



8086/8088/80X86

实用 汇编语言 工具库

TP313
G37

355889

8086 / 8088 / 80X86

实用汇编语言工具库

葛 如 顶 编写

徐 福 培 审校

南京大学出版社

1991 · 南京

JS/74.23
内 容 简 介

本书为配合国内流行的 IBM PC 系列、GW 系列以及与上述机种相兼容的各种微型计算机的应用开发而编写的。书中收编了近 150 个内容丰富、实用性很强的汇编语言子程序，主要包括数制转换、有序或无序序列的操作、多精度算术与逻辑操作、磁盘与文件管理、字符、图形、声音等输入输出以及置/取中断向量与时间操作等。

本书较详细地介绍了汇编语言与各种通用程序设计语言之间的接口，并给出在上述微型机上可运行的实例。

本书对于从事微型计算机的科研、生产、教学和应用开发的软硬件设计人员，均是一本极有价值的实用手册，也可作为高等院校计算机系及有关专业的教学参考书。同时对于微型计算机的初学者也一定会有所帮助。

~~8086~~/8088/80X86
实用汇编语言工具库
葛如顶编写
徐福培审校

南 京 大 学 出 版 社 出 版
(南京大学校内)
江苏省新华书店发行 江苏省高邮市印刷厂印刷
开本: 787×1092 1/16 印张: 25 字数: 614 千
1990 年 8 月第 1 版 1991 年 5 月第 2 次印刷
印数: 5001-10000
ISBN 7-305-00793-5

TP·26

定价: 9.50 元

前 言

汇编语言虽是一种速度快、功能强的程序设计语言，但即使是一个有经验的程序设计者，用汇编语言编写一个高效的汇编语言程序来完成特定功能，也是困难且费时的。而对于初学者或局限于使用高级语言编程的人来说，写一个这样的汇编语言程序更会感到无从入手。为此，本书不仅介绍了汇编语言的编程技术及汇编语言程序之间的相互调用方法，还详细地讨论了汇编语言与各种通用程序设计语言之间的接口技术。使得对于任何一个程序设计者（不管其是否善于使用汇编语言进行编程），都能从本书中获得收益，方便且有效地将本书中的汇编子程序结合到自己的应用程序（无论是汇编语言程序还是高级语言程序）中。本书是作为工具（或手册）提供给读者的，读者毋需从头读到尾，除非你对汇编语言相当有兴趣，并希望从中了解每一个子程序的编程技术。通常情况下，读者仅需从本书中查阅有关内容用于解决你所需要的特定的程序设计问题，为开发自己的实用程序服务。

全书由 9 章和 4 个附录构成，其中四——九章为工具库的主要内容，它们全部在 PC XT、PC AT 和 GW286 等微型机上调试过。它们的每个子程序均按照下列款项进行叙述：

- (1) 子程序名（中文和英文）——调用标识符。
- (2) 功能简介——简要说明子程序所要完成的功能。
- (3) 操作说明——较为详细地介绍子程序中每一个关键性操作，对于较复杂的子程序还给出操作示意图或流程图。
- (4) 输入参数——进入或调用该子程序所必需的入口参数及参数类型等。
- (5) 输出参数——调用该子程序获得什么样的返回参数和寄存器的变化情况等。
- (6) 注解——说明调用该子程序时应注意的事项。
- (7) 例子——给出使用该子程序的示范程序，即应用程序与汇编语言程序的接口。
- (8) 程序清单——这部分按目前较流行的程序设计风格列出，即每个子程序有一个引头简要介绍子程序功能，入、出口参数和寄存器变化情况等。在国际上一些较好的程序例子都采用这种形式。建议读者用汇编语言编制程序时也采用这种程序设计风格。

下面将各章的安排和主要内容简要说明如下：

第一章 微处理器的结构特征 介绍 80X86 系列中主要微处理器的数据通路及其可寻址寄存器的功用。

第二章 汇编语言的使用 简要说明宏汇编语言的语句结构，模块化程序设计方法以及用汇编语言开发应用软件的过程，还列出 80X86 系列的指令表与宏汇编语言的伪操

作符表等。

第三章 高级语言与汇编语言的接口 这一章着重向读者介绍了 MS-PASCAL, TURBO PASCAL, CI-C86, TURBO C, FORTRAN 和编译 BASIC 等高级语言与汇编语言之间的连接问题, 并给出连接的实例, 这些例子均能在 PC/XT, PC/AT 和长城 286, 386 等微型机上运行。

第四章 数制转换 本章给出 10 个二进制数到十进制数、二进制数到十六进制数、二进制数到十进制数的 ASCII 字符串和十进制数的 ASCII 字符串到二进制数的转换子程序。

第五章 串操作 介绍了对无序、有序序列的检索、插入、删除元素的操作, 对字符串的比较、移动、交换和在字符串中查找, 插入、删除与移动另一个字符串的操作等 33 个子程序。

第六章 多精度算术运算与逻辑操作 本章提供 24 个 BCD 数的加、减、乘、除运算和 32 位二进制数的算术运算和移位操作子程序。

第七章 输入输出操作 本章分通用输入输出和 IBM PC 系列等专用输入输出两部分, 前者着重介绍字符与字符串的输入输出。后者给出了包括图形方式下读点、写点、画线、填充和声音输出等操作共 42 个子程序。

第八章 磁盘及文件操作 这一章提供了对磁盘驱动器的操作, 磁盘目录操作以及磁盘文件的读/写等 28 个子程序, 其中文件操作均采用文件管理的扩展功能(即使用文件号方式)完成的。

第九章 中断和其他 本章主要介绍对中断向量的操作和对存储器空间的管理以及加载并执行一个程序等 8 个子程序。

附录一 ASCII 字符与编码对照表。

附录二 IBM PC DOS 中断向量一览表, 较详细地列出部分中断向量的含义与使用方法。

附录三 MS-DOS (3.10) 的系统功能调用, 列出每一个调用的功能、入口和出口参数等。

附录四 工具库子程序索引表, 以英文字母顺序排列子程序名, 方便读者查阅并使用工具库。

与本书配套还提供了四——九章中全部汇编语言子程序的源程序和目标程序工具库软盘片一张, 与本书同时发行。这样, 用户可直接将本书中的汇编语言源程序通过文本编辑从软盘中装入到自己的源程序中, 也可以通过连接程序 (LINK) 把工具库 (ASMT00L.LIB) 中的某些子程序连接到自己的目标程序中, 从而可大大提高应用程序开发的效率。

本书适合于从事微型计算机的科研、生产、教学和应用开发的软硬件设计者, 也可作为高等院校计算机系和有关专业的教学参考书。同时, 也有益于微型计算机的初学者。

在本书的编写过程中，得到了张福炎教授、蔡士杰副教授的热心支持与帮助。承蒙徐福培副教授审阅了全书并提出了许多宝贵意见。全部书稿由万新安同志抄写。值此机会一并致以衷心感谢。

因时间仓促，水平有限，错误在所难免，恳请广大读者批评指教。

编 者

1989.12 于南京大学

目 录

前 言

第一章 微处理器的结构特征	1
1.1 8086 / 8088 微处理器	1
1.2 80286 微处理器	4
1.3 80386 微处理器	7
第二章 汇编语言的使用	13
2.1 汇编语言的语法结构	13
2.2 汇编语言程序应用软件的开发	32
第三章 高级语言与汇编语言的接口	41
3.1 MS-PASCAL 与汇编语言接口	44
3.2 TURBO PASCAL 与汇编语言接口	46
3.3 C/C++ 与汇编语言接口	50
3.4 TURBO C 与汇编语言接口	53
3.5 IBM FORTRAN 与汇编语言接口	57
3.6 编译 BASIC 与汇编语言接口	59
第四章 代码转换	62
4-1 二进制数转换成压缩型 BCD 数 (BIN2BCD)	62
4-2 压缩型 BCD 数转换成二进制数 (BCD2BIN)	63
4-3 十六位二进制转换成压缩型 BCD 数 (B2BCDW)	64
4-4 四位压缩型 BCD 数转换成十六位二进制数 (BCDW2B)	66
4-5 二进制数转换成十六进制的 ASCII 数 (BIN2ASC)	68
4-6 十六进制的 ASCII 数转换成二进制数 (ASC2BIN)	69
4-7 无符号二进制数转换成 ASCII 字符串 (UBIN2\$)	70
4-8 有符号二进制数转换成 ASCII 字符串 (SBIN2\$)	73
4-9 十进制 ASCII 字符串转换成无符号二进制数 (\$2UBIN)	76

4-10 十进制 ASCII 字符串转换成有符号二进制数 (\$2SBIN)	78
--	----

第五章 串操作 82

5.1 无序序列操作 82

5-1 用字节值填充存储器 (FILLMEMB)	82
5-2 用字值填充存储器 (FILLMEMW)	83
5-3 查找一个字节 (FINDBYTE)	85
5-4 查找一个字 (FINDWORD)	86
5-5 求无符号字型序列的平均值 (AVERAGEU)	88
5-6 求有符号字型序列的平均值 (AVERAGES)	89
5-7 求无符号字节型序列的各元素之和 (SUMUB)	90
5-8 求有符号字节型序列的各元素之和 (SUMSB)	92
5-9 求无符号字节型序列中的最大值与最小值 (MAXMINU)	94
5-10 求有符号字节型序列中的最大值与最小值 (MAXMINS)	95
5-11 在无序字节型序列中添加一字节 (ADDB2UL)	97
5-12 从无序字节型序列中删除一字节 (DELBUL)	98

5.2 字符串操作 100

5-13 移动一字节块 (MOVBLOCK)	100
5-14 比较两字节型数据块 (COMPMEMB)	101
5-15 比较两字型数据块 (COMPMEMW)	104
5-16 交换两字符串 (SWITCHB)	105
5-17 交换两字串 (SWITCHW)	107
5-18 转换成小写字母 (TOLOWER)	108
5-19 转换成大写字母 (TOUPPER)	109
5-20 在一字符串末添加一子字符串 (APPEND\$)	111
5-21 在一字符串中插入一子字符串 (INSERT\$)	113
5-22 在一字符串中检索一子字符串 (FINDSUB\$)	115
5-23 在一字符串中删除一子字符串 (DELSUB\$)	118
5-24 在一字符串中复制一子字符串 (COPYSUB\$)	121
5-25 在一字符串中移动一子字符串 (MOVESUB\$)	123

5.3 排序 125

5-26 将一无符号字节型序列排成升序 (BUBBLEBA)	125
5-27 将一有符号字型序列排成升序 (BUBBLEWA)	127

5.4 有序序列的操作 129

5-28 在一升序序列中检索一字节型数据 (FINDBA)	129
5-29 在一升序序列中检索一字型数据 (FINDWA)	132
5-30 在一升序序列中插入一字节 (INSBA)	134

5-31	在一升序序列中插入一字 (INSWA)	137
5-32	从一升序序列中删除一字节 (DELBA)	139
5-33	从一升序序列中删除一字 (DELWA)	141
第六章 多精度算术运算与逻辑操作		144
6.1	十进制算术运算	144
6-1	非压缩型 BCD 数加法 (ADDUD16)	144
6-2	非压缩型 BCD 数减法 (SUBUD16)	146
6-3	非压缩型 BCD 数乘法 (MULUD16)	147
6-4	非压缩型 BCD 数除法 (DIVUD16)	149
6-5	压缩型 BCD 数加法 (ADDPD16)	152
6-6	压缩型 BCD 数减法 (SUBPD16)	153
6-7	压缩型 BCD 数乘法 (MULPD16)	155
6-8	压缩型 BCD 数除法 (DIVPD16)	157
6.2	32 位二进制数算术运算	159
6-9	32 位无符号二进制数加法 (ADDU32)	160
6-10	32 位无符号二进制数减法 (SUBU32)	161
6-11	32 位无符号二进制数乘法 (MULU32)	162
6-12	32 位无符号二进制数比较 (COMPU32)	165
6-13	32 位二进制整数的平方根 (SQRT32)	166
6-14	32 位有符号二进制数加法 (ADDS32)	168
6-15	32 位有符号二进制数减法 (SUBS32)	171
6-16	32 位有符号二进制数乘法 (MULS32)	173
6-17	32 位有符号二进制数比较 (COMPS32)	175
6.3	32 位二进制数移位与循环移位操作	176
6-18	32 位数算术左移 (SAL32)	177
6-19	32 位数算术右移 (SAR32)	179
6-20	32 位数逻辑右移 (SHR32)	180
6-21	32 位数循环左移 (ROL32)	181
6-22	32 位数循环右移 (ROR32)	183
6-23	32 位带进位循环左移 (RCL32)	184
6-24	32 位带进位循环右移 (RCR32)	186
第七章 输入输出操作		188
7.1	通用输入输出	188
7-1	显示一字符串 (LIST\$)	189

7-2	读一字符串 (GET\$)	190
7-3	显示一字符 (LISTCHR)	192
7-4	打印一字符 (PRINTCHR)	193
7-5	读一字符<不带回显> (GETCHR)	194
7-6	读一字符<带回显> (GETCHRE)	197
7-7	发送一字符至串行口 (SENDCAER)	199
7-8	从串行口读一字符 (GETCSER)	200
7-9	响铃 (BEEP)	202
7-10	十六进制数输入 (HEXIN)	203
7-11	十六进制数输出 (HEXOUT)	205
7-12	BCD 数输入 (BCDIN)	206
7-13	BCD 数输出 (BCDOUT)	207
7-14	带有符号的十进制数输入 (SGNDEIN)	209
7-15	无符号十进制数输入 (DECIN)	211
7-16	显示一字符块 (DISPCHRB)	213
7.2	IBM PC 专用输入输出	214
7-17	获得 IBM PC 型号 (GETPCMOD)	215
7-18	置屏幕显示模式 (SETSMODE)	216
7-19	清除屏幕 (CLEAR\$)	218
7-20	读取光标位置 (READCURS)	219
7-21	移动光标 (MOVECURS)	220
7-22	在彩色屏幕上画一点 (SETMDOT)	222
7-23	在彩色屏幕上用异或方式画点 (XORMDOT)	224
7-24	读一点的颜色 (READMDOT)	226
7-25	矩形区域内填充 (RETMBOX)	228
7-26	用异或方式在矩形框内填充 (XORMBOX)	234
7-27	画一条直线 (SETLINE)	239
7-28	任意多边形填充 (PAINT)	244
7-29	画一光栅字符 (DRCHAR)	251
7-30	在彩色屏幕上显示一光栅字符串 (GMESSOUT)	254
7-31	初始化串行通讯口 (COMINIT)	256
7-32	通讯线输入校验 (COMINCK)	258
7-33	通讯线输出 (COMOUT)	260
7-34	通讯线开放 (COMON)	261
7-35	通讯线关闭 (COMOFF)	262
7.3	时间与音响	263
7-36	读取现行时间 (GETTIME)	264
7-37	设置时间 (SETTIME)	266

7-38 延时 (DELAY)	270
7-39 读取日期 (GETATE)	272
7-40 设置日期 (SETDATE)	275
7-41 产生音响 (SOUND)	278
7-42 演奏音乐 (PLAY)	281
第八章 磁盘及文件操作	284
8.1 磁盘设备操作	284
8-1 取当前盘的类型 (GETDRIVE)	284
8-2 选择磁盘驱动器 (SETDRIVE)	286
8-3 读写校验开关 (GETVERIF)	287
8-4 设置或清除写校验开关 (SETVERIF)	288
8-5 获取磁盘剩余空间 (DSKSPACE)	289
8.2 目录操作	291
8-6 显示错误信息 (SHOWERR)	291
8-7 读取现行目录 (GETDIR)	295
8-8 键入路径名 (GETPATH)	297
8-9 建立子目录 (MAKEDIR)	298
8-10 删除子目录 (REMDIR)	300
8-11 改变当前目录 (CHRDIR)	301
8.3 文件操作	303
8-12 文件重命名 (RENAME)	305
8-13 删除文件 (DELFIL)	307
8-14 读文件属性 (GETMODE)	309
8-15 改变文件属性 (CHMODE)	310
8-16 查找第一个匹配文件 (FINDF)	313
8-17 查找下一个匹配文件 (FINDNXTF)	316
8-18 写保护文件 (PROTF)	318
8-19 非写保护文件 (UNPROTF)	321
8-20 隐含文件 (HIDEF)	323
8-21 非隐含文件 (UNHIDEF)	326
8-22 建立文件 (NEWFILE)	328
8-23 打开文件 (OPENFILE)	330
8-24 关闭文件 (CLOSEFILE)	332
8-25 移动文件指针 (MOVEPTR)	333
8-26 读文件 (READFILE)	336
8-27 写文件 (WRITEFILE)	339

8-28 删除文件内容 (EMPTYE)	343
第九章 中断与其他	346
9.1 中断向量	346
9-1 取中断向量 (GETINTV)	346
9-2 置中断向量 (SETINTV)	347
9.2 其他	350
9-3 申请内存块 (REQUIREM)	350
9-4 释放内存块 (FREEMEM)	352
9-5 修改内存 (MODIFYM)	354
9-6 装入一个程序 (LOADPRO)	356
9-7 返回程序终止状态码 (PRCODEC)	360
9-8 取 DOS 版本号 (GETDOSV)	361
附录一 ASCII 字符与编码对照表	363
附录二 IBM PC DOS 中断向量一览表	364
附录三 DOS (3.10) 系统功能调用表	371
附录四 工具库程序索引表 (按字母顺序)	379
主要参考文献	383

第一章 微处理器的结构特征

本书中的所有子程序与举例都可以不加变更地在任何一台具有 MS-DOS 2.0 版以上操作系统的微型计算机上运行。由于是汇编语言子程序，因此它们依赖于构成计算机系统的微处理器。本章从程序设计者的角度来叙述 Intel 80X86 系列主要微处理器的结构特征。着重介绍 Intel 8086/8088, 80286 和 80386 微处理器的数据通路（即内部功能逻辑），汇编语言所能寻址的寄存器，以及对存储器的寻址能力等。

1.1 8086/8088 微处理器

Intel 公司于 1978 年推出了第一种高性能的十六位微处理器——8086。在以后的几年内又很快推出若干种微处理器，以及与之相配套的支持芯片，形成了一个 8086 系列。Intel 8088 就是这个系列中的一员。它采用 8 位外部数据总线接口、16 位内部数据总线的体系结构，既能处理 8 位数据，又能处理 16 位数据，从而架起了 8 位与 16 位微处理器之间的桥梁。这种处理器与 Intel 8086 软件完全兼容，同时又与以前的 8 位微处理器 Intel 8080/8085 的硬件与外围控制芯片兼容，使之具有较强的生命力。特别是 IBM 公司宣布其 IBM PC (XT) 采用 Intel 公司的微处理器及其配套产品以后，Intel 的 16 位微处理器日益得到各方面的重视。Intel 8086 与 8088 除硬件接口有些差别外，从软件设计角度来看没有多大不同。因此，本节主要介绍 Intel 8088 逻辑结构。其内部逻辑结构示于图 1-1。

Intel 8088 功能上可分成总线接口单元 BIU (Bus Interface Unit) 和执行单元 EU (Execution Unit) 两大部分。

BIU 负责与存储器的接口，即 8088 CPU 与存储器之间的信息交往均是由 BIU 完成。其功能为负责从存储器的指定区域内取出指令，送到四个字节（8086 为六个字节）的指令流队列中排队，并负责从内存指定区域中取出指令规定的操作数，传送给 EU 去执行。这样，8088 的取指令与执行指令是分开进行的，其优点是在 EU 执行指令的过程中，BIU 可以预取多条指令放进指令流队列中排队，当 EU 执行一条指令后，可立即执行下一条指令，从而减少了 CPU 取指令的等待时间，提高了整个系统的运行速度。

1. 通用寄存器

8088CPU 为程序员提供了 8 个 16 位的通用寄存器，其中有四个数据寄存器，其名字为 AX, BX, CX 和 DX，用于暂存 16 位操作数，它们也可以把高半部和低半部分开使用，用于暂存 8 位的操作数，此时它们的名字为 AH, AL, BH, BL, CH, CL, DH 和 DL。通常 AX 作为累加器使用，BX 作为基址寄存器使用，CX 作为计数器使用，DX 存

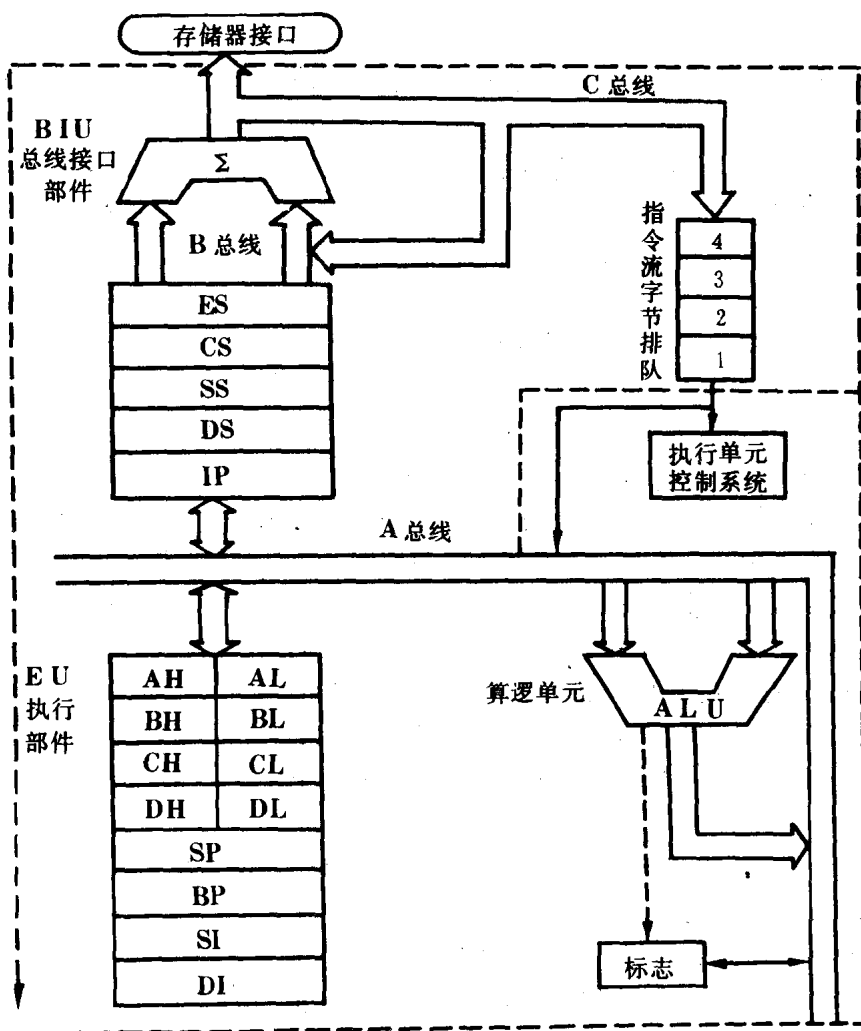


图 1-1 8088 逻辑结构图

放数据。另外四个 16 位的寄存器为指针寄存器 BP 与 SP 和变址寄存器 SI 与 DI。BP 和 SP 寄存器用于对堆栈的管理，SP 为堆栈指针，当对堆栈进行操作时，必须使用 SS 堆栈段寄存器才能确定堆栈的实际地址，它总是指向栈顶单元。BP 寄存器为基址指针，它用于寻址堆栈中的数据，这时同样需用 SS 段寄存器配合寻址。SI 与 DI 用于访问数据存储器中的数据，通常用于串操作，此时，SI 作为源变址寄存器要与 DS 段寄存器配合才能确定实际地址，而 DI 作为目的变址寄存器要与 ES 段寄存器配合才能形成实际地址。BP 和 BX 既可作基址寄存器又可作变址寄存器使用。通用寄存器见图 1-2。

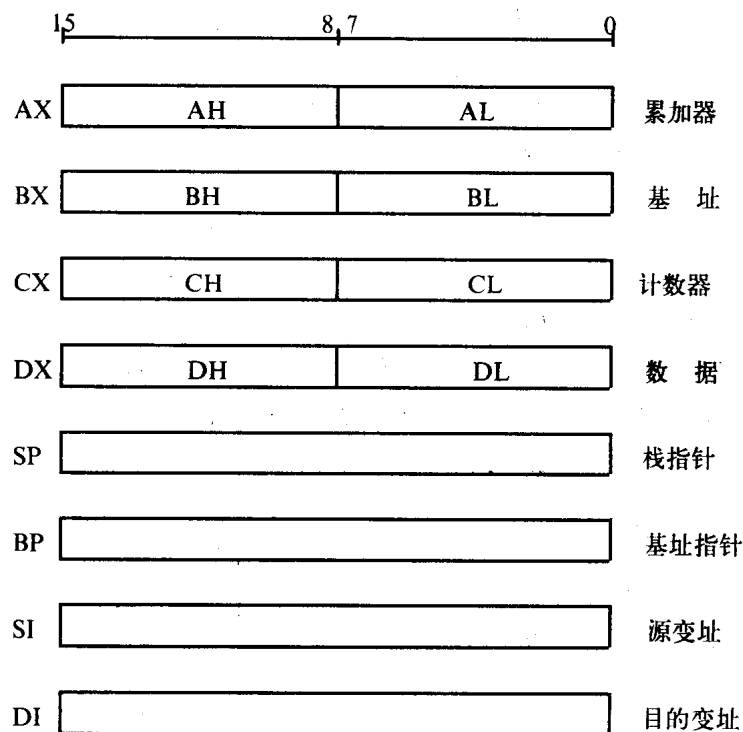


图 1-2 通用寄存器

2. 段寄存器

8088 有四个 16 位的段寄存器（见图 1-3），分别指向当前的四个可寻址存储器段，四个段寄存器记为 CS（代码段），DS（数据段），SS（堆栈段）和 ES（附加数据段）。它们不可以互换使用，CS 识别当前的代码段，DS 识别当前的数据段，SS 识别当前的堆栈段，ES 识别当前的附加数据段。每个段最大空间可达 64K 字节。所有的取指令操作都是在当前代码段内进行的，在执行取指令时，指令指针（IP）的内容总是自动地与 CS 段寄存器的内容左移 4 位后相加，而形成指令的实际地址；取操作数操作时，可选择 DS 或

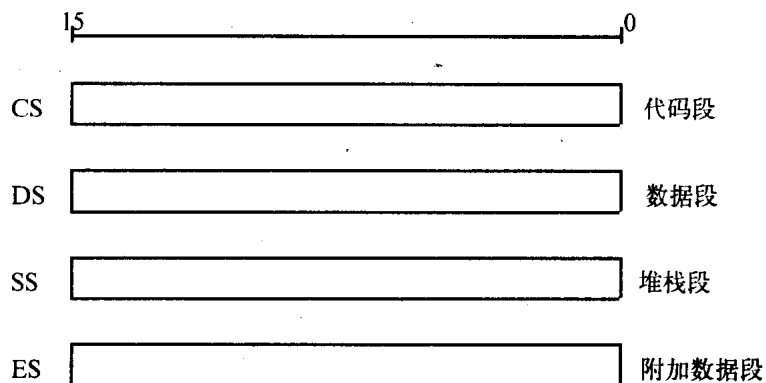


图 1-3 段寄存器

ES 寄存器作为基址，再加上逻辑地址就形成操作数的实际地址，逻辑地址的确定方法由寻址方式决定；在执行堆栈操作时，逻辑地址（SP 寄存器的内容）总是与堆栈段寄存器 SS 的内容左移 4 位相加，形成堆栈单元的实际地址。若存储器的总容量不超过 64KB，可以将所有段寄存器设置成同一数值，即让所有的段完全重迭。

3. 控制寄存器

8088 有两个 16 位的控制寄存器 IP 和 F。指令指示寄存器 IP 用于确定指令地址，它总是与 CS 段寄存器配合使用，形成下一条指令的实际地址。F 称为标志寄存器，用于记录某些指令的执行结果状态，或控制 8088 某些指令的操作流程。控制寄存器见图 1-4。

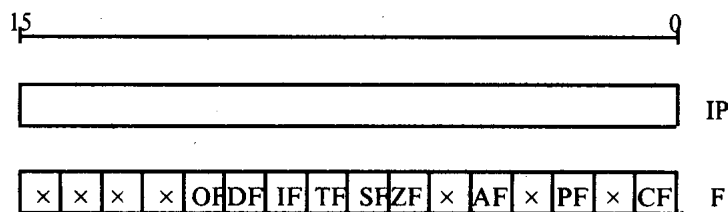


图 1-4 控制寄存器

其中 F 寄存器的各位含义为：

- CF——进位标志。当用 8 位或 16 位操作数完成算术操作产生了进位或借位时，CF 置为 1；否则，它就复位为零。
- PF——奇偶标志。主要用于数据通讯中数据正确性的检验，奇偶性为奇时置 1，奇偶性为偶时清零。
- AF——辅助进位标志。用于 BCD 算术运算，表示最低有效的 4 位 BCD 数值位是否产生了进位或借位，若是则使 AF=1，否则 AF=0。
- ZF——0 标志。当结果为零时自动置 1。
- SF——符号标志。当结果为负时置为 1，正时清零。
- TF——陷阱标志。当 TF=1 时，使微处理器进入单步工作方式。
- IF——中断标志。当 IF=1 时，允许响应外部中断，否则禁止响应外部中断。
- DF——方向标志。当 DF=1 时，串指令自动增量所指定的变址器 SI 和 DI，否则自动减量。
- OF——溢出标志。当结果超出系统所能表示的范围时置 1，否则清零。

1.2 80286 微处理器

80286 是 Intel 公司于 1984 年推出的新一代高性能的微处理器。它具有集成在芯片内部的存储器管理机构，对操作系统和任务程序与任务数据保密。并具有多用户多任务系统的较完善的处理能力。在 10MHz 的 80286 上处理任务的速度比 5MHz 的 8086 快 6 倍左右。

80286 具有很强的寻址能力，并能以两种不同的方式运行（实际地址方式和虚拟地址保护方式）。在实际地址方式下，具有一兆字节的寻址能力；在虚地址保护方式下，能将每个任务的 2^{30} 字节的虚拟地址空间映射到 2^{24} 字节的物理地址空间中。即在这种方式下，具有 16 兆字节的寻址能力。

80286 与现有的 8086 / 8088 具有软件向上兼容的特点。即在实地址方式下，80286 的目标代码与已有的 8086 / 8088 的软件兼容；在虚拟地址方式下，80286 与 8086 / 8088 保持源代码相兼容，这些代码可升级使用由 80286 集成在芯片内的存储器管理和保护机构所支持的虚地址。图 1-5 为 80286 的逻辑结构图。

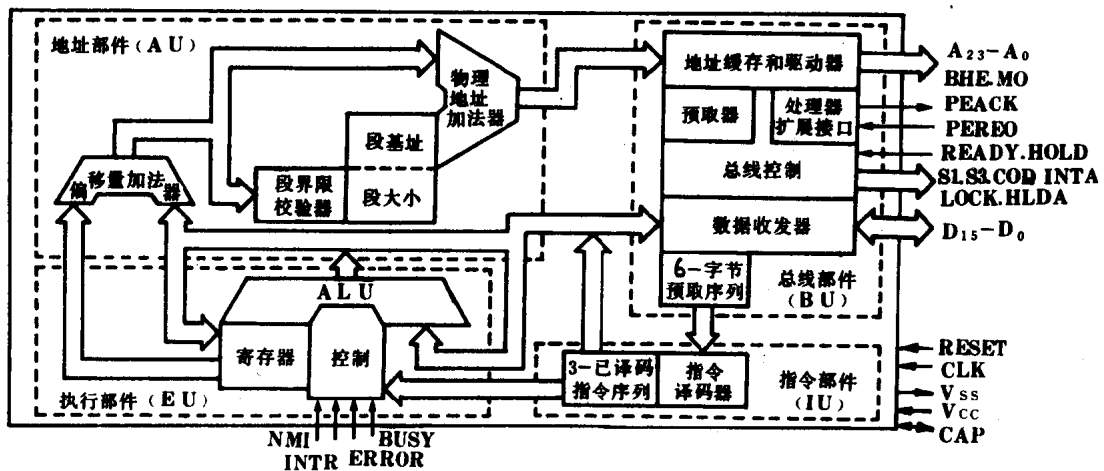


图 1-5 80286 内部逻辑结构

80286 CPU 从功能上看可分成四个部分：地址部件 AU (Address Unit)，总线部件 BU (Bus Unit)，指令部件 IU (Instruction Unit) 和执行部件 EU (Execution Unit)。前三个部件相当于 8086 / 8088 的总线接口部件 (BIU)。这四个部件并行操作提高了系统的吞吐率。80286 内部并行操作的情况如图 1-6 所示。

1. 通用寄存器

80286 为程序员设置了 8 个 16 位的通用寄存器。它们的名字与功能同 8086 / 8088 的通用寄存器。通用寄存器见图 1-2。

2. 段寄存器

80286 设立了四个 16 位的段寄存器 (CS、DS、SS 和 ES)。在实地址方式下，这四个段寄存器的解释与使用同 8086 / 8088，它们中包含了实际的物理地址。在虚地址方式