访问存储器

为了读出程序、处理数据等，必须要在CPU与内存之间进行数据的传送。数据是利用称为数据总线（data bus）的传输途径进行交换的。所谓总线(bus)，就是计算机主板上的某种传输线路。总线的宽度（即传输线的条数），会因CPU而各异。比如，对于32位总线而言，每次可以交换32位的数据。但需要注意的是，总线位数并不是CPU的位数，而是指传输数据时的位数。例如，英特尔公司的奔腾系列为32位的CPU，其数据总线的宽度为64位。总线是与从CPU芯片里面引出来的，一根一根针（pin）脚相对应的。如果能对照CPU芯片或计算机主板的实物，找到称为总线的传输线路，对总线的概念，就会有感觉了。

此外，为了从内在读出数据，或者将数据写入内存，就必须要指定读出的位置和写入的位置。因此，内存中的每个字节（Byte）都被分配了相应的编号（当然不一定都要以字节为单位，也有采用别的单位进行编号的），这种编号就称为地址（address）。当要访问内存时，就需要从CPU传送出指定地址的电信号。这种信号也是通过总线进行传送的，但它与传送数据所用的总线（数据总线）不同，是由地址总线的位数所决定的。

比如，若地址总线只有16条（即16位总线），其所能存取访问的地址是，从0000 0000 0000 0000地址到1111 1111 1111 1111地址为止，只有65636个单元。现在市场上的奔腾系列CPU，其地址总线有32根。它可以直接访问的内在单元为232＝4GB（约40亿个字节）。

CPU和操作系统的位数

常常使用32位或16位（bit）这样的字眼儿，去表示CPU的种类。比如，英特尔公司的奔腾系列CPU，是32位。这个位数，其实是指算术逻辑电路一次所能计算的数据量。在奔腾等系列的CPU中，可以把所有通用器都用于计算，也可以只使用其中的一部分。例如，EAX寄存器的长度为32位，可将其低端的16位，称作AX寄存器，也可以独立加以应用，如果计算的数值比较小的话，就可以使用AX寄存器与BX寄存器。其加法运算的程序指令，则为：

ADD AX，BX

为什么要这样做呢？主要理由，是为了确保CPU的“互换性”或“兼容性”。即使开发出新的（位数更多的）CPU，如果不能确保过去编制的程序，仍能原封不动地在新CPU上运行，那么新CPU就没有存在的意义了。要把迄今所编制的程序，只因出现了新型CPU就全部修改一遍，无论如何也是很不现实的事情。

比如，采用16位的CPU（或16位指令），也能计算32位的数据。只需将其分成高端16位与低端16位，分开计算2次，就可以了。但量，若用32位的CPU，使用32位的指令，只需计算一次就行了。

由于操作系统软件，也是由各种程序组成的，也有32位与16位的区分。过去老的16位操作系统（如windows3.1等），也可以用在32位的CPU上，只是不能充分发挥新32位硬件的优越性能而已。

数据总线x64

（为传输数据所用）

内存

控制芯片

地址总线x32

（用于指定数据存放在内存中的哪个具体位置。）

CPU位数一般与其中通用寄存器的位数是相同的

CPU