第二章

8086CPU

8086CPU内部结构 ,有两个独立的工作部件

- 执行部件 EU
- 总线接口部件 BIU。

执行部件EU

- 通用数据寄存器
  - AX，BX，CX，DX，既可以作 16 位寄存器使用，也可以分成高、低 8 位分别作两个8 位寄存器使用。专用数据寄存器 BP，SP，SI，DI都是 16 位寄存器，一般用来寻访地址信息。
- ALU
  - 核心是 16 位二进制加法器。其功能：一是进行算术/逻辑运算，二是按指令的寻址方式给出所需要操作对象的 16 位（偏移）地址，提供给 BIU，让 BIU 进行对内存储器或 I/O 空间的寻址，传输操作对象。
- 16 位状态标志寄存器
  - 存放操作后的状态特征和设置的控制标志。
- EU 控制器
  - 执行指令的控制电路，实现从队列中取指令、译码、产生 控制信号等

总线接口部件（BIU）

- 负责CPU与存储器、I / O接口之间的信息传送
- 段寄存器
- 指令指针寄存器
- 指令队列
  - 由6个字节的寄存器组成（8088指令队列由4字节组成）。采用“先进先出”原则，暂时存放 BIU 从存储器中预取的指令
- 地址加法器
  - 地址加法器是用来产生 20 位存储器物理地址
    物理地址（20位）　= 段基址（16位）×16偏移地址（16位）
- 总线控制逻辑
  - 8086分配 20 条引脚线分时传送 20 位地址、16 位数据和 4 位状态信息。总线控制逻辑以逻辑控制方法实现分时把这些信息与外部传输

8086CPU的寄存器

通用寄存器

数据寄存器：AX、BX、CX、DX
地址寄存器：SI、DI、SP、BP

寄存器	功能
AX	累加器
BX	基址寄存器
CX	计数器
DX	数据寄存器
SP	堆栈指针
BP	基址指针
DI	目的变址
SI	源变址
IP	指令指针
FLAG	标志寄存器
CS	代码段寄存器
DS	数据段寄存器
SS	堆栈段寄存器
ES	附加段寄存器

EAX——通用寄存器中速度最快，称为累加器（Accumalator）。使用累加器使指令的机器代码更短、执行速度更快。AX、AL同。

EBX——可作为通用寄存器使用，经常用于指示存储器地址，故称基址寄存器（Base）。BX同。

ECX——可作为通用寄存器使用，常用于循环和串指令中的隐含计数，故称为计数器（Count）。CX同。CL常用于移位指令中的隐含计数。

EDX——数据寄存器（Data），用于在4字长运算中存放高位字；DX用于在双字长运算中存放高位字，以及在I/O中存放端口地址。

ESP、EBP、ESI、EDI可以象以上寄存器一样在运算中存放操作数，但主要是用于提供地址。

ESP、SP——Stack Pointer，堆栈指针寄存器 ，永远指向栈顶。通常不用于运算。

EBP、BP——Base Pointer，基址指针寄存器，指向堆栈中任何一数据。

以上ESP、EBP也称为**指针寄存器**

ESI、EDI——称为**变址寄存器。**

在串操作指令中：

ESI、SI源变址寄存器（Source Index） ，表示**源数据串**的地址；

EDI、DI目的变址寄存器（Destination Index），表示**目的源数据串的地址。**

专用寄存器

EIP、IP——Instruction Pointer，指令指针寄存器。与CS寄存器一起指向下一条指令的地址。在程序的执行过程中，EIP、IP的内容自动修改。程序员一般不能使用该寄存器。

EFLAGS——标志寄存器，也称为程序状态字（PSW）。8088的FLAGS是16位的，只用了其中的9位。有6位状态标志（Flag）和3位控制标志。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

标志位

6位**状态标志**
(1) OF（Overflow Flag）溢出标志
运算结果溢出时，OF=1；否则OF=0。只有对有符号数才有意义，有溢出时通过溢出中断处理加以解决。进位与溢出完全不同。

(2) SF（Sign Flag）符号标志
运算结果为负时，SF=1；否则SF=0。用于有符号运算。

(3) ZF（Zero Flag）零标志
运算结果为零时，ZF=1；否则ZF=0。用于分支、循环的转移控制。

(4) CF（Carry Flag）进位标志
运算时，最高有效位有进位或借位时，CF=1；否则CF=0。常用于多字节加、减，也用于条件转移。

(5) AF（Auxiliary Carry Flag）辅助进位标志
运算时，低4位向高4为有进位或借位时，AF=1；否则AF=0。用于十进制算术运算的调整指令，如DAA、DAS、AAA。

(6) PF（Parity Flag）奇偶标志
运算结果的低8位中，1的个数为偶数，PF=1；1的个数为奇数，PF=0。常用于逻辑运算中，也用来检查数据传输中是否有错。

三个**控制标志**用于控制CPU的工作。

(1) TF（Trap Flag）跟踪标志
当TF=1时，用于单步工作方式，便于调试程序；TF=0时，CPU正常工作。

(2) IF（Interrupt Flag）中断允许标志
当IF=1时，允许中断，即可接收外部可屏蔽中断；IF=0时，关闭中断。有指令STI和CLI。

（3) DF（Direction Flag）方向标志
用于字符串操作时，地址变化的方向。DF=1时，地址（SI、DI）减量，即从高地址到低地址；DF=0时，地址（SI、DI）增量，即从低地址到高地址。有指令CLD和STD。

标志位	1	0
OF	OV	NV
SF	NG	PL
ZF	ZR	NZ
CF	CY	NC
AF	AC	NA
PF	PE	PO
DF	DN	UP
IF	EI	DI

段寄存器

用于存放个逻辑段的段基址。

CS——Code Segment，代码段寄存器。划定和控制程序区。

DS——Data Segment，数据段寄存器。划定和控制数据区。

SS——Stack Segment，堆栈段寄存器。划定和控制堆栈区。

ES、FS、GS——Extra Segment，附加段寄存器。划定和控制数据区。

段寄存器的作用：通过“段地址:偏移地址”访问内存单元

存储器

在PC机中，存储器是以“字节”为基本存储单位。为了正确地存放或取得信息，每一个字节单元必须有一个唯一的存储单元地址，称为物理地址或绝对地址。

从存储器“读出”或“写入”数据必需是“字节”的倍数，是字节、字、双字等。

存储器地址组织

存储器地址——从0开始，顺序给存储单元的编号。

- 从0开始编号，顺序加1。
- 机内地址用二进制表示，是无符号整数。
- 为了书写方便，地址一般也用十六进制表示（内容也一样）。

存储器内数据组织

在计算机中，信息的基本单位是一个二进制位（Bit），0或1。

8位（Bit）=1字节（Byte），如10110101
16Bit=2Byte=1字（Word），如10110101 00110110

字节单元

IBM-PC采用字节单元编址。如
（0004H）=78H
（0005H）=56H

字单元

字单元是相邻的（即连续的）两个字节单元存放一个字 (**16位**) 。原则是：**（较）低地址存放低字节；（较）高地址存放高字节**，以其中的低地址来表示字单元的地址。取数据的时候,高字节放在前面,低字节放在后面.

双字单元

双字单元是连续的4个字节单元存放一个双字。原则是：
低地址存放低字节，高地址存放高字节，
以其中的最低地址来表示字单元的地址。

内存的分段结构

规定：每个逻辑段的首地址必须是能被16整除的边界开始，即首地址为xxxx0H。如00000H，00010H，……，41230H，41240H，……，FFFE0H，FFFF0H。

(1) 物理地址——又称绝对地址，表示存储单元距离整个存储器起始地址的字节数。

(2) 偏移地址——是指在段内相对于段起始地址的偏移值（字节数）。

(3) 段地址——逻辑段的起始地址，应该是**20位的物理地址**。由于它的低四位一定是0。所以规定段地址**只取段起始地址的高16位**。

引入段寄存器，使存储器地址的分段成为可能。Intel 8088中有4个16位的段寄存器，每个段寄存器可以确定（存放）一个段的起始地址。
不同的段有不同的用处：

- 代码段：存放当前正在运行的程序的代码；
- 数据段：存放当前正在运行的程序所用的数据；
- 附加段：存放当前正在运行的程序所用的附加数据；
- 堆栈段：定义了堆栈所在的区域。

程序员在汇编语言程序中使用“**段地址：偏移地址**”的逻辑地址表示法。

物理地址=段地址×16+偏移地址

=段地址左移4位+偏移地址

一个段最大定义64K个字节