



北京师范大学  
BEIJING NORMAL UNIVERSITY

信息科学与技术学院

# 第一章 基础知识



# 第1章 基础知识

1.1 机器语言

1.2 汇编语言的产生

1.3 汇编语言的组成

1.4 存储器

1.5 指令和数据

1.6 存储单元

1.7 CPU对存储器的读写

1.8 地址总线

1.9 数据总线

1.10 控制总线

1.11 内存地址空间（概述）

1.12 主板

1.13 接口卡

1.14 各类存储器芯片

1.15 内存地址空间



# 引言

- 汇编语言是直接**在硬件之上**工作的编程语言。
  - 先了解硬件系统的结构，才能有效的应用汇编语言对其编程。
- 本书的原则是，以后用到的知识，以后再说。
  - 当课程进行到需要补充新的基础知识（关于编程结构或其他的）时候，再对相关的基础知识进行介绍和探讨。



# 引言

- 汇编课程的研究重点：

如何利用硬件系统的编程结构和指令集有效灵活的控制系统进行工作。



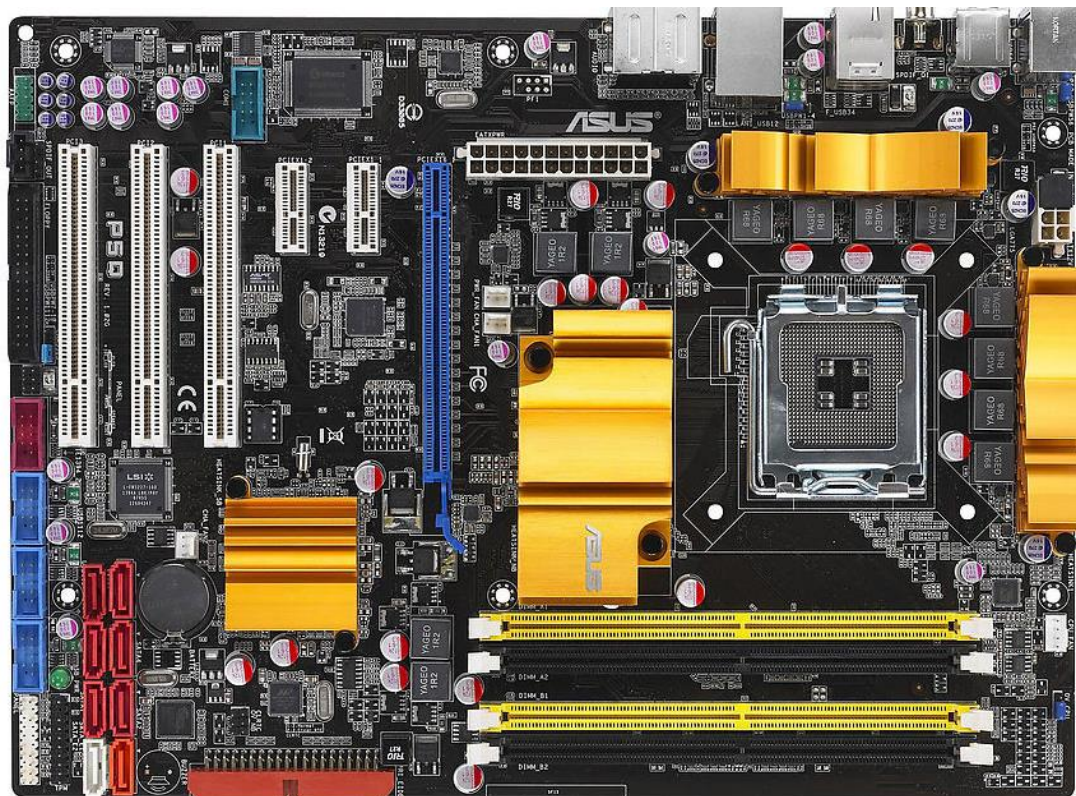
# 1.1 机器语言

- 机器语言是机器指令的集合。
- 机器指令：一台机器可以正确执行的命令。



# 1.1 机器语言

- 计算机: 指由CPU 和其他受CPU 直接或间接控制的芯片、器件、设备组成的计算机系统;
  - 如PC 机。

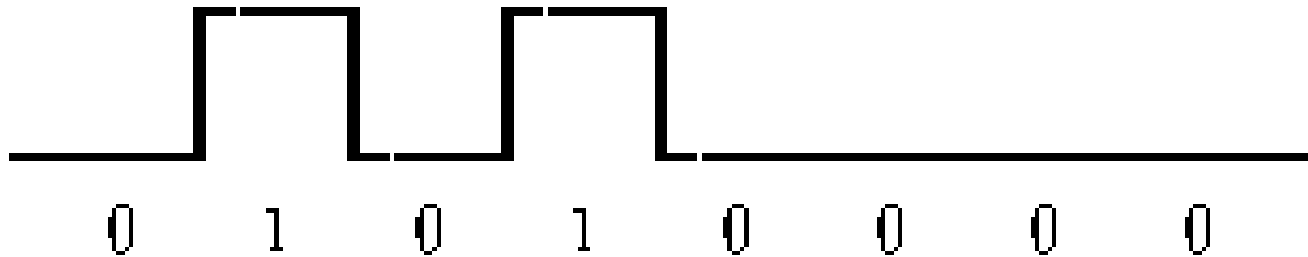




# 1.1 机器语言

- 指令：01010000 (**PUSH AX**)

- 电平脉冲：





# 1.1 机器语言

- 程序员们将 0、1 数字编程的程序代码打在纸带或卡片上，1 打孔，0 不打孔，再将程序通过纸带机或卡片机输入计算机，进行运算。







# 1.1 机器语言

## ■ 示例

应用8086CPU完成运算：

$$S = 768 + 12288 - 1280$$



# 1.1 机器语言

## ■ 示例

应用8086CPU完成运算：

$$S = 768 + 12288 - 1280$$

## ■ 机器码：

```
10110000000000000000000011
0000010100000000000110000
00101101000000000000000101
```



# 1.1 机器语言

- 示例

应用8086CPU完成运算：

$$S = 768 + 12288 - 1280$$

- 机器码：

```
10110000000000000000000011
0000010100000000000110000
0010110100000000000000101
```

- 假如将程序错写成以下这样，请找出错误：

```
10110000000000000000000011
0000010100000000000110000
0001011010000000000000101
```



# 1.1 机器语言

- 在显示器上输出 “welcome to masm”。

- 若某个 “1”误写为 “0”，如何找错？

```
00011110
101110000000000000000000
01010000
101110001100011000001111
1000111011011000
10110100000000110
1011000000000000
1011011100000111
101110010000000000000000
1011011000011000
1011001001001111
1100110100010000
1011010000000010
1011011100000000
1011011000000000
1011001000000000
1100110100010000
1011010000001001
10001101000101100010101000000000
1100110100100001
1011010000001010
10001101000101100011000100000000
1100110100100001
10110100000000110
1011000000010100
1011011100011001
1011010100001011
1011000100010011
1011011000001101
1011001000111100
1100110100010000
1011010000000010
1011011100000000
1011000000001100
1011001000010100
1100110100010000
1011010000001001
10001101000101100000000000000000
1100110100100001
11001011
```



## 1.2 汇编语言的产生

- 汇编语言的主体是**汇编指令**。
- **汇编指令**和机器指令的差别在于指令的表示方法上:
  - 汇编指令是机器指令的助记符(便于记忆的书写格式)。



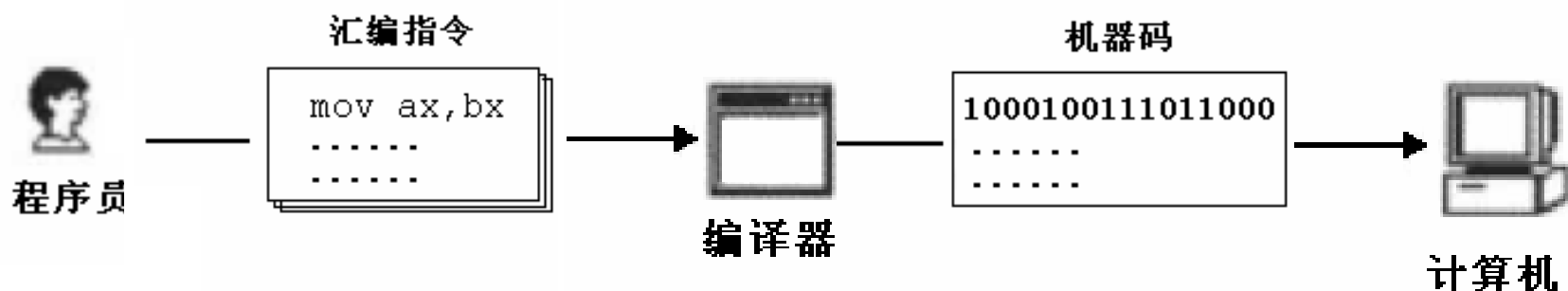
## 1.2 汇编语言的产生

- 机器指令：1000100111011000
- 操作：寄存器BX的内容送到AX中
  - 寄存器：简单的讲是CPU中可以存储数据的器件。
  - 一个CPU中有多个寄存器。
  - AX和BX分别是两个不同寄存器的代号。
  - 更详细的内容我们在以后的课程中将会讲到。
- 汇编指令：MOV AX, BX
  - 接近人类语言，便于阅读和记忆。



## 1.2 汇编语言的产生

- 计算机能读懂的只有机器指令，那么如何让计算机执行程序员用汇编指令编写的程序呢？





## 1.3 汇编语言的组成

■ 汇编语言由以下3类组成：

1. 汇编指令（机器码的助记符）

➤ 这是汇编语言的核心，它决定了汇编语言的特性。

2. 伪指令（由编译器执行）

3. 其它符号（由编译器识别）



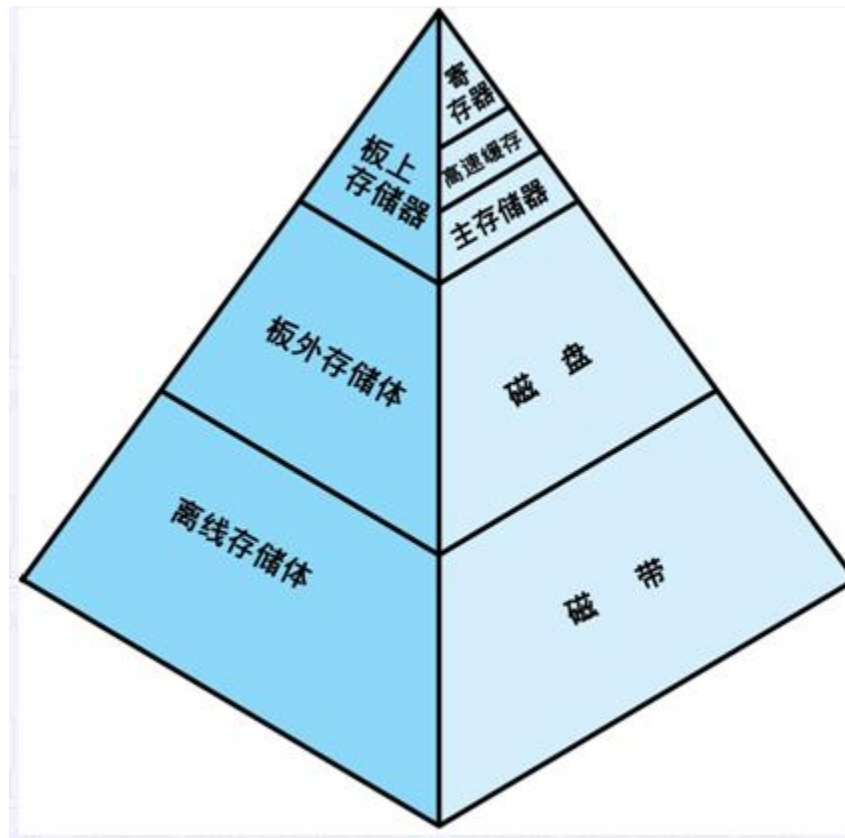


## 1.4 存储器

- CPU 是计算机的核心部件。它控制整个计算机的运作并进行运算。
- 如何控制CPU的工作？向它提供指令和数据。
- 指令和数据在存储器（即内存）中存放。
- 磁盘不同于内存，磁盘上的数据或程序如果不读到内存中，就无法被CPU使用。



# 存储器的层次结构





## 1.5 指令和数据

- 指令和数据是应用上的概念。
  - 在内存或磁盘上，指令和数据没有任何区别，都是二进制信息。



## 1.5 指令和数据

- 指令和数据是应用上的概念。
  - 在内存或磁盘上，指令和数据没有任何区别，都是二进制信息。
  - 二进制信息：1000100111011000
  - 数据：89D8H
  - 程序：MOV AX, BX



- 例如：  
一个存储器有128个存储单元，  
编号从0~127。

**124**  
**125**  
**126**  
**127**



## 1.6 存储单元

- 对于大容量的存储器一般还用以下单位来计量容量（以下用B来代表Byte）：
  - $1\text{KB}=1024\text{B}$
  - $1\text{MB}=1024\text{KB}$
  - $1\text{GB}=1024\text{MB}$
  - $1\text{TB}=1024\text{GB}$
- 磁盘的容量单位同上。



## 1.7 CPU对存储器的读写

- CPU要想进行数据的读写，必须和外部器件（标准的说法是芯片）进行三类信息的交互：
  - 地址信息：存储单元的地址
  - 控制信息：器件的选择，读或写命令
  - 数据信息：读或写的的数据



## 1.7 CPU对存储器的读写

- 那么CPU是通过什么将地址、数据和控制信息传到存储芯片中的呢？
- 专门连接CPU和其他芯片的导线——总线。
  - 物理上：一根根导线的集合；
  - 逻辑上：
    - 地址总线
    - 数据总线
    - 控制总线

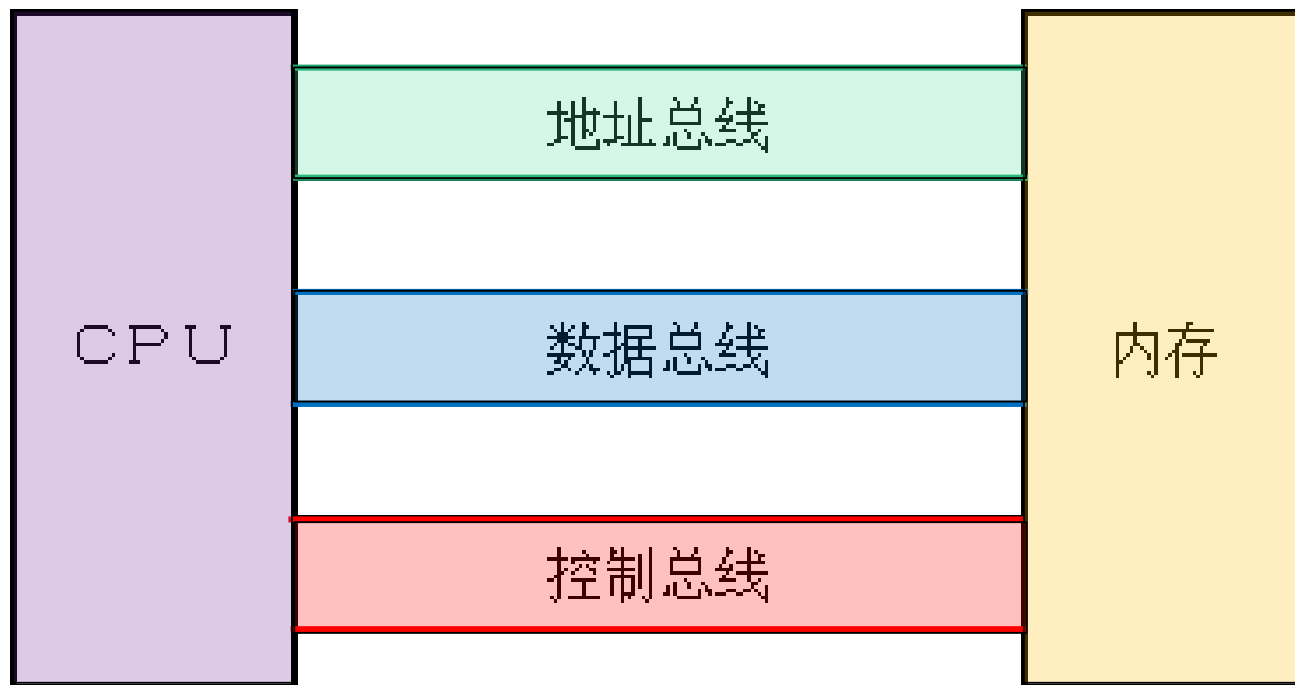




# 1.7 CPU对存储器的读写

## ■ 总线在逻辑上划分

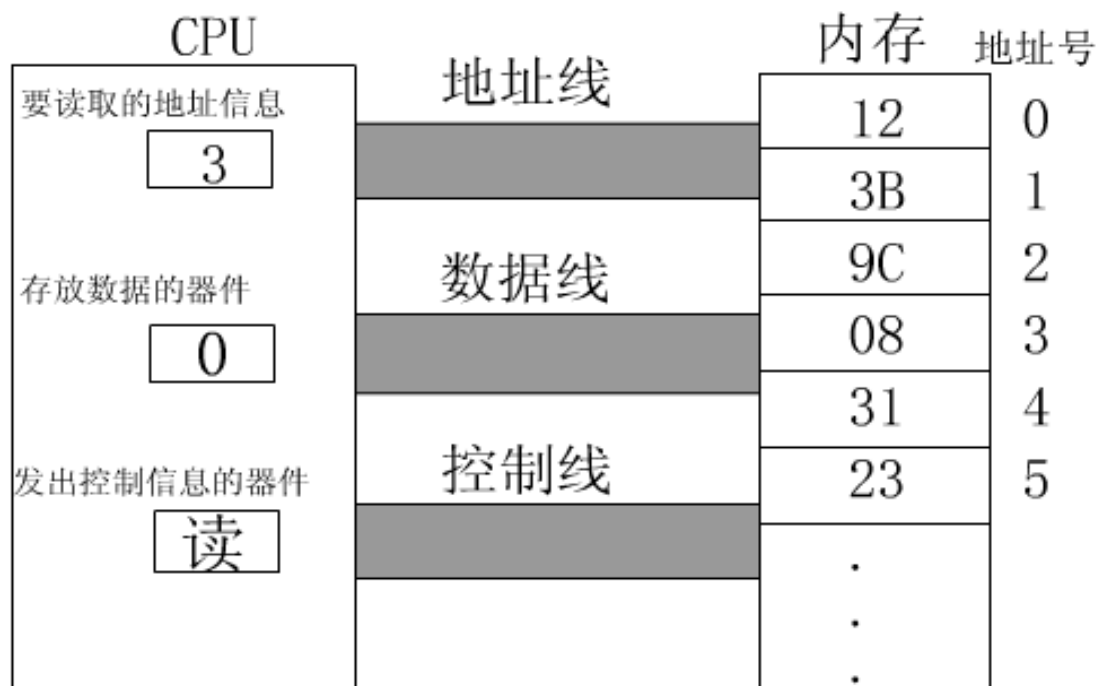
图示：





# 1.7 CPU对存储器的读写

## ■ CPU在内存中**读**数据：



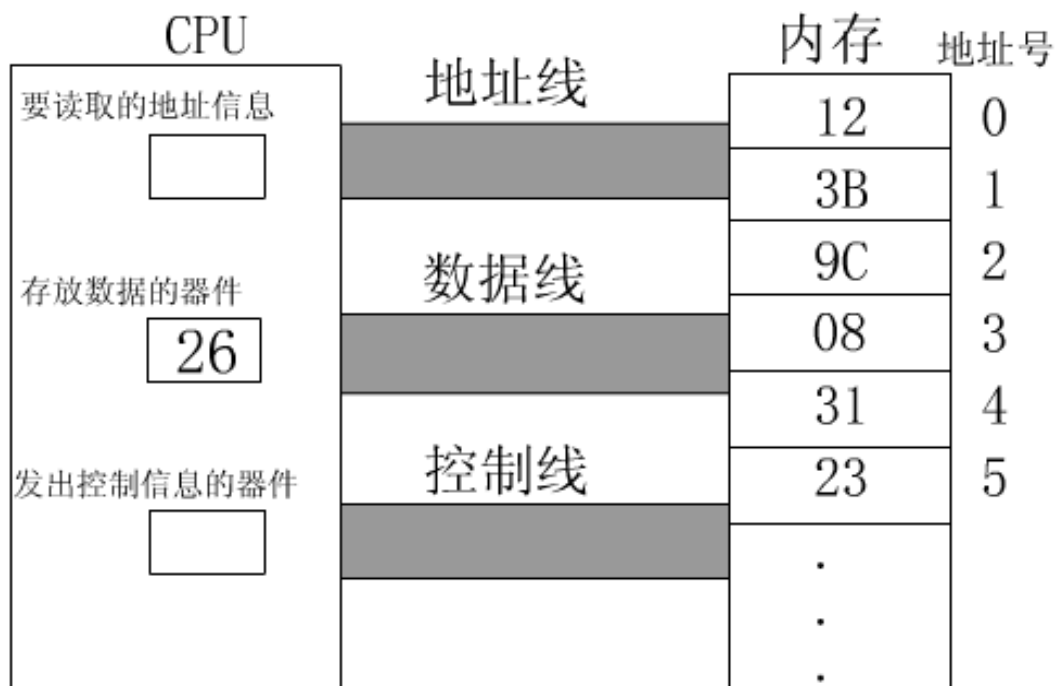
CPU从内存中3号单元处读取数据的过程

▶ play ◀ step ▶ step ◻ stop



# 1.7 CPU对存储器的读写

## ■ CPU在内存中**写**数据：



CPU向内存中3号单元写入数据26的过程



## 1.7 CPU对存储器的读写

- 如何命令CPU进行数据的读写呢？
- 对于8086CPU
  - 机器码：1010000000000001100000000
  - 含 义：从3号单元读取数据送入寄存器AX



# 1.7 CPU对存储器的读写

- 如何命令CPU进行数据的读写呢？
- 对于8086CPU
  - 机器码：1010000000000001100000000
  - 含 义：从3号单元读取数据送入寄存器AX
- 机器码难于记忆，用汇编指令表示
  - MOV AX, [3]



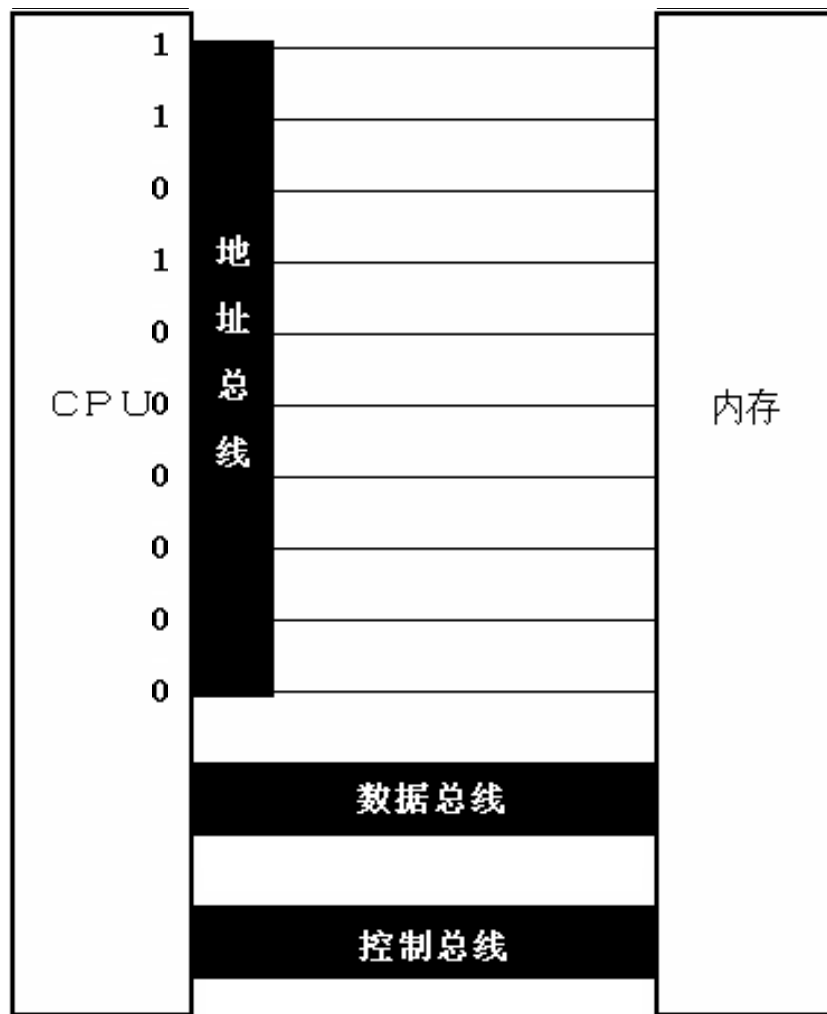
## 1.8 地址总线

- CPU是通过地址总线来指定存储单元的。
- 地址总线上能传送多少个不同的信息，CPU就可以对多少个存储单元进行寻址。



## 1.8 地址总线

### ■ 地址总线发送地址信息





## 1.8 地址总线

- 一个CPU有 $N$ 根地址总线，即地址总线宽度为 $N$ 。
- 地址总线宽度为 $N$ 的CPU最多可寻址内存空间？
  - $2^N$ 个内存单元。





## 1.9 数据总线

- CPU与内存或其它器件之间的数据传送是通过数据总线来进行的。
- 数据总线宽度关系到了CPU和外界的数据传送速度。

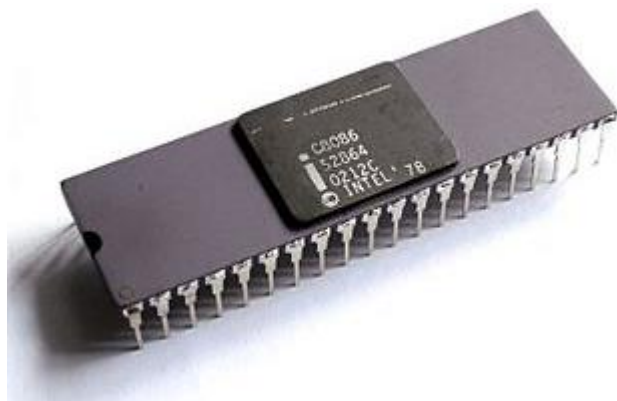


## 1.9 数据总线

- 要向内存中写入数据89D8H时，以下CPU是如何通过数据总线传送数据的：
  - 8088CPU（8位）数据总线上的数据传送情况

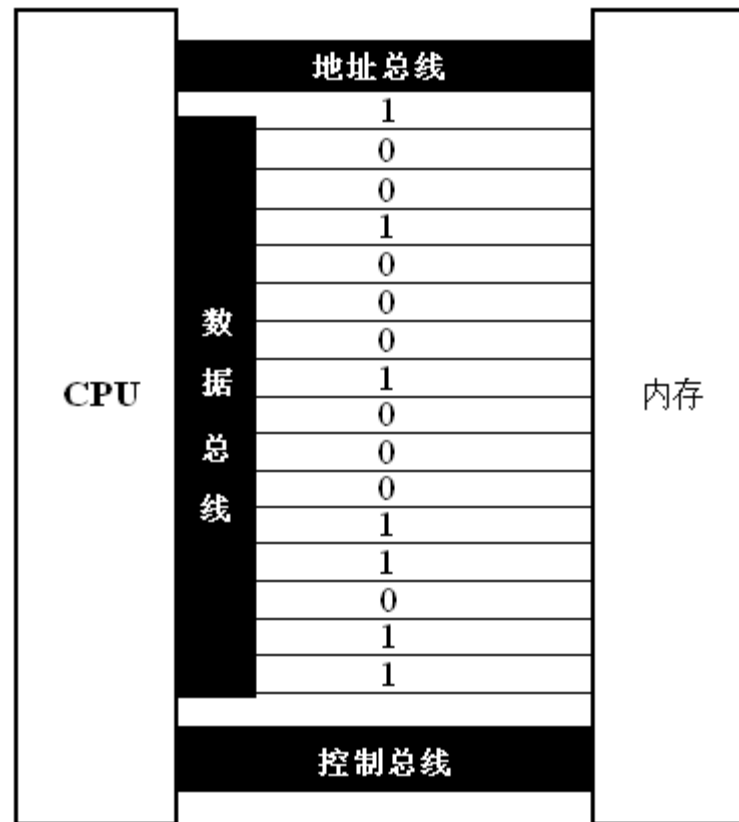


- 8086CPU（16位）数据总线上的数据传送情况





## 1.9 数据总线



8086CPU -16位数据总线上传送的信息



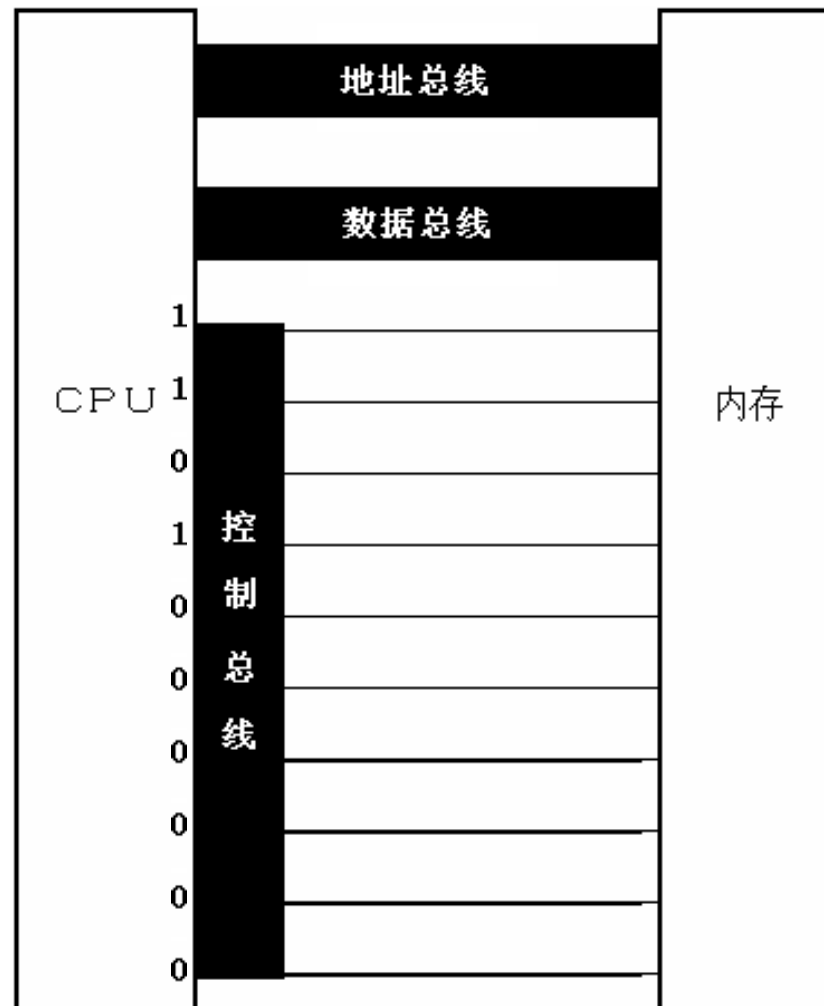
## 1.10 控制总线

- CPU对外部器件的控制是通过控制总线来进行的。
  - 控制总线是个总称，是一些不同控制线的集合。
- 有多少根控制总线，就意味着CPU提供了对外部器件的多少种控制。
  - 控制总线的宽度决定了CPU对外部器件的控制能力。



## 1.10 控制总线

### ■ 控制总线上发送的控制信息





## 1.10 控制总线

- 对内存的**读/写**命令是由几根控制线综合发出的：
  - 其中有一根名为**读信号输出控制线**负责由CPU向外传送读信号，CPU向该控制线上输出低电平表示将要读取数据；
  - 有一根名为**写信号输出控制线**负责由CPU向外传送写信号。



# 1.1节~1.10节 小结

- (1) 汇编指令是机器指令的助记符，同机器指令一一对应。
- (2) 每一种CPU都有自己的汇编指令集。
- (3) CPU可以直接使用的信息在存储器中存放。
- (4) 在存储器中指令和数据没有任何区别，都是二进制信息。
- (5) 存储单元从零开始顺序编号。
- (6) 一个存储单元可以存储 8 个 bits （用作单位写成“b”），即 8 位二进制数。
- (7)  $1\text{B} = 8\text{b}$                        $1\text{KB} = 1024\text{B}$   
 $1\text{MB} = 1024\text{KB}$        $1\text{GB} = 1024\text{MB}$



## 1.1节~1.10节 小结（续）

（8）每一个CPU芯片都有许多管脚，这些管脚和总线相连。也可以说，这些管脚引出总线。一个CPU可以引出三种总线的宽度标志了这个CPU的不同方面的性能：

- 地址总线的宽度决定了CPU的寻址能力；
  - 数据总线的宽度决定了CPU与其它器件进行数据传送时的一次数据传送量；
  - 控制总线宽度决定了CPU对系统中其它器件的控制能力。
- 在汇编课程中，仅从功能的角度介绍这三类总线，对实际的连接情况不做讨论。





# 特别提示

- 检测点 1.1 (Page 9)
- 没有通过检测点请不要向下学习！



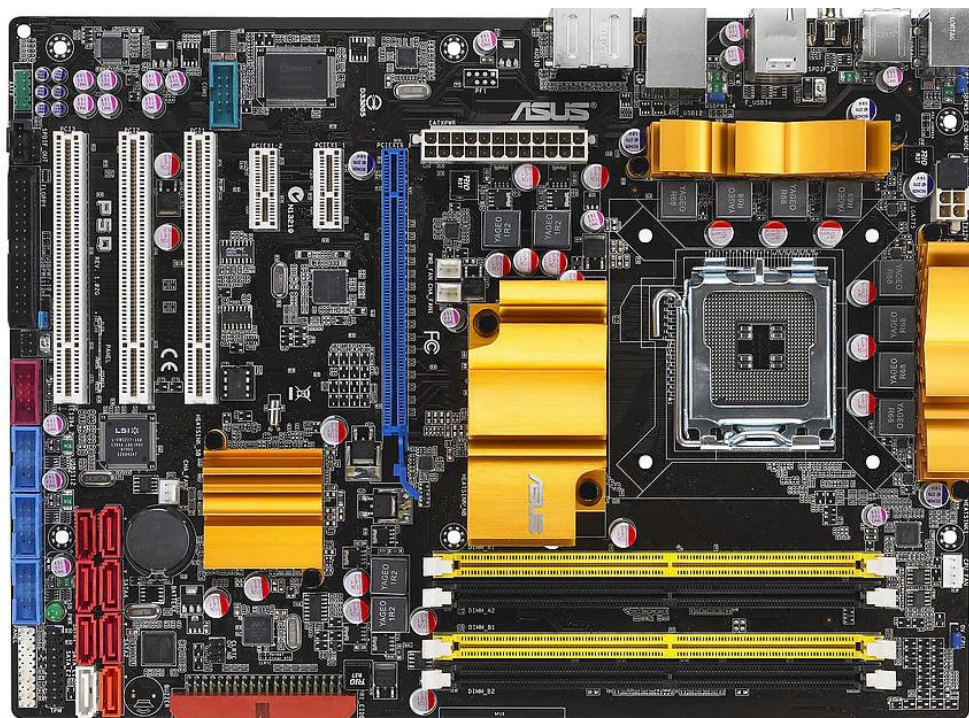
## 1.11 内存地址空间（概述）

- 什么是内存地址空间呢？
- 一个CPU的地址线宽度为10，那么可以寻址1024个内存单元，这1024个可寻到的内存单元就构成这个CPU的内存地址空间。
- 首先需要介绍两部分基本知识，主板和接口卡。



## 1.12 主板

- 在每一台PC机中，都有一个主板，主板上核心器件和一些主要器件。

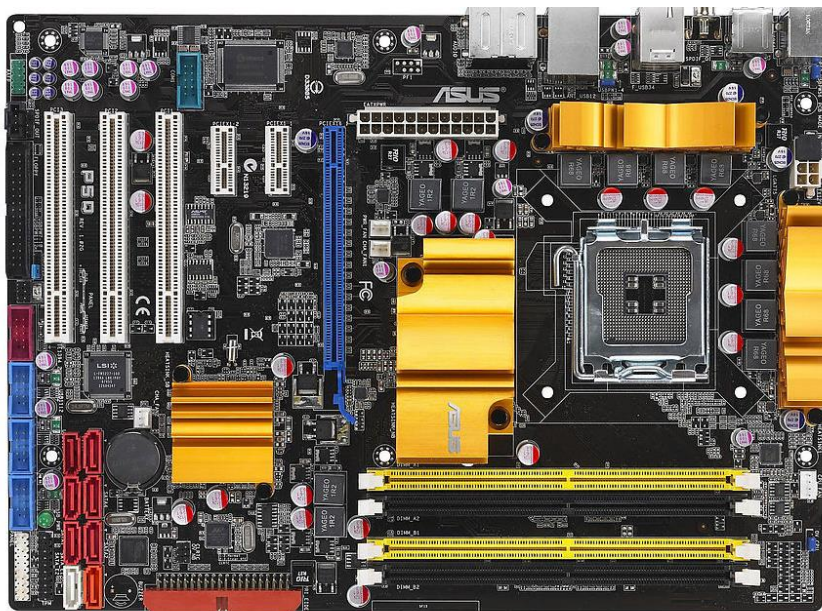


- 这些器件通过总线（地址总线、数据总线、控制总线）相连。



## 1.13 接口卡

- 计算机系统中，所有可用程序控制其工作的设备，必须受到CPU的控制。
- CPU对外部设备不能直接控制，如显示器、音箱、打印机等。直接控制这些设备进行工作的是插在扩展插槽上的接口卡。





## 1.14 各类存储器芯片

- 从读写属性上看分为两类：  
随机存储器（RAM）和只读存储器（ROM）
- 从功能和连接上分类：
  - 随机存储器RAM
  - 装有BIOS的ROM
  - 接口卡上的RAM



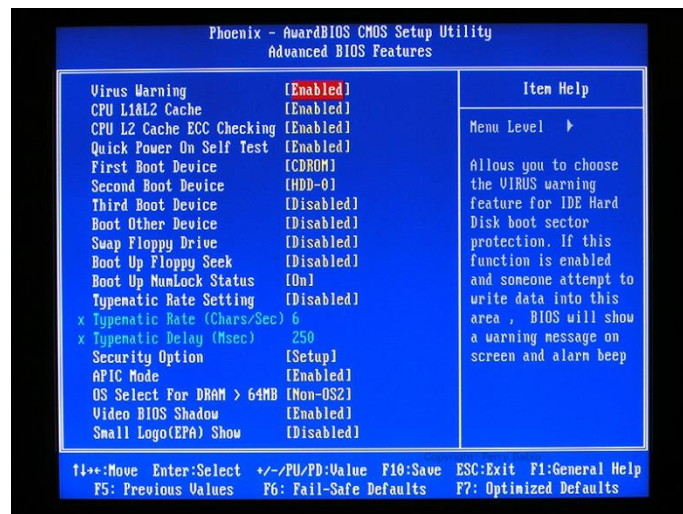


## 1.14 各类存储器芯片

### ■ 装有BIOS的ROM

BIOS: Basic Input/Output System, 基本输入输出系统。

BIOS是由主板和各类接口卡（如：显卡、网卡等）厂商提供的软件系统，可以通过它利用该硬件设备进行最基本的输入输出。在主板和某些接口卡上插有存储相应BIOS的ROM。





## 1.14 各类存储器芯片

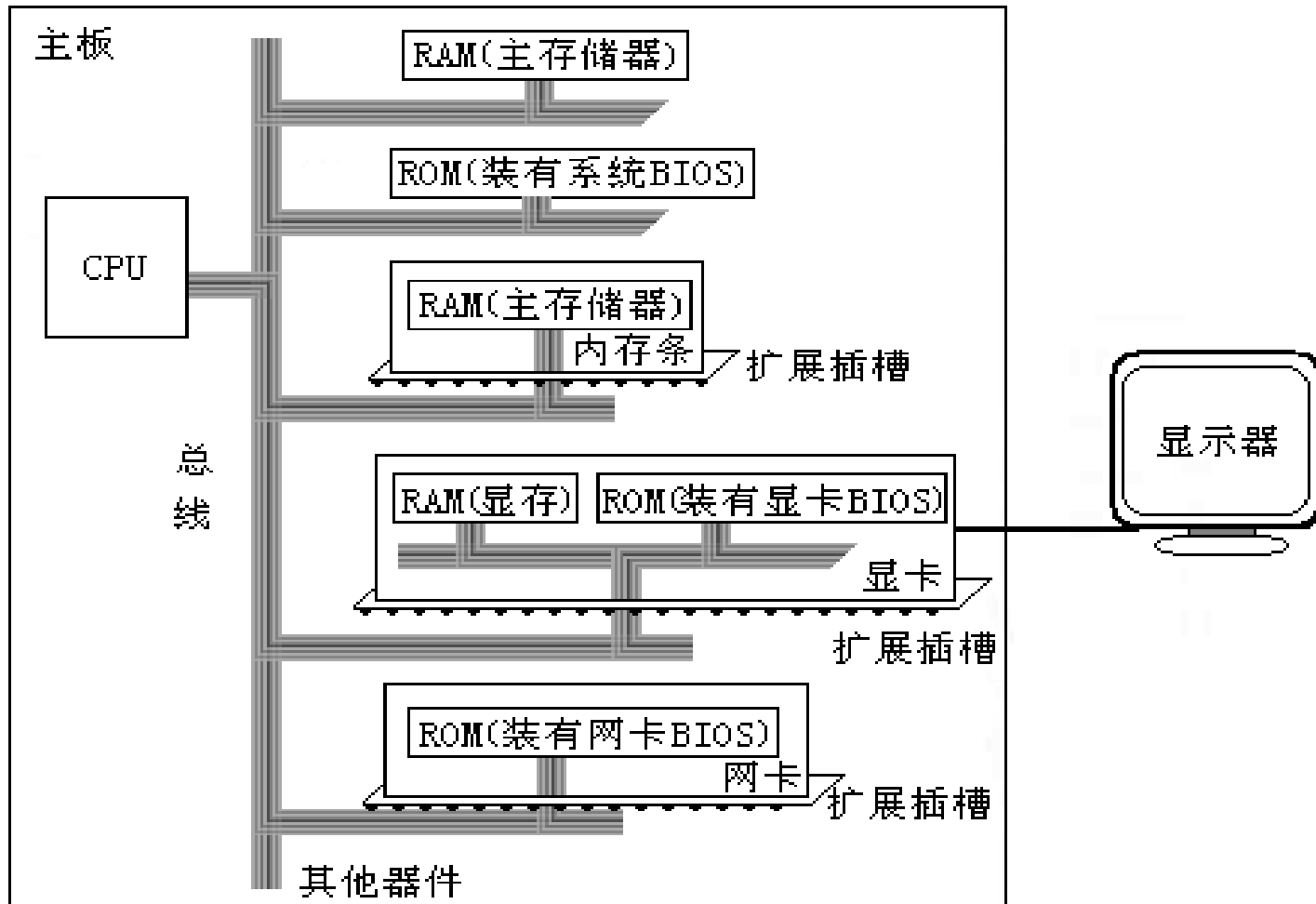
- 新的UEFI标准取代bios





## 1.14 各类存储器芯片

### ■ PC机中各类存储器的逻辑连接情况







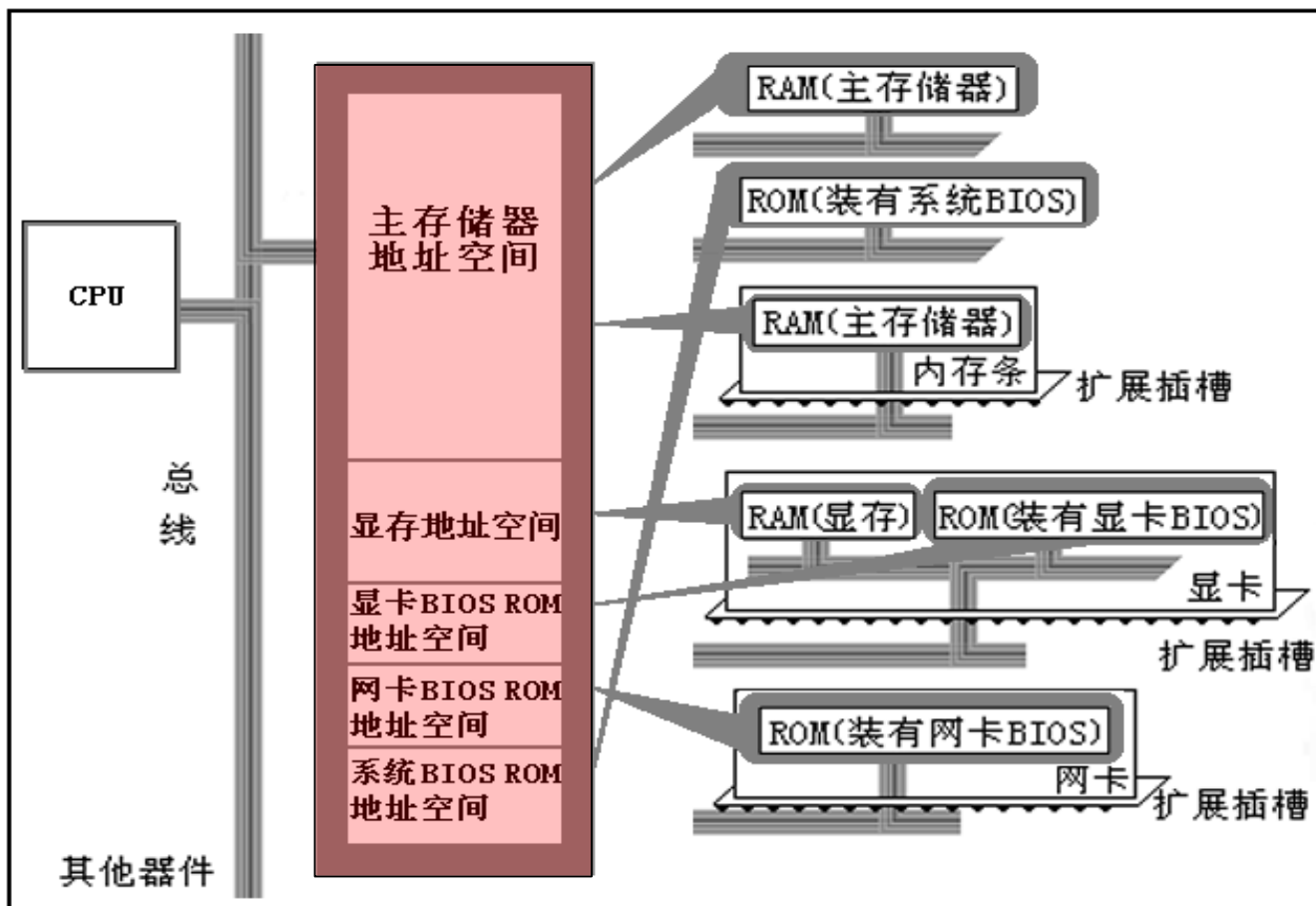
## 1.15 内存地址空间

- 上述存储器在物理上是**独立的器件**。
- 但是它们在以下两点上相同：
  - 1、都和CPU的总线相连。
  - 2、CPU对它们进行读或写的时候都通过控制线发出内存读写命令。



## 1.15 内存地址空间

- 将各类存储器看作一个逻辑存储器：





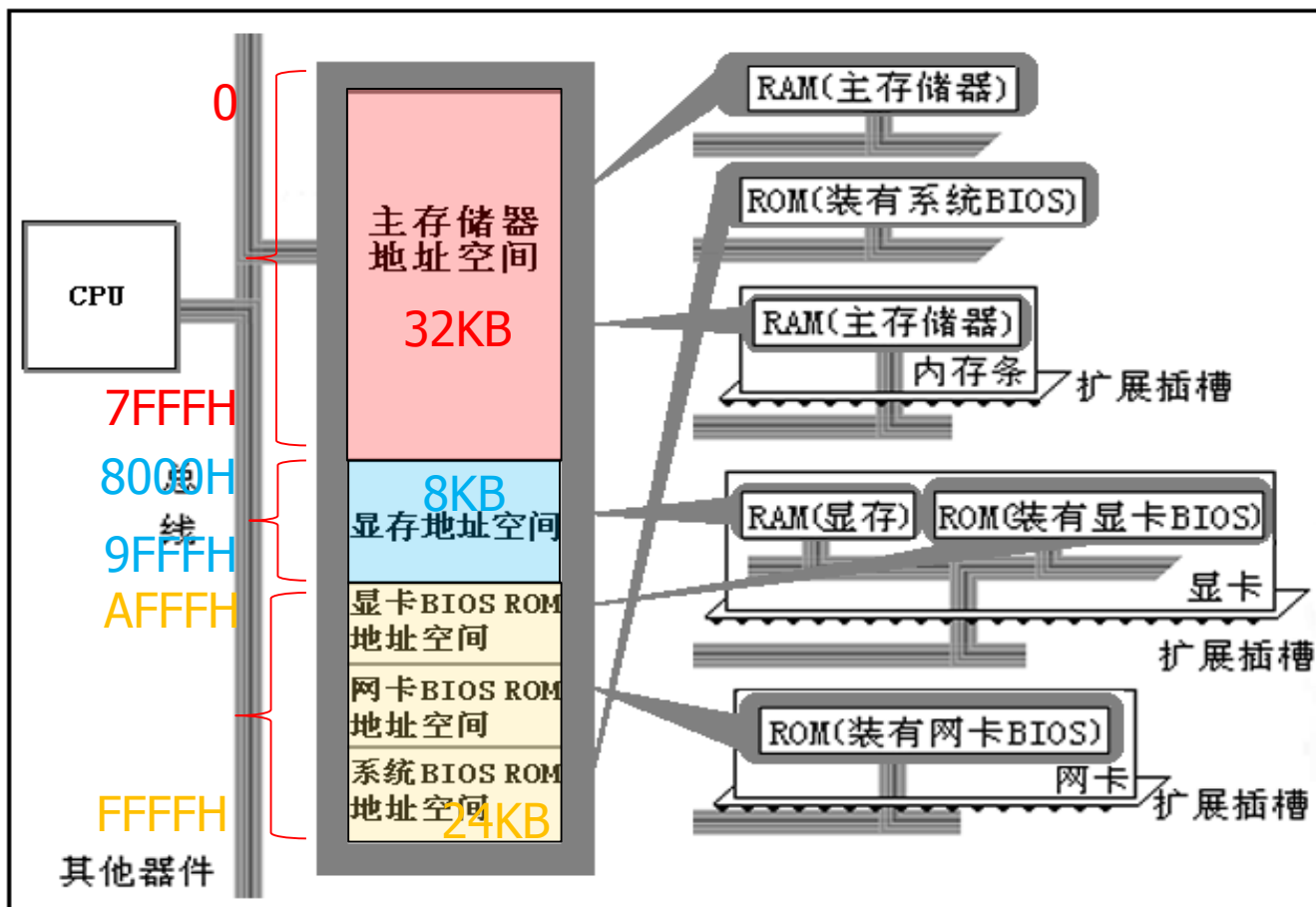
## 1.15 内存地址空间

- 将各类存储器看作一个**逻辑存储器**:
  - 所有的物理存储器被看作一个由若干存储单元组成的逻辑存储器;
  - 每个物理存储器在这个逻辑存储器中占有一个地址段, 即一段地址空间;
  - CPU在这段地址空间中读写数据, 实际上就是在相对应的物理存储器中读写数据。



## 1.15 内存地址空间

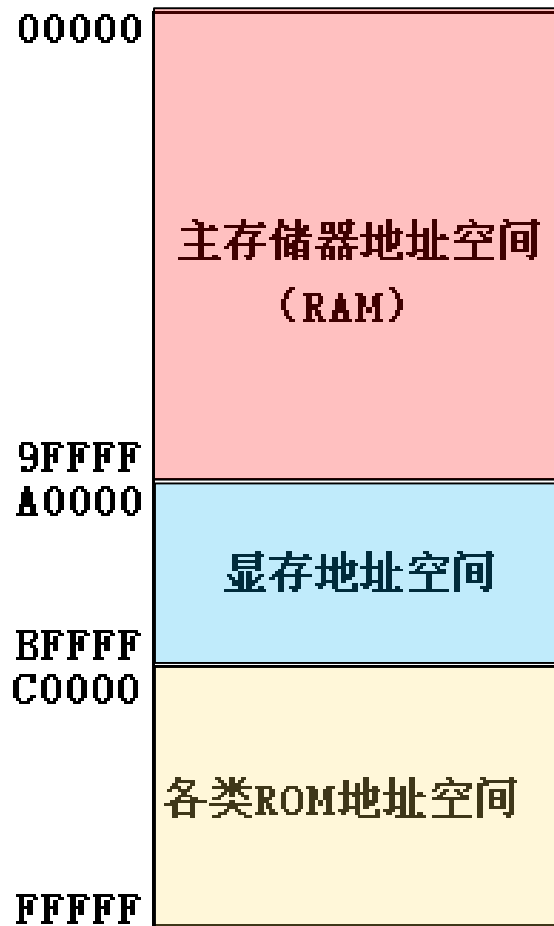
- 例如：内存空间地址段分配如下：





## 1.15 内存地址空间

- 不同的计算机系统的内存地址空间分配情况不同。
- 8086PC机内存地址空间分配的基本情况





## 1.15 内存地址空间

### ■ 内存地址空间：

- 运行程序的是CPU，用汇编编程的时候，必须从CPU角度考虑问题。
- 对CPU来讲，系统中的所有存储器中的存储单元都处于一个统一的逻辑存储器中，它的容量受CPU寻址能力的限制。这个逻辑存储器即是我们所说的内存地址空间。