修订本前言

汇编语言是计算机能提供给用户的最快而又最有效的程序设计语言，它能充分发挥和利用计算机硬件特性，适合于开发对时空效率要求较高、与计算机硬件密切相关的高性能软件的开发，可以实现高级语言难以胜任甚至无法完成的任务。

“汇编语言程序设计”是高等院校计算机软、硬件及应用专业学生必修的核心课程之一。它是学习计算机原理、操作系统、编译原理等其他核心课程的必要先修课。掌握汇编语言知识，对于掌握程序设计技术、加深对计算机系统的理解和掌握程序调试技术都具有重要作用。

本书是在 2003 年出版的《汇编语言程序设计》的基础上重新改编、修订而成的。全书共分 11 章，以 Intel 8086/8088CPU 指令系统与 Microsoft 宏汇编 MASM 5.0 为背景，分别介绍了汇编语言程序设计的基本知识；源程序格式、程序的汇编与连接及程序的调试方法；8086/8088 指令系统；常用伪指令；循环、分支和子程序等基本程序结构及程序设计的基本方法和技术；宏汇编的使用；以中断为主的输入/输出程序设计方法；多模块程序设计等。本书在此次修订时，对原书中的程序实例进行了调整，又增加了大量实例。所有实例都是精心设计并经过上机验证，对学生学习汇编语言均具有一定的指导作用。每章后均附有习题，以便读者复习和检查学习效果。

本书可作为高等院校计算机及相关专业的教材。本教材计划讲授 60 学时左右。

本书适用于初学者使用，读者只要具有数制方面的知识，即可以通过本书的学习，系统掌握汇编语言程序设计的基本方法和技术，如果具有高级语言程序设计的基础，将更有助于对部分内容的深入理解。

在本书修订过程中，天津工业大学计算机技术与自动化学院李兰友教授对本书的编写与修订提供了热情的帮助，并提出了很多宝贵意见，特在此表示感谢。

尽管本书对内容进行了补充和修订，书中难免还有不妥之处，欢迎读者对书中的错误与不妥之处提出批评并给予指正。

编　　者

2007 年 7 月

目　录

第 1 章　汇编语言概述

本章要点：

- ☑ 汇编语言特点
- ☑ 数的二进制补码表示
- ☑ 汇编语言的开发过程及编程工具及环境

1.1　汇编语言简介

程序设计语言通常分为机器语言、汇编语言和高级语言 3 类。其中，机器语言与汇编语言与计算机硬件密切相关，统称为低级语言。

计算机只能识别由 0 和 1 编码的二进制代码。这种二进制代码集合称为该计算机的机器语言，它是计算机唯一能够直接识别并接受的语言。机器语言与计算机的 CPU、内存管理机制和 I/O 机制等有着十分密切的关系，因此，使用机器语言设计程序能够最大限度地利用和发挥计算机的硬件功能和优势。然而，使用机器语言编写的程序可读性差，不易调试和维护，可移植性差，程序的开发效率低。

汇编语言建立在机器语言之上，将机器指令符号化，所以比机器语言直观。然而，汇编语言仅仅是机器语言的符号化，每条汇编语言指令均对应唯一的机器指令，因此，汇编语言同样具有与计算机硬件的密切相关性。正是依赖具体计算机硬件的这一特性，决定了汇编语言是计算机能够提供用户最快而又最为有效的语言，同时，它又是能够利用计算机所有硬件特性且能直接控制硬件的唯一语言。使用汇编语言编写程序可以充分发挥计算机硬件的功能，且还具有占用存储空间少、运行速度快、执行代码短等优点。正是因为汇编语言具有这样的优点，那些需要对计算机硬件进行控制或者对运行时间和效率有要求的应用软件或系统软件，都是用汇编语言编制的。也是因为汇编语言与硬件紧密相关，它可以设计出高级语言无法实现的程序。另外，高级语言为获得汇编语言的优势，通常也增加了调用汇编语言程序的接口或与汇编语言混合编程的能力。

1.2　数制基础

计算机为了方便存储及计算的物理实现，采用了二进制数。然而，二进制数书写冗长，不便于人们阅读、书写和记忆。为此就引入了十六进制数。十六进制数简短、易读，与二进制数之间的转换容易。

二进制数的基数是 2，只有 0 和 1 两个数码，并遵循逢 2 进 1 的规则。

十六进制数的基数是 16，共有 16 个数码，它们是 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F（或 0，1，2，3，4，5，6，7，8，9，a，b，c，d，e，f）。其中，A 或 a 表示十进制数的 10，依此类推。十六进制数遵循逢 16 进 1 的规则。为了与标识符区分，若十六进制数以字母打头，则前面补 0。例如，十进制数的 15，相应的十六进制数应表示为 0FH。

在汇编语言中，为了区分各种不同进制数，在数的结尾以一个字母表示。其中，十进制数书写时以 D 或 d 结尾，也可省略不写；二进制数以 B 或 b 结尾；十六进制数以 H 或 h 结尾。

1.2.1　数据组织

1．位

在计算机中，数据的最小单位是一个二进制位。

2．字节

字节是 8086/8088CPU 可寻址的最小数据单位。一个字节是 8 位。一个字节中位的编号从右到左依次是 0～7，如图 1-1 所示。

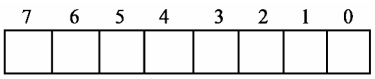

图 1-1　字节的位编号

其中：第 0 位称为最低位或最低有效位，第 7 位称为最高位或最高有效位，一个字节可以表示 2^8 个不同值。

3．字

通常，一个字是 16 位。一个字中位的编号从右到左依次是 0～15，如图 1-2 所示。

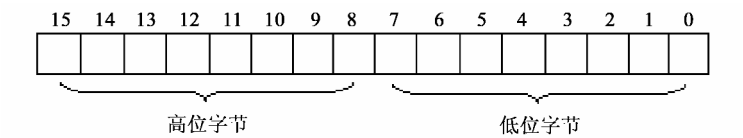

图 1-2　字的位编号

其中：第 0 位称为最低位或最低有效位，第 15 位称为最高位或最高有效位；0～7 位称为低位字节，8～15 位称为高位字节；一个字可以表示 2^{16} 个不同值。

4．双字

通常，一个双字是 32 位。一个双字中位的编号从右到左依次是 0～31，如图 1-3 所示。

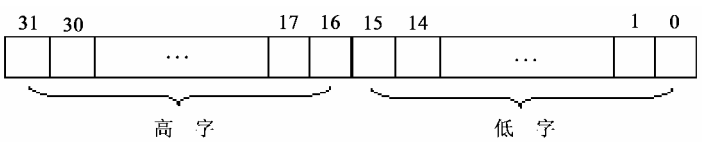

图 1-3　双字的位编号

其中：

第 0 位称为最低位或最低有效位，第 31 位称为最高位或最高有效位；0～15 位称为低字，16～31 位称为高字；一个双字可以表示 2^{32} 个不同值。

1.2.2　数的补码表示

计算机中的数是用二进制表示的，数的符号也是用二进制表示的。一般地，数的符号用最高有效位来表示，正数用 0 表示，负数用 1 表示。这样的数称为带符号数。例如，对于 8 位数，91H 为负数，0FH 为正数。若要处理的数均为正数，此时保留符号位已无意义，最高有效位作为数值处理，这样的数称为无符号整数。例如，对于 8 位数，0FFH 为正数，是十进制的 255。8086/8088CPU 采用二进制补码表示整数。以下介绍补码表示法。

补码表示的规则如下：

- 正数的补码是其本身；
- 负数的补码是先取该负数的绝对值的补码表示，再对该数进行按位取反，末位加 1 的操作。n 位二进制补码数可以表示的带符号数的范围为 -2^{n-1}～$2^{n-1}-1$。n 位二进制补码数可以表示的无符号数的范围为 0～$2^{n-1}-1$。
- 8 位二进制数可以表示的带符号数的范围为－128～127；
- 8 位二进制数可以表示的无符号数的范围为 0～255；
- 16 位二进制数可以表示的带符号数的范围为－32768～32767；
- 16 位二进制数可以表示的无符号数的范围为 0～65535。

【例 1-1】　Pentium 处理器处理的 16 位二进制整数用补码表示，其数值范围是_____。

A．-2^{15+1}～$2^{15}-1$　　B．-2^{15}～$2^{15}-1$　　C．$-2^{16}+1$～$2^{16}-1$　　D．-2^{16}～$2^{16}-1$

答案：B

【例 1-2】　对于 8 位数来说，计算下列各数的补码。

（1）–3　　（2）3　　（3）0

【分析】

（1）计算–3 的补码，须先计算 3 的补码，为 00000011B，然后对 00000011B 按位取反，为 11111100B，再进行末位加 1 的操作，得到 11111101B。即$[-3]_{补}$=11111101B。

（2）3 的补码为$[3]_{补}$=00000011B。

（3）0 的补码为$[0]_{补}$=00000000B。

说明：

① “按位取反，末位加 1”这样的操作称为求补。

② 对–3 的补码进行“按位取反，末位加 1”的操作，即进行求补操作，得到 3 的补码。所以，实质上，求补操作即求相反数。

【例 1-3】　计算下列各数的相反数。

（1）3451H　（2）7FFFH　（3）8000H

【分析】　求相反数即进行求补操作，对二进制数据进行按位取反，末位加 1。

（1）3451H 的二进制形式为 0011 0100 0101 0001B，按位取反得到 1100 1011 1010 1110B，将此数末位加 1 得到 1100 1011 1010 1111B。则 3451H 的相反数为 0CBAFH。

（2）7FFFH（即十进制数+32767）的二进制形式为 0111 1111 1111 1111B，按位取反得到

1000 0000 0000 0000B，将此数末位加 1 得到 1000 0000 0000 0001B。则 7FFFH 的相反数为 8001H（即十进制数–32767）。

（3）8000H（即十进制数–32768）的二进制形式为 1000 0000 0000 0000B，按位取反得到 0111 1111 1111 1111B，将此数末位加 1 得到 1000 0000 0000 0000B。对 8000H 取相反数后仍得到 8000H。–32768 的相反数仍是–32768 当然是不对的，这是因为+32768 超出了 16 位带符号数的表示范围，出现溢出的缘故。

1.2.3　字符的 ASCII 码表示

除了数值外，计算机处理的信息还有字符或字符串，例如，从键盘输入的信息就是以字符方式输入输出的。因此，字符必须按特定规则以二进制方式编码后才能在计算机中表示，常用的编码方式是美国标准信息交换码 ASCII 码（American Standard Code for Information Interchange）。

ASCII 码字符集中的前 128 个 ASCII 码为标准 ASCII 字符，采用一个字节表示字符， 其第 7 位为 0，其余 6 位为字符的 ASCII 码值。标准 ASCII 码字符集见附录 A。

【例 1-4】　查表得到下列各字符的 ASCII 码值。

（1）A　　（2）a　　（3）5

（4）+　　（5）SP（空格）　　（6）CR（回车）

【分析】

（1）A 的 ASCII 码是 41H。　　（2）a 的 ASCII 码是 61H。

（3）5 的 ASCII 码是 35H。　　（4）+的 ASCII 码是 2BH。

（5）SP（空格）的 ASCII 码是 20H。　　（6）CR（回车）的 ASCII 码是 0DH。

【例 1-5】　关于 ASCII 码字符集中的字符，下面叙述中正确的是_____。

A．ASCII 码字符集共有 128 个不同的字符

B．每个字符都是可打印（或显示）的

C．每个字符在 PC 机键盘上都有一键与之对应

D．ASCII 码字符集中大小写英文字母的编码相同

答案：A

1.2.4　BCD 码

BCD 码是一种将十进制数用二进制编码的码制。它是用 4 位二进制数表示一个十进制数码。十进制数码所对应的 BCD 码如表 1-1 所示。

表 1-1　BCD 码

十进制数码	0	1	2	3	4	5	6	7	8	9
BCD 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

表示十进制数的 BCD 码有下列 2 种格式。

1．压缩 BCD 码

压缩 BCD 码是用 4 个二进制位表示一个十进制位，整个十进制数形式为一个顺序的以 4 位为一组的数串。

【例 1-6】　将下列各数用压缩 BCD 码表示。

（1）23D　　（2）5719D

【分析】

（1）23D 用压缩 BCD 码表示为 0010 0011。

（2）5719D 用压缩 BCD 码表示为 0101 0111 0001 1001。

2．非压缩 BCD 码

非压缩 BCD 码是用 8 个二进制位表示一个十进制位。其中，低 4 位与压缩 BCD 码相同，高 4 位无意义。

【例 1-7】　将下列各数用非压缩 BCD 码表示。

（1）23D　　（2）5719D

【分析】

（1）23D 用非压缩 BCD 码表示为 xxxx 0010 xxxx 0011。

（2）5719D 用非压缩 BCD 码表示为 xxxx 0101 xxxx 0111 xxxx 0001 xxxx 1001。

有时，要求非压缩 BCD 码的高 4 位为 0。此时，5719 的非压缩 BCD 码为 0000 0101 0000 0111 0000 0001 0000 1001B。

另外，数字字符（0～9）的 ASCII 码恰好是 0～9 的非压缩 BCD 码。

1.3　基本逻辑运算

二进制逻辑运算包括与运算、或运算、非运算和异或运算。

1．“逻辑与”运算（AND）

0　AND　0　=　0

0　AND　1　=　0

1　AND　0　=　0

1　AND　1　=　1

2．“逻辑或”运算（OR）

0　OR　0　=　0

0　OR　1　=　1

1　OR　0　=　1

1　OR　1　=　1

3．“逻辑非”运算（NOT）

NOT　0　=　1

NOT　1　=　0

4．“逻辑异或”运算（XOR）

0　XOR　0　=　0

0　XOR　1　=　1

1　XOR　0　=　1

1　XOR　1　=　0

所有逻辑运算都是按位操作的。

1.4　汇编语言程序设计过程

1.4.1　编程阶段

1．分析问题

具体编程前，需要对将解决的问题进行全面分析，明确求解问题的内容和任务。例如，规定输入信息的形式和种类；确定给定条件和数据需要进行哪些处理；规定输出结果的形式。

2．确定算法

在明确求解问题的内容和任务之后，应确定解决问题的算法。所谓算法，简单说就是计算机能够实现的有限解题步骤。选择合适的算法是编写正确解题程序的基础。

算法可以使用文字或流程图进行描述。流程图中的主要符号如图 1-4 所示。

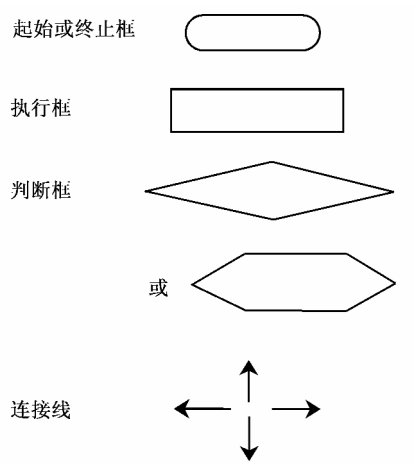

图 1-4　流程图的主要符号

- “起始框和终止框”表示程序的开始和结束；
- “执行框”表示需要完成的某项功能，可以是一条指令或一组指令序列，该框只能有一个入口和一个出口；
- “判断框”表示程序在此处需要根据不同的情况形成分支，框内应写明比较的条件，该框只有一个入口和两个出口；
- “连接线”用来连接两个流程框图并且指明框图的流向。

3．编写程序

采用汇编语言编写程序，实现已经确定的算法。使用汇编语言编写的源程序的后缀为 ASM。

1.4.2　上机阶段

编写好汇编语言源程序后，需要经过上机调试，以便检验程序的正确性并且得到程序的运行结果。8086/8088 汇编语言程序的上机过程可分为 5 个步骤。

1．编写.ASM 源程序

使用文本编辑器编辑汇编语言源程序。

2．对源程序进行汇编

计算机只能识别由机器语言编写的二进制代码，因此使用汇编语言编写的源程序必须被翻译成机器代码才能被执行。将汇编语言源程序翻译成机器语言描述的目标程序的过程称为汇编，实现汇编功能的程序称为汇编程序。本书使用的汇编程序是 Microsoft 公司研制的宏汇编程序 MASM5.0。

在汇编过程中，汇编程序对源程序进行两遍扫描，寻找源程序中存在的语法错误。如果确有语法错误，则扫描结束后，汇编程序将给出源程序中的错误信息而不产生目标文件。此时，用户需继续修改源程序，直到汇编程序扫描源程序不再出现程序错误时，才生成目标文件，即 OBJ 文件。

3．对目标文件进行连接

汇编语言源程序经过汇编后生成目标程序。目标程序是二进制代码文件，但是这个目标程序中的地址是“浮动”的，它只是一种逻辑地址，所以称为浮动二进制文件。目标文件不能直接运行浮动二进制文件，需要使用连接程序将它的逻辑地址转变成能够在计算机上直接运行的物理地址，并且使用连接程序将此目标文件与其他目标文件和库文件连接在一起，生成可执行文件，才能在计算机上直接运行。

与 MASM 配合使用的连接程序是 LINK。

从汇编语言源程序到生成可执行文件的过程如图 1-5 所示。

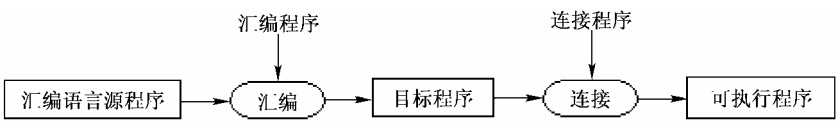

图 1-5　汇编与连接过程

4．使用 DEBUG 程序调试可执行文件

通过 DEBUG 程序调试可执行文件，逐段甚至是逐条指令地调试执行，从中观察是否达到预期的功能或得出预期结果，特别是控制转移指令是否按设想进行跳转等，从而发现程序在设计上的缺陷和错误。

5．在 DOS 环境下运行可执行文件

在 DOS 环境下运行可执行文件，查看运行结果。

1.5　汇编语言编程工具及环境

用于汇编语言的编程工具主要有编辑程序、汇编程序、连接程序和调试程序。

1．编辑程序

编辑汇编语言源程序可以使用任何文本编辑器，例如，DOS 的文本编辑器 EDIT，Windows 的记事本，以及一些高级语言集成开发环境，如 Visual C++，Delphi 等。

2．汇编程序

汇编程序将汇编语言源程序翻译成机器语言描述的目标程序。

汇编程序的主要功能如下：

- 检查源程序文件；
- 检测源程序中的语法错误并给出错误信息；
- 产生源程序的目标文件（.OBJ）；
- 展开宏指令。

3．连接程序

连接程序将目标程序与其他目标文件或库文件连接在一起，生成可执行文件。

连接程序的主要功能如下：

- 完成浮动地址的重定位；
- 实现多个目标文件及库文件的连接。

4．调试程序

调试程序 DEBUG 用于在 DOS 环境下调试运行一个可执行文件。DEBUG 可以逐条或逐段调试、运行可执行文件，同时给出有关结果信息，以便对该文件的功能进行检查。

小结

汇编语言是低级程序设计语言，是机器语言的符号表示。汇编语言可以充分发挥计算机硬件的特性与优势，使用汇编语言编写的程序具有占用存储空间少、运行速度快、执行代码短等优点。因此，汇编语言适宜编写对计算机硬件进行直接控制或者对运行时间和效率有较高要求的应用软件或系统软件。学习汇编语言不仅可以加深对计算机系统，特别是程序工作原理的理解，而且有助于提高程序员软件设计的水平与质量。

汇编语言编写的源程序需要经过汇编和连接处理后才能形成可执行的文件。

计算机系统使用二进制数表示数据。为了描述方便，常采用十六进制形式。计算机系统采用二进制补码表示带符号数。

习题

一、选择题

1．下列四条叙述中，正确的一条是_____。

A．计算机能够直接识别并执行高级语言源程序

B．计算机能够直接识别并执行机器指令

C．计算机能够直接识别并执行数据库语言源程序

D．汇编语言源程序可被计算机直接识别和执行

2．计算机的硬件主要包括 CPU、存储器和_____。

A．显示器和打印机

B．键盘和鼠标

C．输入/输出设备及总线

D．打印机、显示器和鼠标

3．微型计算机采用总线结构，总线通常由三部分组成，分别是_____。

A．数据总线、传输总线和通信总线　　B．地址总线、逻辑总线和信号总线

C．控制总线、地址总线和运算总线　　D．数据总线、地址总线和控制总线

4．下列描述中，正确的是 _____。

A．2 MB=2000 B　B．1 MB=1000 KB　C．1 MB=1024 B　D．2 MB=2048 KB

5．在微机中，应用最普遍的字符编码是_____。

A．BCD 码　　B．ASCII 码　　C．原码　　D．补码

6．某个整数的二进制补码和原码相同，则该数一定_____。

A．大于 0　　B．小于 0　　C．等于 0　　D．大于或等于 0

二、填空

1．十进制数(–123)的补码是_____。

2．PC 微机中单字整数(16 位)的有效范围是_____。

3．所有由两个“1”和六个“0”组成一个 8 位的二进制整数，若采用补码表示，则最小值为_____。

4．汇编语言源程序经汇编、连接后生成 _____。

5．数字字符 4 的 ASCII 码为十进制数 52，数字字符 9 的 ASCII 码为十进制数_____。

三、简述题

1．写出下列十进制数对应的压缩和非压缩 BCD 码形式。

（1）92　　（2）3190

2．写出下列十进制数的二进制补码形式。

（1）–35　　（2）369

3．写出下列补码数的相反数。

（1）11001000　　（2）10111101

（3）10000000　　（4）00000000

4．把下列数和字符用 16 进制表示出来。

（1）字母 Q　　（2）二进制数 10111101

5．将下列 16 进制数转换为二进制数和十进制数。

（1）FA　　（2）FFFE

第 2 章　80x86 计算机组织

本章要点：

- ☑ 8086/8088 CPU 结构
- ☑ 存储器中存储单元的地址和内容
- ☑ 分段内存管理机制
- ☑ 8086/8088CPU 寄存器组

计算机系统主要由中央处理器（CPU）、存储器和输入/输出子系统 3 部分组成，各个部分之间通过系统总线相连，如图 2-1 所示。

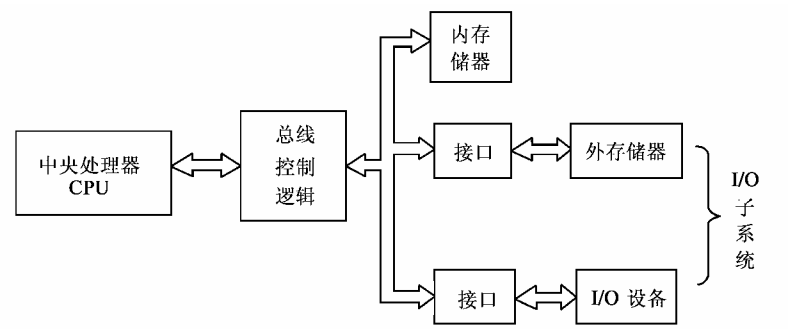

图 2-1　计算机的基本结构

1．中央处理器

中央处理器（CPU）是计算机的核心。程序指令由 CPU 解释并执行。CPU 还负责产生各种控制信号，令各部件协调工作，使整个计算机系统构成一个整体。

2．内存

内存是高速的数据存储部件，即将运行的程序存放在内存中。它也可以存放程序中所用的数据、信息及中间结果。

3．输入/输出子系统

输入/输出子系统完成 CPU 或内存与外存或外部设备的数据交换。外部设备分为输入设备和输出设备。输入设备负责将控制命令、外部采集到的数据传递给 CPU 或内存，典型的输入设备包括键盘、鼠标等；输出设备负责将计算机内部的数据以人可以理解的形式表现出来，或者变成控制信号以控制其他设备，典型的输出设备包括显示器、打印机等。

4．系统总线

系统总线将 CPU、存储器和输入/输出子系统 3 部分相连，用来传送各部分之间的信息。系统总线分为数据总线、控制总线和地址总线 3 类。

（1）数据总线用来传递数据，决定了 CPU 每次存取数据的最大位数。

（2）地址总线用来确定传送的数据在存储器中的具体位置。它决定了最大可编址的内存空间。若有一条地址线，则 CPU 可以产生 0 和 1 两个地址。若有 *n* 条地址线，则 CPU 可以产生 2^n 个不同的地址。

（3）控制总线用来控制 CPU 与内存和外部设备之间的数据传送方式，即控制总线控制 CPU 通过数据总线从内存或外设读取数据还是往内存或外设写入数据。

2.1　中央处理器

CPU 的任务是执行存放在存储器中的指令序列。在结构上，8086/8088CPU 采用流水指令队列及地址计算和算术逻辑运算分布处理的技术。8086/8088CPU 由总线接口单元（BIU）和执行单元（EU）组成，如图 2-2 所示。

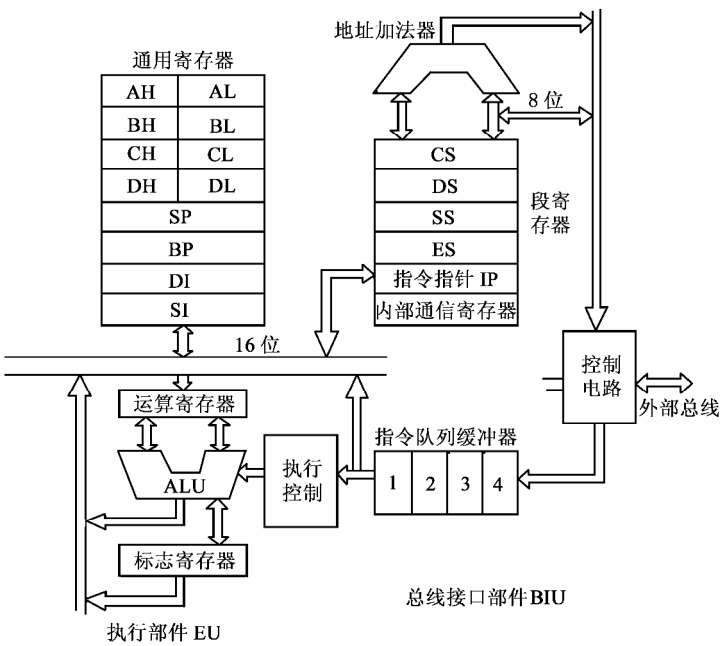

图 2-2　8086 微处理器的内部结构

执行单元 EU 由算术逻辑部件 ALU、通用寄存器组和标志寄存器（PSW）组成。执行单元用来解释并且执行来自指令队列中的指令，完成算术逻辑运算动作。执行单元产生数据在内存中的相对地址，接收来自内部总线的数据。

总线接口部件包括 4 个段寄存器、指令指针寄存器、指令队列缓冲器、地址计算逻辑和执行机构等。总线接口部件专门用于计算实际内存地址，不仅把来自执行单元的数据相对地址转换为实际地址，还负责提供当前取指令的内存地址，即取指令时，从存储器指定地址取出指令送入指令队列排队；执行指令时，根据执行单元命令对指定存储器单元或外设存取数据。

CPU 与内存之间的数据传送需要经过总线接口部件。从内存中取出的指令存放在指令队