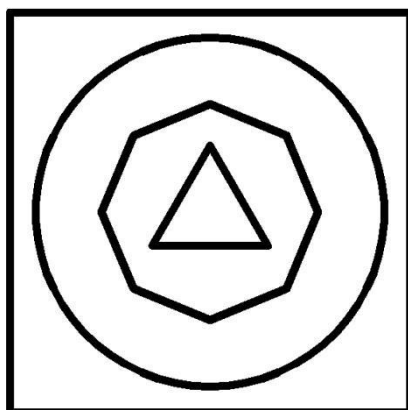


8086:物理地址=段地址\*16D(或 10H)+偏移地址

**8086:物理地址=段地址\*16D(或 10H)+  
偏移地址**



本文件由皇天惊虞制作，免费流通于网络。

制作时间 2023.12.29

8086:物理地址=段地址\*16D(或 10H)+偏移地址

## 8086:物理地址=段地址\*16D(或 10H)+偏移地址

首先我们知道 8086 中段地址是从能被 10H 整除的地址开始的，即地址最后一位都是 0（16 进制表示）。这样在设计 CPU 时就没有必要表示这个最后一个 0 了。例如 FFFF[0]H，这个地址，假设段地址是 FFFFH，那么它的偏移地址就是 0000H 了。因为 CPU 在设计时去掉了一个 16 进制的 0，所以要乘以 10H（十进制的 16）。

8086CPU 有 20 位地址总线，这个总线是指的外部总线（也就是 CPU 与内存交换数据时的总线的宽度），可以传送 20 位的地址，达到 1M 的寻址能力。

8086CPU 又是 16 位的结构，在内部一次处理，暂时存储，传输的地址为 16 位，这里的 16 位指的是 CPU 内部总线（也就是 CPU 内各个寄存器直接传送数据时的总线宽度）。

所以如果从 CPU 内部直接发送数据只能送出 16 位的地址，表现出的寻址能力只有 64KB（小于前面说的 1M）。为了解决这个问题，8086CPU 在内部采用两个 16 位的地址（一个是段地址，一个是偏移地址）合成一个 20 位的地址。所以合成的方法就是:段地址\*16+偏移=物理地址。

段地址\*16 的意思是：让段地址左移 4 位，比如 2000H 左移 4 位，成为 20000H。再加上一个 16 位的偏移地址。这样就是 20 位的物理地址了。

16 位实模式下，一个物理地址由段地址和偏移地址两部分组成，段地址在 16 位段寄存器中，然后在指令中用 16 位偏移地址寻址。物理地址=段地址\*0x10+偏移地址

对于 8086/8088 来说计算实际地址是用绝对地址对 1M 求模。8086 的地址线的物理结构: 20 根, 也就是它可以物理寻址的内存范围为  $2^{20}$  个字节, 即 1M 空间, 但由于 8086/8088 所使用的寄存器都是 16 位, 能够表示的地址范围只有 0-64K, 这和 1M 地址空间来比较也太小了, 所以为了在 8086/8088 下能够访问 1M 内存, Intel 采取了分段寻址的模式: 16 位段基址:16 位偏移 EA。其绝对地址计算方法为: 16 位基址左移 4 位+16 位偏移=20 位地址。

按基址最大 FFFFh, 偏移最大 FFFFh. 计算地址:  $FFFFh * 10h + FFFFh = 10FFEFh$ 。如果访问 100000h~10FFEFh 之间的内存（大于 1M 空间），则必须有第 21 根地址线来参与寻址（8086/8088 没有）。因此，当程序员给出超过 1M（100000H-10FFEFH）的地址时，因为逻辑上正常，系统并不认为其访问越界而产生异常，而是自动从 0 开始计算，也就是说系统计算实际地址的时候是按照对 1M 求模的方式进行的，这种技术被称为 wrap-around

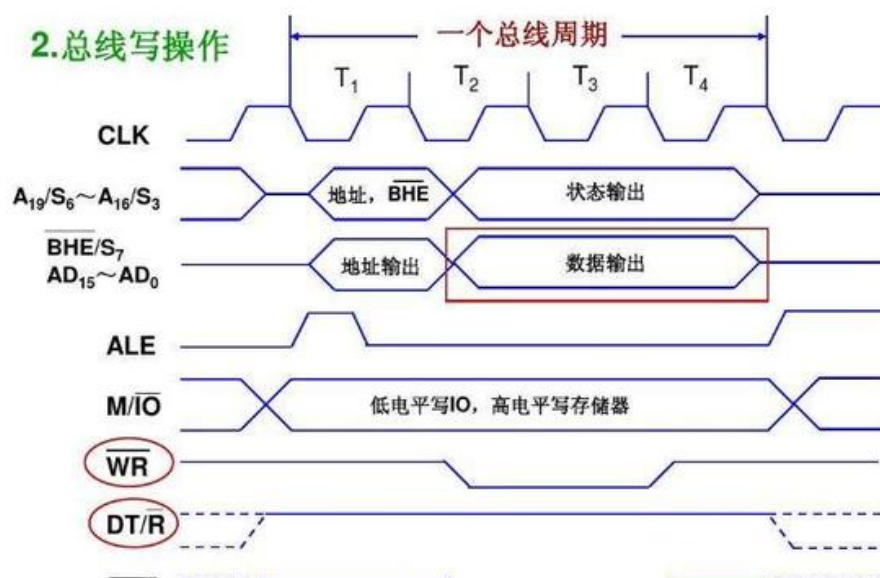
在 80286 以及更高系列的 PC 中，即使 A20 Gate 被打开，在实模式下所能够访问的内存最大也只能为 10FFEFH，尽管它们的地址总线所能够访问的能力都大大超过这个限制。为了能够访问 10FFEFH 以上的内存，则必须进入保护模式。32 位保护模式下，问题变复杂了。首先要明白，保护模式保护什么？保护的是：分清楚各个程序使用的存储区域，不允许随便跨界访问。然后，怎么保护？方式是：为内存里的每段地址空间定义一些安全上的属性，比如可以被多少优先级的代码写入，是不是允许执行等。这个时候，段寄存器远远不能满足要求了。原因有二：段寄存器只有 32 位，保存不了这么多信息；段寄存器个数有限，不能保

8086:物理地址=段地址\*16D(或 10H)+偏移地址

存内存中所有段的信息。

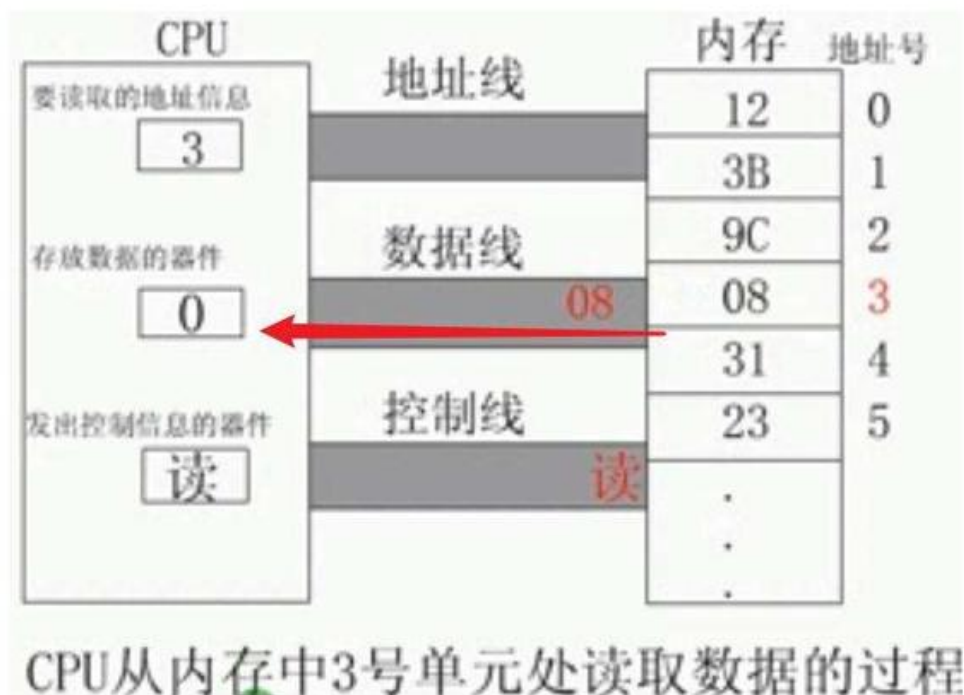
有 16 根数据线和 20 根地址线。

1.推出 8086 的主要目的是为了与当时已有的一套 Inter 外部设备接口芯片直接兼容使用。8086 与 8088 在寄存器结构, 编程结构, 存储器组织及 I/O 端口组织方面是完全一样的或稍有差别。

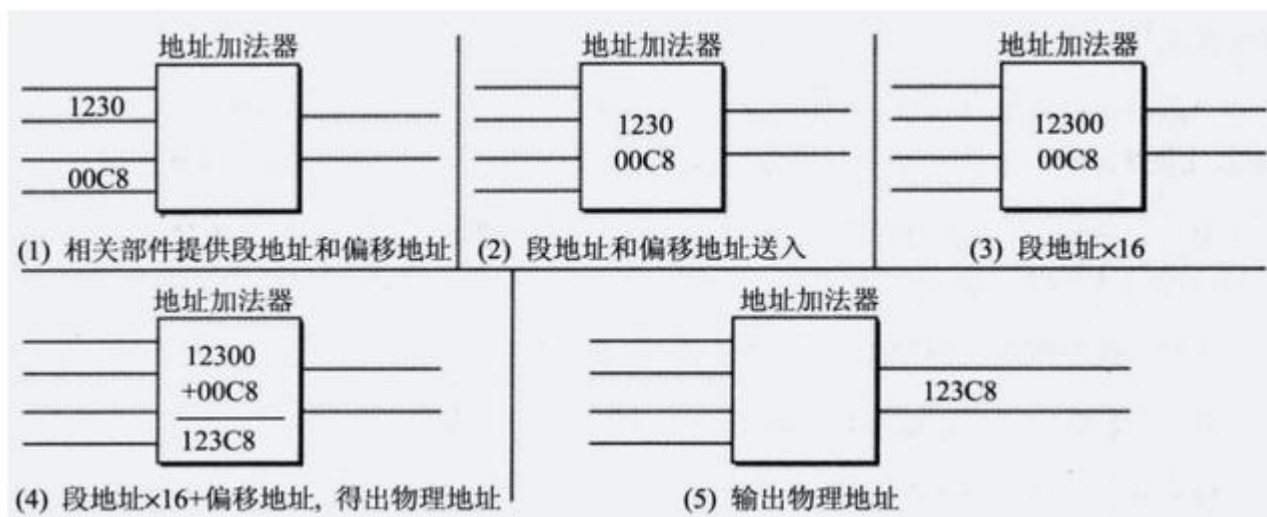


2.Intel 8086 有四个 内存区段 (segment)寄存器, 可以从索引寄存器来设定。区段寄存器可以让 CPU 利用特殊的方式存取 1 MB 内存。8086 把段地址左移 4 位然后把它加上偏移地址。大部分的人都认为这是一个很不好的设计, 因为这样的结果是会让各分段有重叠。

8086:物理地址=段地址\*16D(或 10H)+偏移地址



3. Intel 8086 是一个由 Intel 于 1978 年所设计的 16 位微处理器芯片，是 x86 架构的鼻祖。不久之后，Intel 就推出了 Intel 8088 (一个拥有 8 根外部数据总线的微处理器)。它是以 8080 和 8085 的设计为基础，拥有类似的寄存器组，但是数据总线扩充为 16 位。总线接口单元(Bus Interface Unit)透过 6 字节预存(prefetch) 的队列(queue)位指令给执行单元(Execution Unit)，所以取指令和执行是同步的，8086 CPU 有 20 条地址线，可直接寻址 1MB 的存储空间，每一个存储单元可以存放一个字节 (8 位) 二进制信息。



扩展： 8086 可以和浮点运算器、输入/输出处理器或其他处理器组成多处理器系统，从而极大地提高了系统的数据吞吐能力和数据处理能力；8086 微处理器从功能上可以划分为两个逻辑单元：执行部件 EU (Execution Unit) ；总线接口部件 BIU (Bus Interface Unit) 。8086 微处理器的寄存器结构：8086 可供编程使用的有 14 个 16 位寄存器，按其用途可分为 3 类：通用寄存器；段寄存器；指针和标志寄存器。

8086:物理地址=段地址\*16D(或 10H)+偏移地址

[http://www.360doc.com/content/23/0414/17/99071\\_1076506880.shtml](http://www.360doc.com/content/23/0414/17/99071_1076506880.shtml)

[https://blog.csdn.net/qq\\_43446165/article/details/104625461](https://blog.csdn.net/qq_43446165/article/details/104625461)