

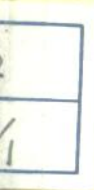
Z80000

中央处理器技术手册

清华大学微计算机教研组 译

Z80000

清华大学出版社



Z 8000

中央处理机技术手册

清华大学微计算机教研组 译

000552

清华大学出版社

内 容 简 介

80年代将是微处理机技术更加蓬勃发展的十年, 其中16位微处理机将起主要作用. Z8000是当前流行的三种16位微处理机中的一种机型. 它已广泛使用在通用和专用的计算机系统中.

本书介绍Z8000的结构、指令系统以及接口技术等主要特点. 第一章为Z8000微处理机性能综述; 第二、三和四章讨论CPU结构和操作方式; 第五、六章详细叙述Z8000的寻址方式和指令系统; 第七章讨论其中断工作; 第八章为刷新技术; 第九章叙述Z8000的外部接口技术.

本手册可供微计算机领域内工程技术人员、高等院校高年级学生、研究生和教师作为参考资料. 对于从事16位微处理机技术的技术人员, 本手册亟有益.

JS400/07

Z8000中央处理机技术手册

清华大学微计算机教研组译

☆

清华大学出版社出版

北京 清华园

轻工业出版社印刷厂排版

河北省固安县印刷厂印刷

新华书店北京发行所发行 · 各地新华书店经售

☆

开本: 787×1092 1/16 印张: 16¹/₂ 字数: 422千字

1984年3月第一版 1984年2月第一次印刷

印数: 1~35,000

统一书号: 15235·99 定价: 2.50元

译 者 的 话

16位微处理机技术，是目前国内微计算机领域内技术人员普遍关心的问题。由于大规模集成电路工艺的迅速发展，微处理机电路的集成度和功能提高得很快。如果说，七十年代中期和末期，8位微计算机系统是占统治地位的话；那末八十年代的初期和中期，16位微计算机系统将占优势。现在，16位微计算机系统产品已经逐渐问世，其功能很多已超过过去的低、中档小型机，可是它的价格要便宜很多。可以预计，16位微处理机和微计算机会得到很广泛的应用。

我们把Zilog公司的“Z8000 CPU 技术手册”翻译出来，其目的是向读者介绍Z8000微处理机技术。读者也许记得，1975年到1976年间8080微处理机手册中译本的出版，曾经在很大程度上推动了我国八位微处理机技术的发展。我们衷心希望：Z8000微处理机手册中译本的出版，将有助于我国16位微处理机技术的发展。

‘Z8000 CPU技术手册’是清华大学计算机工程与科学系微计算机教研组根据Zilog公司的‘Z8000 CPU Technical Manual’ 1980年5月的版本翻译的。其中第一章、第六章大部分由李三立翻译，第二、三、四、五章及第六章的一部分和附录D由潘孝梅翻译，第六章的一部分及第七、八、九章由唐龙翻译，第六章的一部分由王秀玲翻译，附录A、B、C由马群生翻译，全书由林定基校对。

由于译者水平有限，翻译过程一定有很多缺点和错误，敬请读者能给予指正。

目 录

第一章 Z8000 处理机概论	1
1.1 引言	1
1.2 一般结构	1
1.3 结构特点	1
1.3.1 通用寄存器堆	2
1.3.2 指令系统	2
1.3.3 数据类型	3
1.3.4 寻址方式	3
1.3.5 多个存贮地址空间	3
1.3.6 操作的系统/正常方式	3
1.3.7 分开的输入/输出地址空间	4
1.3.8 中断结构	4
1.3.9 多处理	4
1.3.10 Z8000 的巨大地址空间	5
1.3.11 Z8001 分段编址	5
1.3.12 存贮管理	5
1.4 结构的优点	5
1.4.1 程序编码密度	5
1.4.2 编译效率	6
1.4.3 操作系统支援	6
1.4.4 支援多种类型的数据结构	7
1.4.5 两种CPU类型: Z8001和Z8002	7
1.5 扩展指令设备	7
1.6 总结	8
第二章 结构	9
2.1 引言	9
2.2 一般结构	9
2.3 硬件接口	10
2.3.1 地址/数据线	10
2.3.2 段号 (仅 Z8001)	10
2.3.3 总线定时	11
2.3.4 状态	11
2.3.5 CPU 控制	11
2.3.6 总线控制	11
2.3.7 中断	12

2.3.8	分段陷阱请求 (仅 Z8001)	12
2.3.9	多微控制	12
2.3.10	系统输入	12
2.4	定时	12
2.5	地址空间	13
2.5.1	存贮器地址空间	13
2.5.2	I/O 地址空间	13
2.6	通用寄存器	13
2.7	专用寄存器	14
2.7.1	程序状态寄存器	14
2.7.2	程序状态区域指示器 (PSAP)	15
2.7.3	刷新计数器	16
2.8	指令的执行	16
2.9	指令	16
2.9.1	指令格式	17
2.10	数据类型	17
2.11	寻址方式	17
2.12	扩展处理结构	12
2.13	例外	18
2.13.1	复位	18
2.13.2	陷阱	18
2.13.3	中断	19
2.13.4	陷阱与中断服务程序	19
第三章	地址空间	20
3.1	引言	20
3.2	地址空间的类型	20
3.3	I/O 地址空间	21
3.4	存贮器地址空间	21
3.4.1	可寻址数据单元	21
3.4.2	分段和非分段地址	22
3.4.3	分段和存贮器管理	22
第四章	CPU 操作	24
4.1	引言	24
4.2	操作状态	24
4.2.1	运行状态	24
4.2.2	停止/刷新状态	24
4.2.3	总线脱开状态	24
4.2.4	复位的影响	24
4.3	指令执行	25
4.3.1	运行状态的方式	26

4.3.2	分段与非分段方式	26
4.3.3	正常和系统方式	26
4.4	扩展指令	27
第五章	寻址方式	28
5.1	引言	28
5.2	CPU寄存器的用法	28
5.3	寻址方式说明	29
5.4	说明与例子 (Z8002和Z8001非分段方式)	30
5.4.1	寄存器 (R)	30
5.4.2	立即方式 (IM)	30
5.4.3	间接寄存器 (IR)	31
5.4.4	直接地址 (DA)	31
5.4.5	变址 (X)	32
5.4.6	相对地址 (RA)	33
5.4.7	基地址 (BA)	34
5.4.8	基址变址 (BX)	35
5.5	说明与举例 (分段Z8001)	36
5.5.1	寄存器 (R)	36
5.5.2	立即方式 (IM)	37
5.5.3	间接寄存器	38
5.5.4	直接地址 (DA)	39
5.5.5	变址 (X)	40
5.5.6	相对地址 (RA)	41
5.5.7	基地址 (BA)	42
5.5.8	基址变址 (BX)	44
第六章	指令系统	46
6.1	引言	46
6.2	功能综述	46
6.2.1	送入和交换指令	46
6.2.2	算术运算指令	48
6.2.3	逻辑指令	49
6.2.4	程序控制指令	50
6.2.5	按位处理指令	51
6.2.6	循环移位和移位指令	51
6.2.7	成组传送和串处理指令	52
6.2.8	输入/输出指令	54
6.2.9	CPU 控制指令	56
6.2.10	扩展指令	56
6.3	处理机标志	57
6.4	条件码	58

6.5	指令中断和陷阱	59
6.6	表示法和二进制编码	60
6.7	Z8000 指令描述和格式	62
6.8	EPA 指令样板	183
第七章	例外	187
7.1	引言	187
7.2	中断	187
7.2.1	不可屏蔽的中断 (NMI)	187
7.2.2	向量中断 ($\overline{VI}$)	187
7.2.3	非向量中断 (NVI)	187
7.3	陷阱	187
7.3.1	扩展指令的陷阱	188
7.3.2	特许的指令陷阱	188
7.3.3	系统调用陷阱	188
7.3.4	(程序) 分段陷阱	188
7.4	复位	188
7.5	中断的禁止	188
7.6	中断和陷阱的处理	189
7.6.1	认可周期	189
7.6.2	状态的保存	189
7.6.3	加载新的状态	189
7.6.4	执行服务子程序	190
7.6.5	从一次中断或陷阱中返回	190
7.7	优先级	191
第八章	刷新	192
8.1	引言	192
8.2	刷新周期	192
8.3	周期刷新	192
8.4	停止状态刷新	193
第九章	外部接口	194
9.1	引言	194
9.2	总线操作	194
9.3	CPU 引线	195
9.3.1	执行处理的引线	195
9.3.2	总线控制引线	196
9.3.3	中断/陷阱引线	196
9.3.4	多重微 (指令) 引线	196
9.3.5	CPU 控制	196
9.4	操作处理	196
9.4.1	等待 ($\overline{WAIT}$)	198

9.4.2	存储器操作处理	199
9.4.3	I/O操作处理	200
9.4.4	EPU传送操作处理	200
9.4.5	中断/陷阱认可操作处理	201
9.4.6	内部操作和刷新操作处理	202
9.5	CPU和扩展处理单元交互作用	203
9.6	请求	204
9.6.1	中断/陷阱请求	204
9.6.2	总线请求	204
9.6.3	资源请求	205
9.6.4	停止请求	205
9.7	复位	206
附录A	硬件资料	207
附录B	Z8000 系列技术说明	213
附录C	程序员参考资料	228
附录D	术语汇编	251

第一章 Z8000 处理机概论

1.1 引言

本章对Z8000微处理机的先进结构作综合性的描述,尤其着重讨论Z8000CPU所不同于以前其它微处理机的结构特点。第二章将对其结构作完整的讨论,在其后各章中将详细论述处理机的不同方面。

1.2 一般结构

Zilog公司的Z8000微处理机的设计能满足范围很广泛的应用领域,从相当简单到很庞大、很复杂的应用。Z8000 CPU有两种型式:Z8001与Z8002。每种CPU都有一套完整的配套部件系列:一个存储器管理部件,一个DMA控制器,串行与并行I/O控制器,以及扩展处理部件——所有这些都是在Zilog的Z总线相兼容的。高级的Z8000CPU结构与其它Z8000系列部件一起,组合在LSI微处理机设计中,提供了通常小型计算机或大型计算机才具有的灵活性和成熟性。

Z8000CPU用以增强其吞吐率和处理能力的主要结构特点是:它具有一个通用的寄存器堆,系统的和正常的操作方式,多个寻址空间,一个强有力的指令系统,很多种寻址方式,多个堆栈,精致的中断结构,丰富的数据类型组,专门分开的I/O地址空间,以及Z8001还有很大的地址空间与分段存储编址。所有这些特点将在下节中详细讨论。

这些结构特点结合在一起,构成一个强有力的和灵活的微处理机。这些特点所带来的好处有:写程序时编码紧凑、编译效率较高,能支援典型的操作系统操作以及复杂的数据结构。这些将在1.3节中讨论。

设计CPU时,考虑了用一个功能很强的存储管理系统来改善主存储器的利用,并为系统提供保护能力。这将在1.3.12节中讨论。虽然存储管理是一种任选的功能(Z8000CPU是一个没有存储管理部件的非常精巧的微处理机),但是CPU具有易于把外加的存储管理设备结合到Z8000系统中去的显著特点。

最后,由于精心的设计,使得可以通过使用外部设备(叫做扩展处理装置EPU),提供扩展基本指令系统的通用机构。一般来说,EPU专门用来执行复杂和费时的任务,以减轻CPU的负担。对于专门的EPU,其典型任务包括浮点算术运算、数据库搜索与维护操作,网络接口以及其它等等。这个问题将在1.5节讨论。

1.3 结构特点

Z8000 CPU的结构资源包括:十六个16位通用寄存器;七种数据类型,其范围从1位到32位的长字和字节串;八个用户选择的寻址方式;以及一个比大部分小型计算机还要强有力的指令系统。110种不同指令类型,与各种数据类型与寻址方式结合在一起,组成具有414

条指令的丰富指令系统。此外，这个指令系统表现出很高程度的规则性：90%以上的指令可以使用五种主要寻址方式中的任一种，具有8位字节，16位字以及32位长字的数据类型。

CPU产生出一些状态信号，用来指示要进行的总线处理操作的性质。这些信号可用于实现具有多个地址空间（供不同用途专用的存贮区）的高级系统。CPU也有两种操作方式，系统的和正常的，它们可以用来把操作系统功能与正常应用过程区分开来。I/O操作与存贮器访问也是分开的，这样可进一步增强Z8000做成的系统的能力和完整性。还有一个高级中断结构有助于提高外围I/O设备操作的效能。此外，Z8000的扩展处理部件（EPU）能力使CPU把一些费时任务的负荷卸给外部的设备。

Z8000中还引进某些特点，它们有助于实现多个处理机系统。而且，Z8001 CPU具有很大的、分段的寻址能力，这些使该微处理机非常适用于构成很大的系统。

1.3.1 通用寄存器堆

Z8000 CPU结构的核心是一个由十六个16位通用寄存器组成的堆。这些通用寄存器使Z8000功能更强而且灵活，并使指令结构增添规则性。

通用寄存器可以用作累加器、存贮器指针或者变址寄存器。其主要优点是：在程序执行过程中，这些寄存器的特定用途可以随着程序的需要而改变。因此，这种通用寄存器堆可以避免在其它指定或专用寄存器结构中的严重阻塞现象。因为在专用寄存器结构中，当程序所需使用的某些特定类型寄存器的数量超过处理机所能供给的数量时，就必须把专用寄存器的内容保存起来，并且还要恢复。

Z8000 CPU寄存器堆可以有若干种编址方式：作为16个字节寄存器（占据堆的上半部）；或者用寄存器配对的结构来作为16个字寄存器；或者作为8个长字（32位）寄存器；或者作为4个四倍字长（64位）寄存器。鉴于这些寄存器的灵活性，因此，举例说，没有必要使Z8000用户把一个32位的寄存器去保存一个只有一字节长的数据，所以在Z8000中这些寄存器的使用效率很高。

1.3.2 指令系统

Z8000的出色特点之一是它具有强有力的指令系统。指令系统是衡量计算机的灵活性和通用性的一个尺度。在给定的硬件提供的操作情况下，它可节省存贮器空间并且改进速度。此外，对于特定数据类型所能执行的操作应是很完备的，这点比增加一些深奥不大影响计算机性能的指令更加重要。Z8000 CPU提供一套完备的算术、逻辑运算，转移，I/O，移位，循环移位和字符串指令。此外还包括一些特殊指令，有助于多重处理，多个处理机结构，以及典型的高级语言和操作系统功能。指令系统的总的原则是采用双操作数寄存器—存贮器操作，它包括一个作为特殊子组的寄存器—寄存器操作。然而，为了提高编码密度，还使用少量存贮器—存贮器操作来做字符串处理。两地址指令格式能表现最经常进行的操作（诸如 $A \leftarrow A + B$ ）。还有，如果把两个操作数中的一个放在可以快速访问的通用寄存器中，这将有助于使用在计算中产生的中间结果。

大多数的操作是处理字节、字或者长字操作数，因而能提供高度的规则性。在指令系统中还包括最常用操作的紧凑的、单字长的指令，诸如在一个程序中的近距离转移。

比较于过去早期的微处理机标准指令库来说，本指令系统有显著的增加。如装入和交换指令组已经扩展，以支持操作系统的功能和现有微处理机程序的转换。通常的算术运算指令现在可处理高精度的操作数，同时还增加了硬件乘法和除法指令。按位处理指令可以用计算值以确定一个字节或一个字中某位的位置，也可静态地确定在指令中的位置。循环移位和移

位指令要比以前的微处理器灵活得多。字符串指令对于不同字符码之间的转换是很有用的。还有一些特殊指令有利于多个处理机结构。

1.3.3 数据类型

Z8000 结构可以支持很多种数据类型。当一种数据类型具有硬件表示形式并且具有能直接使用它的指令时，可以认为该数据类型获得了支持。新的数据类型都是可以用基本数据类型来表示的，但硬件支持会使操作更快并更方便。基本的数据类型就是字节，它也是基本的可编地址的单元。Z8000 结构还可支持以下数据类型：字（16位），长字（32位），字节串，以及字串。此外，在一个字节或字中的数号可以支持位的数据类型并对它编址。二一十进制（BCD）数字也得到支持，它可以用一个字节中的两个 4 位数字表示。数组是用变址寻址方式支持的（见第 5 章 1.3.4 节）。堆栈是由指令系统以及 Z8001 所配有的外部支援部件（存贮管理部件 MMU）所支持的。

1.3.4 寻址方式

寻址方式（它是一条指令中确定一个操作数的方法）将决定一个地址是如何产生的。Z8000 CPU 具有八种寻址方式。它与大量的指令以及数据类型结合在一起，大大改进了 CPU 的处理能力。八种寻址方式为：寄存器，立即型、间接寄存器，直接地址，变址、相对地址、基数地址以及基数变址。一些特殊指令还有若干其它隐含的寻址方式，其中包括自动增量。以上列举的前五种方式是基本寻址方式，它们使用最频繁，而且适用于具有一种寻址方式以上的大多数指令。（在 Z8002 中，基数地址和变址方式是相同的，在 Z8001 中，使用存贮管理部件后，基数寻址功能可以对所有指令都适用。）

1.3.5 多个存贮地址空间

Z8000 CPU 有利于多个存贮地址空间的使用。当 Z8000 CPU 产生一个地址，它也输出指明某些特殊内部动作的信号。这些信号将请求存贮器做：取指令，访问操作数，或者访问堆栈。这些信息可以有两种用法：增加处理机所能用的存贮空间（例如，把程序放在一个空间，而数据则在另一个空间）；或者保护某一部分存贮器，只允许某种类型的访问（例如，只允许从某个指定为包含专用软件的区域内取指令）。依靠使用 CPU 产生的状态信息，所设计的存贮管理部件可精确提供这些类型的保护性能。

1.3.6 操作的系统/正常方式

Z8000 CPU 既可在系统方式运行，也可在正常方式运行。在系统方式中，所有指令都可执行，所有 CPU 寄存器都可访问。这种方式是为了执行操作系统功能的程序而使用的。在正常方式，某些指令不可执行（如输入/输出操作），而且 CPU 的控制寄存器是不可访问的。通常这种操作方式是为应用程序使用的。这种 CPU 资源的划分，使系统的完整性更好，因为工作于正常方式的程序不能访问处理与时间有关或者系统接口事件的那些 CPU 部分。

在正常方式中执行的程序如有错误，则只要用它原来的数据去重新执行程序，则永远可以使这些错误重现，以便于调试程序，而使用只在系统方式时有用设备的那些程序，会产生一些与时间有关的错误（例如，与磁盘请求频率和磁头臂位置有关的错误），这些错误比较难于调整，因为这些错误不易于重现。因此，在开发程序时宁可采用这种方法，即把任务划分成两部分，一部分是在执行时可以不使用那些只允许在系统方式中访问的资源（它一般应是该任务的大部分），而另外一部分则是需要系统方式资源的。这种划分的办法是来源于当前的小型计算机和大型计算机系统：操作系统在系统方式中运行，而个别用户所写程序则在

正常方式中运行。

为了进一步支持这种系统/正常方式的“两分法”，这里设有双份堆栈指示器（或称堆栈指针）——一个是为系统方式堆栈，另一个是为正常方式堆栈使用的。这两种堆栈有利于发生中断或陷井时任务的切换。为了确保正常堆栈中不会有系统的信息，在发生中断或陷井时要保存的信息，必须新的程序状态送入以前就推入系统堆栈。

1.3.7 分开的输入/输出地址空间

Z8000 结构是把存储器地址和输入/输出地址空间区分开的，这样就需要有专用的输入/输出指令。这种结构上的区分可以有更好的保护性能以及更便于扩充。使用分开的输入/输出空间，也可以节省一些 Z8002 的有限的数据存储空间。实际上还有两种互相区分的输入/输出地址空间：标准输入/输出以及特殊输入/输出。分成两个空间的主要好处是可以提供两种类型的外围支援电路——标准的输入/输出外围电路和特殊的输入/输出外围电路——诸如 Z8010 存储管理部件，它对标准输入/输出命令并不响应，但对特殊输入/输出命令是响应的。这两个空间的第二个好处是可允许 8 位的外围电路可附在处理机地址/数据总线的低 8 位（标准输入/输出），或者高 8 位（特殊输入/输出）。

未来微处理机所要求的高速度大概是用以下这些方法达到的：即把存储器访问和输入/输出访问恰好适合于它相应的访问内容的特点，而且还可以使输入/输出访问和存储器访问同时进行。这些未来的可能性要求今天就做出结构上的区分。当然仍然可以采用存储器对应（映象）输入/输出方式，但这会失去保护的功能，而且缺少可扩展性，这些都是很严重的问题。

1.3.8 中断结构

Z8000 的高级中断结构可以使处理机在等待外部事件要发生的同时，可以继续执行有用的工作。这样可以消除周期性的查询以及空闲的循环（这是用来决定什么时候外设准备好发送数据），从而可以增加系统的吞吐率。CPU 提供三种中断类型。一种是非屏蔽型中断，它表示在灾难事件发生时必须立即加以处理以确保系统完整性。此外，还有两种可屏蔽型中断：非向量中断和向量中断。后者将根据外围给 Z8000 的向量，自动转移到中断处理子程序。

Z8000 对于处理中断提供一个优先权系统。向量中断优先权高于非向量中断。这个优先权线路使得在 Z8000 中的很多外围设备得到有效的控制。

中断会使与当前执行程序有关的信息（即程序状态）保存在一个专门的系统堆栈中，同时还有一个码要说明信息切换的原因。这就允许出现递归任务的切换，而同时使正常堆栈不致受到系统信息的打扰。处理中断的状态程序（新的程序状态）是从存储器中一个由 CPU 内部指针所指定的特殊区域所装入的，这个特殊区域叫程序状态区域。

堆栈以及指向程序状态区的指针的使用是一种特殊安排，如在结构上要另加新的中断或陷井时，它仍能维持结构上的兼容。

1.3.9 多处理

Z8000 所代表的微处理机计算能力的增强，使得具有很多低价微处理机的分布式处理系统的设计变得更为简单。在这种分布系统中，不同微处理机可以处理其专门的过程。

Z8000 设有某些基本的机构，使得不同微处理机之间可以共享地址空间。巨大的、分段的地址空间，以及外部存储管理的支援使这点成为可能。此外，还设有资源请求总线，它与软件结合在一起，就可以很好地利用分享关键性的资源。这些机构，以及设计的新外围电

路, 如Z-FIO, 使得不同CPU之间非同步通信更为方便.

1.3.10 Z8001 的巨大地址空间

对于很多用途, 64K字节的基本地址空间是不够的. 较大的地址空间可以增大应用的范围, 这样可以使一些较大的和复杂的程序和数据组可以直接存放在存贮器中, 而不必把它划分并在需要时交换到小存贮器中去. 一个较大的地址空间可以使程序和数据管理大为简化. 此外, 较大的地址空间和存贮器可以降低对程序规模简化的要求, 从而可以使用高级语言. Z8000的可分段型产品可以产生23位的地址, 每个基本地址空间为8兆字节(8,388,608字节).

1.3.11 Z8001分段编址

Z8000 CPU的可分段型产品(即Z8001)把它的23位地址分成为一个7位的分段号以及一个16位的分段偏置值. 分段号是作为一个分段的逻辑名, 它是不因有效地址计算(如变址)而改变的. 这相应于一般程序使用存贮器的方法, 即分出一部分存贮器存放指令, 而另一部分存放数据. 在一个分段的地址空间中, 指令应放置在一个分段中(或者若干不同程序模块放在不同分段中), 而每个数据组则放置在另外分开的分段中. 分段的一个优点是它可以加速地址的计算和重新定位. 这样分段方法比起线性编址方法来说, 可允许使用较慢的存贮器. 此外, 分段提供划分存贮器的方便办法, 使得每个划分区域都给予一个专门的访问属性(例如, 只读). 如果分段的所有优点实现的话, Z8000(同时访问大量的分段)的分段方案是必需的. 如果一个系统只能直接访问少量的分段(如, 4个分段), 则该系统将缺乏必要的灵活性, 而且将因地址空间限制而受到很大约束.

1.3.12 存贮管理

存贮管理主要包括动态再定位、保护、以及存贮器共享. 它具有以下优点: 对存贮空间提供一个与数据的实际物理位置无关的逻辑结构; 保护用户免受疏忽引起错误的影响, 如企图去执行数据; 防止未受批准的对存贮器资源或数据的访问; 保护操作系统以免被用户破坏.

凡是地址是由程序员所控制的, 由指令所使用的, 以及由分段的Z8000 CPU所输出的, 都称为逻辑地址. 外部的存贮管理系统拿到这逻辑地址, 并且把它们转化为访问存贮器所需要的物理地址. 这种地址转化过程, 称为重新定位, 它使得用户软件可以与物理存贮器(实际的存贮器)无关. 因此, 用户不需要说明信息在物理存贮器中实际装在什么地方.

分段的Z8000 CPU, 其分段的编址方法以及程序状态信息都可以支援存贮管理. 一个分段的编址空间可以使逐个分段得到不同处理.

由CPU产生的程序状态信息, 可使一个外部存贮管理部件来监控所要用的每次对存贮器的访问. 这样可以制止不合法的访问, 并且可以在非计划中的或者不要求有的使用方式时保护存贮器分段. 举例说, 可以保护系统的表格不受直接的用户访问. 由于微处理机已经应用于愈来愈大、愈复杂的任务, 上述所增设的保护功能变得更为重要.

1.4 结构的优点

Z8000结构的一些特点组合在一起, 可以提供以下好处: 改进程序编码的密度, 编译效率、操作系统支援、以及高级数据结构的支援.

1.4.1 程序编码密度

编码密度能影响处理机速度和存贮器利用。编码紧凑可以节省存贮空间，这在小型机中是特别重要的因素，并且由于减少必须取指和译码的指令字数目而改进了处理机速度。Z8000 对于编码密度提供很多有利地方。最经常使用的指令是编成单字格式的。为了完成同样给定任务所需指令数较少，以及有规则的和一致性较好的结构，都能进一步减少所需的指令数目。

编码密度之所以能够很高，部分地还是由于对某些指令使用了特殊的“短格式”，这些指令是在对于汇编程序进行统计分析后，表明它们是最经常使用的。同时还提供一种“短偏置”机构，它可以使一个两字的分段地址缩减成一个单字；这种格式可以由汇编程序和编译程序所使用。

程序规模的大幅度缩减以及工作速度的提高，最根本地是归因于体系结构上的一致性很好以及有规则的结构，同时也归因于非常强有力的指令系统，这些因素极大地减少了一个任务所需的指令条数。由于Z8000的寄存器，地址方式和数据类型可以在一种更有次序的方式下使用，所以比起以前的微处理机来说，它的结构更有规则。任何通用寄存器可以指定为累加器，变址寄存器，或者基数寄存器，除了少数例外，所有指令都可使用所有基本寻址方式，也可用各种数据类型。

通用寄存器不必象指定为专用的寄存器那样经常改变。因为这样不需要经常做装入和存入操作，从而可以减少程序容量。

1.4.2 编译效率

对于微处理机用户来说，如果能使用高级语言而不是汇编语言，则可以使用户大大免于对具体计算机结构的依赖性，而且程序设计更为方便。然而，Z8000的结构并不是专门适用于某一种高级语言，而是作为一个通用的微处理机而设计的。（如果把处理机设计得只对某一种语言很有效，往往会使得对另一些其它语言的效率很低。）对于Z8000，由于所包含的一些设计特点可以使典型的编译过程和生成编码的问题简化，从而提供对语言很好的支援。在这些特点中，尤其是以Z8000寻址方式与数据类型的有规则性最为突出。在过程堆栈中对参数和局部变量的访问，是由“短偏置变址”寻址方式来支持的，当然用基数地址和基数变址寻址方式也可以。此外，地址算术运算是用增量和减量指令来辅助的。

测试，条件码测试，立即数装入存贮器，以及把立即数与存贮器比较等指令，使得数据的测试、逻辑计算，初始化和数据比较做起来很方便。因为编译程序和汇编程序经常处理字符串，所以翻译，翻译和测试，成组比较，以及比较串等指令都能使其速度大为提高。这一点比用软件来模拟这些重要任务要好得多。此外，除了R0寄存器以外，所有寄存器都可被推入和弹出指令当作堆栈指针使用。

1.4.3 操作系统支援

Z8000的中断和任务切换等特点都有利于操作系统的实现。存贮管理和编译支援特点也十分重要。

中断结构有三个级别：非屏蔽的，非向量的和向量的。当一个中断发生，程序状态是保存在堆栈中，并在一个新程序状态以存贮器某特殊区域装入以前要指出状态切换的原因。程序状态包括标志寄存器，控制字以及程序计数器。发生中断的原因用一个向量来编码，这个向量由系统总线读出并保存到堆栈中。如果有向量中断，这向量也决定一个转移表地址，它将指向中断处理子程序。

引入系统方式和正常方式，也可改进操作系统的结构。在系统方式中，所有操作都允许

做：在正常方式中，某些系统指令是禁止的。系统调用指令可以使工作方式有控制地切换，而陷阱的执行将实施这些限制。

陷阱也会引起象中断那样的保存程序状态：在这两种情况中，信息都是推入到一个系统堆栈中去；而保持正常堆栈不受打扰。多装入指令使寄存器内容有效地保存在存贮器中或在堆栈中。在用装入程序状态指令的直接软件控制下，程序运行会引起程序状态的改变。

最后，用“原子”测试和置数指令可以实现排除和串行比，它可以同步那些非同步的合作过程。

1.4.4 支援多种类型的数据结构

数据结构是原始单元的逻辑机构（字节、字、等等），它的格式和访问约定是精心定义的。公共的数据结构包括数组、列表、堆栈、以及串。因为数据结构是在程序设计中经常使用的高级建筑，如果 CPU 能提供可以有效处理它们的机构的话，则处理机的性能可以大为增强。Z8000 可以提供这种机构。

在很多种应用中，最经常遇到的数据结构之一是数组。在 Z8000 中，数组是由基数变址寻址方式和分段的编址方法所支持的。基数变址寻址方式可以把指针用到数组中去（即偏置址构成数组的起始地址）。分段的编址方法可以把一个数组指定到对应的一个分段中去，它只要用分段号就可以访问。

在商业应用和在通用数据处理中，列表比数组要更为频繁地出现。列表是由间接寄存器和基数地址寻址方式所支持的。基数变址寻址方式对于更复杂的列表也是很有用的。

堆栈在所有应用嵌套或例行子程序、成组结构语言，以及中断处理的场合都要使用。堆栈是由推入和弹出指令所支持的，多堆栈可以靠 Z8000 的通用寄存器来实现。此外，可以使用两个硬件堆栈指针，以指定系统和正常操作方式所分开的堆栈。这样可以进一步支持上述的系统和正常操作环境的区分。

字节和字串是由翻译以及翻译与测试指令所支持的。十进制串使用十进制调整指令对于 BCD 数据执行十进制算术运算。并把两个字符组成一个字节。循环转移数字指令也可处理 4 位数据。

1.4.5 两种 CPU 类型：Z8001 和 Z8002

Z8000 CPU 是以两种形式提供的：即 48 条引线的可分段的 Z8001 CPU 以及 40 条引线的非分段的 Z8002 CPU。这两者主要区别之处在于其寻址范围。Z8001 可以直接寻址 8 兆字节存贮器。Z8002 直接寻址 64K 字节。Z8001 中也可以有一种非分段的工作方式，它可以执行为 Z8002 所写的程序。

并不是所有应用都需要 Z8001 很大的地址空间，在这种场合下建议使用 Z8002。此外，一些多处理机系统可以用一个 Z8001 和若干个 Z8002 来实现，而不必只用 Z8001。因为汇编程序是相同的。它可以为两种 CPU 都产生机器码，用户可以只根据所需功能购买某种 CPU。而不必担心处理机之间是否兼容。

1.5 扩展指令设备

Z8000 结构有这种机能，它可以通过使用外部器件来扩展其基本指令。专门设置了一些操作码用来实现这个特点。当 CPU 在它的指令流中遇到具有这些操作码的指令时，它将执行任何指明的地址计算和数据传送。否则它将如同用外部器件执行那样来处理“扩展指令”。

在这些扩展指令中已留出一些字段，它将被外部器件（扩展处理部件—EPU）翻译成操作码。这样，如使用合适的EPU，Z8000的指令系统就可以扩展以包括一些专门指令。

一般来说，一个EPU是专门用来执行复杂和费时的任务，这样可减轻CPU的负担。典型的为专门EPU所适用的任务包括：浮点算术运算、数据库搜索和维护的操作，网络接口，图形支持操作——一个几乎包括大部计算领域的表格。

1.6 总 结

Z8000微处理机结构的高级程度，已可与小型机的水平相比拟。而这些特点：如巨大的地址空间，多存贮器空间，分段地址，以及对多处理机的支援已经超过传统的小型机。这些结构带来的好处——较高的编码密度，编译支援，以及操作系统支援——极大地增强了CPU的功能和通用性。CPU可以支持外部的存贮管理系统，这一特点也使CPU更加适用于大型系统环境。