# MC68000系列微处理器芯片资料

中国科学院成都计算机应用研究所情报室译

Motorola · 1984

# 译 序

这本资料是根据美国 MOTOROLA 公司提供的1984年版本译出的。比起我们以前见到国内译出的 MC68000资料来看，内容增加较多，比较详尽完整。我们相信能对国内从事与微机有关的各行各业的各种工作人员有些帮助。

参加本资料翻译和校对工作的有王又钧、颜莹、邓大毅、罗运民、高正玖、吴浣尘等。全书由吴浣尘统一校对。

# 目 录

# 第一章　引论

在Motorola生产的先进微处理器系列中，MC68000是第一个用超大规模集成电路（VLSI）技术制成的16位微处理器，它有32位寄存器和丰富的基本指令系统，寻址方式灵活适用。

MC68000的异步总线结构有24位地址总线和16位数据总线。

采用MC68000的用户可以利用的资源包括以下几种：

- 17个32位数据和地址寄存器
- 16兆字节直接寻址范围
- 56种功能很强的指令类型
- 对五种主要数据类型进行运算
- 存贮器映象的I／O
- 14种寻址方式

在图1—1上的编程模型中，MC68000有十七个32位寄存器，一个32位程序计数器和一个16位状态寄存器。前八个寄存器（D0—D7）用作字节（8位），字（16位）和长字（32位）运算的数据寄存器。第二个七个寄存器组（A0—A6）和系统堆栈指针可以用作软件堆栈指针和基地址寄存器。此外，寄存器可用于字和长字运算。所有17个寄存器都可以用作变址寄存器。

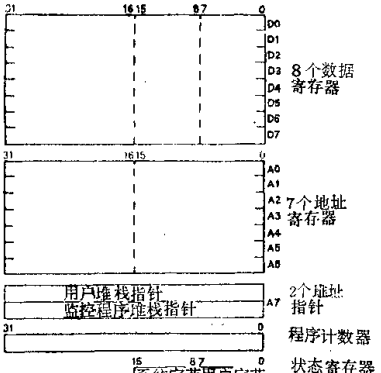



图1—1　编址模型

图1—2上的状态寄存器包括中断屏蔽（八级）以及条件码：扩展（X），负数（N），零（Z），溢出（V）和进位（C）。附加状态位指示处理器在跟踪（T）方式和在管理程序（S）中或用户状态。

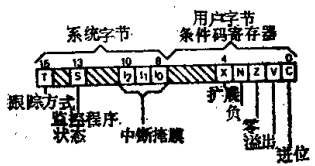



图1—2　状态寄存器

### 1.1　数据类型和寻址方式

可以支持五种基本数据类型。这五种数据类型是：

- 位
- BCD数字（4位）
- 字节（8位）
- 字（16位）
- 长字（32位）

此外，其它如像存贮器地址，状态字数据之类的数据运算都在指令系统中列出。

表1—1中的14种地址方式包括六种基本类型：

- 寄存器直接地址
- 寄存器间接地址
- 绝对地址
- 程序计数器相对地址
- 立即地址
- 隐含地址

在寄存器间接寻址方式中包含后增量，前增量，偏置和变址能力。程序计数器相对寻址方式也可以通过变址和偏置进行修改。

表1—1　　寻址方式

| 方　式 | 产　生 |
|---|---|
| **寄存器直接寻址** | |
| 数据寄存器直接地址 | EA = Dn |
| 地址寄存器直接地址 | EA = An |
| **绝对数据寻址** | |
| 绝对短地址 | EA = (下一个字) |
| 绝对长地址 | EA = (下两个字) |
| **程序计数器相对寻址** | |
| 有偏置的相对地址 | EA = (PC) + $d_{16}$ |
| 有变址和偏置的相对地址 | EA = (PC) + (Xn) + $d_8$ |
| **寄存器间接寻址** | |
| 寄存器间接地址 | EA = (An) |
| 后加一寄存器间接地址 | EA = (An), An←An+N |
| 前加一寄存器间接地址 | An←An−N, EA = (An) |
| 有偏置的寄存器间接地址 | EA = (An) + $d_{16}$ |
| 有偏置的变址寄存器间接地址 | EA = (An) + (Xn) + $d_8$ |
| **直接数据寻址** | |
| 直接数据地址 | DATA = 下一个字 |
| 快直接数据地址 | 固有数据 |
| **隐含寻址** | |
| 隐含寄存器地址 | EA = SR, USP, SP, PC |

注：

EA = 有效地址

An = 地址寄存器

Dn = 数据寄存器

Xn = 地址或数据寄存器用作变址寄存器

SR = 状态寄存器

PC = 程序计数器

(　) = 寄存器的内容

$d_8$ = 8位偏置（位移）

$d_{16}$ = 16位偏置（位移）

N = 1为字节，2为字，3为长字。如An是堆栈指针和操作数是字节，则N = 2以保持堆栈指针在字边界上

← = 代换

## 1.2 指令系统概述

表1—2上列出MC68000的指令系统，在表1—3上是一些附加指令的变化和子系统。特别强调指令系统对结构式高级语言的支持，以便易于编程。除少数例外，各条指令都对字节，字和长字进行运算，大多数指令可采用14种寻址方式中的任何一种方式。组合指令型，数据型和寻址方式提供1,000多条有用的指令。这些指令包括带符号的和不带符号的，乘和除，“快”算术运算，BCD 算术运算和扩展运算（通过陷井）。

表1—2 指令系统一览表

| 助记符 | 说明 |
|---|---|
| ADBC | 带扩展的十进制加 |
| ADD | 加 |
| AND | 逻辑“与” |
| ASL | 算术左移 |
| ASR | 算术右移 |
| Bcc | 条件分支 |
| BCHG | 位测试和改变 |
| BCLR | 位测试和清除 |
| BRA | 经常分支 |
| BSET | 位测试和置位 |
| BSR | 分支到子程序 |
| BTST | 位测试 |
| CHK | 检查寄存器越界 |
| CLR | 清除操作数 |
| CMP | 比较 |
| DBcc | 测试条件，减一和转移 |
| DIVS | 带符号除 |
| DIVU | 不带符号除 |
| EOR | 异或 |
| EXG | 交换寄存器 |
| EXT | 符号扩展 |
| JMP | 转移 |
| JSR | 转移到子程序 |
| LEA | 装入有效地址 |
| LINK | 链路堆栈 |
| LSL | 逻辑左移 |
| LSR | 逻辑右移 |
| MOVE | 移动 |
| MULS | 带符号乘 |
| MULU | 不带符号乘 |
| NBCD | 带扩展的十进制取反 |
| NEG | 取反 |
| NOP | 不操作 |
| NOT | 反码 |
| OR | 逻辑“或” |
| PEA | 推入有效地址 |
| RESET | 外部设备复位 |
| ROL | 不扩展的循环左移 |
| ROR | 不扩展的循环右移 |
| ROXL | 带扩展的循环左移 |
| ROXR | 带扩展的循环右移 |
| RTE | 由异常返回 |
| RTR | 返回并恢复 |
| RTS | 由子程序返回 |
| SBCD | 带扩展的十进位减 |
| SCC | 条件置位 |
| STOP | 停止 |
| SUB | 减 |
| SWAP | 对半交换数据寄存器 |
| TAS | 测试和置操作数 |
| TRAP | 陷阱 |
| TRAPV | 溢出陷阱 |
| TST | 测试 |
| UNLK | 去链接 |

表 1—3 指令类型的各种变化方案

| 指令类型 | 方 案 | 说 明 |
|---|---|---|
| ADD | ADD | 加 |
| | ADDA | 加地址 |
| | ADDQ | 快加 |
| | ADDI | 立即加 |
| | ADDX | 带扩展加 |
| AND | AND | 逻辑与 |
| | ANDI | 立即与 |
| | ANDI到CCR | 立即与条件码 |
| | AND到SR | 立即与状态寄存器 |
| CMP | CMP | 比较 |
| | CMPA | 比较地址 |
| | CMPM | 比较存贮器 |
| | CMPI | 立即比较 |
| EOR | EOR | 异或 |
| | EORI | 立即异或 |
| | EORI到CCR | 立即异或条件码 |
| | EORI到SR | 立即异或状态寄存器 |
| MOVE | MOVE | 移动 |
| | MOVEA | 移动地址 |
| | MOVEM | 移动多个寄存器 |
| | MOVEP | 移动外部数据 |
| | MOVEQ | 快移 |
| | MOVE从SR | 从状态寄存器移动 |
| | MOVE到SR | 移动到状态寄存器 |
| | MOVE到CCR | 移动到条件码 |
| | MOVE USP | 移动用户堆栈指针 |
| NEG | NEG | 取反 |
| | NEGX | 带扩展取反 |
| OR | OR | 逻辑或 |
| | ORI | 立即或 |
| | ORI到CCR | 立即或条件码 |
| | ORI到SR | 立即或状态寄存器 |
| SUB | SUB | 减 |
| | SUBA | 减地址 |
| | SUBI | 立即减 |
| | SUBQ | 快减 |
| | SUBX | 带扩展减 |

# 第二章　数据组织和寻址能力

本章介绍MC68000的寄存器和数据组织。

## 2.1　操作数长度

操作数长度的定义如下：一个字节等于8位，一个字等于16位，一个长字等于32位。每条指令的操作数的长度或者显式地在指令中编码，或者隐式地由指令操作规定。隐式指令可以支持所有三种长度的某些子集。

## 2.2　寄存器中的数据组织

八个数据寄存器可以用以寄存1位，8位，16位或32位的数据操作数。七个地址寄存器与堆栈指针一起可以寄存32位地址操作数。

### 2.2.1　数据寄存器

每个数据寄存器宽为32位。字节操作数占据低8位，字操作数占低16位，而长字操作数占据全部32位。最低有效位的地址编作第零位，而最高有效位的地址编为第31位。

当数据寄存器或者用作源操作数，或者用作目的操作数时，只改变适当的低位部分；余下的高位部分或者不用，或者不变。

### 2.2.2　地址寄存器

每个地址寄存器和堆栈指针宽是32位，可以容纳全部32位地址。地址寄存器不支持标志长度的操作数。因此，当地址寄存器用作源操作数时，究竟使用低位字操作数或者使用长字操作数取决于操作的长短。当地址寄存器用作目的操作数时，不论操作长短如何，全部寄存器都受到影响。如果操作长短是字，在执行操作之前，任何其它操作数都带符号扩展成32位。

## 2.3　存贮器中的数据组织

如图2－1中所示，可以用高位字节单独为字节寻址，高位字节的偶数地址与字的地址是相同的。

低位字节的地址是奇数地址，这个地址比字的地址多一。指令和多字节数据只按字（偶数字节）边界存取。如果一个长字数据被置于地址n（n 偶数），则该数据的第二个字被置于地址n＋2。

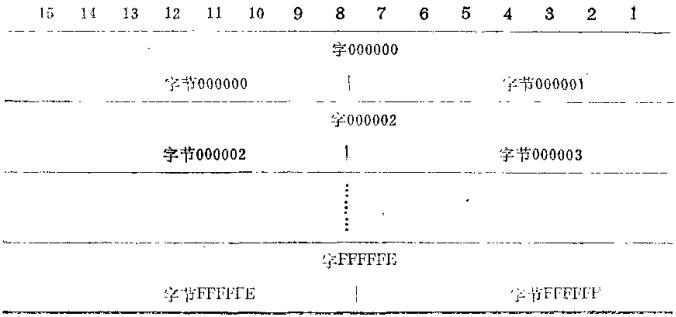
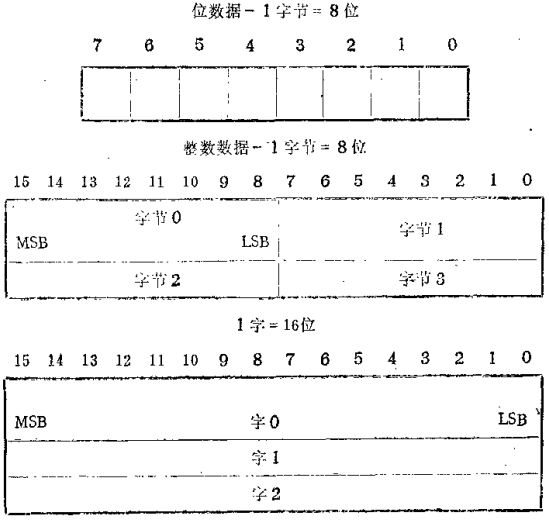



图2—1 存贮器中字的组织

MC68000所支持的数据类型是：位数据，8位，16位或32位地址的整数数据和二进制编码的十进位数据。每个这些类型的数据都如图2—2所示的那样放在存贮器中。编号表示从处理器中存取数据的顺序。



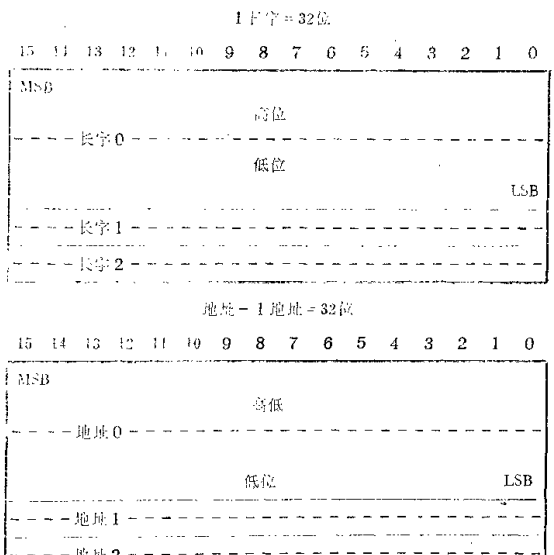



图 2－2 存贮器数据组织

## 2－4 寻址

MC68000的指令包含两种信息：所要执行的功能类型和执行该功能所用的操作数的存贮单元。在以下几节中将说明定位（寻址）操作数所采用的方法。

指令指定一个操作数的存贮单元可采用以下三种办法中的一种：

寄存器技术说明——在指令的寄存器字段中的标出寄存器编号。

有效地址——利用不同的有效寻址方式。

隐含标记——某些指令的定义隐含在专用寄存器的用途中。

### 2.5 指令格式

如图2—3所示，指令长度可以从一到五个字。指令的前一个字叫做操作字，它规定指令长度和所要执行的操作。以下其余的字规定操作数。这些字或者是立即操作数，或者是操作字中所规定的有效地址方式的扩展。

### 2.6 程序／数据标记

MC68000把存贮器标记分为两种：程序标记和数据标记。顾名思义，程序标记是包含要执行的程序的存贮器部分的标记。数据标记是指包含数据的存贮器部分的标记。除了程序记数器相对寻址方式以外，操作数是由数据空间读出的。所有的操作数都写入数据空间。

### 2.7 寄存器技术说明

指令中的寄存器字段规定所使用的寄存器。指令中的其它字段规定选定的寄存器究竟是地址寄存器或是数据寄存器和如何使用寄存器。

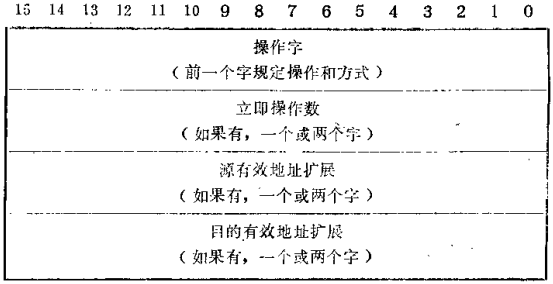



图2—3　指令操作字一般格式

### 2.8 有效地址

大部分指令都用操作字中的有效地址来规定操作数的存贮单元。举例说，图2—4上示出单有效地址指令操作字的一般格式。有效地址由两个3位字段组成：方式字段和寄存器字段。方式字段的数值选择不同的寻址方式。寄存器字段则包含寄存器的编号。

有效地址字段为了充分表明操作数可能需要附加信息。这种附加信息叫做有效地址扩展，它包括在以下的字或字组中，如图2—3所示，它被认为是指令的一部分。有效寻址方式分为三类：寄存器直接地址，存贮器寻址和专用地址。

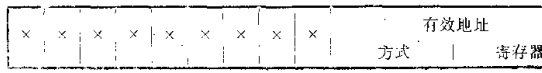



图2—4　单有效地址指令操作字

#### 2.8.1 寄存器直接寻址

这类有效寻址方式规定操作数是16个多功能寄存器之一。

##### 2.8.1.1 数据寄存器直接寻址

操作数在有效地址寄存器字段所指定的数据寄存器中。

##### 2.8.1.2 地址寄存器直接寻址

操作数在有效地址寄存器字段所指定的地址存存器中。

#### 2.8.2 存贮器寻址

这类有效寻址方式规定操作数在存贮器中，并提供操作数的专用地址。

##### 2.8.2.1 地址寄存器间接寻址

操作数地址在寄存器字段所指定的地址寄存器中。标记被分成有异常转移指令的数据标记和转子指令的数据标记。

##### 2.8.2.2 后增量地址寄存器间接寻址

操作数地址在寄存器字段所指定的地址寄存器中。在使用操作数地址以后，按照操作数长度究竟是字节，字或长字加一，二或四。如果地址寄存器是堆栈指针而操作数长度是字节，地址应加二，而不是加一，以保持堆栈指针边在界上。标记分类为数据标记。

##### 2.8.2.3 先增量地址寄存器间接寻址

操作数地址在寄存器字段所指定的地址寄存器中。在使用操作数地址以前，按照操作数长度究竟是字节，字或长字加一，二或四。如果地址寄存器是堆栈指针而操作数长度是字节，地址应加二，而不是加一，以保持堆栈指针在字边界上。标记分类为数据标记。

##### 2.8.2.4 有位移的地址寄存器间接寻址

这种寻址方式需要扩展一个字。操作数地址是地址寄存器中的地址加上扩展字中符号扩展的16位位移整数之和。标记分类为有异常转移指令和转子指令的数据标记。

##### 2.8.2.5 可变址的地址寄存器间接寻址

这种寻址方式需要扩展一个字。操作数地址是地址寄存器中的地址加上扩展字的低八位

中符号扩展位移整数，再加上变址寄存器内容三者之和。标记分类为有异常转移指令和转子指令的数据标记。

### 2.8.3 专用编址方式

专用编址方式使用有效地址寄存器来指定专门的寻址方式，而不采用寄存器编号。

#### 2.8.3.1 绝对短地址

这种寻址方式需要扩展一个字。操作数地址是扩展字。在使用它之前，16位地址是符号扩展地址。标记分类为有异常转移指令和转子指令的数据标记。

#### 2.8.3.2 绝对长地址

这种寻址方式需要扩展两个字。操作数地址由两个扩展字串联起来决定。地址的高位部分是前一个扩展字，地址的低位部分是后一个扩展字。标记分类为有异常转移指令和转子指令的数据标记。

#### 2.8.3.3 有位移的程序计数器

这种寻址方式需要扩展一个字。操作数地址是程序计数器中的地址加上扩展字中符号扩展的16位位移整数之和，程序计数器中的数值是扩展字的地址，标记分类为程序标记。

#### 2.8.3.4 可变址的程序计数器

这种寻址方式需要扩展一个字。地址是程序计数器中的地址，加上扩展字的低八位的符号扩展位移整数，再加上变址寄存器内容三者之和。程序计数器中的数值是扩展字的地址。这种标记分类为程序标记。

#### 2.8.3.5 立即数据

这种寻址方式根据操作长度需要扩展一个或两个字。

字节操作——操作数是扩展字的低位字节

字操作——操作数是扩展字

长字操作——操作数是两个扩展字，高16位在前一个扩展字中，低16位在后一个扩展字中。

#### 2.8.3.6 隐含标记

有些指令给程序计数器（PC），系统堆栈指针（SP），管理程序堆栈指针（SSP），用户堆栈指针（USP）或状态寄存器（SR）加上标记。利用有效地址字段可以用选定的指令集来标记状态寄存器，这些指令是：

| | | |
|---|---|---|
| ANDI到CCR | EORI到SR | MOVE到CCR |
| ANDI到SR | ORI到CCR | MOVE到SR |
| EORI到CCR | ORI到SR | MOVE自SR |