周明德 白晓笛 编著

高档微型计算机

上 册

清华大学出版社

高档微型计算机

（上）

周明德　白晓笛　编著

清华大学出版社

内 容 简 要

本书分上、下两册，上册以Motorola的68020为主，下册以Intel的80386为主，系统地介绍了微处理器的结构，寻址方式，指令系统，汇编语言程序设计，时序，微处理器的状态，中断等异常处理，特权和保护以及主处理器与协处理器的接口，协处理器的原语等。

本书中收集和整理了大量最新资料，又注意到循序渐进的原则，并辅以大量的实例，是作者原《微型计算机硬件、软件及其应用》一书在16位、32位微型计算机方面的延伸和发展。

本书是高档微机干部培训教材，也是计算机专业本科生、研究生的教材，也可作为从事微型机工作的科技人员的参考书。

高档微型计算机

（上）

周明德 白晓笛 编著

清华大学出版社出版

北京 清华园

北京京辉印刷厂印刷

新华书店北京发行所发行

*

开本：787×1092 1/16 印张：19 字数：486 千字

1987年11月第1版 1987年11月第1次印刷

印数：00001—15000

统一书号：15235·334 定价：3.90元

前　言

微型计算机的发展极其迅速，在短短的十五年中已经历了四代。1971年生产的Intel 4004的集成度为2200个管子，而1985年推出的Intel 80386，集成度为27万个管子；字长也由4位发展为全32位(数据总线和地址总线的宽度全是32位的)；运算速度由每秒几万次发展到每秒执行几百万条指令；功能上也有了极大的提高，Intel 80386 的功能已经超过了VAX11/780。

目前，世界上各种、各类微型计算机的年产量已经超过了一千万台，微型计算机得到了空前广泛和深入的应用。一位机、四位机、八位机、十六位机、三十二位机，单片机、单板机、个人计算机等百花争艳各显其能，各在其适用的场合充分发挥作用。但是，微型计算机的生长点是32位机。

32位是比较合理的字长，能有效地综合处理数据、文字、图形和声音等各种信息，在工程计算、数据处理、事务管理、实时控制以及在CAD/CAM方面都能较好地完成任务。

一个较大企业的全面管理或较大范围的办公自动化系统，往往要求有10个或更多的终端，要求有相当大的外存容量等，这就需要用到32位超级微型机。

实时控制常常需要有分级分布系统，另外常需要把控制与管理结合起来。这样，下位机常常可以采用八位（或十六位）的单板机、单片机，而需要控制与管理许多个下位机的上位机，特别是需要有较强的数据处理和管理功能的上位机就需要用到32位超级微型机。

目前，CAD/CAM 已在各行各业得到了广泛应用。但是，若要有较强的较复杂的设计功能，有比较丰富的各种数据库（如零件库、尺寸库等等）支持，也就需要用到32位机这样的档次。

总之，应用的需要迫切要求发展32位超级微型计算机，而 VLSI 的发展也提供了生产 32 位机的可能。

16位、32位微处理器的产品虽然也很多，但有标准化的趋势，最著名的仍是Intel和 Motorola两大系列：Intel的有8086/8088—80186—80286—80386，Motorola的有68000/68008—68010—68020。

考虑到我国的实际情况，国内的半导体和微型计算机专家们主张，国产的16位、32位微型计算机的CPU，走与Intel和Motorola兼容的道路。

微型计算机发展的实践教育了人们，各种档次的CPU应该兼容。所以，Intel和Motorola公司从16位CPU开始注意到了后续产品与先期产品的兼容性。

Intel的8086/8088—80186—80286—80386，Motorola的68000/68008—68010—68020在指令系统和汇编语言上都是向上兼容的，即80386中包含了8086的全部指令，68020中也包含了68000的全部指令。

所以，为了便于由浅入深地学习，也为了实用，本书对Intel系列的论述从8086开始，对Motorola的论述从68000开始。

为了适应我国发展16位和32位微机的需要，为了满足广大读者的要求，本书针对Intel和Motorola 两大系列，分别从16位微处理器入手，详细分析了 16 位和32位微处理器的结构、

指令系统、汇编语言程序设计、信号与时序、工作状态与异常处理等。

本书分为上、下两册，上册论述Motorola系列从68000到68020，下册论述 Intel 系列从8086到80386。

本书编写时收集和整理了大量最新资料，又充分注意到循序渐进的原则，并辅以大量的实例，使本书具有先进性、科学性和实用性。

鉴于作者的水平，书中会有不少缺点和错误，热诚欢迎广大读者不吝赐教。

本书由周明德和白晓笛共同编写。

周明德

一九八六.十.十五

目　录

前言
第一章　MC 68000 的结构 …… 1
第一节　MC 68000 CPU的编程结构 …… 1
第二节　MC 68000 的存储器结构 …… 2
第二章　MC 68000 的寻址方式和指令系统 …… 5
第一节　寻址方式 …… 5
一、寄存器寻址 …… 5
二、立即寻址 …… 6
三、直接寻址 …… 7
四、间接寻址 …… 9
五、程序计数器相对寻址 …… 13
六、寻址方式的规定 …… 14
七、寻址方式的分类 …… 15
第二节　MC 68000 的指令系统 …… 16
一、数据传送指令 …… 16
二、整数运算指令 …… 23
三、逻辑指令 …… 28
四、移位和循环指令 …… 30
五、位操作指令 …… 31
六、BCD指令 …… 31
七、程序控制指令 …… 33
八、连接和恢复指令 …… 38
九、系统控制指令 …… 39
第三章　MC 68000 汇编语言程序设计 …… 42
第一节　MC 68000 宏汇编语言 …… 42
一、汇编语言语句格式 …… 42
二、汇编程序命令 …… 43
三、在操作数场的表达式 …… 45
四、条件汇编 …… 46
五、宏指令（MACROS） …… 47
第二节　汇编语言程序设计及举例 …… 47
一、简单的程序循环 …… 47
二、字符串处理 …… 50
三、码转换 …… 56

四、算术运算……59
五、列和表……65
六、参数传送技术……77
七、子程序……79
第四章　信号和总线操作……84
第一节　MC 68000 的信号……84
第二节　总线操作……87
一、数据传送操作……87
二、总线仲裁操作……92
三、总线错误和暂停操作……95
四、复位操作……97
第五章　处理状态……98
第一节　特权状态 (PRIVILEGE STATES) ……98
第二节　MC 68000 的异常处理……99
一、异常的类型……99
二、异常向量表 ……100
三、异常处理顺序 ……102
四、程序举例 ……107
第六章　支持虚拟存储器的16位微处理器——MC 68010 ……117
一、MC 68010 的编程结构 ……119
二、寻址方式和指令系统 ……119
三、MC 68010 的信号 ……124
四、MC 68010 的总线操作 ……124
五、MC 68010 的工作状态 ……125
六、MC 68010 的指令预取 ……129
第七章　全32位微处理器——MC 68020 ……132
第一节　MC 68020 的结构 ……132
一、MC 68020 的方框图 ……132
二、MC 68020 的编程结构 ……133
三、MC 68020 的状态寄存器 ……134
四、指令的流水线结构 ……134
第二节　数据组织和寻址方式 ……135
一、数据类型 ……135
二、寄存器中的数据组织 ……135
三、在存储器中的数据组织 ……136
四、指令格式 ……137
五、寻址方式 ……139
六、MC 68020 的有效地址编码格式 ……141
七、系统堆栈 ……142

八、用户程序堆栈 ……143
第三节 MC 68020 的指令系统 ……144
一、MC 68020 指令系统摘要 ……144
二、数据传送指令 ……145
三、整数运算指令 ……146
四、逻辑操作指令 ……148
五、移位和循环操作指令 ……149
六、位操作指令 ……149
七、位场操作指令 ……149
八、BCD 码操作指令 ……152
九、程序控制操作指令 ……154
十、系统控制操作指令 ……154
十一、多处理器操作指令 ……158
第四节 信号描述 ……158
第五节 总线操作 ……162
一、操作数传送机构 ……163
二、总线操作 ……171
第六节 处理器状态 ……190
一、特权状态 ……190
二、异常处理 ……192
三、具体的异常处理过程 ……195
四、总线失败恢复 ……200
五、MC 68020 异常堆栈帧 ……202
第七节 在片高速缓冲存储器 ……206
一、Cache 设计和操作 ……206
二、Cache 地址寄存器 ……208
三、Cache 屏蔽输入 ……208
四、Cache 初始化 ……208
第八节 协处理器接口描述 ……208
一、协处理器概念 ……208
二、协处理器状态 ……209
三、协处理器操作 ……209
四、协处理器总线定义 ……210
五、协处理器接口寄存器 ……211
六、协处理器指令 ……214
七、原语/响应 ……218
八、原语系统 ……220
九、协处理器接口协议 ……228
十、通用类指令协议 ……229

十一、通用和条件指令的终结 ……230
十二、协处理器内部状态帧 ……231
十三、SAVE 指令协议 ……232
十四、RESTORE 指令协议 ……233
十五、异常处理 ……233
十六、主处理器检测的异常 ……235
十七、复位 ……236
十八、协处理器指令和原语格式 ……236
第九节 指令执行时间 ……239
一、时间计算因素 ……239
二、指令定时表 ……244
第十节 MC 68020 的一些先进专题 ……261
一、模块支持 ……261
二、访问等级 ……263
三、扩展字 ……265
四、对于系统程序员的CAS/CAS2 ……267
五、程序员观点的 MC 68020 寻址方式 ……270
六、MC 68020 对M 68000 系列的扩展 ……272
附录一 MC 68000 的指令表 ……275
附录二 MC 68020 的指令表 ……278

第一章 MC 68000 的结构

第一节 MC 68000 CPU的编程结构

MC68000 CPU 的内部结构是32位的，它具有17个32位的寄存器，一个32位的程序计数器PC，一个16位的状态寄存器SR (Status Register)，如图1-1所示。

17个32位寄存器可以分成两类：

第一类是 8 个数据寄存器 D0—D7，主要用于存放操作数，可以进行字节(Byte-8位)、字 (Word-16位)和长字(Long Word-32位)操作。

第二类是7个基地址寄存器 A0—A6，主要用作为各种寻址方式的地址指针，以及两个堆栈指针 A7。MC 68000 有两种工作状态：管理状态 (Supervisor State) 和用户状态 (User State) (这两种状态的特点以及相互转换我们在后面详细介绍)。当MC 68000 工作在用户状态时，指令助记符中的A7，就是指用户堆栈指针USP (User Stack Pointer)；当CPU工作在管理状态时，指令助记符中的A7，就是指管态堆栈指针 SSP (Supervisor Stack Pointer)。

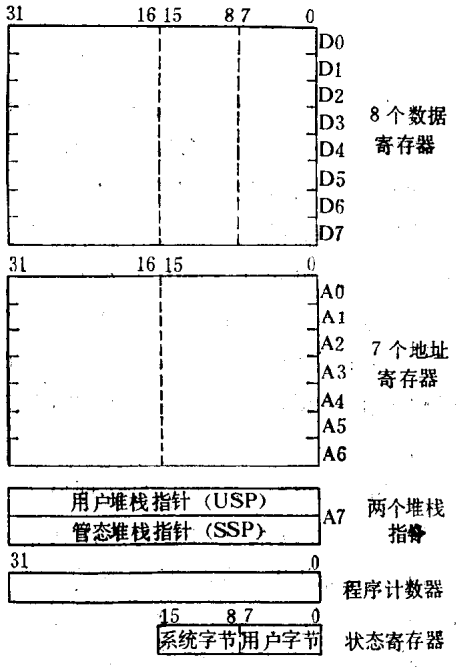


图 1-1 MC 68000 的编程结构

程序计数器是32位的，用以追踪存储器中的程序区，取出所要执行的指令。

MC 68000中的地址寄存器，堆栈指针和程序计数器都是32位的。而MC68000的地址引线是24条，所以实际上只有低24位是有用的。

MC 68000中有一个16位的状态寄存器，它又是由两部分组成，如图1-2所示。

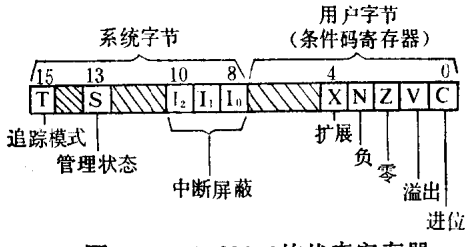


图 1-2 MC68000的状态寄存器

第一部分是低 8 位，实际有用的是 5 个标志位，它们反映了指令执行以后的一些特征，是各种条件转移指令的依据，所以称为条件码寄存器 CCR (Condition Code Register)。

C (Carry) 进位或借位标志——当两个操作数相加时有进位，或两数相减时有借位，则C标志为1；否则，C为零。移位或循环操作时，可以把在寄存器或内存单元中的操作数的某一位移至C中。在作比较操作时，也会影响C标志。

V(Over Flow)溢出标志——这个标志只有对符号数进行操作时才是有意义的。在进行算术运算时，只有当两个同号数相加，或两个异号数相减，当运算的结果超出了指定位数的二进制补码所能表达的范围（字节运算时 $>+127$ 或 <-128，字运算时 $>+32767$ 或 <-32768，长字运算时 $>+2^{31}-1$ 或 $<-2^{31}$）时，V = 1；否则，V = 0。此外，在进行算术移位时，若操作数的最高有效位改变了（若移位前为0，移位后为1；或移位前为1，移位后为0），则V = 1；否则V = 0。

Z（Zero）零标志——当操作的结果为零，则Z = 1；操作的结果不为零，则Z = 0。

N（Negative）负标志——此标志也只有对于带符号数才有意义，当算术运算、逻辑运算、移位或循环操作的结果为负，则N = 1；否则，N = 0。总之，N标志跟踪操作结果的最高有效位，不论此操作数是字节、字或长字，只要指定长度的操作数的最高有效位为1，则N = 1。

X（Extend）扩展标志——在进行加法、减法、求补、移位和循环操作时，X标志与C标志相同，它主要用于多精度（多字节）运算。

状态寄存器的第二部分，是它的高8位的系统字节，它们主要是起一些控制作用。

位10—位8的I2 I1 I0这三位构成了MC 68000的中断屏蔽位。三位二进制对应的值为0—7，这就决定了8种中断优先权等级，7为最高优先权。当在MC 68000的三条中断请求输入线上出现的外设的中断请求等级等于或低于状态寄存器中的I2 I1 I0所构成的中断屏蔽等级时，CPU就不响应（中断等级7例外，这相当于非屏蔽中断，CPU是必须响应的）；只有当输入的中断请求等级高于中断屏蔽等级时，CPU才响应。（关于MC 68000的中断，我们在后面详细介绍）

位13的标志S，是CPU的管理状态（当S = 1时），和用户状态（当S = 0时）的控制位。改变S位，就可以改变CPU的工作状态。

位15的标志T，是追踪状态的控制位。当T = 1时，CPU就工作在单步方式，在每执行一条指令后，CPU就停下来，可由一个服务程序来检查指令执行的结果，主要用于调试。当T = 0时，CPU工作在正常方式。

第二节　MC 68000的存储器结构

MC 68000有24条地址线，故它的直接寻址能力为2^{24} = 16M地址单元。这16M地址单元构成一个线性的存储器结构，最低地址为$000000*，最高地址为$FFFFFF。16M的地址单元不分段。

MC 68000的数据线为16条，它能处理的数据类型为：

位，

BCD数（由4位二进制数，构成1位BCD数），

字节（8位），

字（16位），

长字（32位）。

---

* 在Motorola的宏汇编中规定，16进制数用符号$开头。

一个地址单元为一个字节。因此,MC 68000的最大寻址空间为16M字节。但数据线为16条，故整个存储器分成两块，地址为奇数的字节构成一块，它们的数据线与CPU数据总线的低8位D0—D7相连；地址为偶数的字节构成另一块，它们的数据线与CPU的数据总线中的高8位D8—D15相连，如图1-3所示。

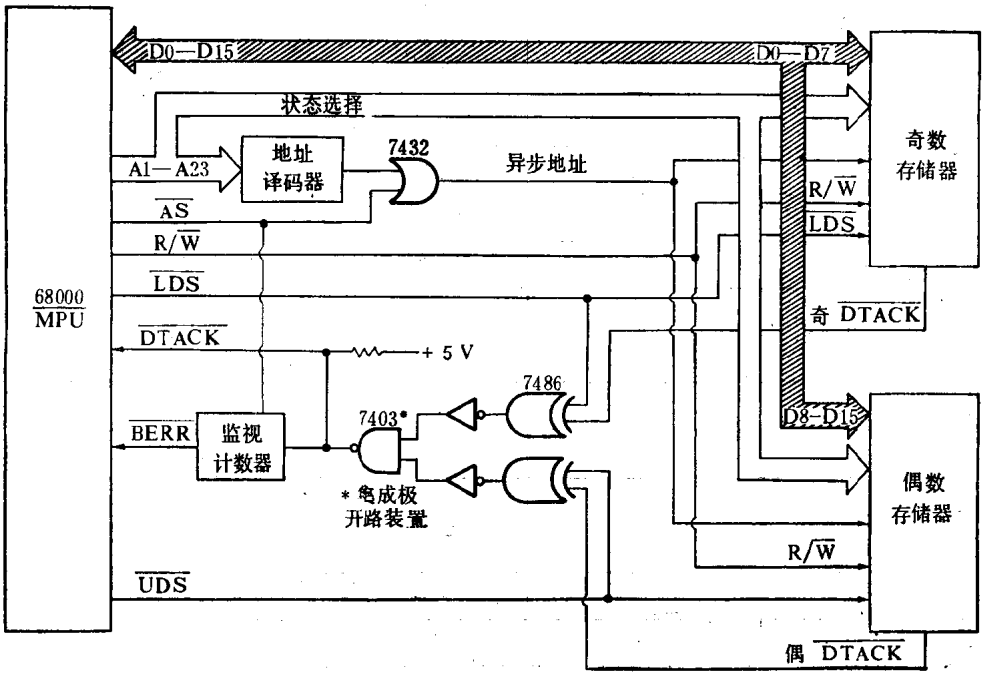


图 1-3 MC 68000字节寻址图

MC 68000存储器中的数据组织如图1-4所示。

由图1-4可以看到，若是要从内存中访问一个（或一串）字节操作数，则可以从奇数地址或偶数地址开始。

但若要访问一个（或一个数据块）字或长字操作数，则必须从偶数地址开始。若从奇数地址访问一个字或长字操作数，则要产生地址错误。

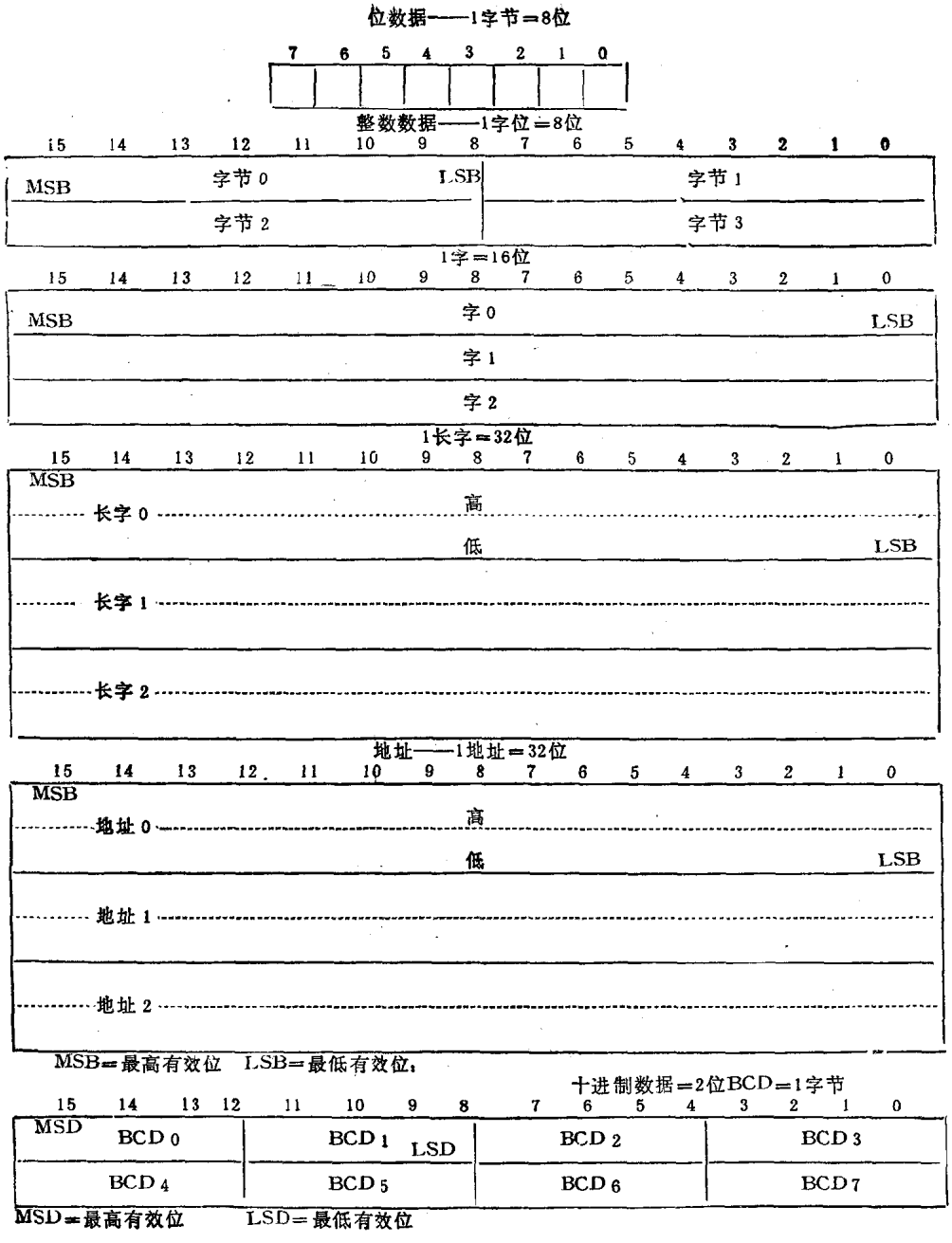


图 1-4 MC 68000存储器中数据组织图

第二章　MC 68000的寻址方式和指令系统

第一节　寻 址 方 式

寻址方式是各种 CPU 指令功能的重要方面。一条指令通常是由规定执行何种操作的操作码（Op code)部分和规定由什么样的数参加操作的操作数（Operand）部分所组成。操作数可能在 CPU 的内部寄存器中，更一般的是在内存储器中，特别是在存储器的数据区中。如何寻找参与操作的操作数，这就是指令的寻址方式。指令的寻址方式越多、越灵活，越能适应高级的或复杂的数据结构，则指令的功能也就越强。

MC 68000有五类12种主要的寻址方式。

一、寄存器寻址 (Register Addressing)

这类寻址方式，操作数在 CPU 内部的某一个寄存器中。CPU 内部有两种寄存器，故也就有两种不同的寄存器寻址方式。

1．数据寄存器直接寻址（Data Register Direct Addressing）

这种寻址方式，是指操作数在某一个数据寄存器内，其示意图如图2-1所示。

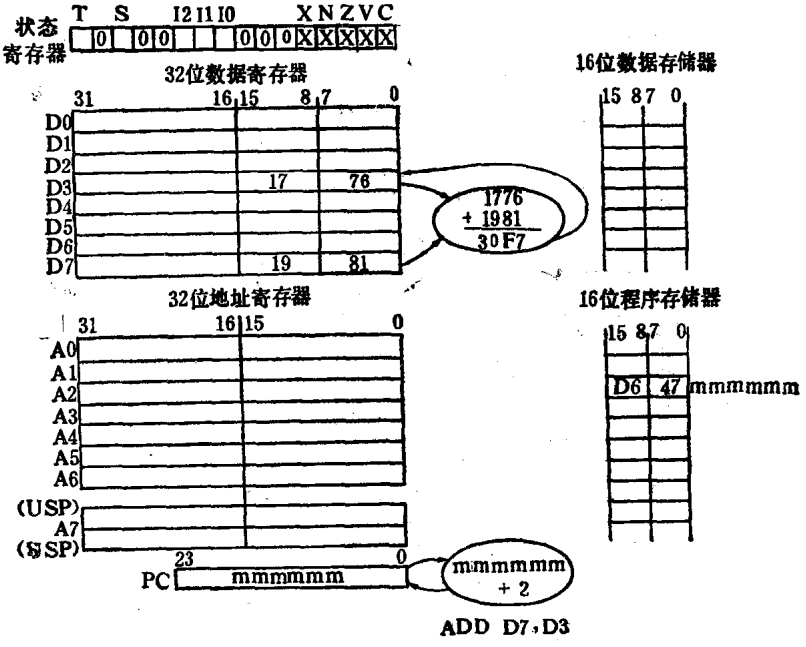


图 2-1　数据寄存器直接寻址

参与加法运算的两个操作数，都在数据寄存器中，一个在 D3 的低字部分（因为指令规定是字运算），另一个在D7的低字部分。

2．地址寄存器直接寻址（Address Register Direct Addressing）

这种寻址方式是指操作数在CPU内部的某个地址寄存器中，如图2-2所示。

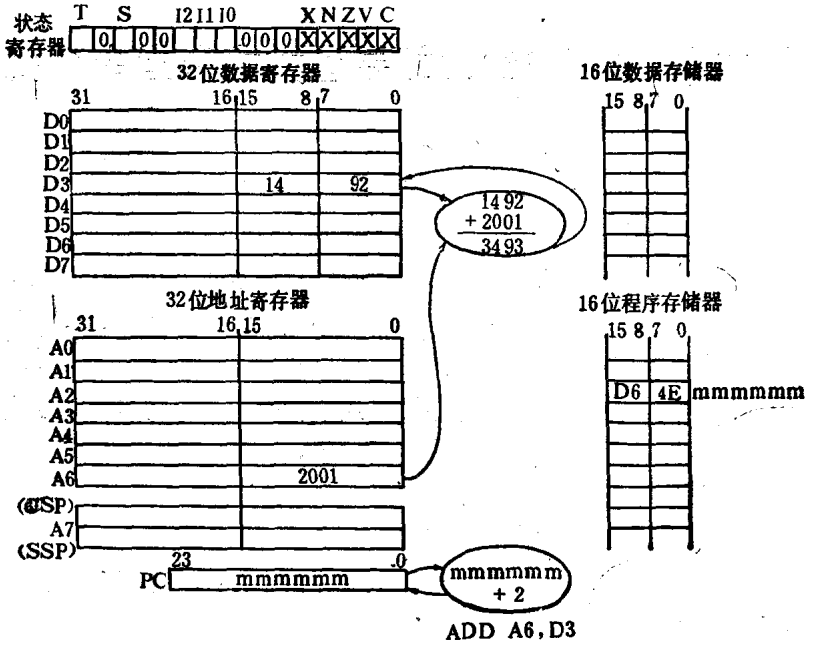


图 2-2 地址寄存器直接寻址

例中指出，参与加法运算的一个操作数（源操作数）在某个地址寄存器(可以是A0—A6中的某一个）中。这说明地址寄存器虽然主要用作地址指针，但是也可用来存放参与运算的操作数。

二、立即寻址 (Immediate Addressing)

这种寻址方式规定操作数是在存储器中，但是在存储器的程序区中，即操作数作为指令

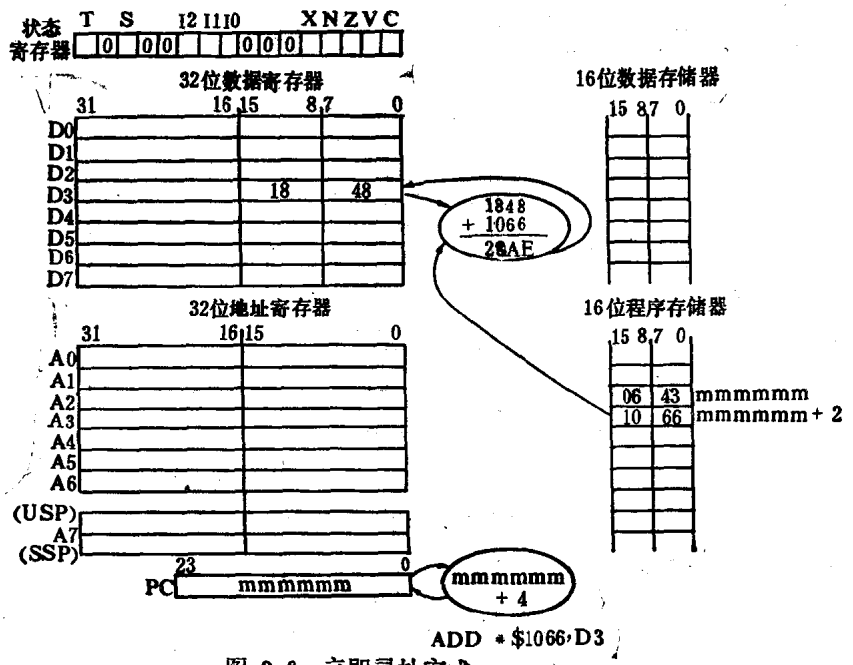


图_2-3 立即寻址方式

的一部分紧跟在指令的操作码部分之后，如图2-3所示。

例中指出，参与加法运算的一个操作数是一个常数（例中的$1066），它紧跟在指令的操作码部分之后。这种寻址方式就称为立即寻址，操作数就称为立即数。

在字操作时，立即数是一个字（16位）；在长字操作时，立即数就是--个长字（32位），它在指令中要占两个字的长度。在 MC 68000 中规定，长字操作数，高16位存放在地址低的存储单元，而低16位存放在地址高的单元，如图 2-4 所示（要注意：这种规定与 8088、Z8000 等中的规定是相反的）。

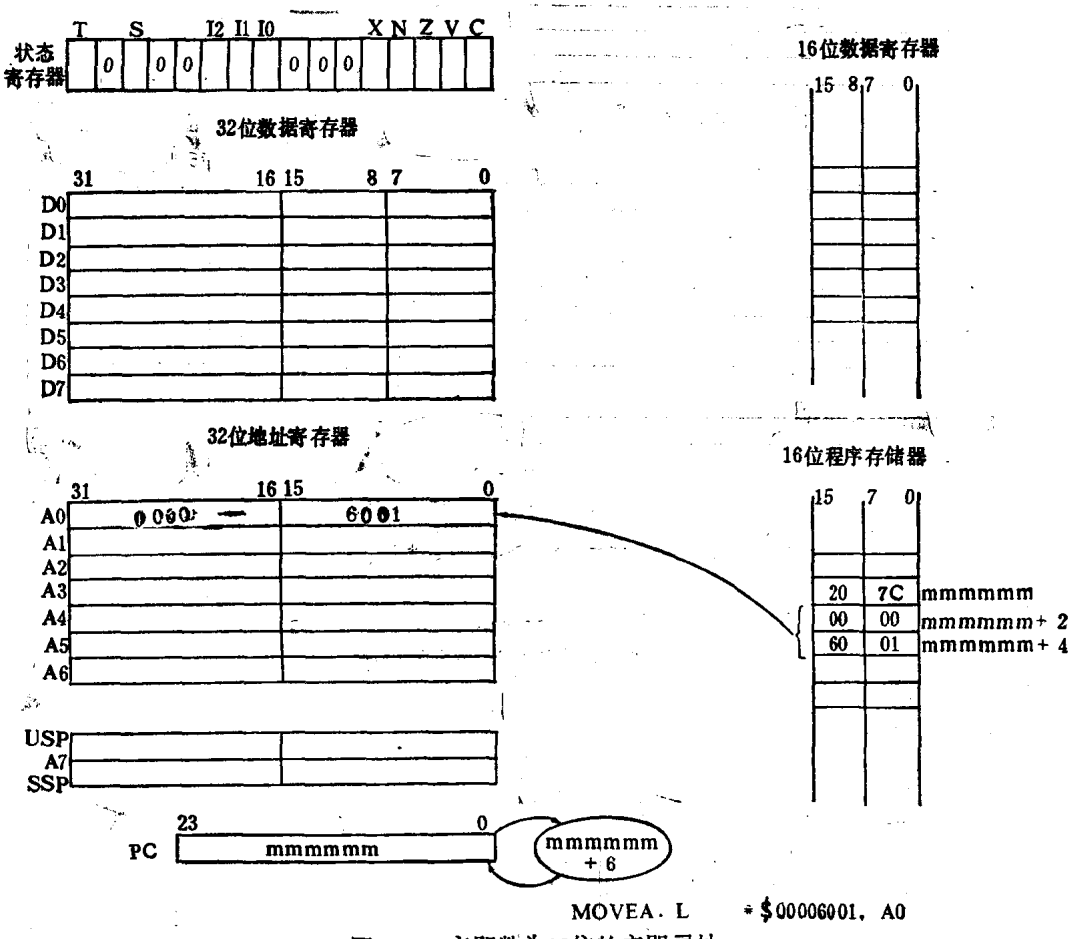
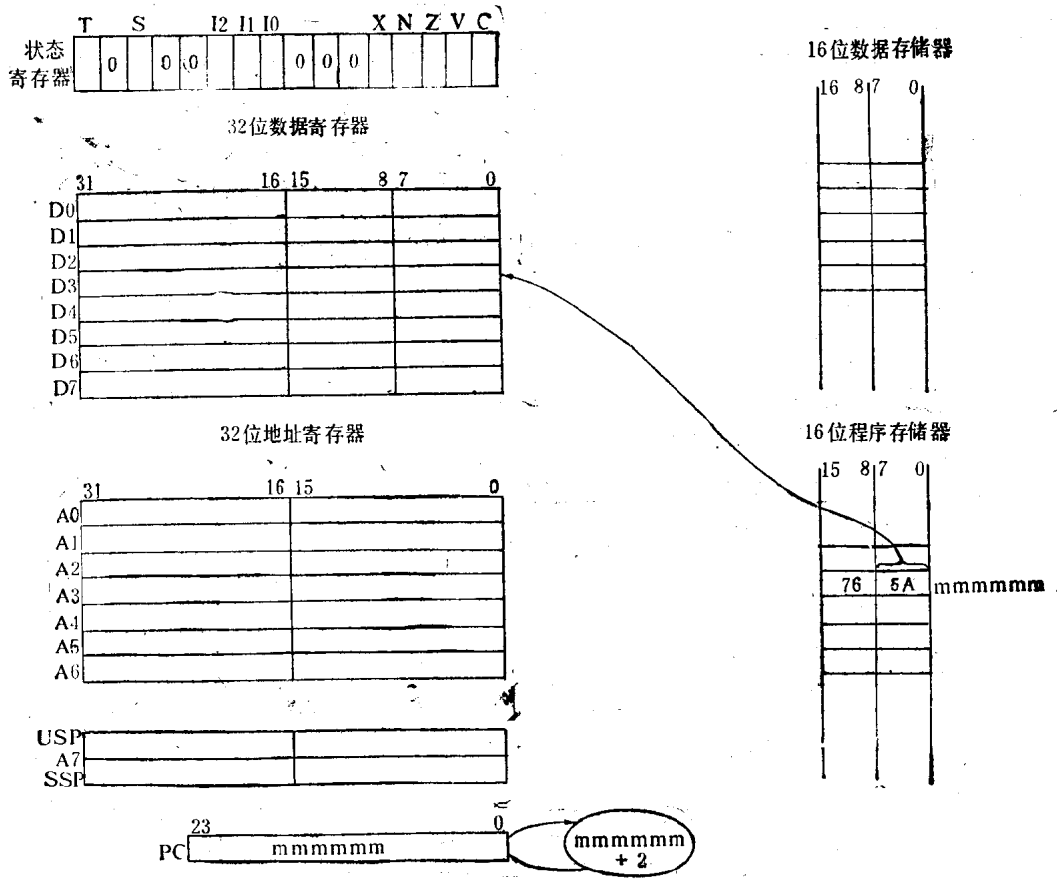


图 2-4 立即数为32位的立即寻址

在MC 68000中还有一种关于立即数的快速操作指令。例如，可以把一个字节的立即数传送给一个目的操作数（例如某个数据寄存器）。这样的指令连操作码在一起共为一个字节，即减少了所占的内存空间，也缩短了执行时间。这种指令的一个例子如图2-5所示。

三、直接寻址（Direct Addressing）

在这种寻址方法中，操作数在内存的数据区，但操作数的地址直接包含在指令中，故称为直接寻址。由于 MC 68000 的地址线是 24 条，所以它的全地址要用一个长字（32位）来表示，这样可以在内存的16 M地址空间的任何部分寻找操作数。但是，通常为了有效地利用地址空间，数据的存放是集中的，用16位地址在64K的范围内寻找操作数已是足够的了。所以，有两种直接寻址方式。

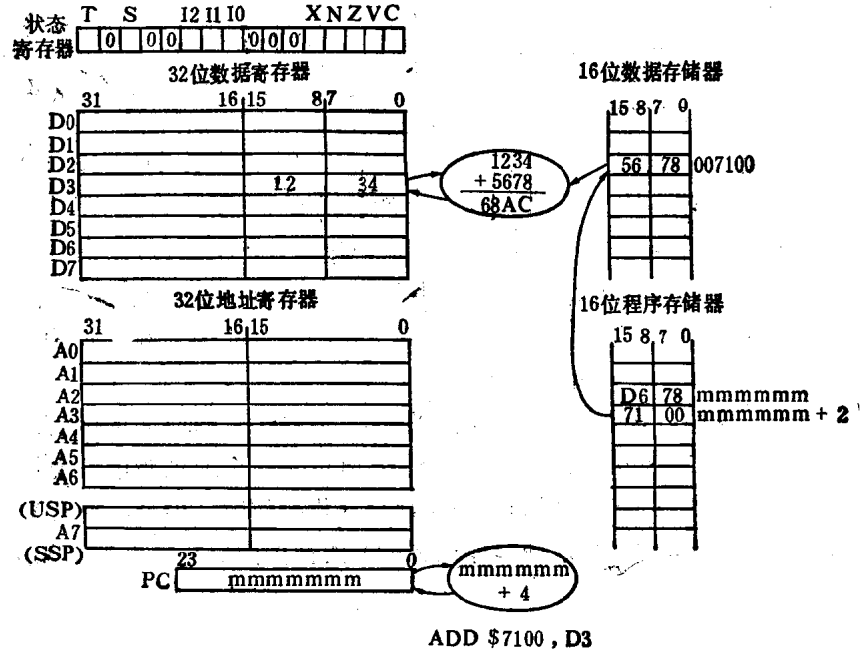


MOVEQ #$5A, D3

图 2-5 快速传送指令的例子

1. 绝对短寻址 (Absolute Short Addressing)

一个例子如图2-6所示。



ADD $7100, D3

图 2-6 绝对短寻址