Intel 微处理器

1 导论

2 微处理器及其体系结构

2.2.2 默认段和偏移寄存器

合用于寻址程序的下一条指令。根据微处理器的操作模式，这种组合是 **CS：IP** 或者 **CS：EIP**。**代码段**（**code segment**）寄存器定义代码段的起点，**指令指针**（**instruction pointer**）指示代码段内的下一条指令的位置。这样的组合（**CS：IP** 或 **CS：EIP**）定位微处理器执行的下一条指令。例如，如果 CS = 1400H 且 IP/EIP = 1200H，则微处理器从存储器的 14000H + 1200H 单元（即 15200H 单元）取下一条指令。

另外一种默认组合用于**堆栈**（**stack**）。通过栈指针（SP/ESP）或者基指针（BP/EBP）寻址堆栈段中某存储单元的堆栈数据。这些组合用 SS：SP（SS：ESP）或者 SS：BP（SS：EBP）表示。例如，如果

- 可重定位程序
- 可重定位数据

2.3保护模式存储器寻址

- 全局描述符（系统描述符）
- 局部描述符（应用描述符）
- 每个描述符表可包含8192个描述符
- 描述符组成：基地址，段界限，粒度位（G），L位（64位），段有效位（AV位），兼容位（D位），访问权限字节

3 寻址方式

数据寻址方式

- `intel` 所有型号微处理器有相同的寻址方式，比例变址寻址方式除外，他只用于`80386 ~ Core2` ...（当时只有`Core2`？），

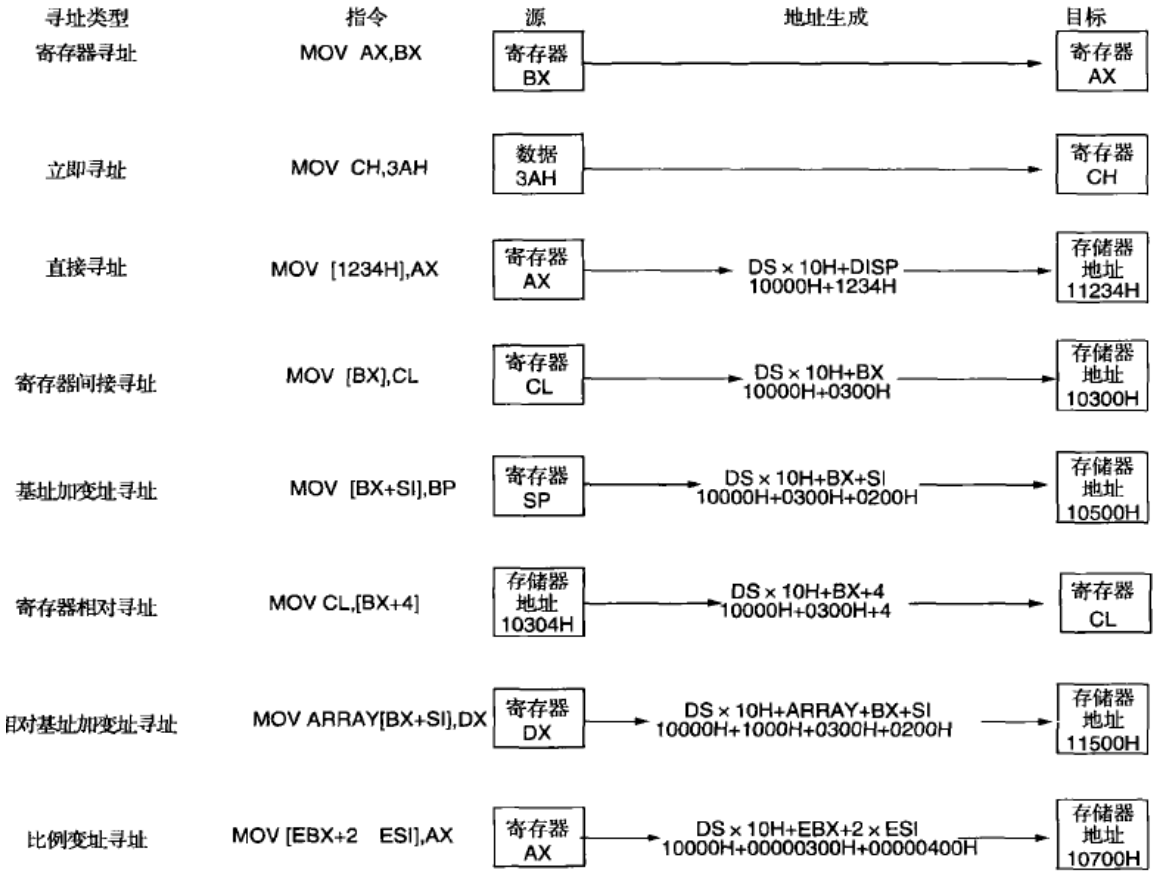
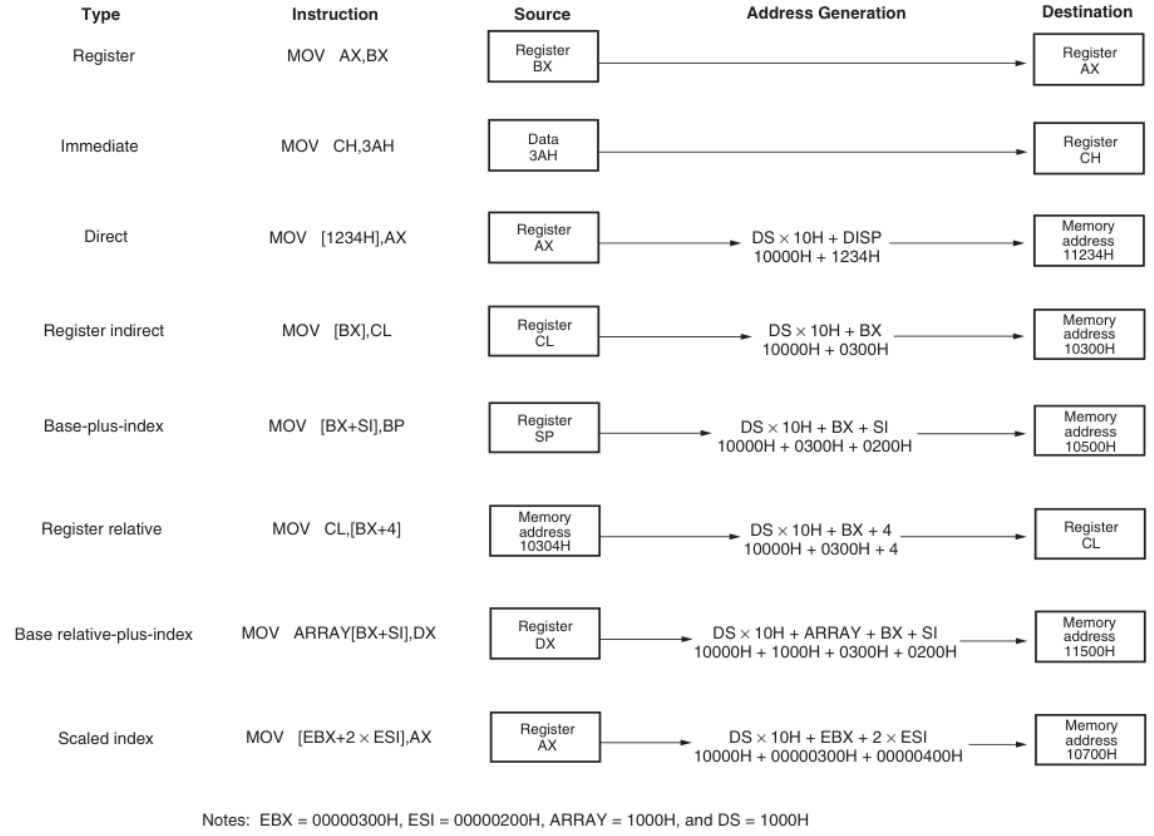


图 3-2　8086 ~ Core2 的数据寻址方式

注：EBX = 00000300H，ESI = 00000200H，ARRAY = 1000H，和 DS = 1000H。



Notes: EBX = 00000300H, ESI = 00000200H, ARRAY = 1000H, and DS = 1000H

- **RIP相对寻址：**

- **寄存器寻址**：即在寄存器之间传输数据，注意不能在段寄存器之间寻址以及不能通过、mov改变代码段寄存器的值。

表 3-1　寄存器寻址指令的例子

汇编语句	长　度	操　作
MOV AL，BL	8 位	把 BL 复制到 AL 中
MOV CH，CL	8 位	把 CL 复制到 CH 中
MOV R8B，CL	8 位	把 CL 复制到为 R8 的字节部分（64 位模式下）
MOV R8B，CH	8 位	不允许
MOV AX，CX	16 位	把 CX 复制到 AX 中
MOV SP，BP	16 位	把 BP 复制到 SP 中
MOV DS，AX	16 位	把 AX 复制到 DS 中
MOV BP，R10W	16 位	把 R10 复制到 BP（64 位模式下）
MOV SI，DI	16 位	把 DI 复制到 SI 中
MOV BX，ES	16 位	把 ES 复制到 BX 中
MOV ECX，EBX	32 位	把 EBX 复制到 ECX 中
MOV ESP，EDX	32 位	把 EDX 复制到 ESP 中
MOV EDX，R9D	32 位	把 R9 复制到 EDX（64 模式下）
MOV RAX，RDX	64 位	把 RDX 复制到 RAX
MOV DS，CX	16 位	把 CX 复制到 DS 中
MOV ES，DS	—	不允许（段到段）
MOV BL，DX	—	不允许（长度不同）
MOV CS，AX	—	不允许（代码段寄存器不能作为目的寄存器）

- **立即寻址**：即通过立即数对寄存器赋值（向存储器中赋值应该是定义数据例如dd,dw, db?）

表 3-2　使用立即寻址的 MOV 指令示例

汇编语句	长　度	操　作
MOV BL，44	8 位	把十进制数 44（2CH）传送到 BL 中
MOV AX，44H	16 位	把十六进制数 44 传送到 AX 中
MOV SI，0	16 位	把 0000H 传送到 SI 中
MOV CH，100	8 位	把十进制数 100（64H）传送到 CH 中
MOV AL，‘A’	8 位	把 ASCII A（41H）传送到 AL 中
MOV AH，1	8 位	64 位模式下不允许，在 32 位和 16 位模式下允许
MOV AX，‘AB’	16 位	把 ASCII 码 BA① （4241H）传送到 AX 中
MOV CL，1100 1110B	8 位	把二进制数 1100 1110 传送到 CL 中
MOV EBX，1234 0000H	32 位	把 12340000H 传送到 EBX 中
MOV ESI，12	32 位	把十进制数 12 传送到 ESI 中
MOV EAX，100B	32 位	把二进制数 100 传送到 EAX 中
MOV RCX，100H	64 位	把 100H 复制到 RCX

① 这不是错误，因为当用一个字存储两个 ASCII 字符时，ASCII 字符存储为 BA。

注意：对于数字来说，左侧为高位，右侧是低位；但是对于字符串来说，左侧低位，右侧高位。

内存寻址：`Effective Address = Base + (Scale * Index) + Disp`

- Direct Data Addressing (Disp)
- Register Indirect Addressing (Base)
- Base-Plus-Index Addressing (Base + Index)
- Register Relative Addressing (Base/Index + Disp)
- Base Relative-Plus-Index Addressing (Base + Index + Disp)
- Scaled-Index Addressing (Base+Scale+Index+Disp)

8种?

- **直接数据寻址**：通过标记或者直接的内存地址格式（段地址:[偏移地址]）进行数据寻址。
- 有两种指令格式：对于与 `AX,AL,EAX,RAX` 来说，由于使用比较频繁，有专门的操作码，使得指令只需要3个字节；而另外的寄存器都需要4个字节。

表 3-3　使用 EAX、AX、AL 和 64 位模式下的 RAX 的直接寻址指令

汇编语句	长　度	操　作
MOV AL，NUMBER	8 位	将数据段存储单元 NUMBER 中的字节内容复制到 AL 中
MOV AX，COW	16 位	将数据段存储单元 COW 中的字内容复制到 AX 中
MOV EAX，WATER①	32 位	将数据段存储单元 WATER 中的双字内容复制到 EAX 中
MOV NEWS，AL	8 位	将 AL 的内容复制到字节存储单元 NEWS 中
MOV THERE，AX	16 位	将 AX 的内容复制到字存储单元 THERE 中
MOV HOME，EAX①	32 位	将 EAX 的内容复制到双字存储单元 HOME 中
MOV ES:[2000H]，AL	8 位	将 AL 的内容复制到附加数据段存储单元 2000H 中
MOV AL，MOUSE	8 位	将 MOUSE 单元的内容复制到 AL；在 64 位模式中 MOUSE 可以是任何地址
MOV RAX，WHISKEY	64 位	将存储单元 WHISKEY 的 8 个字节复制到 RAX 中

① 80386～Pentium 4 微处理器为了在 EAX 与存储器之间移动 32 位数，有时需要多于 3 字节的存储器。

表 3-4　使用位移量的直接数据寻址的示例

汇编语句	长　度	操　作
MOV CH，DOG	8 位	把数据段存储单元 DOG 的字节内容装入 CH 中（DOG 的偏移地址由汇编程序计算）
MOV CH，DS:[1000H]①	8 位	把数据段存储单元 1000H 的字节内容装入 CH 中
MOV ES，DATA 6	16 位	把数据段存储单元 DATA6 的字内容装入 ES 中
MOV DATA7，BP	16 位	把寄存器 BP 的内容复制到数据段存储单元 DATA7 中
MOV NUMBER，SP	16 位	把 SP 的内容复制到数据段存储单元 NUMBER 中
MOV DATA1，EAX	32 位	把 EAX 的内容复制到数据段存储单元 DATA1 中
MOV EDI，SUM1	32 位	把数据段存储单元 SUM1 的双字内容装入 EDI 中

① 多数汇编程序很少使用这种寻址模式，因为在程序中很少访问实际数字的偏移地址。

- 寄存器间接寻址
- 访问数组
- 两个操作数不能同时为寄存器间接寻址。
- 立即数存入内存需要指定长度。

表 3-5　寄存器间接寻址的示例

汇编语句	长　度	操　作
MOV CX，[BX]	16 位	把数据段中由 BX 寻址的存储单元的字内容复制到 CX 中
MOV [BP]，DL①	8 位	把寄存器 DL 的内容复制到堆栈段由 BP 寻址的存储单元中
MOV [DI]，BH	8 位	把寄存器 BH 的内容复制到数据段由 DI 寻址的存储单元中
MOV [DI]，[BX]	—	除了串指令以外，不允许存储器到存储器的传送
MOV AL，[EDX]	8 位	把数据段由 EDX 寻址的存储单元的字节内容复制到 AL
MOV ECX，[EBX]	32 位	把数据段由 EBX 寻址的存储单元的双字内容复制到 ECX
MOV RAX，[RDX]	64 位	把 RDX 中由线性地址确定的存储单元的四字内容复制到 RAX（64 位模式下）

① 由 BP 或 EBP 寻址的数据默认为在堆栈段中，而所有其他间接寻址指令默认使用数据段。

- **基址加变址寻址**

表 3-6　基址加变址寻址的示例

汇编语句	长　度	操　作
MOV CX, [BX + DI]	16 位	把由 BX + DI 寻址的数据段存储单元内的字内容装入 CX
MOV CH, [BP + SI]	8 位	把由 BP + SI 寻址的堆栈段存储单元内的字节内容装入 CH
MOV [BX + SI], SP	16 位	把 SP 的内容存入由 BX + SI 寻址的数据段存储单元
MOV [BP + DI], AH	8 位	把 AH 的内容存入由 BP + DI 寻址的堆栈段存储单元
MOV CL, [EDX + EDI]	8 位	把由 EDX + EDI 寻址的数据段存储单元内的字节内容装入 CL
MOV [EAX + EBX], ECX	32 位	把 ECX 中的双字存入由 EAX + EBX 寻址的数据段存储单元
MOV [RSI + RBX], RAX	64 位	把由 RSI + RBX 寻址的线性存储单元装入 RAX

- 寄存器相对寻址
- 访问结构体元素

表 3-7　寄存器相对寻址的示例

汇编语句	长　度	操　作
MOV AX, [DI + 100H]	16 位	把由 DI + 100H 寻址的数据段存储单元中的字内容装入 AX
MOV ARRAY [SI], BL	8 位	把 BL 中的字节存入由 ARRAY + SI 寻址的数据段存储单元
MOV LIST [SI + 2], CL	8 位	把 CL 中的字节存入由 LIST + SI + 2 之和寻址的数据段存储单元
MOV DI, SET_IT [BX]	16 位	把由 SET_IT + BX 寻址的数据段存储单元的字内容装入 DI
MOV DI, [EAX + 10H]	16 位	把由 EAX + 10H 寻址的数据段存储单元的字内容装入 DI
MOV ARRAY [EBX], EAX	32 位	把 EAX 的内容存入由 ARRAY + EBX 寻址的数据段存储单元中
MOV ARRAY [RBX], AL	8 位	把 AL 的内容存入由 ARRAY + RBX 寻址的存储单元中（64 位）
MOV ARRAY [RCX], EAX	32 位	把 EAX 的内容存入由 ARRAY + RCX 寻址的存储单元中（64 位）

- 相对基址加变址寻址
- 访问结构体数组的内部元素

表 3-8　相对基址加变址寻址的示例

汇编语句	长　度	操　作
MOV DH, [BX + DI + 20H]	8 位	把由 BX、DI 及 20H 之和寻址的数据段存储单元的字节内容装入 DH
MOV AX, FILE [BX + DI]	16 位	把由 FILE、BX 及 DI 之和寻址的数据段存储单元的字节内容装入 AX
MOV LIST [BP + DI], CL	8 位	把 CL 存储到由 LIST、BP 及 DI 之和寻址的堆栈段存储单元中
MOV LIST [BP + SI + 4], DH	8 位	把 DH 存储到由 LIST、BP、SI 及 4 之和寻址的堆栈段存储单元中
MOV EAX, FILE [EBX + ECX + 2]	32 位	把由 FILE、EBX、ECX 及 2 之和寻址的数据段存储单元的双字内容装入 EAX

- 全表达式，比例变址寻址
- `scale`取值为1,2,4,8，原因在于scale值只设置了两个位

表 3-9　比例变址寻址的示例

汇编语句	长　度	操　作
MOV EAX，［EBX +4 * ECX］	32 位	把由 EBX 加 4 倍 ECX 之和寻址的数据段存储单元的双字内容装入 EAX
MOV［EAX +2 * EDI +100H］，CX	16 位	把 CX 的内容存储到由 EAX 加 2 倍 EDI 再加 100H 寻址的数据段存储单元中
MOV AL，［EBP +2 * EDI +2］	8 位	把由 EBP 加 2 再加 2 倍 EDI 寻址的堆栈段存储单元的字节内容装入 AL
MOV EAX，ARRAY［4 * ECX］	32 位	把由 ARRAY 加 4 倍 ECX 寻址的数据段存储单元的双字内容装入 EAX