MC68000

微处理器手册

计算机丛书

[美] G.凯恩 著

郁泉 丁越 译

电子工业出版社

MC68000微处理器手册

〔美〕格里·凯恩 著
郁 泉 丁 越 译

電子工業出版社

内 容 提 要

本书较详细地介绍了MC68000微处理器的结构原理和性能，各信号之间的定时关系，指令系统以及与6800的接口举例。为了便于更好地掌握和使用MC68000的指令系统，译者为本书编译了附录A和附录B，即指令详解和程序设计用表。

68000 Microprocessor Handbook
Gerry Kane
OSBORNE/McGraw-Hill
1981

TS283/03

MC 68000 微处理器手册
〔美〕G·凯恩 著
都 泉 丁 越 译
责任编辑：王惠民
电子工业出版社出版（北京市万寿路）
新华书店北京发行所发行 各地新华书店经售
中国科学技术情报研究所印刷厂
开本：850×1168 1/32 印张：7.75字数：206千字
1984年5月第1版 1984年6月第1次印刷
印数：30500册 定价：1.85 元
统一书号：15290·13

说　明

近年来，以Motorola 68000微处理器为主的16位微计算机，已陆续在国内市场上出现。为了便于了解和使用这种机型，特将1981年〔美〕OSBORNE/McGraw-Hill 公司出版的《68000 MICRO-PROCESSOR HANDBOOK》一书译出，作为我们对我国微计算机事业的一点微薄的贡献。

原书附录部分的编排方法，不大适合于查找和阅读。为此，我们结合 Motorola 公司的用户手册及有关资料编译了附录A。北京工业大学二系微计算机应用研究室徐家栋副教授改编了 Motorola 公司的程序设计卡片，作为附录B。他对译文也提出不少修改意见，在此谨表谢意。

由于译者技术和翻译水平所限，错误之处恳请读者指正。

译　者

1983. 10.

目　录

1 绪 言

MC68000微处理器是Motorola公司生产的一种16位微处理器。它是继Intel公司的8086和Zilog公司的Z8000之后的第三种16位微处理器。

MC68000与Motorola公司的8位微处理器不能兼容。Motorola公司在设计MC68000指令系统时，主要考虑的是如何使其简单，且具有最强的功能，而不是它的兼容性。

同Z8000和8086相比，MC68000有许多使人感兴趣的特性，现分述如下：

1．MC68000在取每条指令的目的码时，与前二条指令的译码及执行相重迭，以获得流水效果。Z8000仅在某些情况下才采用这种方法。8086是用6个字节的目的码队列来实现流水。

2．8086和Z8000两个系列的微处理器都规定在构成"简单"系统或"复杂"系列时器件的用法。8086是通过本身一些具有两种（复合）功能的引脚来实现的，这些双功能引脚在"简单"系统中完成一种功能，在"复杂"系统中完成另一种功能。而Z8000是有两种类型的芯片：Z8001用于"复杂"系统，Z8002用于"简单"系统。MC68000封装在有64根引脚的管壳中，因而无须采用具有双功能引脚结构，就能以"复杂"或"最大"系统结构方式进行操作。

3．MC68000具有在多个CPU配置中，处理总线判优的内部逻辑。8086和Z8000也有相应的逻辑。

4．MC68000有24位地址总线，可直接访问多达16Mb的存储空间。该存储空间通过功能码输出线可扩展到64Mb。相比之下，8086只能直接寻址64Kb，利用段寄存器可寻址到1Mb；Z8000也

是直接寻址64Kb，Z8001采用内部段寄存器和在存储器管理部件中的外部分段，可寻址到多达48Mb。

5．MC68000分为管理方式和用戶方式。特权指令只能在管理方式中执行。管理方式和用戶方式各有其单独的栈指针，这样在程序密集应用时，系统软件（在管理方式执行）可与应用程序（在用戶方式执行）分开。Z8000系列微处理器也具有相似的功能。MC68000的管理方式相当于Z8000的系统方式，而MC68000的用戶方式则相当于Z8000的正常方式。8086沒有类似的操作方式。

6．MC68000共有17个32位寄存器。其中8个作为数据寄存器，可以是8位、16位或者32位。其余9个寄存器作为地址寄存器，其中2个作栈指针（管理的和用戶的）。地址寄存器可以是16位或者是32位。所有寄存器都可以作为变址寄存器。相比之下，Z8000的寄存器，全部都是16位的，成对使用时可以是32位的。8086只有4个16位寄存器和3个独立的16位变址寄存器。

7．MC68000因为是64引脚封装的，有足够的引脚可供连接，所以它的每个数据线和地址输出线都有单独的引脚。Z8000和8086微处理器封装在较小的管壳中，所以它们的数据线和地址线只好共用某些引脚。于是，Z8000和8086都把某些数据和地址信号复合在一根引脚上，使用时必须用外部逻辑将这些信号分开。

MC68000采用N沟HMOS工艺技术，双列直插64引脚封装。单电源+5V供电，所有信号与TTL电平兼容。

MC68000要求外部时钟，最高频率为8MHz。最小指令执行时间为4个时钟周期，最大指令执行时间为158时钟周期（带符号的乘除法）。

有关本书所用信号的约定

信号可以是高电平有效，低电平有效或者两种状态都有效。高电平有效的信号，是指它在高电平状态引起事件发生，而在低电平

状态没有意义。如果一个信号是低电平有效，即它在低电平状态引起事件发生，而在高电平状态没有意义。一个两状态有效的信号，依据其信号电平的高和低，分别引起两类不同事件的发生，这类信号没有不动作的状态。本书中低电平有效信号，均在信号名称的上方加横线。如$\overline{WR}$表示“写选通”信号为低有效，即当CPU准备好为外部逻辑接收的数据时，该信号便跳变到低电平。高电平有效或两状态有效的信号，在它们的名称上不加横线。

时序图的约定

时序图对于描述任何一种微处理器或辅助器件方面都起着很重要的作用。因此，本书中广泛采用了时序图。所有时序图都遵守下列约定：

1．低电平信号等效为没有电压。高电平信号等效为有电压。表示为：

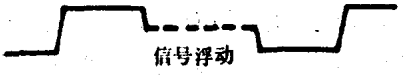
2．由低电平到高电平转换的单个信号表示为：

3．由高电平到低电平转换的单个信号表示为：

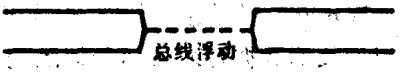
4．当两个或更多个并行信号存在时，有下述表示法：

它表示一个或几个并行信号在改变着电平状态，但对其转换（高到低或低到高）未加规定。

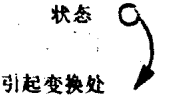
5．三态信号，则表示为浮动。如下所示：



6．三态总线包含二个或多个信号，当总线浮动时表示如下：

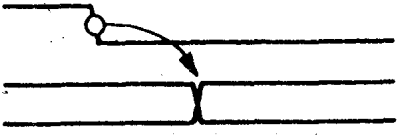


7．当一个信号状态触发另一个信号发生变化时，（箭头指示）其相互间的关系：

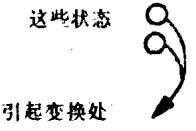


这样，一个从低到高的变换信号能触发产生一个由高到低电平变换的信号。表示为：

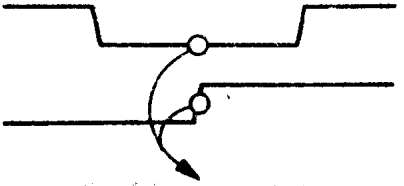
一个从高到低变换的信号，触发总线状态改变时，表示如下：



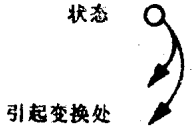
8．当必须有两个或更多的状态存在，才能触发另一个逻辑事件时采用如下的图示：



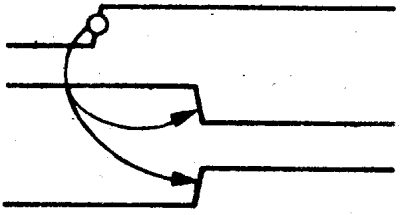
这样，若一个信号发生低到高变换，而另一个信号处于低电平时，才触发第三个事件。其图示如下：



9．当一个触发条件引起二个或多个事件发生时，则采用如下图示法：



一个信号从低到高变换触发另外两个信号电平变化时，其图示如下：



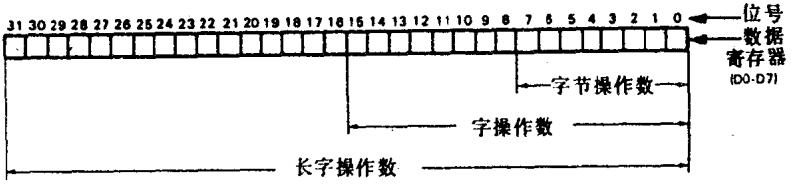
10．所有信号电平的变换都表示为方波，因此不考虑上升和下降的时间。

2 功能概述

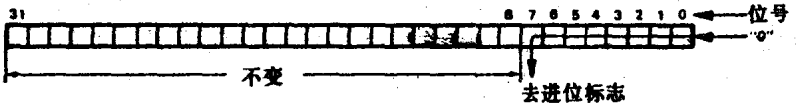
可编程寄存器

图2-1列举了MC68000所提供的寄存器，共有17个32位的数据和地址寄存器；1个32位程序计数器（只用其24位），以及1个16位状态寄存器。MC68000所提供的寄存器和其它16位微处理器的寄存器之间最重要的区别，就在于其数据和地址寄存器全部是32位字长。而8086和Z8000微处理器则为16位字长的寄存器。

数据寄存器　这些数据寄存器可用于处理8位字节（byte），16位字(word)或者32位长字(long)。下图给出了不同长度的操作数是如何放置在数据寄存器里的。

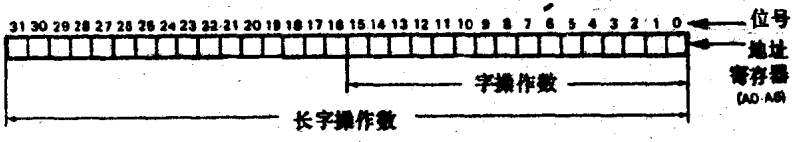


8位字节操作数占用数据寄存器的0到7位，而字操作数占用0到15位。长字操作数占用该寄存器整个32位。当数据寄存器用作源或者目的操作数时，只有该寄存器相应的低位部分通过规定的操作发生变化，那些高有效位并不受影响。例如，若规定一个8位长的操作数作算术左移指令，则只有数据寄存器最低的8个有效位（0～7位）被移位，8至31位不因执行这条指令而变化。



数据寄存器除被指令用作源或目的寄存器外，还可以作为变址寄存器或数据计数器。

地址寄存器　　共有7个通用地址寄存器（A0～A6）。这些寄存器可以处理16位字或者32位长字的操作数。当采用其中一个地址寄存器来提供源操作数时，不是用其低16位（如果已规定为字操作数），就是用整个32位（如果已规定为长字操作数）。若地址寄存器用作字的源操作数，则高16位有效位（16～31位）不受影响。但是，如果地址寄存器用作目的操作数时，则不管规定为字或长字操作数，整个寄存器都受影响。如果地址寄存器指定为字目的操作数，则这个字输入该寄存器之前自动将符号扩展至32位。



正如我们已经指出的那样，MC68000所有的数据和地址寄存器都是32位长，而Z8000和8086只是16位长。MC68000和8086的寄存器之间另一个重要的区别是MC68000寄存器为通用性的。这点和Z8000所采用的方法相似，它提供给程序员以更大的灵活性。虽然数据和地址寄存器（A0～A6）在处理不同数据长度的方式上有点差别，但每个寄存器用法确相似。专用寄存器有堆栈指针寄存器（A7，管理和用户方式）、程序计数器和状态寄存器。现在我们就来讨论这些寄存器。

堆栈指针　　MC68000可以工作在管理（或系统）方式或者用户(或正常)方式。状态寄存器中S位的状态决定MC68000选择哪种工作方式。管理方式通常是用于操作系统软件，用户方式一般是用

于应用程序。有些指令规定为特权指令，它们只有当处理器处于管理方式时才能执行。管理方式和用戶方式都有单独的堆栈指针。如图 2－1 所示，这两个堆栈指针作为地址寄存器A 7 来寻址。当 MC68000工作在管理状态时，用戶堆栈指针不能被访问。反之，当MC68000处于用戶状态时，管理堆栈指针不能被访问。用戶和管理堆栈指针二者的操作方法是相同的，即这些系统堆栈都是从高地址单元填充到低地址单元。调用子程序时，程序计数器的内容推入到相应的系统堆栈（管理或用戶堆栈）。当从子程序返回时，程序计数器的内容便从该堆栈中弹出，并重新送到程序计数器。因为程

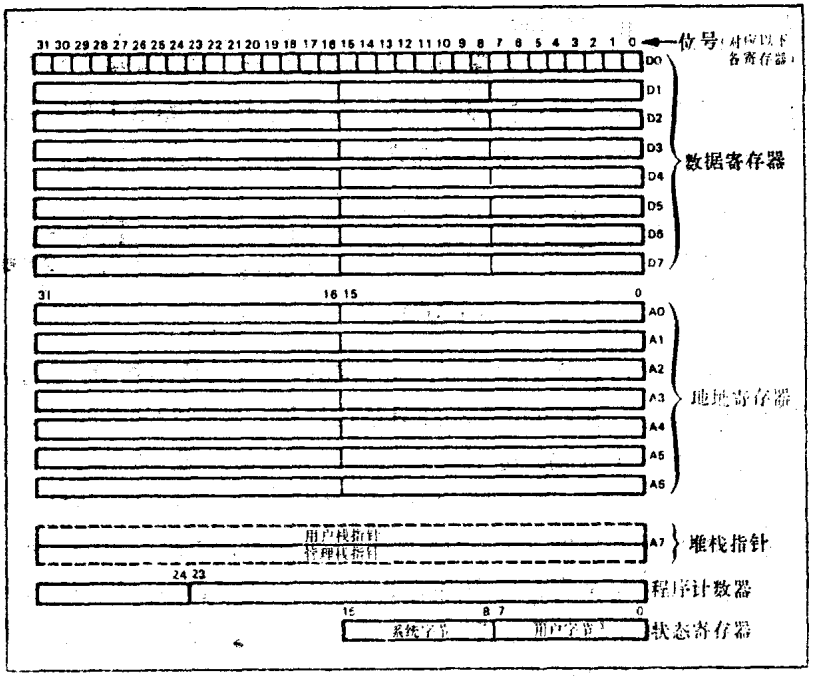
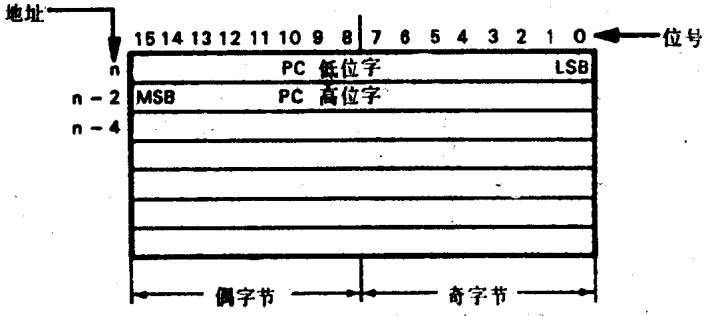


图 2 — 1　　可编程寄存器

序计数器是一个32位寄存器，所以堆栈需要占存储器 4 个字节（二个字）来存程序计数器的内容。调用子程序后，程序计数器的内容

在系统堆栈内的配置情况如下图所示：



推入堆栈的数据常按字界（偶地址）来写入，即写入到偶地址存储单元。所以，当数据字节推入堆栈时，它们写入到存储器的高位字节部分，低位字节（对应存储器奇地址）不变。

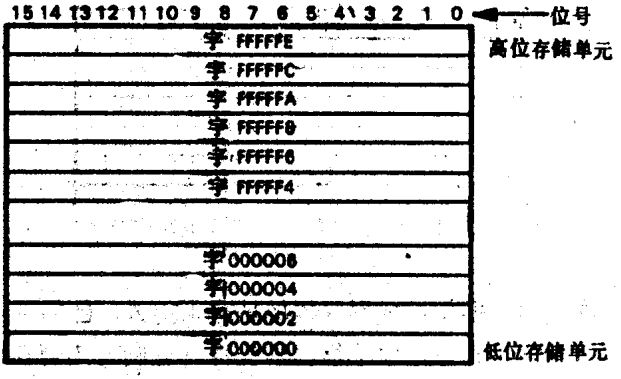
MC68000可按字节或者按包括二个字节的字来寻址存储器。所有的字必须去访问偶地址，否则，如微处理器在存储器奇地址进行字操作时，就会发生错误。虽然，这类问题在任何16位微处理器中都存在，但只有MC68000微处理器能自动检测，以保证访问存储器偶地址的所有字。如果访问发生在存储器奇地址，则MC68000便开始异常处理序列，关于这一点将会在以后讲到。

下图给出字节在存储器中的放置情况：

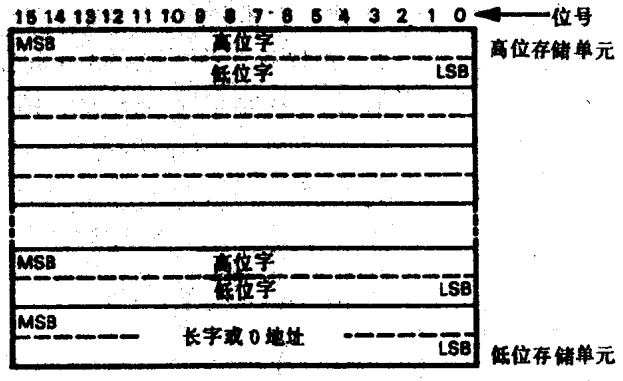
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ←位号

字节 FFFFFE	字节 FFFFFF	高位存储单元
字节 FFFFFC	字节 FFFFFD	
字节 FFFFFA	字节 FFFFFB	
字节 FFFFF8	字节 FFFFF9	
字节 FFFFF6	字节 FFFFF7	
字节 FFFFF4	字节 FFFFF5	
字节 000006	字节 000007	
字节 000004	字节 000005	
字节 000002	字节 000003	
字节 000000	字节 000001	低位存储单元

可见，存储器中第一个字节（如地址000000）为存储字的高有效字节。当字存入存储器时正如上边谈到的那样，它们只在偶地址寻址。如下图所示：



当32位长字（如32位地址）存入存储器时，它们占两个相邻的16位存储单元或四个字节。长字的高位字存在存储器的高位单元。如下图所示：



状态寄存器　MC68000有一个16位的状态寄存器。它分成二个字节：系统字节和用户字节。图 2-2 给出了状态寄存器中各位的用途。其中进位，溢出，零和负等位是多数微处理器所具有的。

如果最高有效位在加法运算之后有进位，或者减法运算时要求从最高有效位借位时，进位位（C）便置位。此状态位也可用某些

移位和循环指令来改变。

溢出位（V）是在算术运算中操作数最高有效位的进位和次高位进位的“异或”。溢出位置位，说明运算结果已不能用规定的操作数字长来表示了。

零位（Z）是当运算结果为零，它即被置位，否则便复位。

负位（N）相当于其它微处理器的符号位。该位等于算术运算结果的最高有效位的值。如果带符号的二进制算术运算完成后，负状态位是“0”则表示结果是正或是零；反之，负状态是“1”则说明结果为负。

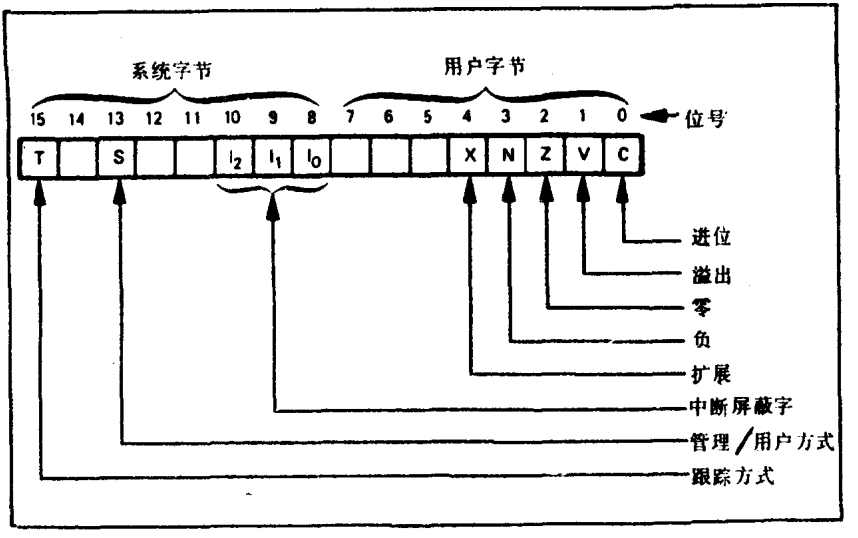


图 2—2　状态寄存器

扩展位(X)用于多精度运算。当指令使其改变时，它便被置成和进位位同样状态。状态寄存器用户字节的三个高位（5、6、7位）一般不用，因而总是零。

状态寄存器中的“系统字节”装有与系统有关的状态信息。“用户字节”则装有与指令或程序有关的状态位（X，N，Z，V，