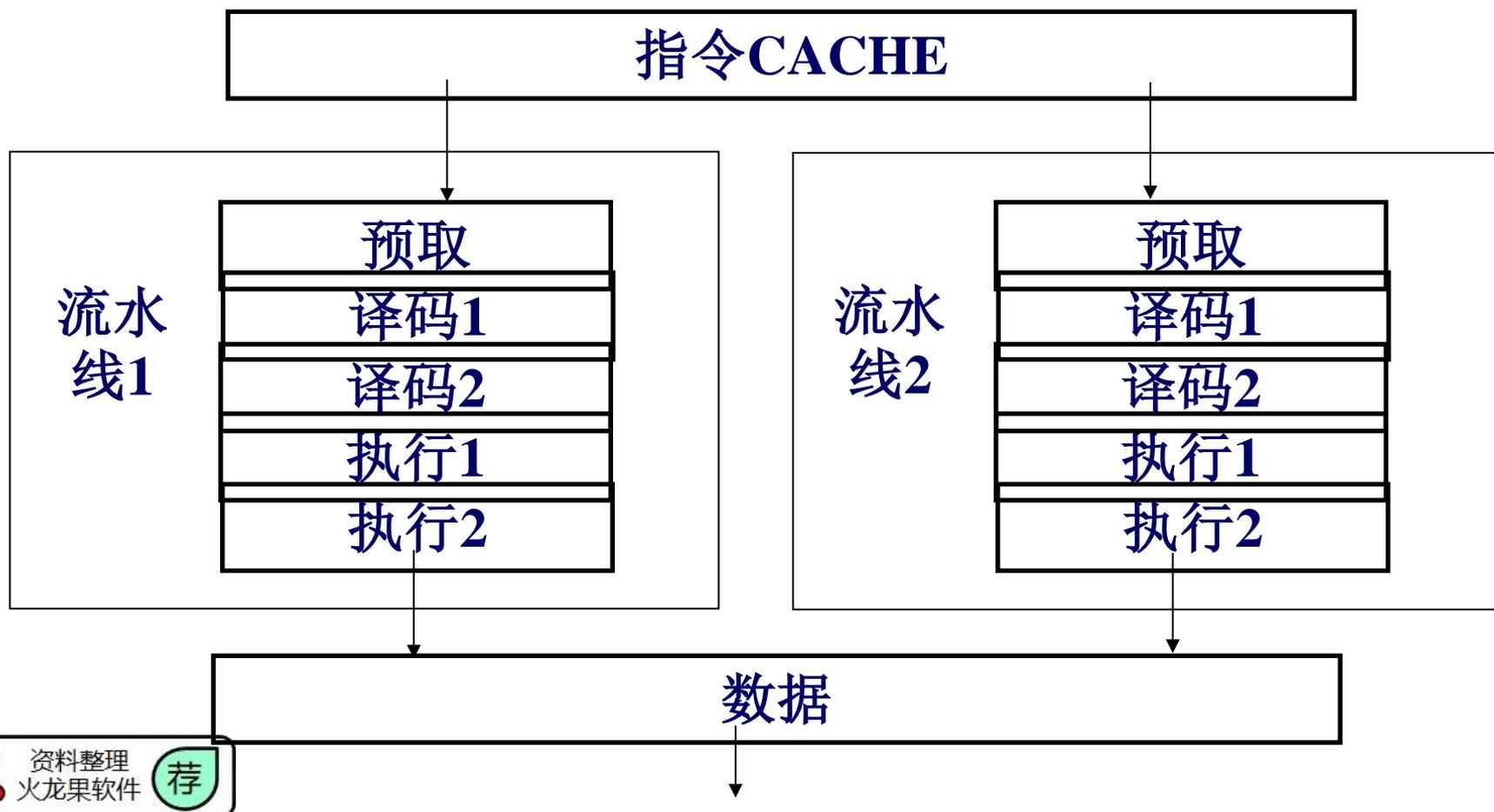


时间

- 在一些微处理器上，如**ARM**系列处理器、**DSP**等，指令实现流水线作业，指令过程按流水线的数目来进行划分。如**5级**流水线的处理器将指令分**5个**阶段执行。

## (2) 超标量执行

**超标量执行：**就是在处理器内部设置多个平行的处理单元，将多个相互无关的任务在这些处理部件中分别进行独立处理。



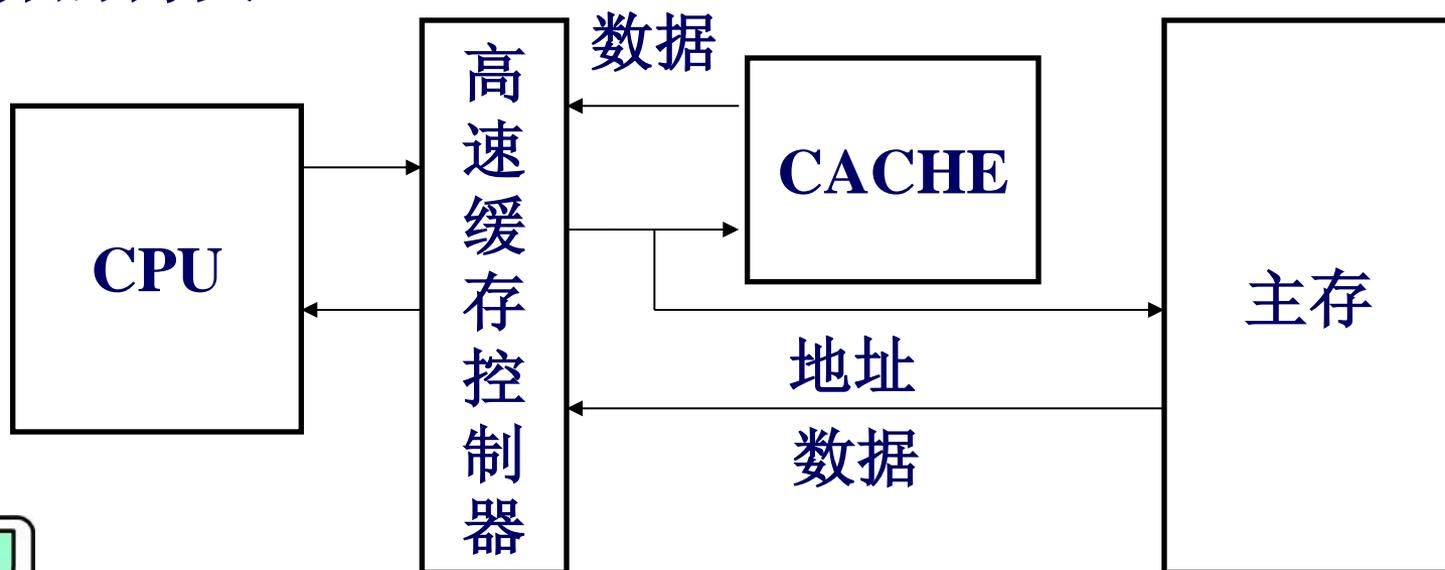
# (3) 高速缓存 (CACHE)

## 1、为什么采用高速缓存

微处理器的时钟频率比内存速度提高快得多，高速缓存可以提高内存的平均性能。

## 2、高速缓存的工作原理

高速缓存是一种小型、快速的存储器，它保存部分主存内容的拷贝。



- 通常用静态RAM来设计

- 因此，速度快但比较贵

- 通常和处理器同在一个芯片上

- 高速缓存的操作方式:

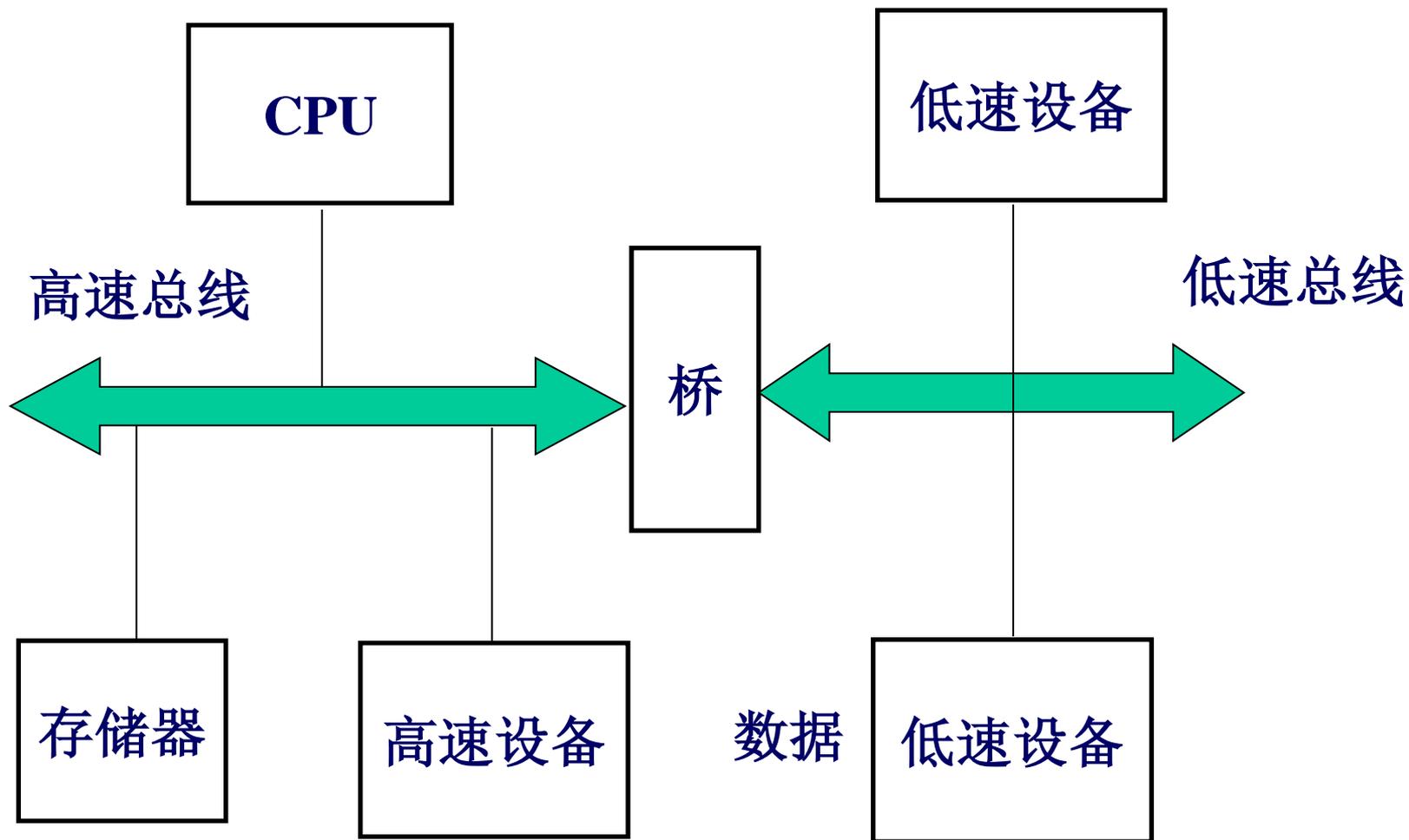
- 要求对主存储器进行访问 (读或写)

- 首先，检查高速缓存是否有相应的拷贝

- 如果有，称为高速缓存命中——拷贝在缓存中，可以快速访问

- 如果没有，称为高速缓存失误——拷贝没在缓存中，需要将该地址及其相邻的多个地址的数据读入高速缓存

## (4) 高速总线和总线桥



# 处理器信息存储的字节顺序

处理器信息存储的字节顺序主要分为大端存储法和小端存储法

## ● 大端模式

- ◆ 字数据的高位字节存储在低地址中
- ◆ 字数据的低字节则存放在高地址中

## ● 小端模式

- ◆ 低地址中存放字数据的低字节
- ◆ 高地址中存放字数据的高字节

# 处理器信息存储的字节顺序

- 以 `unsigned int value = 0x12345678` 为例，分别看看在两种字节序下其存储情况，我们可以用 `unsigned char buf[4]` 来表示 `value`:

**Big-Endian:** 低地址存放高位，如下:

- 高地址

```
-----  
buf[3] (0x78) -- 低位  
buf[2] (0x56)  
buf[1] (0x34)  
buf[0] (0x12) -- 高位  
-----
```

低地址

- **Little-Endian:** 低地址存放低位，如下:

- 高地址

```
-----  
buf[3] (0x12) -- 高位  
buf[2] (0x34)  
buf[1] (0x56)  
buf[0] (0x78) -- 低位  
-----
```

低地址

# 存储器系统

## 1. 存储器的分类

### ● 按存储介质分类

- 半导体存储器、磁表面存储器、光表面存储器

### ● 按存储器的读写功能分类

- 只读存储器（ROM）、随机存储器(RAM)

### ● 按在微机系统中的作用分类

- 主存储器、辅助存储器、高速缓冲存储器

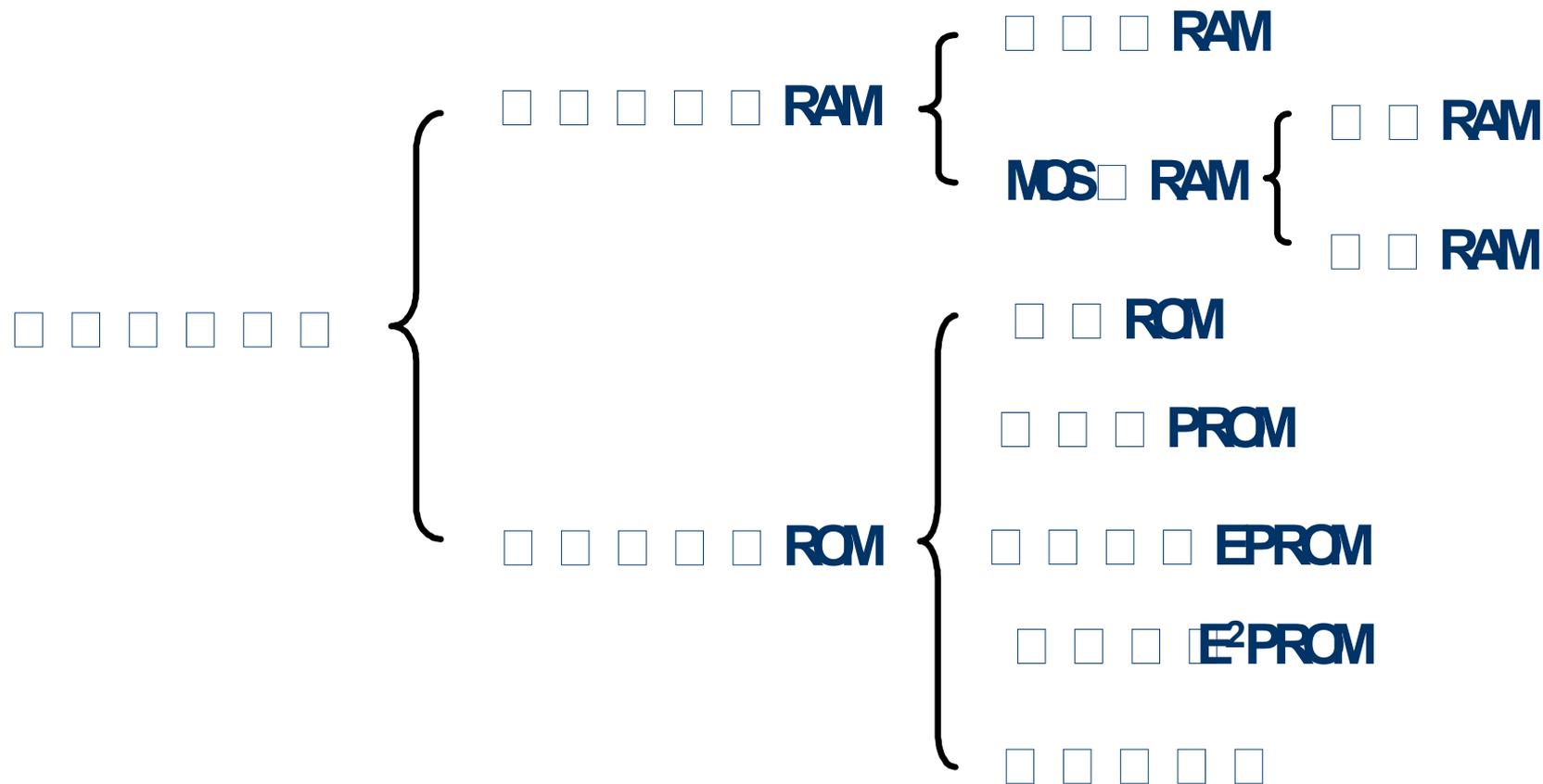


为了解决**CPU**与主存储器速度差所采取的措施有：

- (1) **CPU**内部设置多个通用寄存器
- (2) 采用多存储模块交叉存取
- (3) 采用高速缓冲存储器（**Cache**）

# 半导体存储器

半导体存储器主要包括随机存储器RAM和只读存储器两类ROM



**RAM:** 随机存取存储器, **SRAM:** 静态随机存储器,  
**DRAM:** 动态随机存储器

- 1) **SRAM**比**DRAM**快
- 2) **SRAM**比**DRAM**耗电多
- 3) **DRAM**存储密度比**SRAM**高得多
- 4) **DRAM**需要周期性刷新

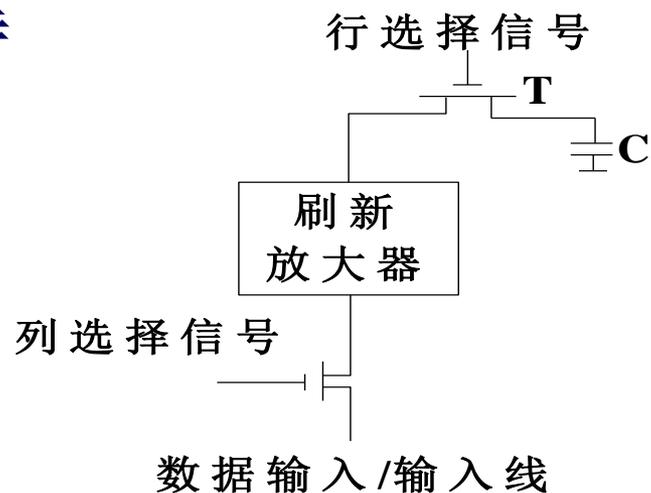
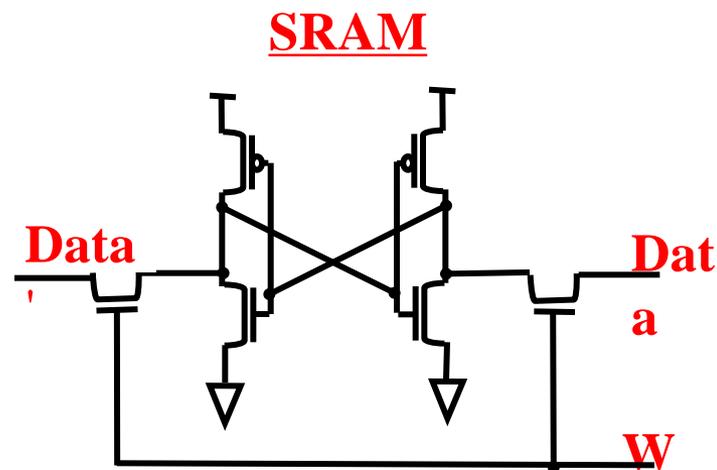
**ROM:** 只读存储器

**FLASH:** 闪存

# (1) 随机存储器RAM

- 静态RAM——SRAM: Static RAM
  - 存储单元用触发器来存储数据位
  - 要求6个晶体管
  - 需要电源保持数据
- 动态RAM——DRAM: Dynamic RAM
  - 存储单元用MOS晶体管来存储数据位比SRAM更紧凑
  - 由于电容的泄漏，需要更新来保持数据
  - 典型的刷新频率是15.625 us.
  - 比SRAM访问更慢

存储器内部结构

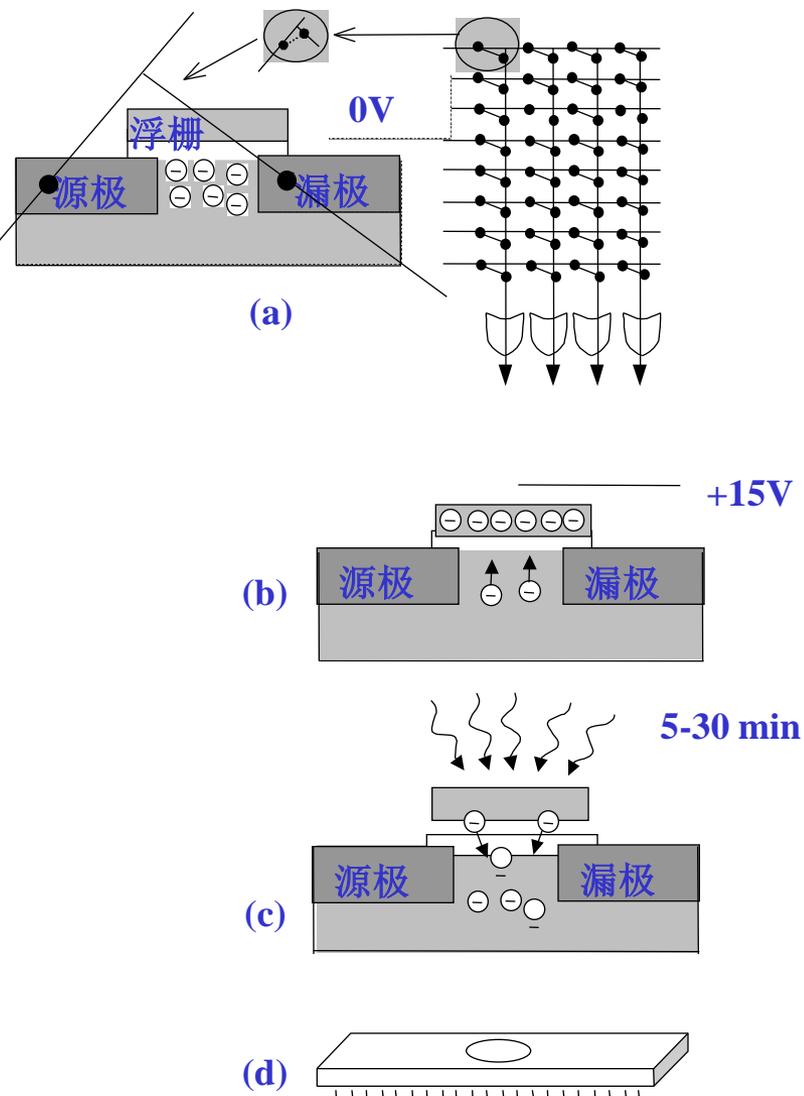


## (2) 只读存储器ROM

**EPROM: 可擦除的可编程ROM**

可编程部分是一个**MOS**晶体管

- 晶体管有一个绝缘体包围的“浮栅”
- (a) 负电荷在源极和漏极之间形成一个隧道
- (b) 较大的正电压在栅极使负电荷移出隧道进入栅极形成逻辑0
- (c) 擦除——紫外线在栅极表面的照射使负电荷从栅极回到隧道保持逻辑1
- (d) EPROM有一个**紫外线**可以通过的石英窗



# EEPROM: 电擦除的可编程ROM

- 电可编程和擦除(**Programmed and erased electronically**)
  - 使用电压比正常的高
  - 能单个字进行擦除和编程
- 较好的写入能力(**Better write ability**)
  - 通过内部电路提供较高电压能在系统内编程
  - 由于写入需经过擦除和编程两个步骤, 因此写入较慢
  - 可重复擦除和编程数万次
- 存储永久性和**EPROM**相近(大约10年)
- 比**EPROM**方便得多, 但更贵

# 快闪存储器 (Flash Memory)

## ●EEPROM的扩展

- 同样利用浮栅原理
- 具有和EEPROM的写入能力和保存永久性

## ●擦除更快

- 每次可以擦除存储器一块区域，但EEPROM每次只能擦除一个字
- 每个块通常几千个字节

## ●进行单字写入时可能更慢

- 必须先进行整块读取后，对需修改的字修改更新后再整块写入

## ●用在需要大量存储数据在非易失性存储器的嵌入式系统

- 如数码相机，机顶盒，移动电话等

两种主要的Flash Memory技术是NOR和NAND结构



## 2.2.3 处理器输入输出接口

输入输出接口又称I/O接口，它是主机与外围设备之间交互信息的连接口，它在主机和外围设备之间的信息交换中起着桥梁和纽带作用。

### 1. I/O接口与CPU交换的信息类型

输入输出通道与CPU交换的信息类型有三种：

- (1) 数据信息：反映生产现场的参数及状态的信息，它包括数字量、开关量和模拟量。
- (2) 状态信息：又叫做应答信息、握手信息，它反映过程通道的状态，如准备就绪信号等。
- (3) 控制信息：用来控制过程通道的启动和停止等信息，如三态门的打开和关闭、触发器的启动等。

## 2. I/O的编址方式

由于计算机系统一般都有多个过程输入输出通道，因此需对每一个输入输出通道安排地址。I/O口编址方式有两种：

### (1) I/O与存储器统一编址方式

这种编址方式又称存储器映像方式，它从存贮器空间划出一部分地址空间给过程通道，把过程通道的端口当作存贮单元一样进行访问，对I/O端口进行输入输出操作跟对存储单元进行读写操作方式相同，只是地址不同。

### (2) I/O与存储器独立编址方式

这种编址方式将过程通道的端口地址单独编址，有自己独立的过程通道地址空间，而不占用存储器地址空间。

# 例如USB

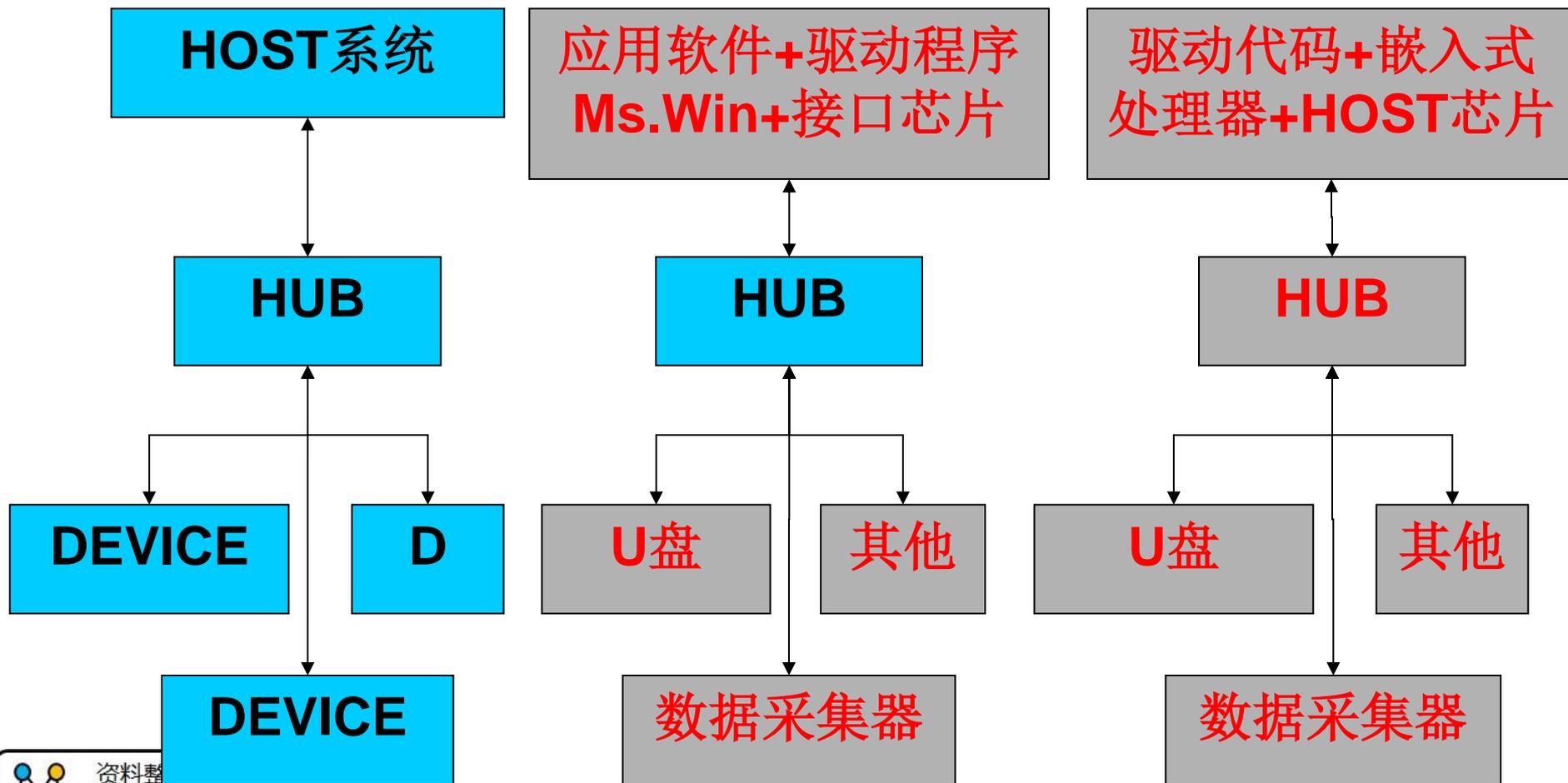
- **USB: Universal Serial Bus**, 通用串行总线
- 大家生活中常见的与**USB**有关的东西有:
  - ◆ **U盘、移动硬盘、无驱型的MP3 (U盘)**
  - ◆ **USB接口的键盘、Mouse、打印机、数码相机.....**
- 即插即用, 热插拔, 系统不需重启便可工作, 且易于扩展 (127个)
- **USB2.0**以低成本实现高达**480Mb/s**的传输率 ( **USB1.1**的全速设备可达**12Mb/s**)
- 接口标准统一、端口供电

# 一个典型的USB通讯系统

## 通用系统模型

## PC机中的情况

## 嵌入式系统应用



# 嵌入式处理器分类

嵌入式处理器是嵌入式系统的核心，是控制、辅助系统运行的硬件单元。范围极其广阔，从最初的4位处理器，目前仍在大规模应用的8位单片机，到最新的受到广泛青睐的32位，64位嵌入式CPU。嵌入式处理器主要包括以下几类。

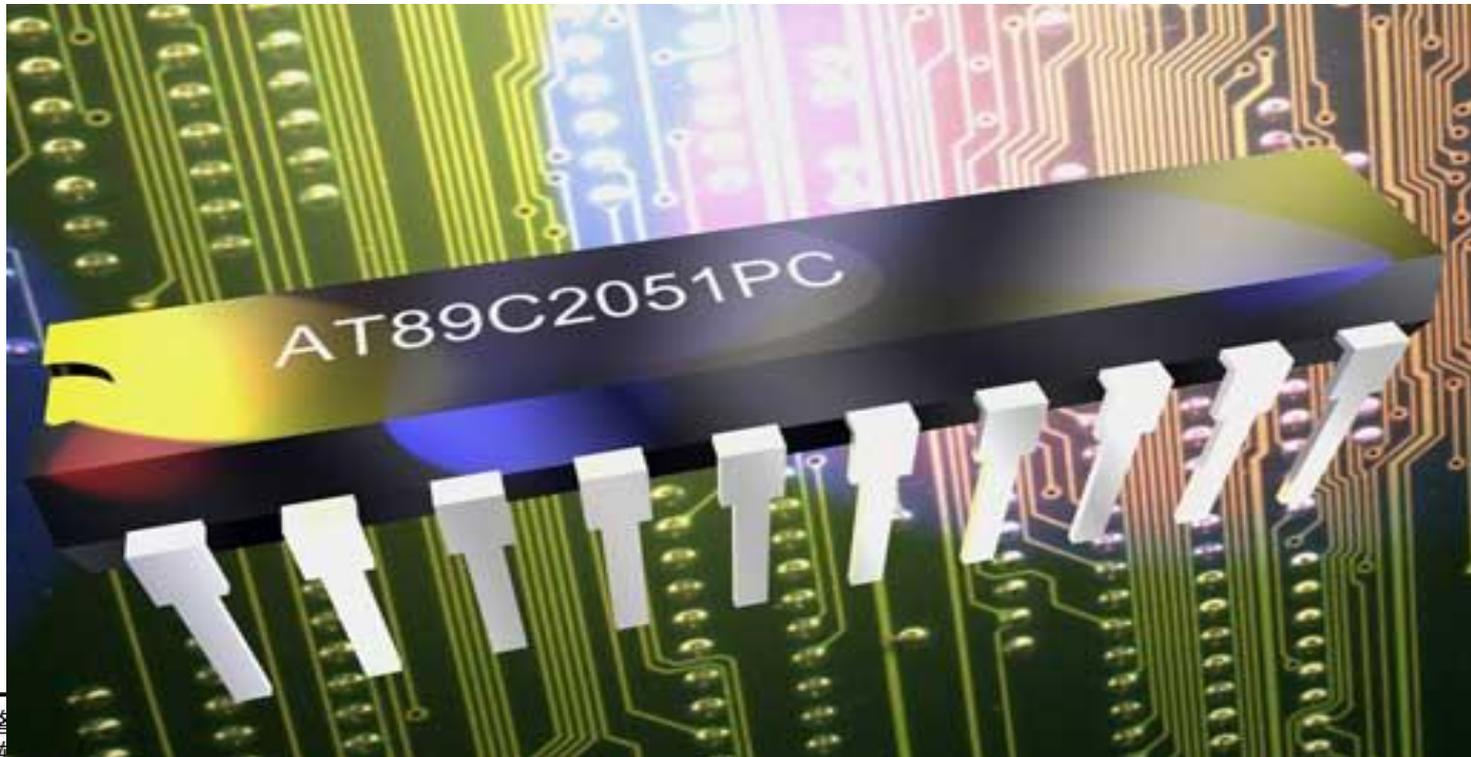
- (1) 嵌入式微控制器
- (2) 嵌入式DSP处理器
- (3) 嵌入式微处理器
- (4) 嵌入式片上系统(System On Chip)

# (1) 嵌入式微控制器

- 嵌入式微控制器的典型代表是单片机，这种 8 位的电子器件目前在嵌入式设备中仍然有着极其广泛的应用。
- 单片机芯片内部集成ROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制(PWM)输出、A/D、D/A、Flash ROM、EEPROM等各种必要功能和外设。

# 嵌入式微控制器

- 微控制器的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。
- 微控制器是目前嵌入式系统工业的主流。微控制器的片上外设资源一般比较丰富，适合于控制，因此称为微控制器。



## (2) 嵌入式DSP处理器

- **DSP**处理器是专门用于**信号处理**方面的处理器，其在系统结构和指令算法方面进行了特殊设计，在数字滤波、**FFT**、频谱分析等各种仪器上**DSP**获得了大规模的应用。
- **DSP**的理论算法在**70**年代就已经出现，但是由于专门的**DSP**处理器还未出现，所以这种理论算法只能通过**MPU**等由分立元件实现。**1982**年世界上诞生了首枚**DSP**芯片。在语音合成和编码解码器中得到了广泛应用。**DSP**的运算速度进一步提高，应用领域也从上述范围扩大到了通信和计算机方面。
- 目前最为广泛应用的嵌入式**DSP**处理器是**TI**的**TMS320C2000/C5000**系列，另外如**Intel**的**MCS-296**和**Siemens**的**TriCore**也有各自的应用范围。

### (3) 嵌入式微处理器 (Micro Processor Unit)

- MPU嵌入式微处理器是由通用计算机中的CPU演变而来的。
- 与计算机处理器不同的是，在实际嵌入式应用中，只保留和嵌入式应用紧密相关的功能硬件，去除其他的冗余功能部分，这样就以最低的功耗和资源实现嵌入式应用的特殊要求。

# 嵌入式微处理器

- 和工业控制计算机相比，嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点。目前主要的嵌入式处理器类型有**Am186/88**、**386EX**、**SC-400**、**Power PC**、**68000**、**MIPS**、**ARM/StrongARM**系列等



## (4) 嵌入式片上系统(System On Chip)

- **SoC(System on Chip)**, 就是把整个电子系统功能全部集成到一块集成电路里面,SoC嵌入式系统微处理器就是一种电路系统。
- 它结合了许多功能区块, 将功能做在一个芯片上, 像是**ARM RISC**、**MIPS RISC**、**DSP**或是其他的微处理器核心, 加上通信的接口单元, 像是通用串行端口(**USB**)、**TCP/IP**通信单元、**GPRS**通信接口、**GSM**通信接口、**IEEE1394**、蓝牙模块接口等等, 这些单元以往都是依照各单元的功能做成一个个独立的处理芯片。

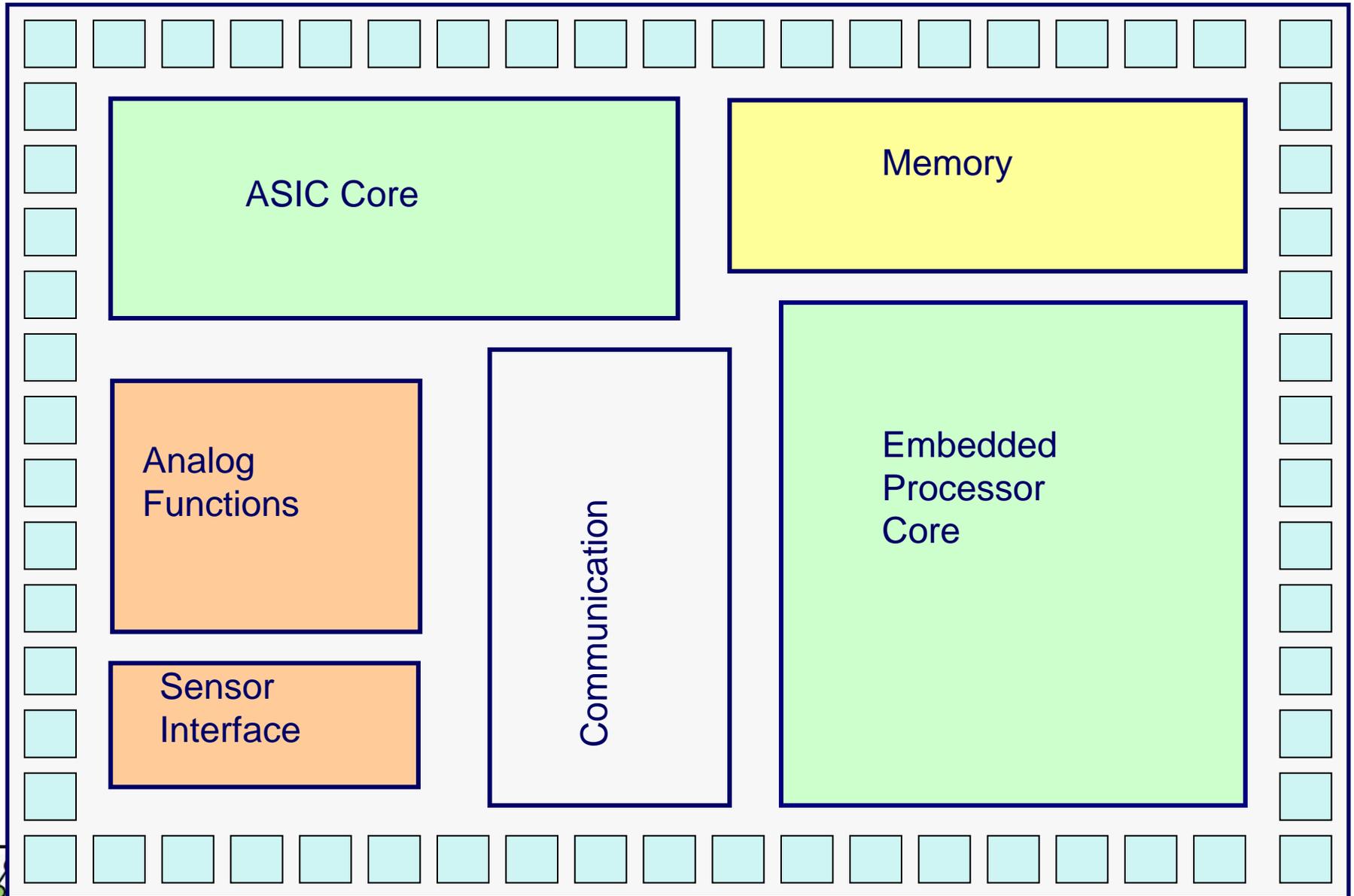
- **SoC**是追求产品系统最大包容的集成器件，**SoC**最大的特点是成功实现了软硬件无缝结合，直接在处理器片内嵌入操作系统的代码模块。
- 运用**VHDL**等硬件描述语言不需要再像传统的系统设计一样，绘制庞大复杂的电路板，一点点的连接焊制，只需要使用**精确的语言**，**综合时序设计**直接在器件库中调用各种通用处理器的标准，然后通过仿真之后就可以直接交付芯片厂商进行生产。
- 也可以采用**FPGA**来实现**SoC**功能

- 由于**SoC**往往是专用的，所以大部分都不为用户所知，如 **Philips**的**Smart XA**。 **Siemens**的**TriCore**， **Motorola**的**M-Core**， 某些**ARM**系列器件， **Echelon**和**Motorola**联合研制的**Neuron (神经元)**芯片等。
- **SoC**芯片也将在声音、图像、影视、网络及系统逻辑等应用领域中发挥重要作用。

## SoC嵌入式系统微处理器所具有的其他的的好处可以分为下列几点：

- 利用改变内部工作电压，降低芯片功耗。
- 减少芯片对外管脚数，简化制造过程。
- 减少外围驱动接口单元及电路板之间的信号传递，可以加快微处理器数据处理的速度。
- 内嵌的线路可以避免外部电路板在信号传递时所造成系统杂讯。

# 新的发展方向：SoC



# 多核处理器CMP (Chip multiprocessors)

多核处理器主要具有以下几个显著的优点:

- **控制逻辑简单:** 相对超标量微处理器结构, 单芯片多处理器结构的控制逻辑复杂性要明显低很多。
- **高主频:** 由于单芯片多处理器结构的控制逻辑相对简单, 包含极少的全局信号, 因此线延迟对其影响比较小, 因此, 在同等工艺条件下, 单芯片多处理器的硬件实现要获得比超标量微处理器和超长指令字微处理器更高的工作频率。
- **低通信延迟:** 由于多个处理器集成在一块芯片上, 且采用共享**Cache**或者内存的方式, 多线程的通信延迟会明显降低, 这样也对存储系统提出了更高的要求。
- **低功耗:** 通过动态调节电压/频率、负载优化分布等, 可有效降低**CMP (Chip multiprocessors)** 功耗。
- **设计和验证周期短:** 微处理器厂商一般采用现有的成熟单核处理器作为处理器核心, 从而可缩短设计和验证周期, 节省研发成本。

- 多核处理器主要包括两类：同构多核处理器和异构多核处理器。
  - 同构多核处理器是集成多个相同的处理器核在一个芯片上，这种处理器能很好地实现一个任务在不同处理器核上的并行执行。
  - 异构多核处理器是集成不同构架的处理器到一块芯片上，用于满足不同应用的需要，可以实现多个任务在不同处理器核上的并行处理。
- 在嵌入式系统中，异构多核处理器比同构多核处理器应用更为广泛。如TI的**OMAP**系列、达芬奇系列，**IBM**的**Cell**系列处理器等。

# 3 ARM处理器概述

●ARM——Advanced RISC Machines

●ARM——32位RISC结构IP核提供商

# ARM的发展历程-1

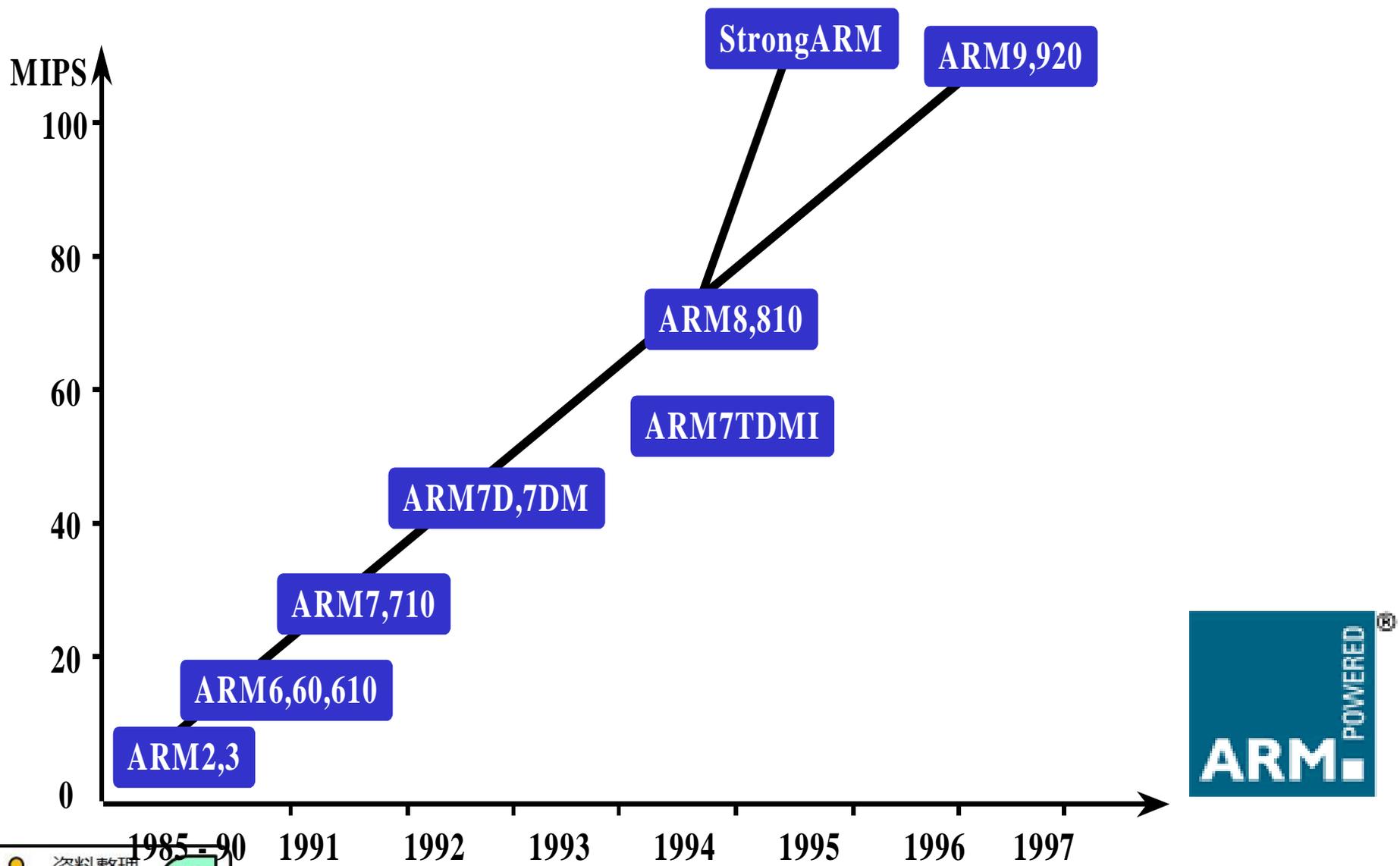
- ARM即Advanced RISC Machines的缩写。
- 1985年4月26日，第一个ARM原型在英国剑桥的Acorn计算机有限公司诞生。
- 20世纪80年代后期，ARM很快开发成Acorn的台式机产品，形成英国的计算机教育基础。
- 1990年成立了Advanced RISC Machines Limited。
- 20世纪90年代，ARM32位嵌入式RISC（Reduced Instruction Set Computer）处理器扩展到世界范围，占据了低功耗、低成本和高性能的嵌入式系统应用领域的领先地位。

- **32位RISC处理器受到青睐，领先的是ARM嵌入式微处理器系列。**
- **ARM公司虽然只成立20多年，但在1999年因移动电话火爆市场，其32位RISC处理器占市场份额超过了50%，2001年初，ARM公司的32位RISC处理器市场占有率超过了75%。ARM公司是知识产权供应商，是设计公司。由合作伙伴公司来生产各具特色的芯片。**

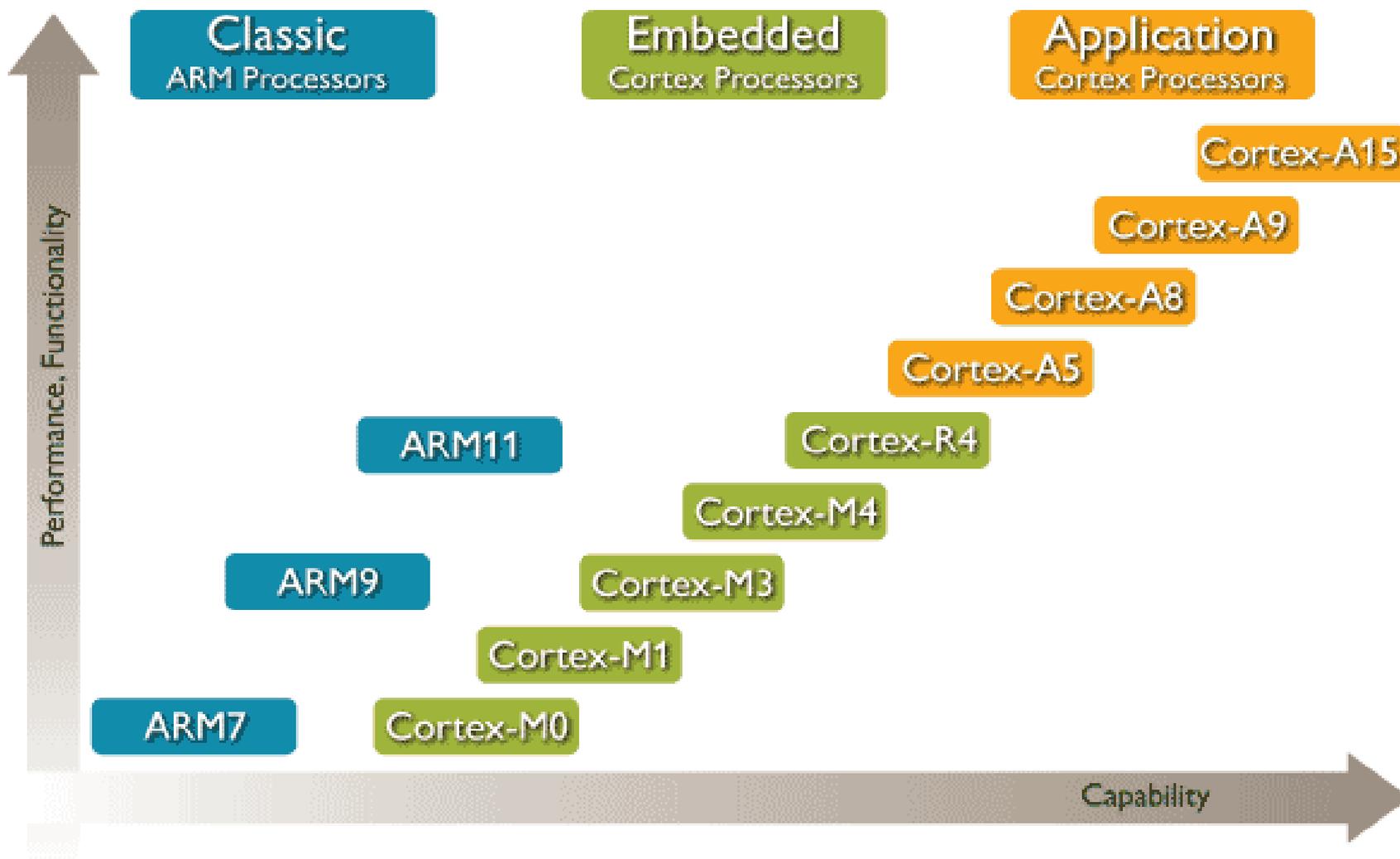
## ARM的发展历程-2

- **1991 - ARM 推出第一款RISC嵌入式微处理器核 ARM6**
- **1993 - ARM 推出 ARM7 核**
- **1995 – ARM的Thumb扩展指令集结构为16位系统增加了32位的性能,提供业界领先的代码密度**

# ARM的发展历程-3



# ARM的发展历程-4



# ARM处理器的特点

## ARM处理器的3大特点如下：

- 小体积、低功耗、成本低、高性能；
- 16位/32位双指令集；
- 全球众多的合作伙伴。

## ARM处理器系列提供的解决方案包括：

- 在无线、消费电子和图像应用方面的开放平台；
- 存储、自动化、工业和网络应用的嵌入式实时系统
- 智能卡和SIM卡的安全应用。

# ARM体系结构版本 - 1

- **Version 1 (obsolete)**

- 基本数据处理
- 字节,字以及多字 load/store
- 软件中断
- 26 bit 地址总线

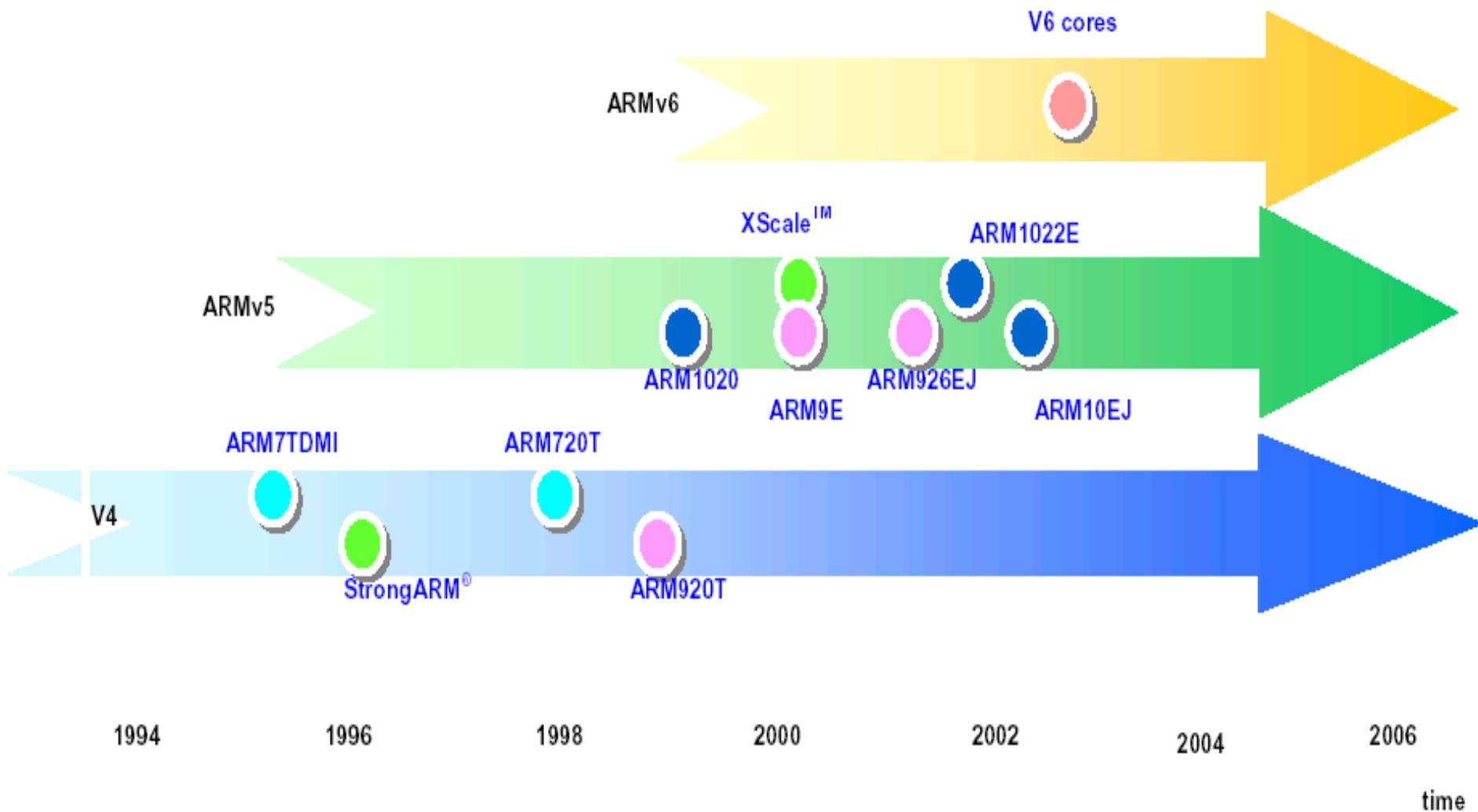
- **Version 2 (obsolete)**

- Multiply & Multiply-accumulate
- 支持协处理器
- 支持线程同步
- 26 bit 地址总线

# ARM体系结构版本 - 2

- **V3版本推出32位寻址能力,结构扩展变化为**
  - ◆ **T—16位压缩指令集**
  - ◆ **M—增强型乘法器,产生全64位结果  
( $32 \times 32 \Rightarrow 64$  or  $32 \times 32 + 64 \Rightarrow 64$ )**
- **V4版本增加了半字load和store指令**
- **V5版本改进了ARM和Thumb之间的交互,结构扩展变化为:**
  - ◆ **E--增强型DSP指令集,包括全部算法操作和16位乘法操作**
  - ◆ **J--支持新的JAVA,提供字节代码执行的硬件和优化软件加速功能**

# ARM 体系结构更新



# ARM微处理器的应用领域

- 工业控制领域
- 无线通讯领域
- 网络应用
- 消费电子产品
- 成像和安全产品

在ARM处理器内核中有多个功能模块可供生产厂商根据不同用户的要求来配置生产。这些模块分别用**T**、**D**、**M**、**I**、**E**、**J**、**S**等来表示，这些模块一般从处理器的内核版本上可以区分出来。

**T**: 表示支持**Thumb**指令，说明该内核可从**16位Thumb**指令集扩充到**32位ARM**指令集。

**D**: 表示支持**Debug**，说明该内核中放置了用于调试的结构，通常它为一个边界扫描链**JTAG**，可使**CPU**进入调试模式，从而方便地进行断点设置、单步调试。

**M**: 表示**Multiplier**，说明处理器内部带有**8位**乘法器。

**I**: 表示**Embedded ICE Logic**，用于实现断点观测及变量观测的逻辑电路部分，其中的**TAP**控制器可接入到边界扫描链。

除了以上一些特性外，**ARM**处理器内核中还有一些处理器内核带**EJ-S**模块。

**E**: 表示**DSP Enhancement**，即增加了前导零处理和饱和运算等一些常用的**DSP**运算指令，极大地改善音、视频处理程序的性能。

**J**: 表示**Jazelle DBX (Direct Bytecode eXecution)**，这是**ARM**公司推出的**Java**加速解决方案。**Jazelle**不是一个简单的加速硬件，它是融入于处理器流水线之中的一项专门针对**Java**指令执行的硬件功能，使得**CPU**可以直接接收一部分**Java**指令，并加以译码执行。

## 4.1.2 ARM内核种类分类

### 1.ARM7系列

- ARM7处理器采用了ARMV4T（冯·诺依曼）体系结构
- 分为三级流水,空间统一的指令与数据Cache,平均功耗为0.6mW/MHz,时钟速度为66MHz,每条指令平均执行1.9个时钟周期。其中的ARM710,ARM720和ARM740为内带Cache的ARM核。
- ARM7体系结构是小型、快速、低能耗、集成式的RISC内核结构。
- 该产品的典型用途是数字蜂窝电话和硬盘驱动器
- 主流的ARM7内核是ARM7TDMI、ARM7TDMI-S、ARM7EJ-S、ARM720T
- 常用的有Samsung公司的S3C44BOX与S3C4510处理器、Atmel公司的AT91FR40162系列处理器、Cirrus公司的EP73xx系列等

## 2.ARM9系列

- **ARM9处理器采用ARMV4T（哈佛）体系结构**
- **ARM9采用五级流水处理及分离的Cache结构，平均功耗为0.7mW/MHz。时钟速度为120MHz~200MHz，每条指令平均执行1.5个时钟周期**
- **与ARM7处理器系列相似，其中的ARM920、ARM940和ARM9E处理器均为含有Cache的CPU核，性能为132MIPS（120MHz时钟，3.3V供电）或220MIPS（200MHz时钟）**
- **常用于无线设备、仪器仪表、联网设备、机顶盒设备、高端打印机及数码相机应用中**
- **主流的ARM9内核是ARM920T、ARM922T、ARM940**
- **Samsung公司的S3C2510、S3C2410，Cirrus公司的EP93xx系列等**

## 3.ARM10E系列

- ARM10E处理器采用ARMVST体系结构
- 可以分为六级流水处理，采用指令与数据分离的Cache结构，平均功耗1000mW，时钟速度为300MHz，每条指令平均执行1.2个时钟周期
- 其系统时钟高达300MHz时钟，指令Cache和数据Cache分别为32KB，数据宽度为64位
- 能够支持多种商用操作系统，适用于下一代高性能手持式因特网设备及数字式消费类应用
- 主流的ARM10内核是ARM1020E、ARM1022E、ARM1026EJ-S等

## 4. SecurCore处理器

- **SecurCore**系列处理器提供了基于高性能的**32位RISC**技术的安全解决方案，该系列处理器具有体积小、功耗低、代码密度大和性能高等特点。
- 采用软内核技术，以提供最大限度的灵活性，以及防止外部对其进行扫描探测，提供面向智能卡的和低成本的存储保护单元**MPU**，可以灵活地集成用户自己的安全特性和其他的协处理器
- 目前含有**SC100、SC110、SC200、SC210** 4种产品。

## 5.Intel的StrongARM

- 采用ARMV4T的五级流水体系结构。
- 目前有SA110、SA1100、SA1110等3个版本。

另外Intel公司的基于ARMv5TE体系结构的XScale PXA27x系列处理器，与StrongARM相比增加了I/D Cache，并且加入了部分DSP功能，更适合于移动多媒体应用。

目前市场上的大部分智能手机的核心处理器就是XScale系列处理器。

## 6. ARM11系列

- **ARM11处理器系列可以在使用130nm代工厂技术、小至2.2mm<sup>2</sup>芯片面积和低至0.24mW/MHz的前提下达到高达500MHz的性能表现。**
- **ARM11处理器系列以众多消费产品市场为目标**
- **主要的ARM11处理器有ARM1136JF-S、ARM1156T2F-S、ARM1176JZF-S、ARM11 MCORE等多种。**

## 7. ARM Cortex 系列

- **A R M 3**种系列。
- **Cortex-M3**的定位是：向专业嵌入式市场提供低成本、低功耗的芯片。在成本和功耗方面，**Cortex-M3**具有相当好的性能，特别适用于汽车和无线通信领域。迄今为止，已经有多家芯片制造商开始生产基于**Cortex-M3**内核的微控制器。
- **ARM Cortex-M3**是一种基于**ARM V7**架构的最新**ARM**嵌入式内核，它采用哈佛结构，使用分离的指令和数据总线（冯诺伊曼结构下，数据和指令共用一条总线）。

## 7. ARM Cortex 系列

- **Cortex-M3**只支持最新的 **Thumb-2**指令集。
- **Thumb-2**指令集的设计是专门面向**C**语言的。
- **Thumb-2**指令集允许用户在**C**代码层面维护和修改应用程序，**C**代码部分非常易于重用。
- 如果在低成本的情况下寻求更好的性能和改进功耗，您最好的选择是选用**Cortex-M3**

● 这里对各类处理器核的**ARM**流水线做一下对比

ARM7	预取 (Fetch)	译码 (Decode)	执行 (Exec)
------	---------------	----------------	--------------

ARM9	预取 (Fetch)	译码 (Decode)	执行 (Exec)	访问 (Memory)	写回 (Write)
------	---------------	----------------	--------------	----------------	---------------

ARM10	预取 (Fetch)	发射 (Issue)	译码 (Decode)	执行 (Exec)	访问 (Memory)	写回 (Write)
-------	---------------	---------------	----------------	--------------	----------------	---------------

ARM11	预取 (Fetch)	预取 (Fetch)	发射 (Issue)	译码 (Decode)	转换 (Snnny)	执行 (Exec)	访问 (Memory)	写回 (Write)
-------	---------------	---------------	---------------	----------------	---------------	--------------	----------------	---------------

## 表4-1 ARM处理器分类

处理器内核分类	具体的处理器 IP 核	应用市场
Embedded Core	ARM7TDMI、ARM946E-S、 ARM926EJ-S	无线、网络应用、 汽车电子
Application Core	ARM926EJ-S、ARM1026EJ-S、ARM11	消费类市场、多媒 体数码产品
Secure Core	SC110、SC110、SC200、SC210	智能卡、身份识别