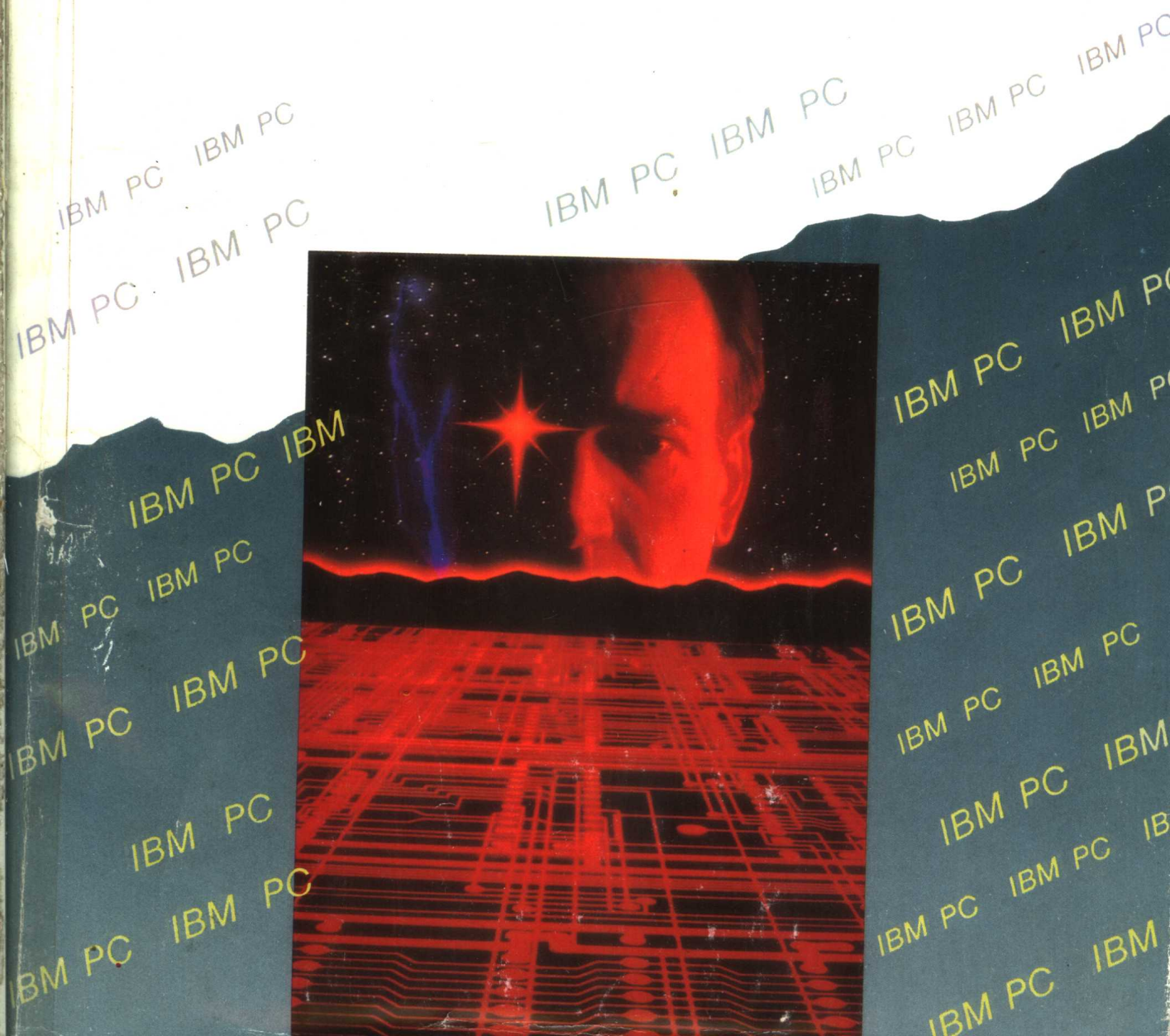


程卫东 主编

IBM PC系列 微机芯片手册

福建科学技术出版社



程卫东 主编

IBM PC系列 微机芯片手册

福建科学技术出版社

(闽)新登字 03 号

IBM PC 系列微机芯片手册

程卫东 主编

*

福建科学技术出版社出版、发行

(福州得贵巷 59 号)

福建省新华书店经销

福建省科发电脑排版服务公司排版

沙县印刷厂印刷

开本 787×1092 毫米 1/16 34.25 印张 2 插页 864 千字

1994 年 9 月第 1 版

1994 年 9 月第 1 次印刷

印数:1—5 200

ISBN 7—5335—0790—2/TN·85

定价:26.80 元

书中如有印装质量问题,可直接向承印厂调换

前 言

自从 1971 年第一个微处理器问世以来,已历时 20 多年。20 多年来,随着 VLSI 工艺技术及 AD 设计自动化技术的进步,微处理器获得了惊人的发展。集成度从最早的 Intel4004 的 2250 个 MOS 晶体管发展到今天的 Pentium 的 320 多万个 MOS 晶体管。工作速度从最早的 3 万次操作/s 急增到今天的 400MIPS。时至今日,微型计算机系统的性能已足以同几年前的大型机、巨型机相比美了。实际上,微型计算机正在逐渐取代大型机和巨型机的市场,一直注重于大型机的 IBM 的衰落也恰恰说明了这一点。

20 几年来,微处理器一直沿着 RISC 结构和 CISC 结构两个分支发展。CISC 结构的微处理器历史较长,从 1971 年的 Intel4004 算起,已历经 5 代的变化。典型的产品有 IntelX86 系列和 Montarala 的 680X0 系列。RISC 结构的微处理器 80 年代中期才推出。迄今,第三代 RISC 微处理器已问世。典型的 RISC 微处理器产品有 IBM 的 PowerPC、Intel 的 i860、MIPS 的 RX000 和 DEC 的 Alpha AXP 系列。尽管 RISC 微处理器的研究起步较晚,但其性能却高出 CISC 微处理器一头。例如,1992 年推出的 RISC 结构的 Alpha 系列的 21066-AA 其主频为 200MHz,指令执行速度为 400MIPS,浮点运算能力为 200MFLOPS。而 1993 年推出的 CISC 结构的 Pentium 其主频仅为 66MHz,指令执行速度为 112MIPS。孰优孰劣,一目了然。

RISC 结构的微处理器一般用于工作站系统,CISC 结构的微处理一般用于 PC 计算机系统。性能卓越的 RISC 微处理器不能取代 CISC 微处理器的原因就在于支持 CISC 微处理器的几百亿美元的软件资源不能在 RISC 微处理器上得到利用。并且,未来的微处理器的发展,RISC 芯片与 CISC 芯片的差别将越来越小。CISC 微处理器将越来越多地采用 RISC 芯片的技术,诸如,超级流水线技术、超标量技术、超长指令字技术等。Intel 公司的 Pentium (P5) 正是因为采用了超级流水线和超标量技术,使其性能大大提高,超过了 SUN 公司的 Supper SPARC 而与 MIPS 公司的 R4000 和 IBM 公司的 PowerPC 相当。可以预料,Intel 公司正在进行的“P6”和“P7”计划,将采用更多的 RISC 微处理器技术,其性能也将大幅度地提高。

到本世纪末,集成电路工艺的光刻技术将达到 $0.12\mu\text{m}$ 。新的半导体材料 GaAs 也可望实用化。微处理器的单片集成度将达到 5000 万到 1 亿个 MOS 晶体管,指令的执行速度将达到 2000MIPS。实用化的神经网络微处理器、光电合一的微处理器、具有强大的多媒体功能的微处理器也可望问世。届时,微处理器的性能将有极大的提高。微型计算机与工作站之间的差别将消失。微型计算机将主导着人们的工作、生活、娱乐。一个不懂计算机的人,将会感到无所适从。

本书详细介绍了 Intel 公司 X86 系列的微处理器, 从 Intel8086 一直到 Intel Pentium, 重点放在微处理器的体系结构、接口时序方面, 以期满足不同档次的读者的需要。之所以介绍 Intel 系列的微处理器, 这是因为 20 几年来, Intel 公司一直称雄微处理器市场, 主导着微处理器的发展方向。X86 系列是 Intel 的 CISC 结构的微处理器的芯片, 广泛用作 PC 计算机系统的 CPU, 其有着极丰富的支持软件。

读者可能会问, 今天的 80386 已经属于淘汰产品了, 为什么还要从 8086/8088 介绍起呢? 这是因为 8088 是 IBM PC/XT 的主处理器, 曾有着广泛的市场和丰富的软件, 并且 X86 的所有的升级产品都具有向上兼容的特性, 所有在 IBM PC/XT 机上开发的软件可以不加修改地在 PC/AT, 以至 Pentium 系统上运行。80286、80386、80486、Pentium 都有一种模拟 8086 运行的方式。更好地了解 8086/8088, 将有助于更好地理解更高层次的微处理器。

本书由程卫东主编, 刘海黎、管宪生、王平、林青、郑云国、王少峰、张峰、李冰、郑秋云、林云峰、王岚参加了本书的编著工作, 何敏、李川、陈晓霞、姚琼、夏雨、张秀、陈冰等为本书的绘图作了大量的工作。由于教学、科研繁忙, 成书仓促, 书中难免有错误, 还望各位读者不吝指教。

程卫东

1994 年 1 月于中国科大

目 录

第一章 Intel 8086 微处理器

1.1	概述	(1)
1.2	基本体系结构	(1)
1.3	寄存器组	(3)
1.4	总线接口	(6)
1.5	寻址方式与存储器组织	(13)
1.6	指令系统	(19)
1.7	中断系统	(26)
1.8	输入/输出	(30)
1.9	8086 的各种运行状态	(30)
1.10	8080 与 8086 的比较	(34)
1.11	特性数据、图表	(36)

第二章 Intel 8087 数值协处理器

2.1	概述	(59)
2.2	基本体系结构	(59)
2.3	引脚介绍	(66)
2.4	指令系统	(69)
2.5	异常出错与处理	(76)
2.6	8087 与主处理器的接口	(77)
2.7	特性数据、图表	(79)

第三章 Intel 80186 微处理器

3.1	概述	(88)
3.2	基本体系结构	(88)
3.3	总线接口	(116)
3.4	指令系统	(122)
3.5	中断系统	(127)
3.6	80186 的各种运行状态	(130)
3.7	80186 与 8086 的比较	(134)
3.8	80186 与 80188 的比较	(136)
3.9	特性数据、图表	(137)

第四章
Intel 80286
微处理器

4.1	概述	(157)
4.2	基本体系结构	(157)
4.3	寄存器组	(158)
4.4	总线接口	(161)
4.5	指令系统	(171)
4.6	实地址方式	(179)
4.7	保护虚地址方式	(182)
4.8	任务转换	(207)
4.9	特性数据、图表	(212)

第五章
Intel 80287
数值协处理器

5.1	概述	(225)
5.2	基本体系结构	(225)
5.3	引脚介绍	(231)
5.4	异常与异常处理	(232)
5.5	指令系统	(234)
5.6	与主处理器的连接	(238)
5.7	80287 与 8087 的比较	(239)
5.8	特性数据、图表	(240)

第六章
Intel 80386
微处理器

6.1	引言	(248)
6.2	基本体系结构	(248)
6.3	寄存器组	(250)
6.4	指令系统	(256)
6.5	寻址方式	(257)
6.6	数据类型	(259)
6.7	存储器空间和 I/O 空间	(260)
6.8	输入/输出	(262)
6.9	中断和异常	(266)
6.10	复位和初始化	(271)
6.11	可测试性	(274)
6.12	调试	(276)
6.13	实地址方式	(281)
6.14	保护方式	(283)
6.15	虚拟 8086 方式	(301)
6.16	总线接口	(306)
6.17	特性数据、图表	(326)

第七章
Intel 80387
数值协处理器

7.1	概述	(349)
7.2	内部结构	(349)
7.3	外部结构	(351)
7.4	接口协议	(352)

第八章
Intel 80486
微处理器

8.1	概述	(357)
8.2	寄存器组	(358)
8.3	指令系统	(373)
8.4	存储器结构	(378)
8.5	I/O 空间	(379)
8.6	寻址方式	(379)
8.7	数据类型	(381)
8.8	中断和异常	(384)
8.9	操作方式	(385)
8.10	存储管理	(386)
8.11	任务管理	(388)
8.12	超高速缓存	(390)
8.13	总线接口	(394)
8.14	总线操作	(404)
8.15	可测试性	(426)
8.16	调试支持	(432)
8.17	特性数据、图表	(435)

第九章
Intel Pentium
微处理器

9.1	概述	(458)
9.2	引脚功能说明	(460)
9.3	超高速缓存	(467)
9.4	总线接口和总线周期	(473)
9.5	数据完整性	(494)
9.6	可测试性和调试	(497)
9.7	探针方式	(511)
9.8	系统管理方式	(524)
9.9	指令系统	(527)
9.10	存储管理	(533)
9.11	异常和中断	(537)
9.12	调试	(538)
9.13	虚拟 8086 方式	(539)

第一章 Intel 8086 微处理器

1.1 概述

Intel 8086 是 Intel 公司于 1978 年推出的第三代微处理器。它是采用 HMOS 工艺技术制成的 40 引脚封装的双列直插式 VLSI 芯片。

在 8086 的设计中,引入了 2 个重要的结构概念:存储器分段和指令流水线。这些概念在以后的 Intel 系列微处理器中一直被沿用与发展着。

8086 由两大基本部件组成:总线接口部件和执行部件。

总线接口部件负责:

- (1) 从内存取指令和存取数据;
- (2) 与外部设备交换数据;
- (3) 指令预取与排队;
- (4) 基本总线控制信号;
- (5) 地址重定位。

执行部件负责:

- (1) 从总线接口部件的指令队列中获取指令,并对其译码;
- (2) 执行指令所要求的各种操作;
- (3) 在要访问存储器操作数时,向 BIU 部件提供操作数的偏移地址。

总线接口部件 (BIU) 和执行部件 (EU) 异步并行的工作,极大地提高了处理器的效率。8086 的寻址空间达 1MB,外部引脚支持单 CPU 系统与多 CPU 系统设计,与 Intel 公司的第二代微处理器产品 8080A 在汇编语言一级向上兼容;与 8087 数值协处理器相结合,可构成具有强大的数学计算和浮点数据处理能力的系统。

8086 有几种不同工作频率的产品,分别为 5MHz 的标准的 8086,8MHz 的 8086-2,10MHz 的 8086-1,4MHz 的 8086-4。这些产品除了工作频率不同外,其它结构完全一样。

1.2 基本体系结构

8086 的结构如图 1-1 所示。由图可知,8086 可分成两大部分:总线接口部件 BIU 和执行部件 EU。

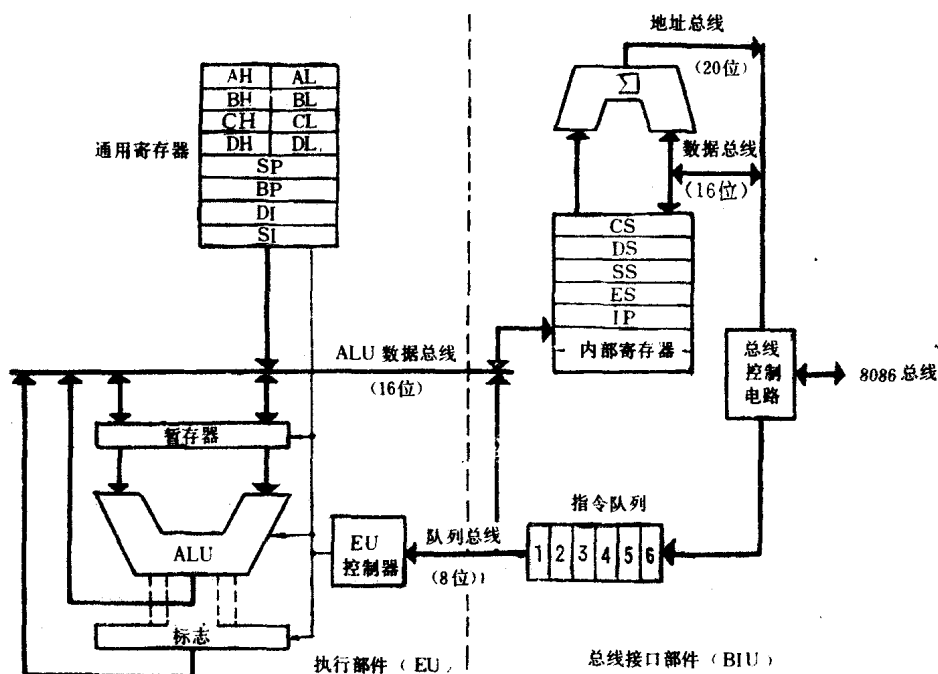


图 1-1 8086 微处理器结构图

1.2.1 总线接口部件

总线接口部件是 8086 与外部的接口，其由总线控制逻辑、6 字节长的指令队列寄存器、4 个段寄存器、指令指针、内部通讯寄存器及地址加法器组成。

总线控制逻辑产生各种总线控制及状态信号。指令队列寄存器是一组先进先出 (FIFO) 寄存器组，用以存放 BIU 预取的指令。4 个段寄存器分别存放当前代码段、数据段、附加段、堆栈段的段基址的高 16 位。指令指针指向当前指令代码在代码段中的位置。地址加法器把 16 位段基址和 16 位偏移量按一定规则相加，形成 20 位物理地址。

BIU 负责处理所有的总线操作。完成取指、指令预取与排队、操作数读写、地址重定位等功能，提供 16 位双向数据总线，20 位地址输出总线，所有的总线控制信号与状态信号。

BIU 利用指令流排队原理构成系统的并行流水线结构。在 EU 部件不需要进行存储器或 I/O 操作时，若指令队列寄存器中有 2 个以上的空闲字节，BIU 自动执行取指总线周期，取得的指令填入指令队列寄存器，直至指令队列只有一个空闲字节或无空闲字节为止。在 EU 部件需要进行存储器或 I/O 操作时，EU 发出命令，BIU 进入存储器或 I/O 总线访问周期。因为 8086 数据总线是 16 位的，所以 BIU 的数据操作一般都是 2 字节的。仅当操作码或操作数存放在奇地址单元时，数据操作才是 1 字节的。

1.2.2 执行部件

执行部件由算术逻辑部件 (ALU)、标志寄存器、通用寄存器、暂存器和控制器组成。

算术逻辑部件 (ALU) 完成各种算术和逻辑运算, 并根据运算结果修改标志寄存器。标志寄存器的一些位用来控制处理器的运行方式, 一些用来表征处理器的运行状态。通用寄存器用来存放操作数、操作数地址、运行结果等。暂寄存器用来存放中间结果。控制器产生 EU 执行各种操作的相应控制信号。

EU 负责解码和执行所有的指令。EU 从 BIU 指令队列寄存器的顶部取得指令, 对它们进行译码, 如果需要操作数的话, 产生操作数的地址并把它传送给 BIU, 要求 BIU 进行相应的总线操作, 并进行指令所要求的操作。在指令执行期间, EU 测试状态寄存器并根据指令执行结果修正相应的位。如果指令队列已空, 则 EU 等待 BIU 取下一个指令字节并把它移到指令队列的顶部。由此可知, EU 本身并不直接与存储器和 I/O 打交道, 而是通过 BIU 间接的来进行。

当 EU 执行一条转移指令时, 它把控制转移到新的地址单元, BIU 清除指令队列寄存器, 从新的地址单元取指令填充指令队列。

1.3 寄存器组

8086 内部有 14 个用户可访问的 16 位寄存器。这些寄存器按其性质与用途可分成四组: 通用寄存器组、段寄存器组、指令指针和标志寄存器。

这些寄存器用来提供参加运算的操作数、控制指令的执行、处理内部寻址等。

1.3.1 通用寄存器组

通用寄存器组包括 8 个 16 位寄存器, 它们分别是 AX、BX、CX、DX、BP、BS、SI、DI。这些寄存器位于 EU 部件内, 按其用途可进一步分成数据寄存器组和地址指针与变址寄存器组。

数据寄存器组包括 AX、BX、CX、DX 寄存器。每个寄存器既可以作为 1 个 16 位寄存器使用, 也可以作为 2 个 8 位寄存器使用。作为 8 位寄存器时, 分别为 AH、AL、BH、BL、CH、CL、DH、DL 寄存器。除 AH 寄存器外, AL~DL 寄存器分别对应 8080A 的 A~E 寄存器。一般而言, 数据寄存器可作为源操作数、目的操作数、源操作数的地址、目的操作数的地址。但每个寄存器都有其特殊的用途。AX 寄存器专门用以存放所有 I/O 操作、字符串操作、算术逻辑运算的结果, 故其称之为累加器。BX 寄存器在基址寻址方式中指向一个数据块的基地址, 故称之为基址寄存器。CX 寄存器在循环、移位、字符串操作中作为计数器使用, 故称之为计数器。DX 寄存器在乘、除运算中作为辅助累加器使用, 故称之为数据寄存器。

地址指针与变址寄存器组包括 BP、SP、SI、DI 寄存器。这组寄存器不能像数据寄存器组的寄存器那样可以拆成 2 个 8 位寄存器使用, 只能作为 16 位寄存器使用。这 4 个 16 位寄存器除了能存放 16 位数据, 作累加器使用外, 还具有各自不同的功能。SP 寄存器在堆栈操作中指向栈顶, 故称之为栈指针寄存器。BP 寄存器在间接寻址中用作基址指针, 故称之为基址指针寄存器。SI 寄存器在字符串操作及基址变址寻址中用作源变址寄存器, 故称之为源变址寄存器。DI 寄存器在类似的操作中用作目的变址寄存器, 故称之为目的变址寄存器。

通用寄存器的功能如表 1—1 所示。

表 1—1 通用寄存器功能一览表

寄存器名称	用 法	隐 含 属 性
AX、AL	在算术运算中，用作累加器	加减不隐含， 乘除隐含
	在输入/输出指令中用作数据寄存器	隐 含
AH	在 LAHF 指令中作目标寄存器	隐 含
AL	在 BCD 码及 ASCII 码运算指令中作累加器	隐 含
	在 XLAT 指令中作累加器	隐 含
BX	在加减等运算中用作累加器或数据寄存器	不隐含
	在寄存器间接寻址时用作地址寄存器	不隐含
	在基址寻址时用作基址寄存器	不隐含
	在 XLAT 指令中用作基址寄存器	隐 含
CX	在一些算术运算中用作累加器或数据寄存器	不隐含
	在循环移位及字符串操作中用作计数器	隐 含
CL	在移位及循环移位指令中作位数计数器。指令执行后，CL 内容不变	隐 含
DX	在间接寻址的输入/输出指令中用作地址寄存器	不隐含
	在乘除法指令中用作辅助累加器（当乘积或被除数为 32 位，存放高 16 位）	隐 含
	在加、减法指令中用作累加器或数据寄存器	不隐含
BP	在加、减法指令中用作累加器或数据寄存器	不隐含
	在寄存器间接寻址时用作地址寄存器	不隐含
	在基址（或基址变址寻址）时用作基址寄存器	不隐含
SP	在堆栈操作中用作堆栈指针	隐 含
SI	在字符串指令中用作源变址寄存器	隐 含
	在算术指令中用作累加器或数据寄存器	不隐含
	在寄存器间接寻址中用作地址寄存器	不隐含
	在变址（基址变址）寻址中用作变址寄存器	不隐含
DI	在算术运算中用作累加器或数据寄存器	不隐含
	在字符串指令中用作目的变址寄存器	隐 含
	在寄存器间接寻址中用作地址寄存器	不隐含
	在变址（基址变址）寻址中用作变址寄存器	不隐含

1.3.2 段寄存器组

8086 首先在微处理器中引入存储器分段管理的概念，8086 外部有 20 根地址线，寻址空间达 1MB，但 8086 内部的指令指针及其寄存器都是 16 位的，最大寻址范围只有 64KB。为了把 8086 的寻址范围扩大到 1MB，在 8086 的设计中引入了段寄存器的概念。8086 在进行存储器访问时，存储器的物理地址等于段寄存器内容左移 4 位加上 16 位偏移量。这实际上等于把 1MB 的物理存储器划分成若干个逻辑段，每段 64KB。段的起始地址是 16 的整数倍。通过把段寄存器置入不同的段起始地址，就可以随意访问 1MB 存储器中任意的 64KB 空间。

段寄存器组包括 CS、DS、ES、SS 四个 16 位寄存器。CS 寄存器称之为代码段寄存器，用以存放指令代码段的段基址（物理地址/16）。SS 寄存器称之为堆栈段寄存器，指向堆栈段的段基址。DS 寄存器称之为数据段寄存器，用以存放数据段的段基址。ES 寄存器称之为附加段寄存器，这是因为 ES 用以存放附加数据段段基址的缘故。

8086 最多可同时使用 4 个段寄存器，但这并不是必要的。对 8086 的运行来说，其至少要寻址一个段。

8086 段寄存器功能如表 1—2 所示。

表 1—2 段寄存器功能一览表

段寄存器名	用 法	隐含属性
CS	取指令时提供段基址	隐 含
	取数据或变量时提供段基址	不隐含
	串操作时为源串提供段基址	不隐含
	在基址（或基址变址）寻址中，由 BP 提供基址时，可作为段寄存器	不隐含
DS	取数据或变量时作为段寄存器	对一些寻址方式隐含，另一些寻址方式不隐含
	串操作时为源串提供段基址	隐 含
SS	堆栈操作时提供段基址	隐 含
	取数据或变量时提供段基址	不隐含
	串操作时为源串提供段基址	不隐含
	在基址（或基址变址）寻址中，由 BP 提供基址时，可作为段寄存器	不隐含
ES	取数据或变量时可作为段寄存器	一般不隐含
	串操作中可为源串，目的串提供段基址	源串不隐含 目的串隐含
	在基址（或基址变址）寻址中，由 BP 提供基址时，可作为段寄存器	不隐含

1.3.3 指令指针寄存器

8086 的指令指针寄存器 IP 是一个 16 位寄存器。在程序执行期间，BIU 自动修正 IP，使其始终指向下一条要取的指令。因此在程序顺序执行期间，IP 存放 BIU 将要取的指令的偏移量；当程序发生可返回转移时，8086 先自动调整 IP 的内容，使 IP 指向返回后第一条要执行的指令，然后把 IP 压入栈顶。程序不能直接访问 IP，但可以通过某些指令间接地修改 IP。

1.3.4 标志寄存器

8086 的标志寄存器是个 16 位寄存器。其中只有 9 个标志位有效。它们用于存放 8086 的运行状态信息（状态标志）和控制 8086 运行方式的控制信息（控制标志）。8086 的标志寄存器格式如图 1—2 所示。

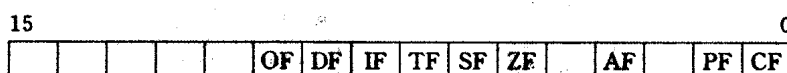


图 1—2 8086 的标志寄存器

状态标志一般在 CPU 执行某个操作后设置，表示 CPU 执行完该操作后的状态。各标志位的含意如下：

(1) 进位标志位 (CF) 表示指令执行是否在最高位产生一个进位或借位。当在一条算术指令执行期间, 最高位上产生了借位或进位, 则 $CF=1$, 否则 $CF=0$ 。

(2) 辅助进位标志位 (AF) 表示指令执行是否在低 4 位 (8 位数据运算) 或低 8 位 (16 位数据运算) 上产生了一个进位或借位, 该标志用于十进制算术运算指令。若有进位或借位产生, 则 $AF=1$, 否则 $AF=0$ 。

(3) 溢出标志位 (OF) 表示指令执行的结果是否产生了超出范围的带符号的结果。若 $OF=1$, 则表示发生了有效数字的丢失。

(4) 零标志位 (ZF) 表示指令执行是否产生了一个为 0 的结果。当指令执行结果为 0 时, $ZF=1$, 否则 $ZF=0$ 。

(5) 符号标志位 (SF) 表示指令执行结果的符号。 $SF=1$, 操作结果为负; $SF=0$, 操作结果为正。

(6) 奇偶标志位 (PF) 表示运算结果中 1 的个数的奇偶性。若运算结果中含偶数个 1, 则 $PF=1$, 否则为 0。该标志位用于数据传送指令的奇偶检验。

控制标志一般在处理器执行某个操作前设置。当处理器执行相应的指令时, 就完成所要求的操作。各标志位的含意为:

(1) 方向标志位 (DF) 该位控制串操作指令的字符串操作方向。 $DF=1$, 表示字符串操作中地址指针自动增量。 $DF=0$ 则相反。

(2) 中断允许标志位 (IF) 控制处理器是否对外部可屏蔽中断请求进行响应。 $IF=1$, 允许处理器响应外部可屏蔽中断请求。 $IF=0$, 则屏蔽外部可屏蔽中断请求。

(3) 单步标志位 (TF) 控制处理器是否进入单步工作方式。 $TF=1$, 处理器进入单步工作方式, 每执行完一条指令就自动产生一次内部中断, 使程序员可以逐步检查程序执行情况。

1.4 总线接口

1.4.1 引脚介绍

8086 的外部封装形式为 40 引脚的双列直插式结构, 其外部封装结构如图 1—3 所示。

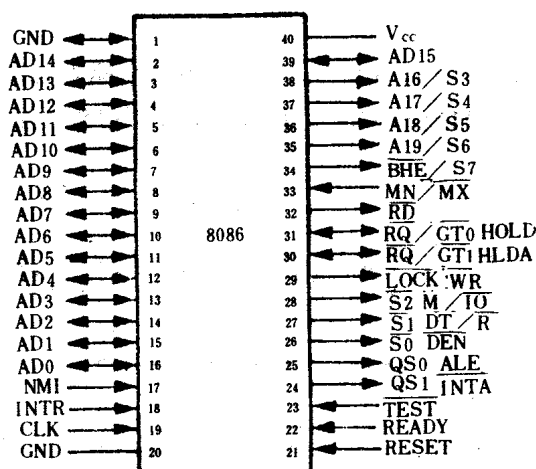


图 1—3 8086 的外部封装结构图

8086 通过这些引脚完成地址及控制信号的传送、数据的存取、接收外部中断请求、中断应答、与其它总线主控设备进行局部总线控制权切换等功能。为了减少芯片的外部引脚, 8086 的设计采用了引脚多路信号分时复用的方法。8086 的各引脚的功能如下:

(1) $AD_0 \sim AD_{15}$ 地址/数据总线, 输入/输出/三态。

$AD_0 \sim AD_{15}$ 为分时复用的地址/数据总线。在一个总线周期的 T_1 时间内, 处理器在这些线上输出存储器或 I/O 口的地址。在其它时间内, 则作为收发数据的数据线。在第一个中断应答周期和局部总线“保持应答”期间呈高阻状态。

(2) $A_{16}/S_3 \sim A_{19}/S_6$ 地址/状态线, 输出/三态。

在总线周期的 T_1 时间内, 对于存储器访问, 其提供 20 根地址线的最高 4 位; 对于 I/O 端口访问, 其保持为 0; 在总线周期的其它时间内, 输出 8086 的状态信号。作为状态线使用时, 表示的意义如表 1-3 所示。

表 1-3 状态信号线编码

状态信号	取值	含义
(S_4, S_5)	(0, 0)	当前所用的段寄存器为 ES
	(0, 1)	当前所用的段寄存器为 SS
	(1, 0)	(1) 当前所用的段寄存器为 CS (2) 对 I/O 端口寻址 (3) 对中断向量寻址
	(1, 1)	当前所用的段寄存器为 DS
S_6	0	标志寄存器中的 IF 位清零 (禁止 CPU 响应外部可屏蔽中断)
	1	标志寄存器中的 IF 位置位 (允许 CPU 响应外部可屏蔽中断)
S_7	0	该位为 0, 指示 8086 在使用总线

当 8086 处于局部总线保持应答状态时, 这些引脚置成高阻状态。

(3) $\overline{BHE}/S_7$ 高位字节允许/状态线, 输出/三态。在总线周期的 T_1 时间内, 其输出 $\overline{BHE}$ 信号, 若为低电平, 则表明可以在数据总线 $D_8 \sim D_{15}$ 上传送一个字节的数据。在总线周期的其它时间内, 输出 S_7 状态信号。在 8086 处于局部总线“保持应答”状态时, 其浮空至高阻状态。 $\overline{BHE}$ 和 A_0 的编码表如表 1-4 所示。

表 1-4 $\overline{BHE}$ 、 A_0 线的编码

$\overline{BHE}$	A_0	数据传送性质
0	0	在 $AD_0 \sim AD_{15}$ 上进行 16 位字传送
0	1	在数据总线高 8 位 $AD_8 \sim AD_{15}$ 上和奇地址间进行字节传送
1	0	在数据总线低 8 位 $AD_0 \sim AD_7$ 上和偶地址间进行字节传送
1	1	不用

(4) $\overline{RD}$ 读选通信号线, 输出/三态。当 $\overline{RD}$ 线为低电平时, 表示处理器正在进行读存储器或 I/O 端口的操作。具体是选中存储器单元还是 I/O 端口, 由 $M/\overline{IO}$ 引脚的输出信号决定。当处理器处于局部总线“保持应答”状态时, 其浮空至高阻状态。

(5) READY 准备好信号线, 输入。当 Ready 信号为高电平时, 表示存储器或外部设备准备就绪, 处理器可以与之交换数据。当 Ready 信号为低电平时, 处理器自动插入一个或几

个等待周期 T_w ，直至 Ready 线出现高电平为止。

(6) INTR 可屏蔽中断请求线，输入。当 INTR 线为高电平（电平触发型）时，表示外部设备向处理器提出了中断请求。处理器在每条指令的最后一个时钟周期对 INTR 信号进行检测，以便决定是否进入中断应答周期。8086 响应中断时，从中断接口电路读取中断向量类型码，并根据它在存储器中的中断向量表找到相应的中断服务程序入口地址。这根输入线上的中断请求信号可由标志寄存器中的 IF 位加以禁止。

(7) $\overline{\text{TEST}}$ 测试信号线，输入。当处理器执行 WAIT 指令时，每 5 个时钟周期就对 $\overline{\text{TEST}}$ 线进行一次检测。如果 $\overline{\text{TEST}}$ 线为高电平，处理器继续等待，并重复检测过程。若 $\overline{\text{TEST}}$ 为低电平，则处理器接着执行下一条指令。 $\overline{\text{TEST}}$ 信号在每个时钟周期的上升沿由内部时钟同步。 $\overline{\text{TEST}}$ 信号线主要用于实现 8086 与 INTEL 8087 数值协处理器间的同步。

(8) NMI 非可屏蔽中断请求线，输入。NMI 请求线上一个从低到高的电平变化（边沿触发型）使得处理器在当前指令结束后立即进入一个类型 2 的中断。NMI 不能通过软件改写标志寄存器的 IF 位加以禁止。

(9) RESET 复位信号线，输入。当 RESET 信号线为高电平时，处理器立即终止当前正在进行的操作，并把标志寄存器、段寄存器、指令指针和指令队列寄存器复位到起始状态。为了处理器可靠地复位，RESET 信号应保持高电平至少 4 个时钟周期。当 RESET 变成低电平后，处理器就开始执行再启动过程。

(10) $\text{MN}/\overline{\text{MX}}$ 最小/最大工作方式选择线，输入。 $\text{MN}/\overline{\text{MX}}$ 线输入电平的高低决定了处理器的工作模式。当 $\text{MN}/\overline{\text{MX}}$ 线接 +5V 电源时，8086 工作于最小模式，亦即 8086 组成一个单处理器系统。所有的总线控制信号均由处理器本身产生。当 $\text{MN}/\overline{\text{MX}}$ 接低电平时，8086 工作于最大模式，亦即构成一个多处理器系统，8288 总线控制器译码 8086 的状态线 $S_0 \sim S_2$ ，产生总线控制信号。

(11) CLK 时钟信号线，输入。时钟输入线提供处理器及其所构成的系统的基本定时信号。定时脉冲是非对称的，占空比为 33%。

(12) V_{cc} 工作电源线，输入。 V_{cc} 线为 8086 微处理器提供基本的工作电源，对电源的要求为 $+5 \pm 10\%V$ 。

(13) GND 地线

(14) $\text{M}/\overline{\text{IO}}$ 存储器/I/O 端口选择线，输出/三态。 $\text{M}/\overline{\text{IO}}$ 线是 8086 工作于最小模式时，用来选择当前总线周期访问的是存储器还是 I/O 端口。当 $\text{M}/\overline{\text{IO}}$ 为高电平时，表示当前访问的是存储器；为低电平时是 I/O 端口。当 8086 处于局部总线“保持应答”状态时， $\text{M}/\overline{\text{IO}}$ 线浮空至高阻状态。

(15) $\overline{\text{WR}}$ 写选通信号线，输出/三态。8086 工作于最小模式下的写选通信号。当 $\overline{\text{WR}}$ 为低电平时，表示处理器当前正在进行写操作总线周期。与 $\text{M}/\overline{\text{IO}}$ 相配合，处理器可以把数据写入存储器或 I/O 端口。当 8086 处于局部总线“保持应答”状态时， $\overline{\text{WR}}$ 线浮空至高阻状态。

(16) $\overline{\text{INTA}}$ 中断应答信号线，输出。8086 工作于最小模式下的中断应答信号输出线。在每个中断应答周期的 T_2 、 T_3 和 T_w 期间，其输出为低电平。一般用作读中断向量的选通信号。

(17) ALE 地址锁存允许信号线，输出。8086 工作于最小模式下的地址锁存允许信号线。当 ALE 为高电平时，外部接口电路利用它的后沿（下降沿）把地址/数据总线上地址信号锁入地址锁存器，从而腾出地址/数据总线来存取数据。

(18) $\text{DT}/\overline{\text{R}}$ 数据收发选择信号线，输出/三态。8086 工作于最小模式下的数据收发选择

信号线。用它来控制总线收发器 8286/8287 的数据流向。当 $\overline{DT}/\overline{R}=1$ 时,处理器发送数据;当 $\overline{DT}/\overline{R}=0$ 时,处理器接收数据。当 8086 处于局部总线“保持应答”期间,其浮空至高阻状态。

(19) $\overline{DEN}$ 数据允许信号线,输出/三态。8086 工作于最小模式下的数据允许信号线。用它作为总线收发器 8286/8287 的定时选通信号。在写总线周期内,当 $\overline{DEN}$ 有效时,表示数据总线上的数据有效。在读总线周期内,当 $\overline{DEN}$ 有效时,表示处理器已准备好接收数据。当 8086 处于局部总线“保持应答”期间,其浮空至高阻状态。

(20) $\overline{HOLD}$ 总线保持请求信号线,输入。8086 工作于最小模式下的总线保持请求输入信号线。当 $\overline{HOLD}$ 为高电平时,表示系统中的其它总线主控设备请求 8086 放弃总线控制权,以使其它总线主控设备使用总线。

(21) $\overline{HLDA}$ 总线保持应答信号线,输出。8086 工作于最小模式下的总线保持应答输出信号线。当 $\overline{HLDA}$ 为高电平时,表示 8086 已让出总线控制权,其它总线主控设备可以使用总线。当 $\overline{HOLD}$ 变成低电平时, $\overline{HLDA}$ 也变成低电平,禁止其它总线主控设备使用总线。

$\overline{S_0}, \overline{S_1}, \overline{S_2}$: 总线周期状态信号线,输出/三态。

8086 工作于最大模式下的总线周期状态输出信号线。总线控制器 8288 接收这些状态信号,并根据这些信号产生相应的控制信号。 $\overline{S_0}, \overline{S_1}, \overline{S_2}$ 的状态编码意义如表 1—5 所示。

表 1—5 $\overline{S_0}, \overline{S_1}, \overline{S_2}$ 的状态线编码

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	意 义
0	0	0	INTA 中断应答
0	0	1	IOR I/O 端口读
0	1	0	IOW I/O 端口写
0	1	1	HALT CPU 已执行 HALT 指令,处于暂停状态
1	0	0	IFETCH 处理器取指令代码字节
1	0	1	MEMR 存储器读
1	1	0	MEMW 存储器写
1	1	1	NONE 系统总线处于无源状态

(22) $\overline{RQ}/\overline{GT_0}, \overline{RQ}/\overline{GT_1}$ 请求/授予信号线,输入/输出/三态。8086 工作于最大模式下的请求/授予信号线。其与 8086 最小模式下的 $\overline{HOLD}$ 、 $\overline{HLDA}$ 信号的功能相似。不同的是 $\overline{HOLD}$ 和 $\overline{HLDA}$ 为高电平有效,且为单向的。而 $\overline{RQ}/\overline{GT_0}, \overline{RQ}/\overline{GT_1}$ 都是低电平有效,且为双向的。 $\overline{RQ}/\overline{GT_0}$ 与 $\overline{RQ}/\overline{GT_1}$ 信号线的功能一样,只不过 $\overline{RQ}/\overline{GT_0}$ 的优先级更高。请求/授予信号的时序如下:

①请求。其它总线主控设备利用 $\overline{RQ}/\overline{GT_0}$ 。(或 $\overline{RQ}/\overline{GT_1}$) 线向 8086 发一个时钟宽度的请求脉冲(低电平有效);

②检测。8086 在每个时钟的上升沿对 $\overline{RQ}/\overline{GT_0}, \overline{RQ}/\overline{GT_1}$ 线采样;

③授予。在 8086 检测到请求信号后,在当前总线周期的 T_1 状态或空闲状态 T_1 的下降沿,在同一条信号线上送出一个时钟宽度的授予脉冲信号,通知其它总线主控设备,从下一个时钟周期开始,可以使用系统总线。在交出总线控制权后,处理器处于保持状态;

④释放。其它总线主控设备使用完总线后,再通过 $\overline{RQ}/\overline{GT_0}$ 。(或 $\overline{RQ}/\overline{GT_1}$) 线发出一个时钟宽的释放脉冲,表示其已放弃总线,处理器可以重新使用总线。

(23) $\overline{LOCK}$ 锁定信号线,输出/三态。8086 工作于最大模式下的锁定信号输出线。当 8086 执行带 $\overline{LOCK}$ 前缀的指令时, $\overline{LOCK}$ 线变为低电平时,处理器对总线进行封锁,不响应