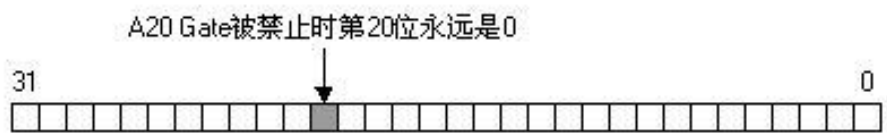
激活A20 地址线详解

A20地址线是一个很不容易理解的地方，这主要是由于历史原因造成的。在8086/8088中，有20根地址线，所以可以访问的地址是2^20=1M，但由于8086/8088是16位地址模式，能够表示的地址范围是0-64K，为了在8086/8088下能够访问1M 内存，Intel 采取了分段的模式：16位段基地址:16位偏移。但是这种方式有一个问题,他的最大的访问空间为：0xFFFF：0xFFFF=0x10FFEF=1M+64K-16Bytes，但8086/8088只有20位地址线，如果访问100000h~10FFEFh 之间的内存，则必须有第21根地址线。所以当程序员给出超过1M（100000H-10FFEFH）的地址时，系统并不认为其访问越界而产生异常，而是自动从重新0开始计算。到了80286，系统的地址总线发展为24根，这样能够访问的内存可以达到2^24=16M。在实模式下，80286和其后续系统所表现的行为应该和8086/8088所表现的完全一样，但是，80286芯片却存在一个BUG：如果程序员访问100000H-10FFEFH 之间的内存，系统将实际访问这块内存，而不是重新从0开始。为了解决这个问题，IBM 使用键盘控制器上剩余的一些输出线来管理第21根地址线，即 A20 Gate。如果 A20 Gate 被打开，则当程序员给出100000H-10FFEFH 之间的地址的时候，系统将真正访问这块内存区域；如果 A20 Gate被禁止，则当程序员给出100000H-10FFEFH 之间的地址的时候，系统仍然使用8086/8088的方式。从80286开始，系统出现了一种新的机制，被称为保护模式。那为什么进入保护模式一定要打开A20呢，它对保护模式有什么影响？如果A20 Gate被禁止，对于80286来说，其地址为24bit，其地址表示为 EFFFFF；对于80386极其随后的32-bit 芯片来说，其地址表示为FFEFFFFF 。这种表示的意思是如果A20 Gate 被禁止，则其第20-bit 在CPU 做地址访问的时候是无效的,永远只能被作为0;如果 A20 Gate被打开,则其第20-bit 是有效的，其值既可以是0，又可以是1。



所以，如果 A20被禁止，可访问的内存只能是奇数段（2N+1）M，只有当 A20被打开

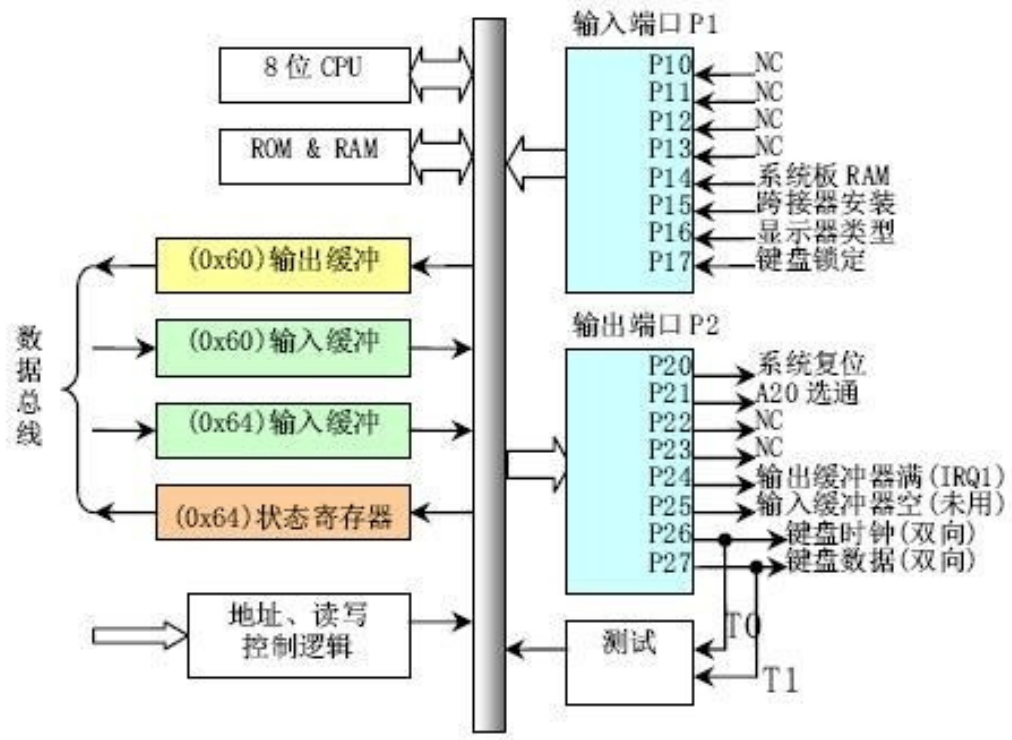
的时候才能访问连续的内存。

下面讨论一下如何打开A20地址线：

通常的方法是通过设置键盘控制器的端口值，不过有些系统觉得键盘控制器很慢，为此引入了一个Fast Gate A20，它用IO端口的0x92来处理A20信号线。还有一种方法是通过读取0xee端口来开启A20地址线，写端口则会禁止地址线。

我给大家介绍的是最普通的方法

先看看804x的逻辑图



由此可见，端口p2的第1位（从0开始）就是控制A20地址线的地方，如果它置位，则开通A20地址线。分配给IO控制器的端口为0x60-0x6f，但IBM PC只用0x60和0x64，如下图：

端口	读/写	名称	用途
0x60	读	数据端口或输出缓冲器	是一个 8 位只读寄存器。当键盘控制器收到来自键盘的扫描码或命令响应时，一方面置状态寄存器位 0 = 1，另一方面产生中断 IRQ1。通常应该仅在状态端口位 0 = 1 时才读。
0x60	写	输入缓冲器	用于向键盘发送命令与/或随后的参数，或向键盘控制器写参数。键盘命令共有 10 多条，见表格后说明。通常都应该仅在状态端口位 1=0 时才写。
0x61	读/写		该端口 0x61 是 8255A 输出口 B 的地址，是针对使用/兼容 8255A 的 PC 标准键盘电路进行硬件复位处理。该端口用于对收到的扫描码做出应答。方法是首先禁止键盘，然后立刻重新允许键盘。所操作的数据为： 位 7=1 禁止键盘；=0 允许键盘； 位 6=0 迫使键盘时钟为低位，因此键盘不能发送任何数据。 位 5-0 这些位与键盘无关，是用于可编程并行接口(PPI)。
0x64	读	状态寄存器	是一个 8 位只读寄存器，其位字段含义分别为： 位 7=1 来自键盘传输数据奇偶校验错； 位 6=1 接收超时(键盘传送未产生 IRQ1)； 位 5=1 发送超时(键盘无响应)； 位 4=1 键盘接口被键盘锁禁止；[??是=0 时] 位 3=1 写入输入缓冲器中的数据是命令(通过端口 0x64)； =0 写入输入缓冲器中的数据是参数(通过端口 0x60)； 位 2 系统标志状态：0 = 上电启动或复位；1 = 自检通过； 位 1=1 输入缓冲器满(0x60/64 口有给 8042 的数据)； 位 0=1 输出缓冲器满(数据端口 0x60 有给系统的数据)。
0x64	写	输入缓冲器	向键盘控制器写命令。可带一参数，参数从端口 0x60 写入。键盘控制器命令有 12 条，见表格后说明。

系统在向0x60发送一字节的时候，就是发送键盘命令，键盘在接受命令后20ms 内给予反应，并返回一个0xfa相应码。

键盘命令如下图：

命令码	参数	功能
0xed	有	设置/复位模式指示器。置 1 开启，0 关闭。参数字节： 位 7-3 保留全为 0； 位 2 = caps-lock 键； 位 1 = num-lock 键； 位 0 = scroll-lock 键。
0xee	无	诊断回应。键盘应回送 0xee。

0xf0	有	读取/设置扫描码集。参数字节等于： 0x00 - 选择当前扫描码集； 0x01 - 选择扫描码集 1(用于 PCs，PS/2 30 等)； 0x02 - 选择扫描码集 2(用于 AT，PS/2，是缺省值)； 0x03 - 选择扫描码集 3。
0xf1		保留不用。
0xf2	无	读取键盘标识号(读取 2 个字节)。AT 键盘返回响应码 0xfa。
0xf3	有	设置扫描码连续发送时的速率和延迟时间。参数字节的含义为： 位 7 保留为 0； 位 6-5 延时值；令 C=位 6-5，则有公式：延时值=(1+C)*250ms； 位 4-0 扫描码连续发送的速率；令 B=位 4-3；A=位 2-0，则有公式： 速率=1/((8+A)*2^B*0.00417)。 参数缺省值为 0x2c。
0xf4	无	开启键盘。
0xf5	无	禁止键盘。
0xf6	无	设置键盘默认参数。
0xf7-0xfd		保留不用。
0xfe	无	重发扫描码。当系统检测到键盘传输数据有错，则发此命令。
0xff	无	执行键盘上电复位操作，称之为基本保证测试(BAT)。操作过程为： 1. 键盘收到该命令后立刻响应发送 0xfa； 2. 键盘控制器使键盘时钟和数据线置为高电平； 3. 键盘开始执行 BAT 操作； 4. 若正常完成，则键盘发送 0xaa；否则发送 0xfd 并停止扫描。

系统向0x64写一字节，就是发送一个键盘控制器命令，可带参数，参数由0x60发送。

键盘控制命令如图：

命令	参数	功能
0x20	无	读给键盘控制器的最后一个命令字节，放在端口 0x60 供系统读取。
0x21-0x3f	无	读取由命令低 5 比特位指定的控制器内部 RAM 中的命令。
0x60-0x7f	有	写键盘控制器命令字节。参数字节：(默认值为 0x5d) 位 7 保留为 0； 位 6 IBM PC 兼容模式(奇偶检验，转换为系统扫描码，单字节 PC 断开码)； 位 5 PC 模式（对扫描码不进行奇偶校验；不转换成系统扫描码)； 位 4 禁止键盘工作（使键盘时钟为低电平)； 位 3 禁止超越(override)，对键盘锁定转换不起作用； 位 2 系统标志；1 表示控制器工作正确； 位 1 保留为 0； 位 0 允许输出寄存器满中断。
0xaa	无	初始化键盘控制器自测试。成功返回 0x55；失败返回 0xfc。

0xab	无	初始化键盘接口测试。返回字节： 0x00 无错； 0x01 键盘时钟线为低(始终为低，低粘连)； 0x02 键盘时钟线为高； 0x03 键盘数据线为低； 0x04 键盘数据线为高；
0xac	无	诊断转储。804x 的 16 字节 RAM，输出口、输入口状态依次输出给系统。
0xad	无	禁止键盘工作（设置命令字节位 4=1）。
0xae	无	允许键盘工作（复位命令字节位 4=0）。
0xc0	无	读 804x 的输入端口 P1，并放在 0x60 供读取；
0xd0	无	读 804x 的输出端口 P2，并放在 0x60 供读取；
0xd1	有	写 804x 的输出端口 P2，原 IBM PC 使用输出端口的位 2 控制 A20 门。注意，位 0(系统复位)应该总是置位的。
0xe0	无	读测试端 T0 和 T1 的输入送输出缓冲器供系统读取。 位 1 键盘数据；位 0 键盘时钟。
0xed	有	控制 LED 的状态，置 1 开启，0 关闭。参数字节： 位 7-3 保留全为 0； 位 2 = caps-lock 键； 位 1 = num-lock 键； 位 0 = scroll-lock 键。
0xf0-0xff	无	送脉冲到输出端口。该命令序列控制输出端口 P20-23 线，参见键盘控制器逻辑示意图。欲让哪一位输出负脉冲(6 微秒)，即置该位为 0。也即该命令的低 4 位分别控制负脉冲的输出。例如，若要复位系统，则需发出命令 0xfe(P20 低)即可。

从理论上讲，打开A20 Gate的方法是通过设置8042芯片输出端口（64h）的2nd-bit，但事实上，当你向8042芯片输出端口进行写操作的时候，在键盘缓冲区中，或许还有别的数据尚未处理，因此你必须首先处理这些数据。

所以，激活A20地址线的流程为：

1.禁止中断；

2.等待，直到8042 Input buffer为空为止；

3.发送 Write 8042 Output Port命令到8042 Input buffer

4.等待，直到8042 Input buffer为空为止；

5.向 P2写入数据，将 OR2置1；

相关程序如下：

```
act20addr:
    call wait_8042free      ;;等待8042为空
    mov al,0x0D1            ;;D1的意思是向8042端口的 P2写数据
```

```
        mov dx,0x064
        out dx,al
        call wait_8042free
        mov al,0x0DF        ;;DF 为11011111，写入 P2，根据图示，A20位置1，开通了A20地址线
        mov dx,0x060
        out dx,al
        ret
    wait_8042free:
    .ll_begin:
        in al,0x64
        test al,0x02   ;;测试指令，与运算。判断第二位是否为0，如果为0，则代表8042是空的
        jnz .ll_begin     ;;如果运算为0，即8042为空，则返回，不为空，则继续读取缓冲器内容
        ret
```

到此为止，A20的激活就已经完成了。