## 基础知识

汇编语言是一种位于低级语言(机器语言)和高级语言(c语言，python)之间

汇编语言的主体是汇编指令

机器指令是一堆 01

汇编指令 MOV AX，BX

把寄存器BX的内容送到AX中

寄存器：CPU中可以存储的器件

汇编语言由：

汇编指令 机器码的助记符

伪指令 编译器执行

其他符号 编译器识别

核心是汇编指令

汇编指令经过编译器变成机器码让计算机执行

存储器在一台PC中作用仅次于CPU

离开内存 再好的CPU也无法使用

存储单元：8bit(比特)=1bytes(字节)

KB MB GB TB 以1024为倍数

CPU进行数据读写，必须与外部器件（标准说法是芯片）进行三类信息交换

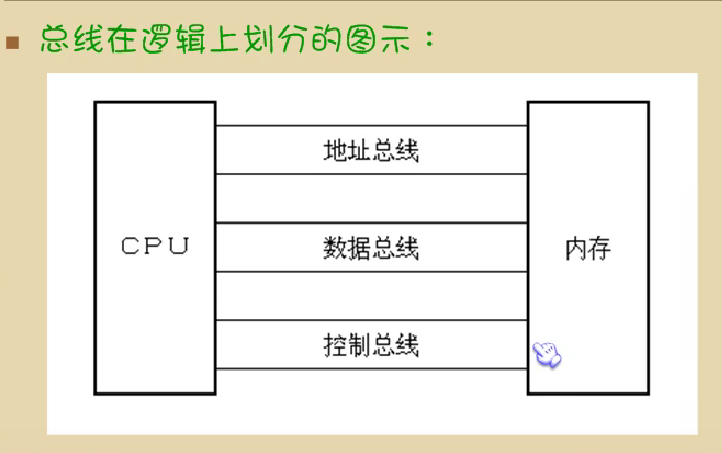
存储单元的地址

器件的选择 都或者写的命令

读或者写的命令

计算机中有连接CPU和其他芯片的导线

地址 数据 控制 （总线）



通过地址总线指定存储单元

宽度为N 可存储2N个内存单元

（索引地址）

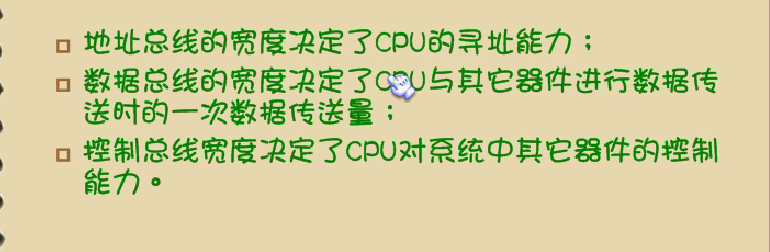
数据通过数据总线传输

数据总线宽度决定CPU和外界的数据传送速度

（类比为路的宽度）

控制总线是各种不同控制线的集合

有多少跟控制线 意味着CPU对外部多少器件进行控制



存储器

随机存储器（RAM）

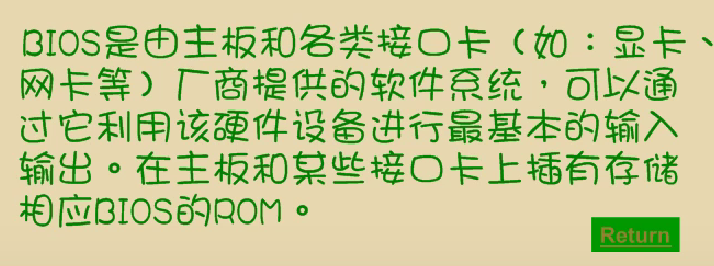
只读存储器（ROM）

功能和连接上分为：

随机存储器RAM

装有BIOS的ROM

BIOS：基本输入/输出系统



接口卡上的RAM

一个CPU由运算器、控制器、寄存器等器件组成 这些器件靠内部总线相连

内部总线实现CPU内部各个器件之间的联系

外部总线实现CPU和主板上其他器件的联系

## 寄存器

8086CPU的所有寄存器都是16位 可以存放两个字节

AX BX CX DX存放一般数据 被称为通用寄存器

字在寄存器中的存储 一字=二字节

1word=2bytes

几条简单的汇编指令

|  |  |  |
| --- | --- | --- |
| 汇编指令 | 控制CPU完成的操作 | 高级语言描述 |
| Mov ax，18 | 将18送入寄存器AX | AX=18 |
| Mov ah，78 | 将78松日寄存器AH | AH=18 |
| Add ax，8 | 将寄存器AX中的数值加8 | AX=AX+8 |
| Mov ax，bx | 将bx中数据送入寄存器AX | AX=BX |
| Add ax，bx | 将ax和bx中的数据相加，结果保存在ax中 | AX=AX+BX |

物理地址

**CPU访问内存单元时需要知道内存单元的地址。**

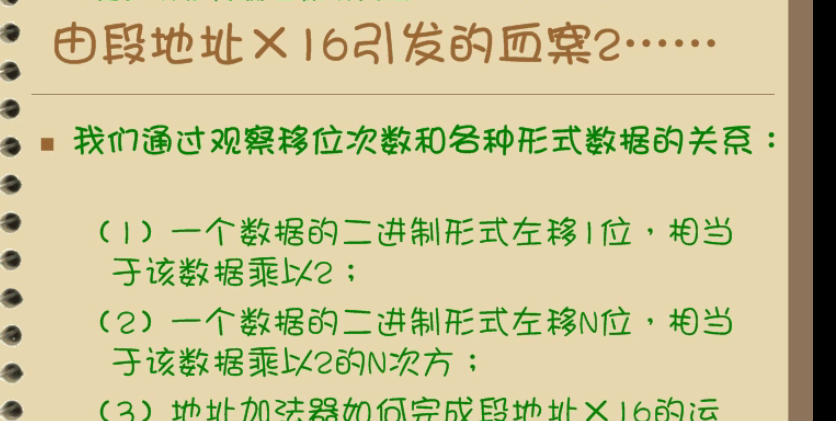
16位的CPU：

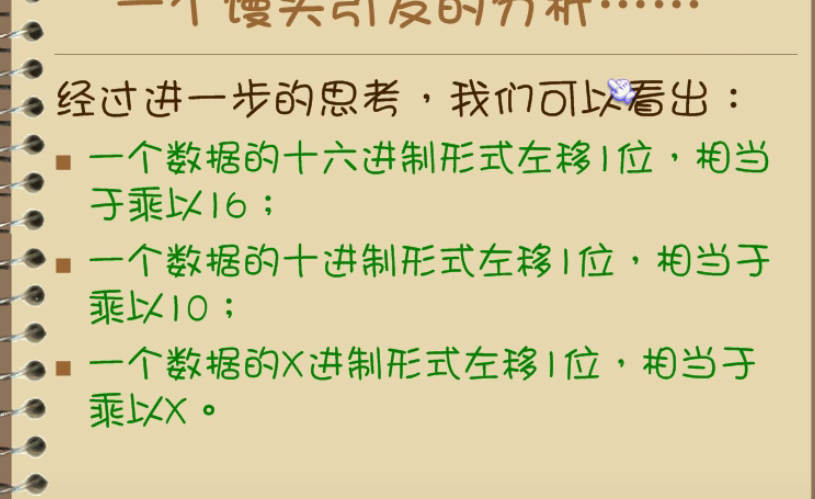
运算器一次性可处理16位的数据

寄存器最大宽度位16位

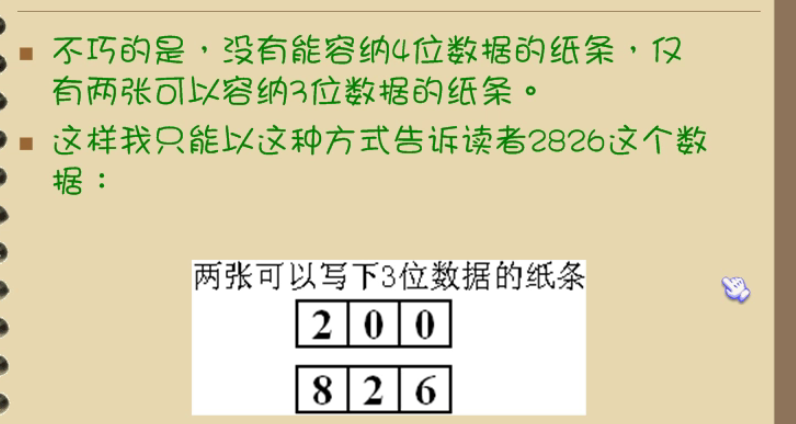
寄存器和运算器之间最大通路是16位

地址加法器采取 **物理地址=短地址X 16 + 偏移地址** 的方法合成物理地址





通过以下这个方法 可以更好的理解物理地址的构建思想



在8086PC机中 存储单元的地址由两个元素描述 即段地址和偏移地址

|  |  |  |
| --- | --- | --- |
| **CS** | **Code**segment | 代码 |
| **DS** | **data** segment | 数据 |
| **SS** | **stack** segment | 栈 |
| **ES** | **extra** segment | 额外 |

段寄存器提供段地址：

segment

代码

CS位代码段寄存器

IP是偏离地址寄存器

CPU读取和执行相关指令的过程最好结合视频或者书本上的图例进行理解学习

