Intel80386CPU

一、80386 概述

80386处理器被广泛应用在1980年代中期到1990年代中期的IBM PC相容机中。这些PC机称为「80386电脑」或「386电脑」，有时也简称「80386」或「386」。80386的广泛应用，将PC机从16位时代带入了32位时代。80386的强大运算能力也使PC机的应用领域得到巨大扩展，商业办公、科学计算、工程设计、多媒体处理等应用得到迅速发展。它的数据总线和地址总线都是32位，直接寻址的内存空间4GB,虚拟地址空间为64TB。芯片上集成了27.5万个晶体管，主频16-33MHz。它是X86第一个真正的32位CPU，它能提供真正的多任务处理和建立虚拟系统的能力。

二、80386的引脚及功能

80386 DX有132根引脚，采用PGA(Pin Grid Array，引脚网格阵列)封装，采用这种封装工艺单根引脚所占用的面积较双列直插时小，因此引脚数目可以多一些，不必再采用引脚复用技术。因此，在80386中数据线和地址线是分开设置的，控制信号和状态信号也不再复用引脚。其中34 条地址线(A31～A2、BE3～BE0)，32 条数据线(D31～D0)，3 条中断线，1条时钟线，13 条控制线，20 条电源线VCC，21条地线VSS，还有8 条为空。

与8086/8088 相比，需要说明以下几点：

1）时钟(CLK2)： 80386 的基本定时信号由CLK2 提供。CLK2 的频率是80386 内部时钟信号频率的两倍，输入该信号与82384 时钟信号同步，经80386 内部2 分频之后得到80386 的工作基准频率信号。

2）数据总线(D31～D0)：为80386 和其他设备之间提供数据通路，32 位数据总线，双向三态，一次可传送8 位、16 位或32 位数据，由输入信号(BE3～BE0)和BE16确定。在任何写操作周期(包括暂停周期和停机周期)，80386 总是驱动数据总线的所有32 位信号，而不管当前总线的实际宽度。

3）地址总线(A31～A2，BE3～BE0) 。

- A31～A2：地址总线，输出三态，和BE3～BE0相结合起到32位地址的作用。80386地址总线包含A2～A31地址线和字节选通线BE3～BE0。BE3～BE0线的功能与8086和80286的A0和BHE非常相似，它们是内部地址信号A0和A1的译码。由于80386有一个32位数据总线，所以内存可以建立4B宽的存储体。BE3～BE0信号是用来选通这4B个存储体。这些单独选通可以使80386 的内存传送或者接收字节、字或者双字。
- BE3～BE0：字节选通信号。用于选通在当前的传送操作要涉及4B数据中的哪几个字节。BE0对应于D0～D7，BE1对应于D8～D15，BE2对应于D16～D23，BE3对应于D24～D31。

4）总线周期定义信号(M/IO，W/R，D/C，LOCK，三态，输出，用来定义正在进行的总线周期类型)。

- M/IO：存储器/输入输出选择信号，输出信号。高电平时访问存储器，低电平时访问I/O 端口。80386 直接I/O 端口简单地把8086 和80286 端口结构扩充成32 位端口。32 位I/O 端口可以通过并联8 位I/O 端口设备（如8255A）来构成。80386 可以使用所有8 位端口地址的IN 或OUT 指令来编址256 个8 位端口、128 个16 位端口、64 个32 位端口。使用DX 寄存器存放16 位端口地址，80386 可以编址64K 个8 位端口、32K 个16 位端口或8K 个32 位端口。
- W/R：读/写控制输出信号，高电平时写入，低电平时读出。
- D/C：数据/指令控制信号，输出。高电平时传送数据，低电平时传送指令代码，D/C指示总线操作是一个数据读/写还是控制字传输(如取一个操作码)。
- W/R、D/C、M/IO是总线周期定义信号。当80386 驱动ADS(地址状态)输出信号有效时，这3个信号被驱动为有效，根据3 个信号的功能可得到总线周期定义，见表3-8。
- LOCK：总线周期封锁信号，低电平有效。

5）总线控制信号(ADS，READY，NA，BE16)。

这组信号用来表示总线周期何时开始，以及数据总线的宽度和总线周期的终结。

- ADS：地址选通信号，三态输出，低电平有效。当有效时，表示总线周期中地址信号有效。当有效地址、BE信号和总线周期定义信号均在总线上时，ADS信号将被设置。因为80386 地址总线是不可复用的，所以8086 类型的ALE 信号是不需要的。但是，在某些80386中，ADS信号用于一种称为地址流水线的模式，将地址传送到外部锁存器。地址流水线的原理：如果一个地址保持在外部锁存器的输出端，80386 就可以把地址引脚上的“老”地址清除，并在总线周期的前期输出下一个操作的地址。外部控制芯片通过设置下一个地址信号来通知80386 何时为下一个操作输出地址。对一个有SRAM 高速缓冲的系统，流水线地址模式通常不是必需的，因为SRAM 高速缓冲已足够快了，不需要等待状态。
- READY：准备就绪，输入信号，低电平有效。READY有效时表示当前总线周期已完成。信号用来在总线周期中根据低速的内存或I/O设备接口的需要插入等待状态。
- NA：下一个地址请求信号，输入信号，低电平有效。允许地址流水线操作，当其有效时，表示当前执行中的周期结束之后，下一个总线周期的地址和状态信号可变为有效。
- BE16：输入信号，低电平有效，指定16 位数据总线。BE16输入端允许80386以16位和/或32位数据总线工作。如果设置了BE16，那么80386只将数据传送到32位数据总线的低16位上。如果设置了BE16并且要从16位宽内存中读一个32位的操作数，那么80386将自动产生一个第二总线周期来读第二个字。对于未调整的传输，如果设置了BE16，那么80386 也产生所需数目的总线周期。

6）总线仲裁信号(HOLD，HLDA) ：由总线请求主设备来控制该组信号。

- HOLD：总线请求信号，输入信号，高电平有效。

- HLDA：总线保持响应信号，输出信号，有效时，CPU 让出总线。

7）协处理器接口信号(PEREQ，BUSY，ERROR)：控制80386 同80287 或80387 之间的通信。

- PEREQ：来自协处理器的请求信号，输入信号，表示80387 要求80386 控制它们与存储器之间的信息传送。PEREQ 信号是由一个像80387 浮点处理器这样的协处理器输出的，它通知80386 为协处理器取数据字的第一部分，然后协处理器将接管总线并读数据字的其余部分。
- BUSY：协处理器忙，输入信号，低电平有效。BUSY信号由协处理器使用。以避免80386 在协处理器结束当前指令之前又继续下一条指令。
- ERROR：协处理器错误信号，输入信号，低电平有效。如果协处理器设置了ERROR 信号，80386 将执行类型为16 的异常中断。

8）中断信号(INTR，NMI，RESET)：用来引起中断或中止80386 正在执行的指令流。

- INTR：可屏蔽中断请求，输入信号。80386 响应INTR 请求时，完成两个连续的中断响应周期，在整个响应周期，LOCK信号有效。在第二个周期末，D0～D7数据线上送出8位中断类型码，以识别中断源。INTR信号可以由80386的标志寄存器中的IF位屏蔽。
- NMI：非屏蔽中断请求，输入信号。80386对NMI的处理不运行中断响应周期，而是自动产生一个中断类型2。
- RESET：复位信号，输入信号，当RESET 有效时，将中止80386 正在执行的一切操作，并置于一个已知的复位状态。复位期间的80386 的有关引脚的状态见表3-9。

80386有许多VCC 脚，也有许多标为VSS 的地线，这些引脚均被接到PC板合适的电平上。

三、80386内部结构

上图可简化为如下图:

- 总线接口部件 它通过数据总线、地址总线、控制总线来与外部环境联系，包括从存储器中预取指令、读写数据，从I/O端口读写数据，以及其他的控制功能。数据总线和地址总线都是32位的，由于它们是分开的，所以从存储器中存储数据最快也需要两个时钟周期内完成。

- 指令预取部件 IPU：它将存放在存储器中的指令经BIU取到16字节长的预取指令队列中，并向指令译码部件输送指令。
- 指令译码部件 IDU :从IPU中取出指令进行译码分析，然后将其放入IDU中的译码指令队列中，供执行部件使用。（容纳3条以译码的指令）
- 执行部件 EU: 执行部件EU包含算数逻辑单元ALU，8个32位的通用寄存器，一个64位的多位移位加法器，执行数据处理和运算操作
- 存储管理部件(MMU)由分段部件和分页机构组成，实现了从逻辑地址到物理地址的转换，既支持段式存储管理、页式存储管理，也支持段页存储管理。它存储器采用段、页式结构，80386首次将分页机制引入到80X86结构，每页大小为4KB。
- 分段部件 SU: 按指令要求，分段部件SU将指令中的逻辑地址转换成线性地址。
- 分页部件 PU:分页部件PU将分段部件SU产生的线性地址转换成物理地址，每页容量4KB.当系统不使用分页功能时，线性地址就是物理地址。

四、80386的寄存器

80386的寄存器结构 80386微处理器共有7类34个寄存器，通用寄存器组、段寄存器、指令指针和标志寄存器、系统地址寄存器、控制寄存器、调试寄存器、测试寄存器。

(1)通用寄存器组：共有8个32位寄存器，EAX， EBX，ECX，EDX，ESP，EBP，ESI，EDI。它们由8086的16位寄存器扩展而来，它们的低16位与8086使用方法相同。

(2)段寄存器：共有6个16位的段寄存器CS、DS、SS、ES、FS、GS。与这6个段寄存器对应的有6个64位描述符寄存器，它是80X86处理器提供的一种附加的非编程的寄存器，用来装64的段描述符，每当一个段选择符被装入段寄存器是，相应的段描述符就由内存装入到对应的非编程的CPU寄存器。其中CS、DS、SS、ES与8086的段寄存器完全相同，在实地址方式下，使用方法也与8086相同；在虚地址保护方式下，这些寄存器中的值是“段选择符”，需要查全局描述符表（GDT）或者局部描述符表（LDT）来获得段的基地址，再加上偏移地址才能得到线性地址。 FS和GS是新加的附加数据段寄存器，可以由用户将FS、GS定义为其他数据段。

(3)指令指针和标志寄存器：指令指针寄存器EIP，由8086的IP寄存器扩展而来。标志寄存器EFLAGS包含一组状态标志、一个控制标志、一组系统标志，图四定义该寄存器中的标志位。

标志寄存器EFLAGS的低12位与8086的标志寄存器FLAGS一样。IOPL位表示特权标志位，定义当前任务的特权层。NT位表示任务嵌套标志位，当NT位为1时表明当前执行的任务嵌套在另外一个任务中，否则NT位为0。RF位表示重新启动标志位，与调试寄存器一起用于断点和单步操作，RF位为1时表明下一条指令的调试故障将被忽略，不产生中断异常；RF位位0时表示调试故障被接受并产生中断异常。由于调测失败后强迫程序恢复执行；在每条指令成功执行后，RF自动复位。VM位表示虚拟模式标志位，VM位为1时表明80386工作在保护虚拟地址方式。前4个定义从80286开始，后面的2个定义从80386开始存在。另外的三个标志是Pentium以后的CPU才有的。VIF(Virtual interrupt flag)表示虚拟中断标志。当VIF=1时，可以使用虚拟中断，当VIF=0时不能使用虚拟中断。该标志要和下面的VIP和CR4中的VME配合使用。VIP(Virtual interrupt pending flag)表示虚拟中断挂起标志。当VIP=1时，VIF有效，VIP=0时VIF无效。ID(Identification flag)表示鉴别标志。该标志用来只是Pentium CPU是否支持CPUID的指令。
(4) 系统地址寄存器和系统段寄存器：系统地址寄存器有全局描述符表寄存器GDTR、中断描述符表寄存器IDTR。系统段寄存器有局部描述

符表寄存器LDTR和任务寄存器TR。这些寄存器保存相应的描述符表的地址。
(5) 控制寄存器：4个32位的控制寄存器CR0，CR1，CR2，CR3，它们保存全局性的机器状态，其基本定义如图五。从Pentium开始，又增加了一个CR4。下面来简单介绍控制寄存器中的位。

1)CR0的低16位包含了与80286的MSW一致的位定义，保持了和80286的兼容，同时也兼容了从80286开始的两条指令LMSW/SMSW。指令LMSW和SMSW分别用于装入和保存机器状态字信息，可以通过MOV指令对CR0进行读写操作。CR0中各位含义如下：

PE(Protection Enable)保护模式允许位，用来启动CPU进入虚地址保护方式。PE=0表示CPU工作在实地址方式；PE=1表示CPU工作在虚地址保护方式。

MP(Monitor Coprocessor)监控协处理器，MP=1表示协处理器在工作；MP=0表示协处理器未工作。

EM(Emulation)协处理器仿真，当MP=0，EM=1时，表示正在使用软件仿真协处理器工作。

TS(Task Switched)任务转换，每当进行任务转换时，TS=1；任务转换完毕，TS=0。TS=1时不允许协处理器工作。

ET(Extension Type)处理器扩展类型，反映了所扩展的协处理器的类型，ET=0为80287，ET=1为80387。

PG(Paging)页式管理机制使能，PG=1时页式管理机制工作，否则不工作。

NE(Numeric Error)数值异常中断控制，NE=1时，如果运行协处理器指令发生故障，则用异常中断处理，NE=0时，则用外部中断处理。

WP(Write Protect)写保护，当WP=1时，对只读页面进行写操作会产生页故障。

AM(Alignment Mask)对齐标志，AM=1时，允许对齐检查，AM=0时不允许，关于对齐，在EFLAGS的AC标志时介绍过，在80486以后的CPU中，CPU进行对齐检查需要满足三个条件，AC=1、AM=1并且当前特权级为3。

NW(Not Write-through)和CD(Cache Disable)，这两个标志都是用来控制CPU内部的CACHE的，当NW=0且CD=0时，CACHE使能，其它的组合比较复杂。

前4个定义从80286开始，接着的2个定义从80386开始存在, 后面4个是从80486开始定义的。

2)CR1寄存器用来保留给Intel微处理器将来开发使用；CR2寄存器包含一个32位的线性地址，指向发生最后一次也故障的地址，只有在PG=1时，CR2才有效，当页故障处理程序被激活时，压入页故障处理程序堆栈中的错误码提供页故障的状态信息;CR3寄存器中包含页物理目录表的物理基地址，由于每4KB为一页，80386中的页目录表总在页的整数边界上，CR3的低13位总是为0，只有当CR0中的PG=1时，CR3的页目录基地址才有效。
(6) 调试寄存器：共8个排错寄存器DR0~DR7。DR0~DR3可以分别设置4个断点的线性地址，DR4~DR5保留未用，DR6是断点状态寄存器，DR7是断点控制寄存器(包括断点类型、断点长度，断点开放/禁止)。
(7) 测试寄存器：2个32位的测试寄存器TR6和TR7，用于控制转换后援缓冲器中的RAM测试，其中TR6为命令测试寄存器，TR7为测试数据寄存器。

五、80386工作模式

有三种工作模式：实地址模式、保护虚拟地址模式和虚拟8086模式。

常用的WINDOWS,LINUX 就是工作在处理器的保护模式底下.
为了兼容以前在实模式底下工作的软件,80386支持实模式,但是在实模式底下不能支持多任务处理,所以V86模式应运而生.
下面分别介绍几种模式:
1. 实模式
80386工作在时模式底下是 A0--------A19的20根地址线是可用的,寻址空间为1MB,这个时候80386和8086,8088的寻址方式是一样的,即段寄存器内容左移4位作为段地址,在加上段内偏移地址就构成了20位的物理地址,每个段的最大长度是64K,所以实模式底下物理地址的最高为是0XFFFFFH,若超除了就会被丢弃.在这种模式底下,80386不支持优先级,所以程序都可执行特权指令,不支持硬件上多任务的切换,是单操作系统,DOS既运行在实模式底下.定位中断服务子程序,需要中断向量表,中断向量号乘以4得到中断向量号,再在表中查找中断向量,中断向量有4个字节组成,分别是两个字节的段地址和两个字节的偏移地址,就可得到中断服务程序的入口地址. 这种模式底下可以直接对I/O地址空间,数据段,代码段进行读写.这时的80386和8086非常相似,80386就象是一个快速的8086,只不过80386有32位的数据线和32位的通用寄存器而8086是16位的.准确的说是准16位的系统
2 保护模式
在保护模式底下,逻辑地址由段寄存器和偏移地址组成,不过要得到物理地址可不是实模式底下的方法,首先,段寄存器中存放的不是段基址,而是选择子,系统通过选择子来得到真正的段基地址,然后段基地址和偏移地址相加后得到线性地址,这是通过分段机制实现的,然后再通过分页机制把线性地址转化成物理地址.但是分页机制是可选的.在保护模式底下,32条地址线全部有效,最大寻址空间可达4GB.支持多任务处理,使用一条指令或者一个中断就可以在任务内或任务间切换.提供了0---3共4个特权级,操作系统运行在最高级上即0级,应用程序运行低级上,不但实现资源共享,而且实现数据和代码的安全.不再使用中断向量表来实现中断功能,而是通过各种控制描述符来和特权级检查来完成多任务的实现即中断功能.以前那些I/O操作的指令不能对I/O端口进行读写,通过特权级和I/O许可位来进行安全检查.
3 V86模式
估计大家已经猜出来了(其实你在前面也讲了!),不管是实模式还是保护模式都有弊端,V86就充当了这两者之间的桥梁.说白了V86模式就是想利用保护模式的优点来运行8086下的程序.它是在保护模式底下工作的,也称虚拟8086模式.v86模式底下寻址和实模式相同,但20位地址不是真实的物理地址,而是线性地址,寻址空间为1MB,为了使多个虚8086任务不使用同一个位置的1MB地址空间,系统使用了分页机制将不同的V8086任务映射到不同的物理空间上去.每个V86任务认为自己在0-1MB的空间上运行.这样8086下的程序可以在80386的V86模式底下运行,但是它的一部分指令是是受到保护的,如果执行将发生异常,V86模式受到了V86监控程序的控制,v86监控程序和硬件组成了"8086虚拟机"。V86监控程序控制外部界面,中断,I/O,硬件提供最底端的1MB虚拟存储.,在80386中每个V86模式是相对的,这样就充分发挥了处理器的能力