

硬件接口开发系列

PC Interfacing, Communications
and Windows Programming

PC 接口通信 与 Windows 编程

William Buchanan 著
精英科技 译



ADDISON-WESLEY



中国电力出版社

www.infopower.com.cn

硬件接口开发系列

PC Interfacing, Communications
and Windows Programming

PC 接口通信 与 Windows 编程

William Buchanan 著

精英科技 译

中国电力出版社

内 容 提 要

本书是“硬件接口开发系列”丛书的一本。

本书内容包括了计算机系统的各个方面，从低层硬件到高层软件。本书包括七个主要部分：硬件、接口、RS-232/并行端口、Windows 编程、Win32 编程、Windows 接口和网络，讲述了软件和硬件工程的交叉知识。

本书结构严谨、内容丰富、思路清晰，书中例子易于理解，每章后都有习题，适于作为大中专院校教材，也是电脑爱好者的必备参考书。

图书在版编目（CIP）数据

PC 接口通信与 Windows 编程/（美）布坎南 编著；精英科技 译.
-北京：中国电力出版社，2001
ISBN 7-5083-0695-3

I .P… II.①布…②精… III. ①个人计算机-接口②窗口软件，
Windows-软件接口-程序设计 IV.TP3

中国版本图书馆 CIP 数据核字（2001）第 045607 号

北京版权局著作权登记号 图字 01-2000-1696

本书英文版原名：PC Interfacing, Communications and Windows Programming

Published by arrangement with Addison Wesley Longman, Inc.

All rights reserved.

本书中文版由美国培生集团授权出版，版权所有。

中国电力出版社出版、发行

（北京三里河路 6 号 100044 <http://www.infopower.com.cn>）

三河市实验小学印刷厂印刷

各地新华书店经售

*

2001 年 10 月第一版 2001 年 10 月北京第一次印刷

787 毫米×1092 毫米 16 开本 35 印张 785 千字

定价 59.00 元

版 权 所 有 翻 印 必 究

（本书如有印装质量问题，我社发行部负责退换）

前 言

本书是《Applied PC Interfacing》和《Graphics and Interrupts》的后续部分，希望能够使读者对计算机系统有更深入的理解。在过去几年中，我已经从全世界的学生和学术工作者那里接收到大量的电子邮件，其中的主要问题是（按询问次数排序）：

- 如何使用串行端口通信？
- 如何使用并行端口通信？
- 如何写中断驱动软件？
- 如何通过网络通信？
- 如何在 Microsoft Windows 下编程？
- 连接到外设最好的总线是什么（串行或并行）？
- 如何使用 Win32 编程？
- 如何创建一个菜单？
- 哪个更适合编程，是 C++ 还是 VB？
- PC 机使用什么设备？

我认为需要有这样一本书，它包括计算机系统所有的不同方面，从低层硬件（例如处理器和接口设备）到高层软件（例如 Win32 编程）。一个软件工程师不需要考虑硬件的时代已经过去了，同样，硬件工程师不需要理解软件的时代也已经过去了。

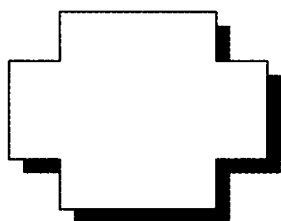
本书可以被分成七个主要部分；它们是：

- 硬件。8086 微处理器、8086 指令、80386/80486/Pentium、接口、UART、PIC、PPI、PTC、中断和 BIOS。
- 接口。标准、PC/ISA、本地总线、PCI、主板设计、IDE、SCSI 和 PCMCIA。
- RS-232/并行端口。RS-232、中断驱动 RS-232、并行端口、中断驱动并行端口、ECP/EPP 模式和调制解调器。
- Windows 编程。VB 编程、窗体、菜单、时间、图形、定时器和串行通信。
- Windows 接口。注册表、体系结构。
- Win32 编程。消息、输出、绘图、网络、对话框和用户输入。
- 网络。TCP/IP、以太网和 WinSock。

最后，我要感谢 Addison Wesley Longman 的 Elaine Richardson（高级责任编辑）和 Karen Mossman（编辑），感谢他们的辛勤工作和一贯支持。我也要感谢我的家庭，Julie、Billy、Jamie 和 David，感谢他们的爱和理解。

更多的信息和源代码可以在 Addison Wesley Longman 的网站或者在 http://www.eece.napier.ac.uk/~bill_b/ad_pc.html 上获得。

需要帮助可以通过电子邮件 w.Buchanan@napier.ac.uk 与我联系。



目 录

前 言

1	8086 处理器	1
1.1	序言	1
1.2	8088 微处理器	2
1.3	内存段	6
1.4	处理器内部透视	8
1.5	机器代码和汇编语言	9
1.6	练习	10
1.7	内存地址索引	11
2	8086/8088 指令	12
2.1	汇编语言助记符	12
2.2	字符和数值	19
2.3	注释	19
2.4	传送 (MOV)	19
2.5	存储器寻址	20
2.6	加法和减法	22
2.7	比较 (CMP)	22
2.8	一元指令	23
2.9	逻辑位运算指令	23
2.10	移位/循环指令	24
2.11	无条件跳转	26
2.12	条件跳转	26
2.13	子程序调用	27
2.14	PUSH 和 POP	28
2.15	在内存中移动数据	29
2.16	汇编伪指令	30

2.17	数据定义	30
2.18	等值 (EQU)	32
2.19	练习	32
3	80386/80486	38
3.1	序言	38
3.2	80486 引脚图	38
3.3	80386/80486 寄存器	42
3.4	高速缓存	43
3.5	DMA	48
4	Pentium/Pentium Pro	50
4.1	序言	50
4.2	Intel 处理器的发展	50
4.3	有关术语	52
4.4	Pentium II 和 Pentium Pro	53
4.5	系统概述	53
4.6	MMX 技术	57
5	Pentium II	58
5.1	序言	58
5.2	微体系概况	59
5.3	系统总线概况	64
5.4	操作模式	71
6	8086 接口和时序	73
6.1	序言	73
6.2	与内存接口	73
6.3	内存映射 I/O 端口	74
6.4	独立 I/O 端口	74
6.5	计时	79
7	UART 和 PIC	83
7.1	序言	83
7.2	通用异步数据收发器 (8250)	83
7.3	可编程中断控制器 (8259)	93

8	PPI 和 PTC	94
8.1	序言	94
8.2	可编程外围接口 (8255)	94
8.3	可编程时钟控制器 (8254)	105
8.4	练习	115
9	中断与 BIOS	119
9.1	序言	119
9.2	BIOS 和操作系统	120
9.3	中断向量	121
9.4	处理器中断	124
9.5	硬件中断	124
9.6	触发软件中断	131
9.7	练习	136
10	接口标准	139
10.1	序言	139
10.2	PC 总线	139
10.3	ISA 总线	141
10.4	MCA 总线	145
10.5	EISA 总线	147
10.6	总线类型之间的比较	147
10.7	练习	148
11	局域总线	150
11.1	序言	150
11.2	VESA VL 局域总线	150
11.3	PCI 总线	152
11.4	练习	163
11.5	厂商和即插即用编号举例	164
12	主板设计	165
12.1	序言	165
12.2	Pentium 处理器	166
12.3	82381 SB PCI ISA X 加速芯片 (PIIX3)	167
12.4	82438 系统控制器 (TXC)	170
12.5	错误检测和纠正	174

12.6	PCI 接口	174
12.7	82091AA (AIP)	174
12.8	DRAM 接口	175
12.9	时钟速度	176
12.10	ISA/IDE 接口	177
12.11	DMA 接口	178
12.12	间隔定时器	178
12.13	中断控制器	178
12.14	鼠标功能	179
12.15	电源控制	179
12.16	USB	180
12.17	鼠标和键盘接口	180
12.18	ATX 主板实例	181
12.19	练习	182
13	IDE 和海量存储设备	183
13.1	序言	183
13.2	磁道和扇区	183
13.3	软盘	184
13.4	固定磁盘	185
13.5	驱动器规格	186
13.6	硬盘/CD-ROM 接口	186
13.7	IDE 接口	188
13.8	IDE 通信	189
13.9	文件系统	195
13.10	处理磁盘错误	198
13.11	光学存储设备	200
13.12	磁带	204
13.13	练习	205
14	SCSI	207
14.1	序言	207
14.2	SCSI 类型	207
14.3	SCSI 接口	209
14.4	SCSI 操作	212
14.5	SCSI 指针	215
14.6	消息系统简述	215

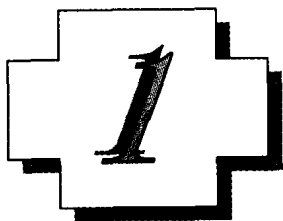
14.7	SCSI 命令	216
14.8	状态	219
14.9	练习	220
15	PCMCIA	222
15.1	序言	222
15.2	PCMCIA 连接	222
15.3	PCMCIA 信号	223
15.4	PCMCIA 寄存器	224
15.5	练习	228
16	RS-232	230
16.1	序言	230
16.2	电气特性	230
16.3	帧格式	232
16.4	两个节点之间的通信	232
16.5	RS-232 编程	237
16.6	RS-232 程序	237
16.7	标准 Windows 串行通信程序	242
16.8	练习	247
17	中断驱动的 RS-232	252
17.1	中断驱动的 RS-232	252
17.2	Win32 程序	252
17.3	DOS 下的 RS-232 程序	253
17.4	练习	263
18	并行端口	266
18.1	序言	266
18.2	数据握手	267
18.3	I/O 寻址	270
18.4	练习	275
19	中断驱动的并行端口	279
19.1	序言	279
19.2	中断	279
19.3	实例程序	279

19.4	程序说明	283
19.5	练习	285
20	增强的并行端口	286
20.1	序言	286
20.2	IEEE1284 数据传输模式	286
20.3	兼容模式	287
20.4	半字节模式	287
20.5	字节模式	289
20.6	EPP	290
20.7	ECP	291
20.8	1284 协商	294
20.9	练习	295
21	Modem	297
21.1	序言	297
21.2	RS-232 通信	298
21.3	Modem 标准	299
21.4	Modem 命令	300
21.5	Modem 配置	303
21.6	Modem 指示器	304
21.7	配置文件查看	304
21.8	测试模式	305
21.9	数字调制	308
21.10	典型的 Modem	310
21.11	传真发送	311
21.12	练习	313
22	使用 VB 的串行通信编程	314
22.1	VB 简介	314
22.2	表单	315
22.3	菜单	324
22.4	事件	327
22.5	图像和定时器	328
22.6	串行通信	330
22.7	练习	343

23	Windows 注册表	344
23.1	序言	344
23.2	Window 95/98 和注册表	345
23.3	Windows NT 和注册表	349
23.4	INF 文件	350
23.5	练习	355
23.6	INF 文件范例	356
24	Win32 简介	360
24.1	序言	360
24.2	Win32 和 Window 3.x	361
24.3	异常事件	362
25	Win32 基础	364
25.1	序言	364
25.2	主程序	364
25.3	创建窗口	366
25.4	应用程序实例	378
25.5	其他的 Windows 配套函数	385
25.6	练习	387
26	Windows 消息系统	389
26.1	序言	389
26.2	Message 结构	389
26.3	Message 函数	390
26.4	消息	392
26.5	练习	404
27	Windows 输出系统	405
27.1	序言	405
27.2	设备描述表	405
27.3	文本输出	406
27.4	GDI 函数	407
27.5	练习	415
28	绘图和着色	416
28.1	序言	416

28.2	颜色.....	416
28.3	绘图和着色函数.....	417
28.4	结构.....	433
28.5	练习.....	434
29	网络基础.....	437
29.1	序言.....	437
29.2	OSI 模型.....	438
29.3	传输标准和 OSI 模型.....	441
29.4	标准机构.....	441
29.5	网络电缆型号.....	442
29.6	LAN 拓扑结构.....	443
29.7	互联网络连接.....	444
29.8	Internet 布线协议.....	448
29.9	网络拓扑结构.....	452
29.10	练习.....	454
30	以太网.....	455
30.1	序言.....	455
30.2	IEEE 标准.....	456
30.3	以太网-MAC 层.....	456
30.4	IEEE 802.2 和以太网 SNAP.....	458
30.5	OSI 和 IEEE 802.3 标准.....	460
30.6	以太网收发器.....	461
30.7	NIC.....	462
30.8	标准以太网的局限性.....	466
30.9	以太网类型.....	468
30.10	双绞网络集线器.....	469
30.11	100Mbps 以太网.....	470
30.12	接线器和交换集线器.....	476
30.13	各快速以太网的比较.....	479
30.14	练习.....	479
31	TCP/IP.....	481
31.1	序言.....	481
31.2	TCP/IP 网关和主机.....	482
31.3	IP 的功能.....	482

31.4	互联网数据报	483
31.5	ICMP	484
31.6	TCP/IP 互联网	486
31.7	域名系统	490
31.8	因特网命名结构	490
31.9	DNS	491
31.10	bootp 协议	493
31.11	网络实例	495
31.12	IP v6	498
31.13	TCP	499
31.14	TCP/IP 命令	502
31.15	练习	508
32	WinSock 编程	510
32.1	序言	510
32.2	Windows 套接字	510
32.3	实践 Win32 编程	522
32.4	练习	522
附录 A	RS-232/并行端口	523
A.1	RS-232	523
A.2	并行端口	525
A.3	PC 连接	527
A.4	并行端口连接	528
附录 B	Modem 代码	530
B.1	AT 命令	530
B.2	结果代码	535
B.3	S 寄存器	536



8086 处理器

1.1 序 言

Intel (Intel 是 *Integrated Electronics* 的缩写) 公司首次推出的微处理器被命名为 4004, 它在电子工业中引发了一场革命, 原因是早先的电子系统都只有一种固定的功能性。随着处理器的出现, 这种功能性就可以用软件编程来实现了。但是以现在的标准看, 它只有 2000 个晶体管, 46 条指令, 只允许有 4kB 的程序编码和 1kB 的数据, 每次也只能处理一次四位数据的运算 (半个字节)。就是从这样简陋的开始, PC 机从此进入了使用 Intel 微处理器的全新发展时期。

第二代 Intel 微处理器于 1974 年面世, 一次可以处理八位数据的运算, 它们被命名为 8008、8080 和 8085。它们比早先的四位处理器功能要强大得多, 在早期的微机和其他一些设备, 例如打印机和电子仪器中得到了广泛的运用。8008 处理器有十四位的地址总线, 这样就可以为 16kB 内存空间寻址 (8080 地址总线有 16 位, 最多可以为 64kB 寻址)。

第三代微处理器是从十六位处理器投入使用开始的。Intel 公司推出的 8086 微处理器主要是 8080 处理器原型的扩展, 这样就在一定程度上保留了软件的兼容性。IBM 公司的设计人员注意到 8086 微处理器的功能, 并将它用于最初的 IBM PC 机和 IBM XT (eXtended Technology) 机中。8086 处理器有十六位数据总线 (data bus) 和二十位地址总线, 这样最大就有 1MB 的寻址空间。8086 处理器可以同时处理八位数据的运算和十六位数据的运算 (虽然是以一种混乱的方式进行的)。

一种稍许退步的、只有八位外部数据总线的微处理器也出现了, 被命名为 8088。这种处理器允许让设计人员生产出较为简单, 也较为便宜的计算机系统。1982 年, 一种改进了体系结构的微处理器——80286 面世, 并被用于 IBM AT (Advanced Technology) 机中。

1985 年, Intel 公司推出了首款三十二位微处理器——80386DX。这款微处理器与以前的 8088/8086/80286(80x86)微处理器是兼容的, 并且有极佳的性能——可以同时处理八位数据、十六位或三十二位数据的运算。它有完整的三十二位数据总线和十六位地址总线, 这样就可以为 4GB 的物理内存寻址。一种稍许退步的、只有十六位外部数据总线和二十四位地址总线的处理器——80386SX, 也于 1988 年面世, 这种处理器只能为 16MB 的物理内存

寻址。

1989 年, Intel 公司推出了 80486DX 处理器。80486DX 其实基本上只是 80386DX 的改进版本, 它将高速缓冲存储器 (memory cache) 和数学协处理器 (math co-processor) 集中到了一块芯片上。它的内部结构有所改进, 这使得它比同等的 80386 要快大约 50%。80486SX 也面世了, 它仅仅是将 80486DX 的数学协处理器断开后加了个连接而已。具有时钟脉冲双倍/三倍倍频的 80486 处理器也面世了, 这种处理器的运算速度比系统时钟脉冲速度要快。一般来说, 装有时钟脉冲双倍倍频处理器的系统要比同等的装有非时钟脉冲双倍倍频处理器的系统快 75%。典型的时钟脉冲双倍倍频处理器是 DX2-66 和 DX2-50, 它们运行的时钟脉冲分别为 33MHz 和 25MHz。Intel 公司也推出了 80486 系列产品, 它们运行的速度为系统时钟脉冲速度的三到四倍, 被称为 DX4 处理器, 其中包括 Intel DX4-100 (时钟脉冲 25MHz) 和 Intel DX4-75 (时钟脉冲 25MHz)。

Pentium 处理器 (或称为 P5) 是六十四位超标量 (superscalar) 处理器, 它可以一次处理一条以上的指令。P5 有完整的六十四位 (八字节) 数据总线 and 三十二位地址总线。在性能方面, 它的运算速度几乎是同等 80486 的两倍, 它改进了浮点运算 (大概快了三倍), 并且与以前的 80x86 处理器完全兼容。

Pentium II 处理器 (或 P6) 是 P5 的增强版本。它的总线可以同时支持四个处理器, 而且无需额外逻辑电路的支持, 它的时钟脉冲增值速度超过 300MHz, 而且 P6 省电性能和 EMI (electro magnetic interference), 抗电磁干扰性能都很好。P6 总线一个极大的增强功能是它既可以检测和纠正所有的单位数据总线错误, 也可以检测多位数据总线错误。

1.2 8088 微处理器

十六位 8086 处理器引发了一场处理能力的重要革命。8086 处理器拥有二十位地址总线和十六位数据总线, 而 8088 处理器拥有八位外部数据总线。图 1-1 所示的是 8088 处理器的引脚和处理器的主要连接方式。8086 处理器的四十个引脚中大部分都有双重功能。例如: 引脚 AD0~AD7 既可以用作地址总线的低八位 (A0~A7), 也可以用于数据总线的低八位 (D0~D7)。引脚 A16/S3~A19/S6 也有双重功能, S3~S6 在正常情况下不会被 PC 机使用, 这样它们通常被用作地址总线的高四位。当引脚 ALE (address latch enable) 从高电位跳到低电位时, 从上述引脚传输的数据就被锁定为地址。

总线控制器 (8288) 根据 8088 处理器的状态引脚 $\overline{S0}$ ~ $\overline{S2}$ 产生所需的控制信号。例如: 如果 $\overline{S0}$ 是高电位、 $\overline{S1}$ 是低电位, $\overline{S2}$ 是低电位, 那么引脚 $\overline{MEMR}$ 就变成低电位。主要的控制信号如下:

- $\overline{IOR}$ (I/O 读): 该信号表示处理器正在读入 I/O 总线上的地址内容。
- $\overline{IOW}$ (I/O 写): 该信号表示处理器正在将数据总线上的内容当作地址写入 I/O 总线。
- $\overline{MEMR}$ (内存读): 该信号表示处理器正在读入地址总线上的地址内容。
- $\overline{MEMW}$ (内存写): 该信号表示处理器正在将数据总线上的内容写到地址总线所

指定的地址中去。

- $\overline{\text{INTA}}$ （中断肯定应答）：该信号处理器用于肯定应答中断（ $\overline{\text{S0}}$ 、 $\overline{\text{S1}}$ 和 $\overline{\text{S2}}$ 都跳到低电位）。当外围设备有指令需要处理器来处理时，它就发一个中断请求给 8259，如果允许的话，8259 将 $\overline{\text{INTR}}$ 置为高电位。

处理器既可以通过引脚 $\overline{\text{MEMW}}$ 和 $\overline{\text{MEMR}}$ 直接与内存通信，也可以通过独立的 I/O 端口（信号 $\overline{\text{IOR}}$ 和 $\overline{\text{IOW}}$ ）与外围设备通信。

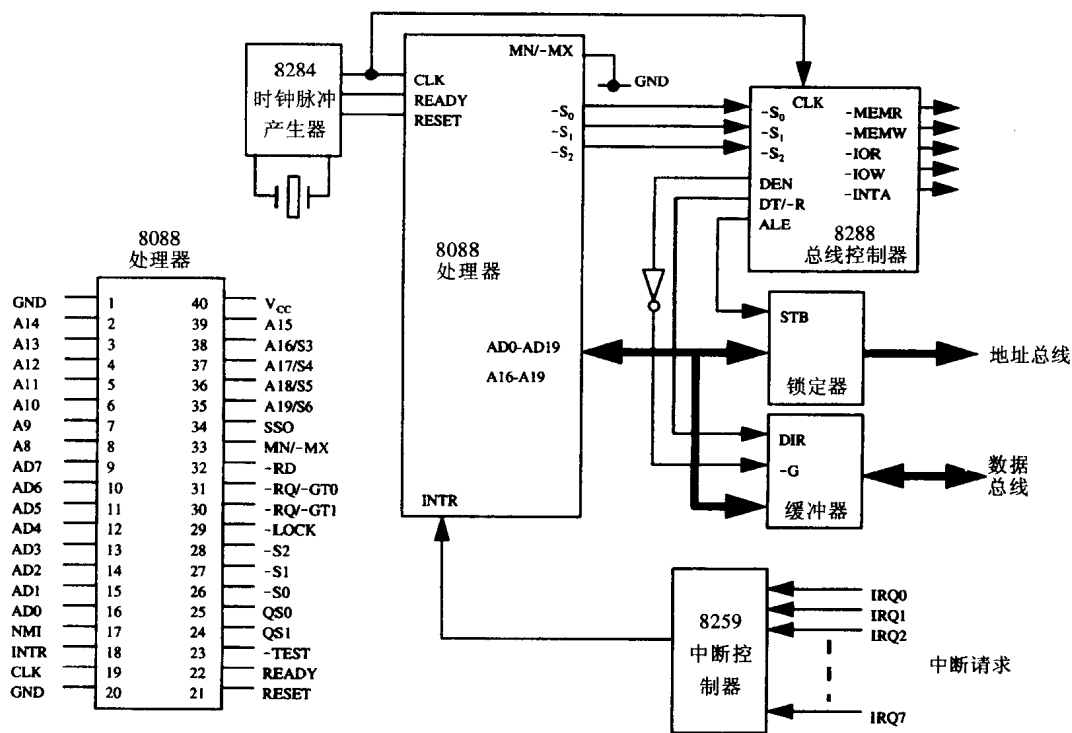


图 1-1 8088 的主要连接方式

1.2.1 寄存器

每个基于 PC 机的 Intel 微处理器都与最初的 8086 处理器兼容，并且一般也与其以前的各款处理器兼容。举例来说，Pentium 可以运行 8086、80386 和 80486 处理器的代码。微处理器是通过使用寄存器（register）来实现其功能的，这些寄存器基本上是处理器中拥有特殊名称的专用内存储存单元。8088/8086 处理器有十四个寄存器，可以分成四类，如图 1-2 所示。

※ 通用寄存器

共有四个通用寄存器（general-purpose register），分别是 AX、BX、CX 和 DX。每一个通用寄存器都可以用于处理一个十六位的字（word）或两个八位的字节（byte），这两个八位字节分别被称为低段字节和高段字节。每个这样的寄存器都可以被用作两个八位寄

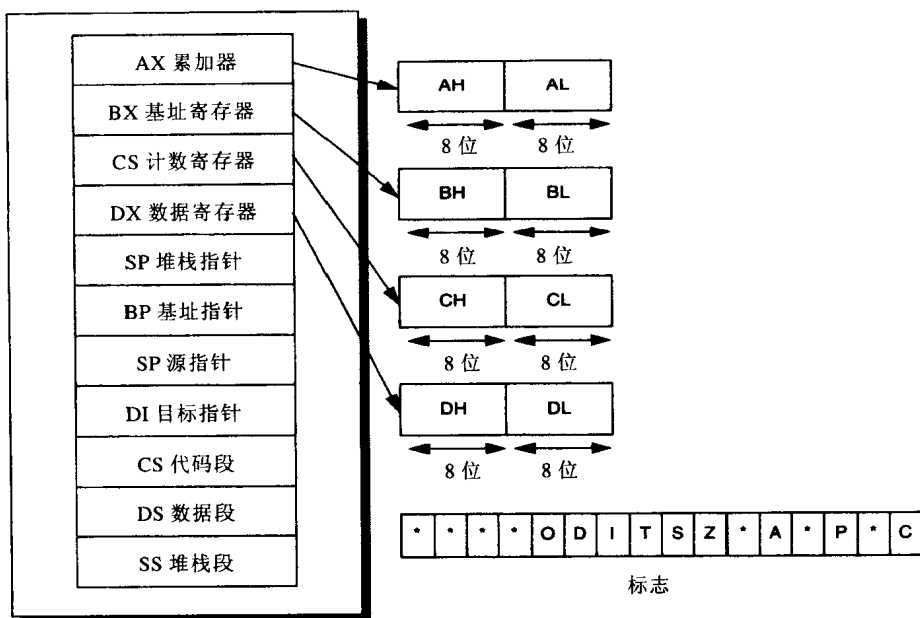


图 1-2 8086/8088 寄存器

寄存器。例如，AL 表示一个 AX 的低八位组成的八位寄存器，而 AH 表示一个 AX 的高八位组成的八位寄存器。

AX 寄存器是四个寄存器中最常用的，它通常用于各种类型的操作运算中。其他每个寄存器都有一个或是多个暗含的功能，具体如下：

- AX 是累加器。它用于所有的输入输出操作和一些算术操作中，例如，乘、除和翻译指令都要用到 AX 寄存器。
- BX 是基址寄存器。它被用作是地址寄存器。
- CX 是计数寄存器。它被用于需要计数的指令中，典型的就是用于控制循环次数和移位操作中。
- DX 是数据寄存器。它用于一些输入输出操作中，也用于乘和除的操作中。

※ 地址寄存器

地址寄存器（addressing register）用于内存寻址操作中，例如存储内存的源地址和目的地址。这些寄存器被命名为 BP、SP、SI 和 DI，具体如下：

- SI 是源变址寄存器，用于扩展寻址命令。
- DI 是目的变址寄存器，用于某些寻址方式中。
- BP 是基址指针寄存器。
- SP 是栈指针寄存器。

※ 状态寄存器

状态寄存器（status register）用于检验一个操作的各种不同状态，例如“结果为负吗？”、