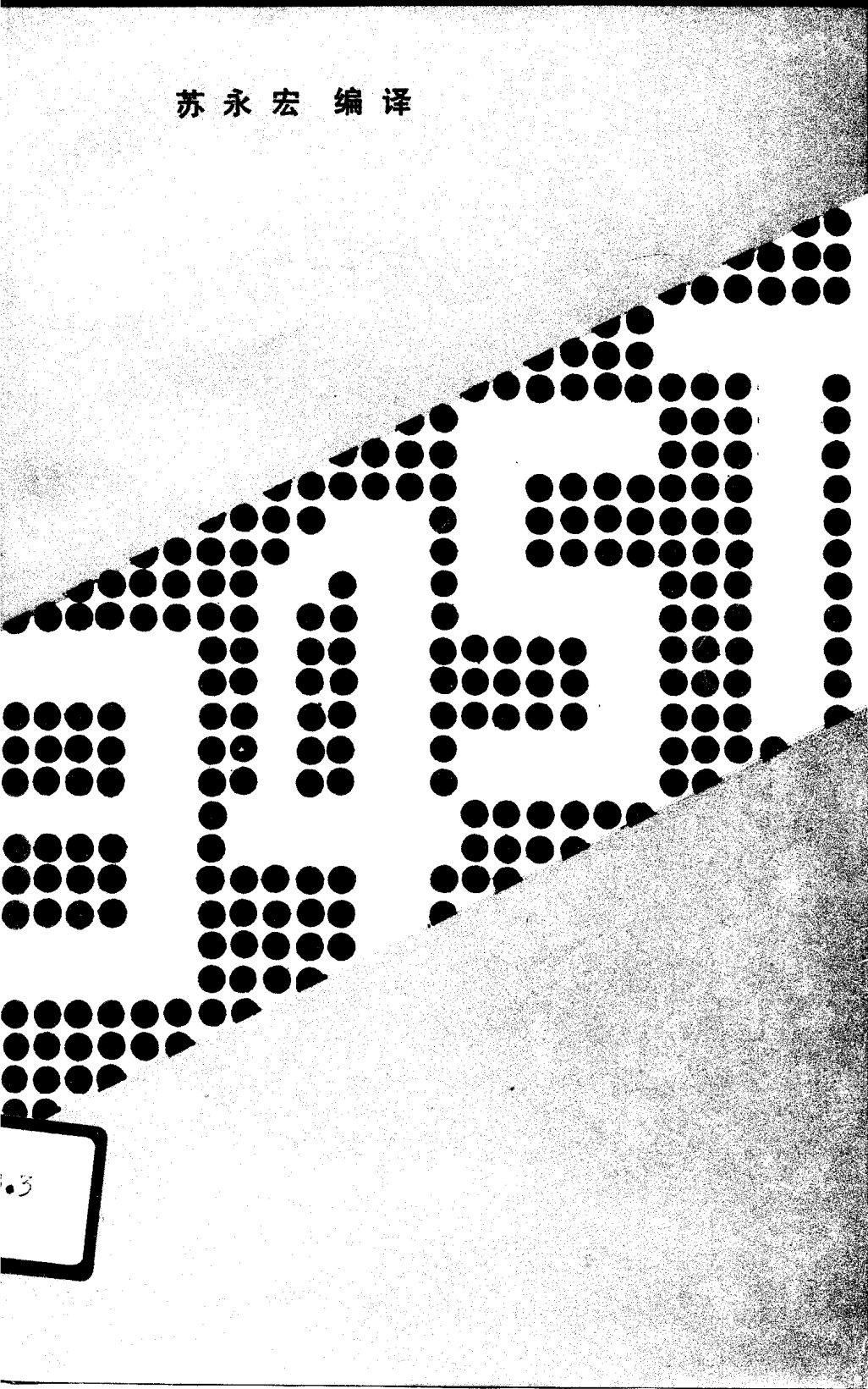


微型计算机及其在蒸馏控制中的应用

苏永宏 编译



内 容 提 要

本书从工程应用的角度出发,综合论述了微型计算机及其在蒸馏控制中的应用。书中概括地论述了国内外近年来在蒸馏控制中应用微型计算机的最新成就,介绍了蒸馏过程计算机控制的现状,阐述了蒸馏塔计算机最优化控制、多变量解耦、多变量补偿控制、自适应控制和数字仿真等问题。

读者对象:炼油、石油化工自控专业工程技术人员和有关大专院校师生。

微型计算机及其在蒸馏控制中的应用

苏永宏 编译

中国石化出版社出版

北京朝阳区太阳宫路甲1号 邮政编码:100029)

海丰印刷厂排版

海丰印刷厂印刷

新华书店北京发行所发行

787×1092毫米 16开本 21¹/₄印张 547千字 印1—1600

1990年10月北京第1版 1990年10月北京第1次印刷

ISBN 7-80043-099-5/TP·004 定价:9.40元

前 言

随着石油化学工业的迅速发展,生产规模不断扩大,经济效益逐步提高,对蒸馏过程控制提出了更高的要求。过去我们往往以经典反馈控制理论为基础,对蒸馏过程进行PID定值调节,这在当前已经难以满足生产上更为严格的要求了。因此,如何根据蒸馏过程的特点,运用现代控制理论来开发新的控制方案,已成为目前蒸馏过程控制的一个重要研究方向。

近年来,人们注意到,微型计算机的应用为在蒸馏过程中运用现代控制理论,实施新的控制方案开辟了广阔的前景。在蒸馏过程控制中引进、推广和应用微型计算机,开展这方面的工作,是石油化工企业技术改造的一个经济而又有效的途径。目前,这方面虽然有了新的进展,但问题还是很多的。为此,需要吸取国内外的先进经验,力求在蒸馏过程控制中更好地应用微型计算机。

本书的编译目标,是概括地论述国内外近年来在蒸馏控制中应用微型计算机的最新成就。因此,本书是根据近年来国内外发表的论著、科技报告以及期刊杂志上的有关文献,经过归纳整理之后编写而成的。

考虑到一些工程技术人员应掌握有关微型计算机方面的基本知识,本书第一章简要介绍微型计算机结构。限于篇幅起见,这章是以读者已经掌握数字运算和数字电路的基本知识为起点的,并就目前国内引进和使用较多的TRS-80机作典型结构分析,使读者了解微型计算机硬件的基本知识。第二章从过程控制应用的实际出发,在一般性地介绍微型计算机软件的基础上,着重介绍汇编语言程序设计,同时,介绍怎样使用BASIC语言调用汇编语言程序等应用软件。为了建立微型计算机硬件、软件与工业生产过程的联系,第三章介绍了微机扩展接口装置、磁盘操作系统、S-100总线、常用接口片子等,并且介绍一种多通道外围接口。第四章在蒸馏过程分析的基础上,对几种常用的控制方案进行了讨论。第五章对蒸馏塔计算机控制作了全面综述,介绍了蒸馏过程计算机控制的现状,并提出了用系统工程方法规划计算机的应用问题。第六章和第七章分别介绍了蒸馏过程最优化和蒸馏塔多变量解耦及多变量补偿等控制方案。第八章内容为蒸馏塔的自适应控制;讨论了两类重要的自适应控制系统及其在蒸馏塔上的应用。第九章介绍如何应用微型计算机对蒸馏过程进行数字仿真。这里着重讨论了蒸馏过程动态数字仿真的基本理论和实验研究方法。

本书从工程应用出发,着重引用现代控制理论综合论述微型计算机及其在蒸馏控制中的应用。

本书致力于系统性、先进性和完整性。同时,对一些专业性较强的理论问题亦作了深入浅出的阐述,有利于广大读者掌握这方面专业知识。

本书编译时,曾经得到宋崇武、庄兴稼、和庄富山、刘继舜、许治平等同志的支持和帮助,并且分别校阅了初稿的部分章节。庄富山同志还为本书编写了第八章内容。在此,表示衷心地感谢。

由于编者水平有限,书中一定存在不妥和错误之处,敬请读者批评指正。

目 录

第一章 微型计算机硬件

§ 1-1 微型计算机的结构特点	2
§ 1-2 微处理器 (MPU)	3
§ 1-3 地址选择器	8
§ 1-4 系统RAM及ROM	12
§ 1-5 视频RAM及视频信号发生器	19

第二章 微型计算机软件

§ 2-1 软件的分​​类	28
§ 2-2 Z80的寻址方式	30
§ 2-3 Z80的指令系统	32
§ 2-4 汇编语言程序设计	43
§ 2-5 汇编语言程序的调用	57

第三章 微型计算机扩展系统

§ 3-1 扩展接口装置	63
§ 3-2 磁盘操作系统	78
§ 3-3 S-100总线及D+7A功能部件	85
§ 3-4 常用接口片子	92
§ 3-5 多通道外围接口	105

第四章 蒸馏过程控制基础

§ 4-1 常规蒸馏过程分析	112
§ 4-2 蒸馏过程控制	126

第五章 蒸馏塔计算机控制概述

§ 5-1 使用微型计算机控制蒸馏塔的现状	141
§ 5-2 用系统工程方法规划微型计算机的应用	148
§ 5-3 微型计算机系统在塔控制方面的应用方案	161
§ 5-4 蒸馏塔控制中的节能方法	166

第六章 蒸馏过程最优化控制

§ 6-1 最优化控制目标	171
§ 6-2 蒸馏过程最优化算法	172
§ 6-3 自动稳定系统的最优化	177
§ 6-4 计算机在线最优化方案	190
§ 6-5 最优状态反馈控制	193

第七章 蒸馏塔多变量解耦及多变量补偿控制

§ 7-1 相对增益和回路相关.....	202
§ 7-2 理想解耦和简化解耦.....	207
§ 7-3 蒸馏塔多变量纯滞后补偿控制.....	211
§ 7-4 蒸馏塔多变量鲁棒补偿控制.....	219

第八章 蒸馏塔自适应控制

§ 8-1 引言.....	229
§ 8-2 蒸馏塔自校正控制的基础知识.....	234
§ 8-3 蒸馏塔的自校正控制之一.....	253
§ 8-4 蒸馏塔的自校正控制之二.....	261
§ 8-5 复杂分馏塔的前馈自校正控制.....	267
§ 8-6 模型参考自适应系统及其在蒸馏塔上的应用.....	281

第九章 蒸馏过程动态数字仿真

§ 9-1 动态建模及仿真.....	295
§ 9-2 多元蒸馏动态仿真.....	299
§ 9-3 抽提蒸馏塔的建模、仿真及控制.....	305

附录

附录A 按源码顺序排列的Z-80指令表	
附录B 按操作码顺序排列的Z-80指令表	
附录C Z-80与8080指令对照表	
附录D 英制单位与法定计量单位换算表	
附录E 组分浓度换算公式	

第一章 微型计算机硬件

通常，数字计算机由五个基本部分组成，如图1-1所示。它包括运算器、控制器、存储器、输入和输出设备。

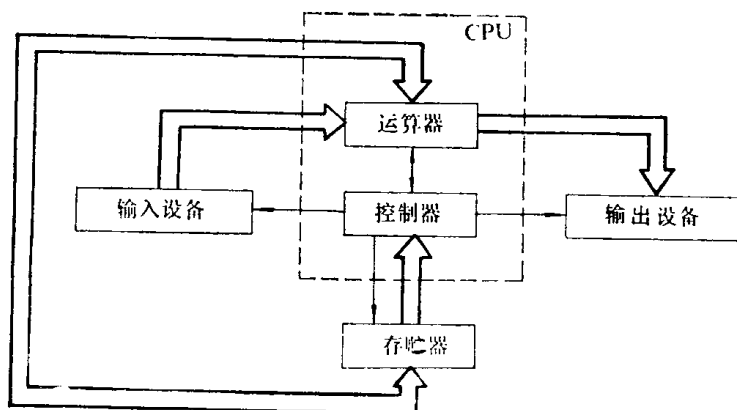


图1-1 数字计算机结构示意图

其中运算器与控制器往往统称为中央处理单元或中央处理机（CPU）。随着大规模集成电路工艺技术的发展，原来占满中心大柜的中央处理机竟然可以微缩在一个或几个大规模集成电路芯片上。这种具有CPU功能的大规模集成电路芯片，不同厂家有不同的型号。例如，Intel 公司的8080;Zilog 公司的Z-80;Motorola 公司的 M6800 等等，都称之为微处理器（也叫微处理机），通常为了区别于大型中央处理机，而表示为MPU。

微处理器还不是一台微型计算机，它只是微型计算机的一部分。但它与适当容量的存储器、输入输出接口电路以及必要的外部设备结合在一起，就组成了一台微型计算机（MC）。

实际应用的微型计算机往往是一套微机系统，如图1-2所示。

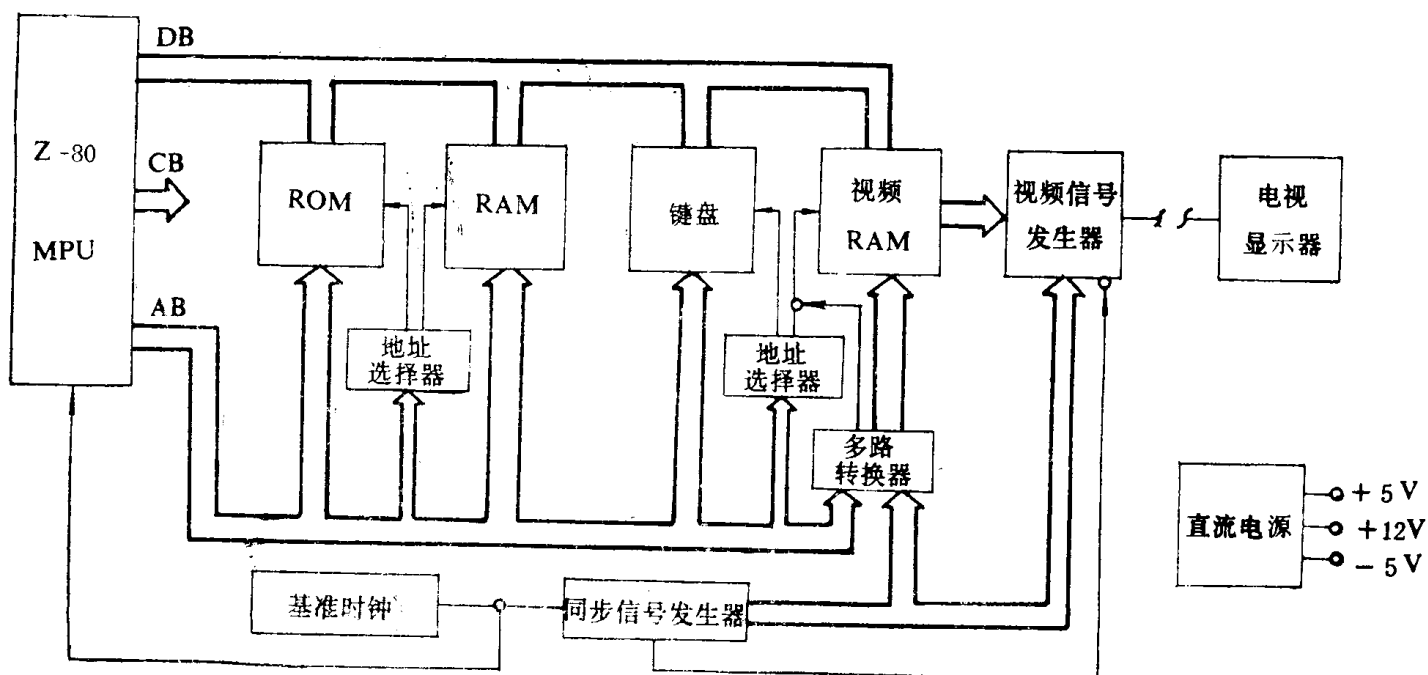


图1-2 微型计算机基本系统结构示意图

§ 1-1 微型计算机的结构特点

微型计算机是70年代发展起来的一种新型计算机。它具有速度快、功能强、成本低、体积小、功耗低、重量轻、可靠性高等优点。而这些优点又是与微型计算机的结构特点分不开的。下面分别介绍微型机的外部和内部结构特点。

§ 1-1-1 微型机的外部结构特点

微机系统中，外部信息的传送是通过总线进行的，如图1-3所示。通常有三种总线：地址总线（AB）、数据总线（DB）和控制总线（CB）。

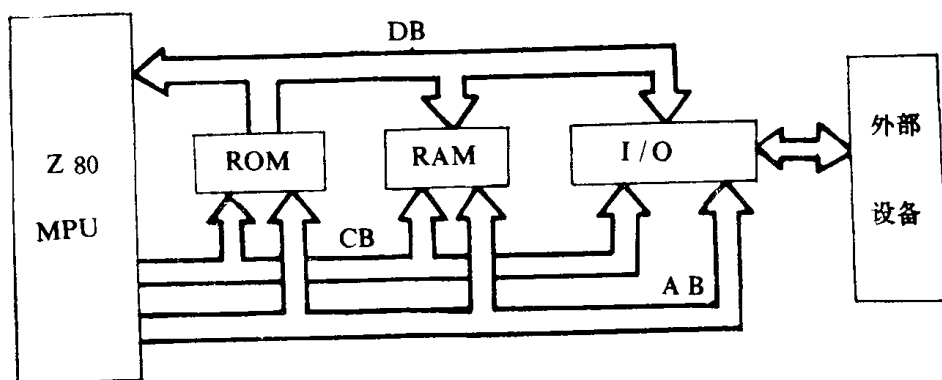


图1-3 微型计算机简化结构图

地址总线 通常为16位，即 $A_{15} \sim A_0$ ，可寻址的内存单元为 $2^{16} = 64K$ 。I/O接口也是通过地址总线的低8位来寻址的，故可对256个外设接口寻址。

数据总线 通常为8位，即 $D_7 \sim D_0$ ，可传送一个字节数。数据在MPU与存储器及MPU与I/O接口之间的传送是双向的，故数据总线为双向总线。

控制总线 它传送各种控制信号，包括MPU到存储器和外设接口的控制信号。例如Z80中的复位信号 $\overline{RESET}$ ，存储器请求 $\overline{MREQ}$ ，I/O接口请求 $\overline{IORQ}$ ，读信号 $\overline{RD}$ ，写信号 $\overline{WR}$ 等等。也有的是由外设到MPU的信号，如Z80中的 $\overline{WAIT}$ 和 $\overline{INT}$ 等。

早期的计算机，输入输出是通过运算器进行的，在输入输出设备与存储器之间不能实现信号的直接联系。现在的微型计算机系统中，由于采用了总线结构，所以可在存储器与外设之间直接进行信息的传送，即实现直接存储器存取（DMA-Direct Memory Access）。

§ 1-1-2 微型机的内部结构特点

微型机的内部结构特点主要由微处理器芯片的结构所决定。尽管微处理器品种繁多，功能各异，但其芯片结构通常有如下特点：

（1）微处理器的结构严格地受到大规模集成电路工艺的约束，因为芯片的面积不能过大，否则生产的成品率下降，成本增加。也就是说，必须在较小的芯片上（如： $5 \times 5mm^2$ ）制作微处理器。

（2）由于微处理器往往集中在单片上，其封装引出端的数量就受到限制。如果引出端增多，测试设备也相应复杂得多，成本增加。所以，目前通用微处理器的引出端都采用40个管脚。这就进一步约束了控制总线的数量。

（3）芯片内部为了减少连接线所占用的面积，采用内部单总线，即内部所有单元电路都挂在内部总线上。总线分时使用，如图1-4所示。

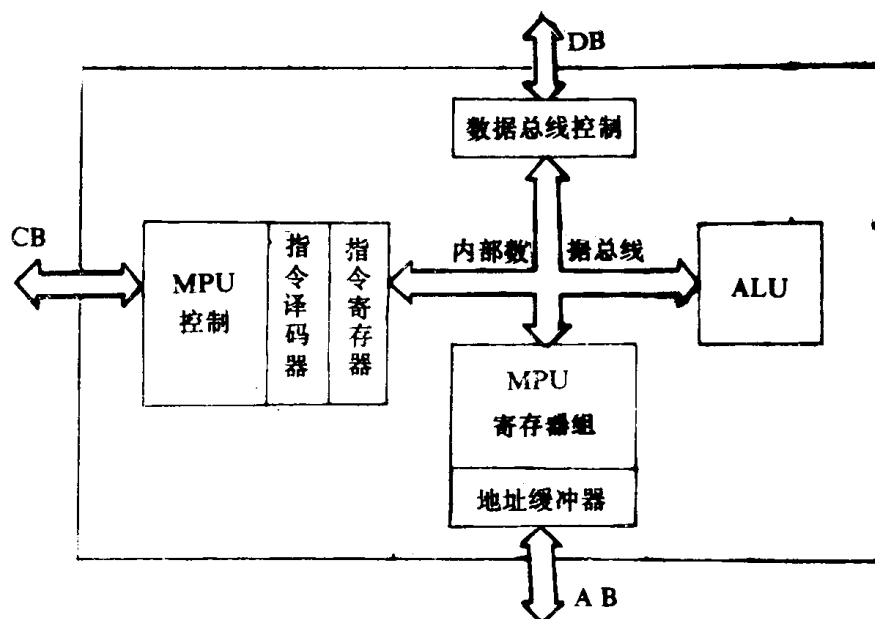


图1-4 Z80-MPU 内部结构示意图

§ 1-2 微处理器 (MPU)

微处理器是微型计算机的核心部件，其作用是从存储器中取出指令，并执行指令所要求的操作，实现运算和控制功能。

目前，微处理器品种繁多。这里仅结合Z80-MPU的内部结构、外部引线功能和时序分析来阐述微处理器的逻辑结构及其工作原理。

§ 1-2-1 Z80-MPU 的内部结构

Z80-MPU 的内部结构框图示于图1-4。它表示了MPU的全部主要部件和数据传送通路的情况。

1. MPU内部寄存器组

Z80-MPU芯片包含了可供使用的十八个8位寄存器和四个16位寄存器。内部寄存器组大体可以分为三类，其中各寄存器的名称和结构示于图1-5。

1) 通用寄存器

Z80-MPU中有两组一样的通用寄存器，每组都有六个8位寄存器可供使用，也可以成对地用作16位寄存器，即形成BC、DE、HL这样三对16位的寄存器对。它们主要用来寄存参与运算的8位数据，或操作数的16位地址。通常只有一组寄存器参与操作，但有时可以用一条简单的交换指令来选用另一组寄存器。

2) 累加器和状态标志寄存器

MPU中包含两个独立的8位累加器（A与A'）和两个相连的8位标志寄存器（F与F'）。进行算术和逻辑运算操作时，累加器A中的内容为一个操作数，且操作的结果放在累加器中。另外，算术和逻辑运算操作结果的一些特征，例如操作结果是否为零，有没有进位等等都要存放在标志寄存器中。程序人员可用简单的交换指令来选择两个累加器中的任何一个，但在复位状态后只能是累加器A。

3) 专用寄存器

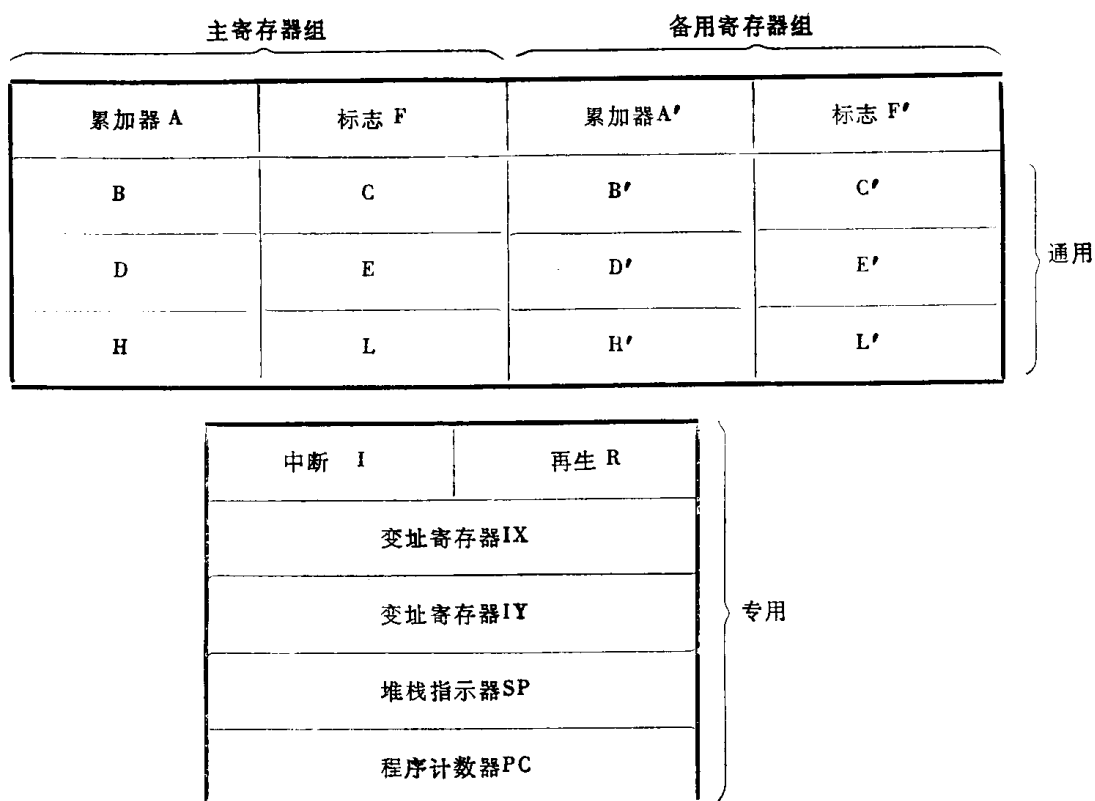


图1-5 Z80-MPU 寄存器组

它包括如下六个寄存器：

程序计数器 (PC) 程序计数器也叫指令计数器，它保存有待于从存储器取出的指令的16位地址码。通常指令是顺序执行的，所以在一般情况下，当取出一条指令（更确切地说是一个指令字节），并送到地址总线后，PC自动加1；只有在执行转移指令或子程序调用（或处理中断）时，才把要转向的地址赋给PC。

堆栈指示器 (SP) 堆栈是在外部（MPU的外部）存储器中，按照后进先出的原则建立的存储区域。堆栈指示器保存着16位地址码，它始终指向堆栈的顶部。利用PUSH和POP指令，可以把内部寄存器对的内容推入堆栈（推入到SP所指示的地址单元），或者把堆栈的内容（SP所指示的地址单元内容）弹出到内部寄存器中去。利用堆栈可实现多级中断、子程序嵌套以及其它操作。

两个变址寄存器 (IX和IY) 两个独立的变址寄存器存放变址寻址方式中所用的16位基地址，由它加上变址指令中给定的偏移量以形成操作数的有效地址。这种变址寻址方式使许多类型的程序大为简化，特别是用在数据表格的场合。

中断页面地址寄存器 (I) 中断页面地址寄存器也叫中断向量寄存器。为了扩大中断功能，Z80-MPU在响应中断时可以采用中断方式2，转向中断服务子程序并调用任一存储单元。它允许有128个中断服务子程序的入口，这些入口地址在存储器中形成了一个表格，占用256个内存单元，称为一页。若内存64K字，就分为256页。因此，这个表格存放在内存的哪一页，就由I寄存器中的内容确定。I寄存器中的内容是中断服务子程序入口地址表的页面地址，即用于存放间接地址的高8位，而中断设备则提供间接地址的低8位。

存储器刷新寄存器 (R) 存储器刷新寄存器也叫存储器再生寄存器。为了增加存储器的集成度，广泛采用了半导体MOS动态存储器，它利用寄生电容来存储信息。由于信息易被泄漏掉，所以要定期（一般为2ms）对动态存储器进行刷新。Z80-MPU中设置了存储器刷新寄

寄存器，每次刷新的内存单元地址由R寄存器提供，而在每刷新一行之后，R的内容自动加1，进行刷新计数。

2. 算术和逻辑运算单元 (ALU)

Z80-MPU中的8位算术和逻辑运算指令在ALU单元中执行。ALU通过内部数据总线与内部寄存器和外部数据总线交换信息。ALU所能完成的功能有：加法、减法运算；逻辑与、逻辑或、异或逻辑；比较数的大小；左或右的移位或循环；加1、减1；位操作等等。

3. 指令寄存器和MPU控制

这部分相当于控制器功能。从存储器中取出指令，要送到指令寄存器IR中，然后由指令译码器译码，通过定时控制电路，在规定的时刻发出信息。这些信息包括寄存器传送、存储器读写、执行各种算术和逻辑运算等所需要的全部控制信息，以及所需要的MPU外部的控制信息。

§ 1-2-2 Z80-MPU引线及其功能

要使用微处理器，先熟悉Z80-MPU引线规定及其功能是必要的。MPU芯片封装在一个有40条引脚的双列直插式组件中，该组件的引脚连线如图1-6所示。

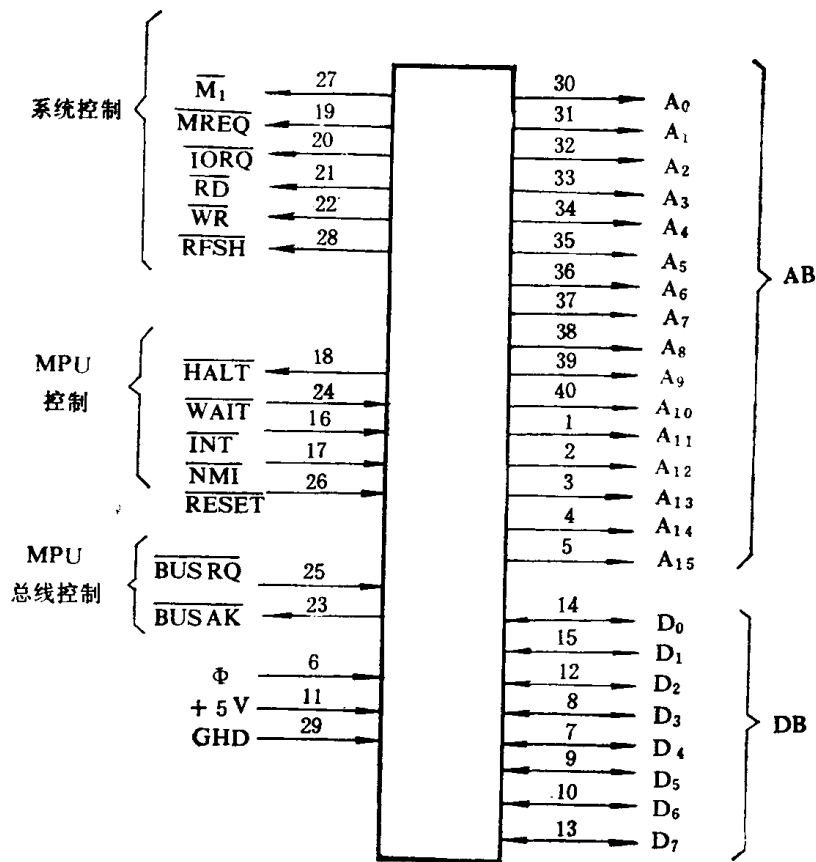


图1-6 Z80-MPU引脚图

引脚中有十六条为地址总线 $A_0 \sim A_{15}$ ；有八条为双向数据总线 $D_0 \sim D_7$ ；有两条作为电源+5V和地线；有一条时钟信号线；有十三条控制信号线。这些引线的功能如下：

$A_0 \sim A_{15}$ 地址总线 形成16位地址信息。三态输出，高电平有效。它为存储器数据交换和I/O设备数据交换提供地址。对存储器而言，它可用来对64K字节的存储单元进行寻址。对I/O来说，寻址应用地址总线中的低8位，允许用户选择多至256个I/O设备端口。在刷新期间，低7位由R寄存器提供，R中存放一行有效的刷新地址。

$D_0 \sim D_7$ 组成8位双向数据总线。三态输入或输出，高电平有效。它用于和存储器、I/O设

备进行数据传送和交换。

图中所有控制信号线中的信号名称上面均有符号“—”，它表示这些信号逻辑电平为低时才有效，同时箭头方向指明信号传送方向。

$\overline{M}_1$ (机器周期I) 指明现行机器周期是指令执行中的取操作码周期 (或者叫做取指周期)。当操作码为两字节时, 每取一个操作码字节都产生一个 $\overline{M}_1$ 信号。此外, $\overline{M}_1$ 和 $\overline{IORQ}$ 信号相结合, 指明该周期为中断响应周期。

$\overline{MREQ}$ 为存贮器请求信号, 三态输出。当存贮器进行读操作或写操作时, 该信号指示地址总线上所提供的地址是有效的。

$\overline{IORQ}$ 为I/O请求信号, 三态输出。该信号指示地址总线低8位中提供一个读或写的有效的I/O端口地址。当中断响应时, 也产生一个 $\overline{IORQ}$ 信号, 它与 $\overline{M}_1$ 信号相结合, 表示中断响应, 通知外设应把中断矢量放到数据总线上。

$\overline{RD}$ 为读信号, 三态输出。该信号指示微处理器从存贮器或I/O设备读取数据。被寻址I/O设备或存贮器, 利用这个信号打开数据传送逻辑门, 使数据进入数据总线。

$\overline{WR}$ 为写信号, 三态输出。它控制将数据从MPU写入存贮器或I/O设备, 而MPU的数据总线上应保持有写入存贮器或I/O设备的有效数据。

$\overline{RFSH}$ 为刷新信号, 用于动态RAM的再生控制, 它指示地址总线的低7位保持有动态RAM的刷新地址。这个信号与 $\overline{MREQ}$ 信号一起用于刷新动态RAM。

$\overline{HALT}$ 为暂停信号, 它指明MPU执行了一条HALT软件指令, 并且正在等待可屏蔽或不可屏蔽中断。暂停期间, MPU执行空操作指令 (NOP), 以维持存贮器刷新操作。

$\overline{WAIT}$ 为等待信号, 它向MPU指明, 所寻址的存贮器或I/O设备还没有为数据传送做好准备, 因而MPU继续处于等待状态, 直到此信号变为有效为止。利用这个信号, 可实现任意速度的存贮器和I/O设备与MPU相同步。

$\overline{INT}$ 为中断请求信号, 它发自I/O设备而指向MPU。若在MPU内部由软件控制的中断开放触发器 (IFF) 处于允许状态, 并且信号 $\overline{BUSRQ}$ 有效, 则现行指令执行结束时, 该中断请求将得到响应。当MPU接受中断时, 在下一个指令周期开始, 送出一个中断响应信号 (在 $\overline{M}_1$ 周期有 $\overline{IORQ}$ 信号)。MPU能用三种不同方式响应中断。

$\overline{NMI}$ 为非屏蔽中断请求信号, 它比 $\overline{INT}$ 有较高的优先权, 且不论IFF状态如何, 在现行指令执行结束时它总得到响应。 $\overline{NMI}$ 信号迫使Z80-MPU转向0066H地址单元。此时, PC内容自动地保存在堆栈中, 保证用户能返回到中断前的程序。

$\overline{RESET}$ 为复位信号, 它迫使程序计数器清零, 并且使MPU置为初始状态。MPU的初始状态包括如下内容: 清除中断开放触发器; 置I寄存器为00H; 置R寄存器为00H; 采取中断方式0。复位期间, 地址总线和数据总线都转为高阻状态, 所有控制输出信号处于无效状态, 不发生刷新。

$\overline{BUSRQ}$ 为总线请求信号, 它请求MPU把地址总线、数据总线 and 三态输出控制信号转为高阻状态, 以便其它设备能利用这些总线。

$\overline{BUSAk}$ 为总线响应信号, 它向提出申请的设备指明MPU地址总线、数据总线 and 三态控制线已经处于高阻状态, 外部设备现在可以使用这些总线。利用 $\overline{BUSRQ}$ 和 $\overline{BUSAk}$ 这两个信号, 可以实现存贮器与外设之间信息的直接传送 (DMA)。

Φ 为MPU所需的单相时钟输入信号。对于Z80, 时钟频率为2.5MHz; 对于Z80A, 时钟频率为4.0MHz。

§ 1-2-3 Z80-MPU典型时序分析

微型计算机在统一的时钟脉冲信号 Φ 作用下，各部件进行有节奏而协调的工作。也可以说，微型计算机工作过程就是严格执行指令的动作过程。这些动作过程包括一些基本操作。例如，存储器读或写；I/O设备读或写；中断响应等。每个基本操作可在3~6个时钟周期内完成，也可以延长，以便使MPU和外设动作速度达到同步。基本时钟周期称为T周期（或叫T状态），而基本操作称为M周期（或叫机器周期）。图1-7给出了一个典型的Z80-MPU时序的例子。它表示出一条指令周期是怎样由一系列特定的M和T周期组成的，图示指令包含三个机器周期。

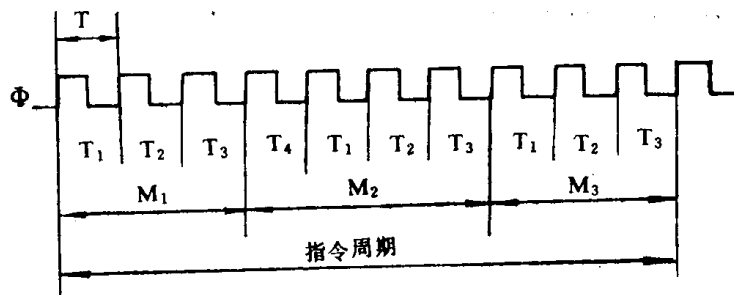


图1-7 Z80-MPU基本时序举例

任何指令的第一个机器周期都是取指周期（其实是取指令的操作码周期），它包含4~6个T周期，如果不够还可以通过等待信号来延长若干个T周期。后续的机器周期则用于在MPU和存储器或I/O设备之间传送数据。

每条指令都有一定的时序，但它们可以分解为一些典型的简单时序。下边仅以取指周期时序为例来说明。

任何一条指令的第一个机器周期必然是取指周期。为了与别的周期相区别，用 M_1 信号表示，故取指周期也叫做 M_1 周期，其周期时序如图1-8所示。

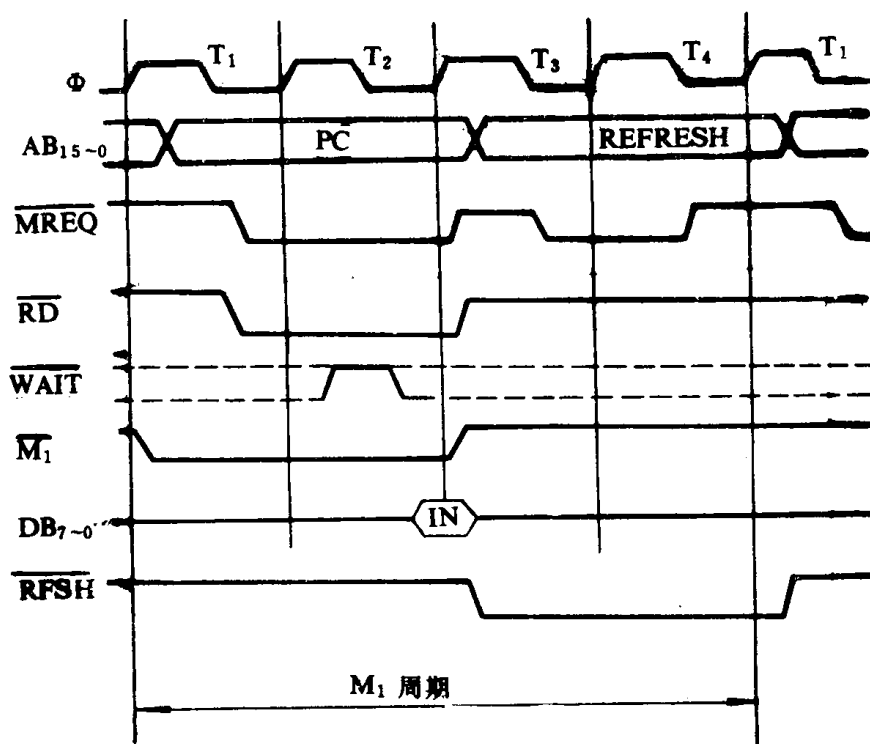


图1-8 取指周期时序图

在没有 $\overline{\text{WAIT}}$ 信号请求等待的情况下，取指周期由4个T状态组成。

取指周期开始时，在 T_1 上升沿后一段时间，程序计数器PC的内容被放到地址总线上。 T_1 的后半周期， $\overline{\text{MREQ}}$ 信号变为有效，此时存贮器的地址在地址总线上已经稳定，所以此信号可直接作为存贮器的选择信号。 $\overline{\text{RD}}$ 信号与 $\overline{\text{MREQ}}$ 信号几乎同时有效，它命令存贮器执行读操作，把存贮器的读出信号与数据总线接通。MPU是在 T_3 的上升沿接受数据总线指令的。同时，它使 $\overline{\text{M}_1}$ 、 $\overline{\text{MREQ}}$ 和 $\overline{\text{RD}}$ 信号失效。这样就保证了MPU是在 $\overline{\text{MREQ}}$ 和 $\overline{\text{RD}}$ 有效时获取指令。

在取指周期的 T_3 和 T_4 状态，在MPU内部对已经获取的指令进行译码和执行一些MPU内部操作；同时存贮器进行动态RAM的再生。地址总线的低7位为存贮器刷新地址。同时刷新信号 $\overline{\text{RFSH}}$ 变为有效。在刷新期间 $\overline{\text{MREQ}}$ 信号仍有效，但由于此时 $\overline{\text{RD}}$ 无效，所以信号不会被读到数据总线上。

若存贮器的工作速度较低，则利用 $\overline{\text{WAIT}}$ 信号可在 T_2 状态后插入任意个 T_w 周期。MPU在 T_2 以及每一个 T_w 周期的脉冲下降沿，对 $\overline{\text{WAIT}}$ 线进行采样。若其为低电平则插入 T_w 周期，否则停止插入。具有 T_w 周期的取指周期时序如图1-9所示。

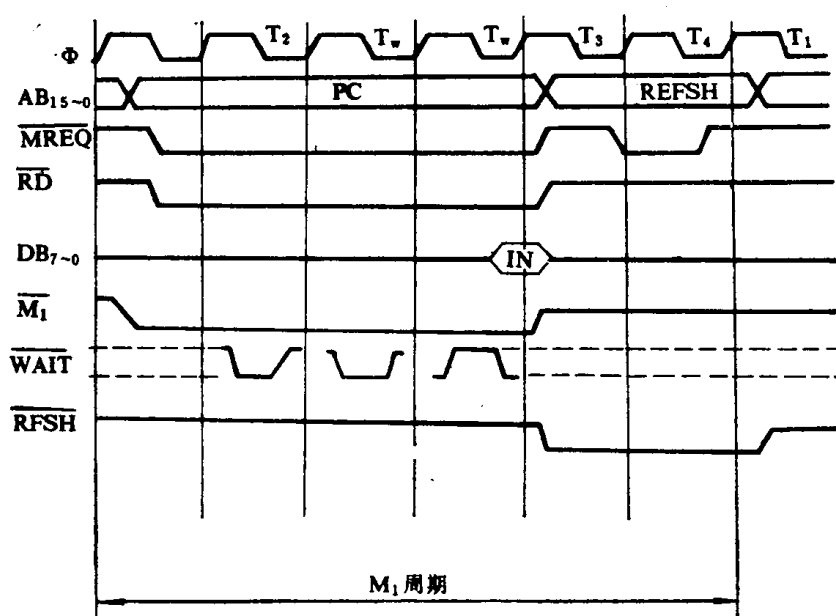


图1-9 具有 T_w 周期的取指周期时序图

§ 1-3 地址选择器

如图1-2所示，在微型计算机内部，要与微处理器MPU进行信息交换的部件除了随机存贮器RAM之外，还有只读存贮器ROM、键盘以及视频RAM等。这些部件同微处理器取得联系，首先应该由MPU地址总线给出地址信号，然后发出相应的控制信号，最后才能在数据总线上进行信息交换。

当由MPU送出地址信号时，选择哪一个地址要根据地址选择器的输出来确定。下面以一个实用的地址选择器的部分逻辑线路图（图1-10）为例来加以说明。

§ 1-3-1 ROM地址选择

图中译码器Z21可以看做是3个输入（ A_{12} 、 A_{13} 、 A_{14} ）、8个输出的译码器。它的启动端输入由 A_{15} 和信号 RAS^* （Z80MPU的 $\overline{\text{MREQ}}$ 输出信号）相“与”的结果来确定。因此，存贮器请

求信号 ($\overline{MREQ}$) 和 A_{15} 都为 “0” 时 Z21 译码器就开始动作。16 脚插座 X3, 有 6 根线已被切断, 其中 Z21 的引脚 9 连接到插座 X3 的接点 10 和 7 上。这样 $ROMA^*$ 的译码逻辑为

$$\begin{aligned} ROMA^* &= \overline{RAS^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \\ &= \overline{RAS^*} + \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}} + \overline{A_{12}} \end{aligned}$$

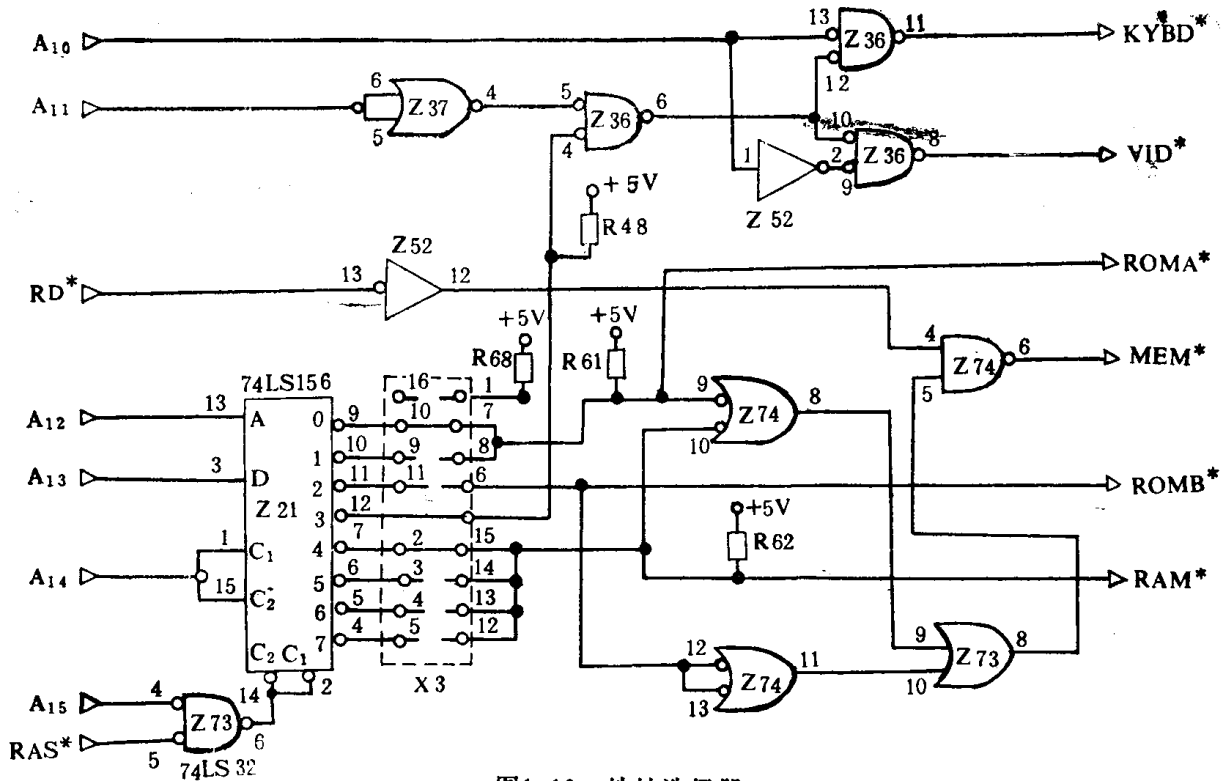


图1-10 地址选择器

由于 $ROMA^*$ 低电平为有效信号, 所以它选定的地址范围为 $0000H \sim 0FFFH$ 。也就是说, $ROMA^*$ 只能被指定到 4K 字节的 ROM 里, 这正是 BASIC I ROM 被选择的地址。

该 $ROMA^*$ 信号经过 Z74 的引脚 9, 从引脚 8 输出, 然后作为或门 Z73 的输入, 从 Z73 的引脚 8 输出, 该输出又通过或门 Z74 与读入信号 RD^* 相 “与”。当信号 RD^* 为 “0” 时, 通过 Z52 使 Z74 的引脚 4 输入信号变成 “1” 状态, Z74 的引脚 6 则呈现 “0” 状态, 于是产生信号 MEM^* 。该信号用于打开 ROM 或 RAM 的缓冲器三态输出门, 因而允许从 ROM 取出数据送到数据总线, 然后由 Z80-MPU 将数据读入。

§1-3-2 键盘地址选择

如上所述, 译码器 Z21 的启动条件为 Z73 的引脚 6 处于低电平。当由 MPU 地址总线给出的地址信号 A_{14} 为 “0”、 A_{13} 为 “1”、 A_{12} 为 “1” 时, 译码器 Z21 的引脚 12 输出低电平。如果地址信号 A_{11} 为 “1”, A_{10} 为 “0”, 经过 Z36 选择输出信号 $KYBD^*$ 。其译码逻辑关系式为

$$\begin{aligned} KYBD^* &= \overline{RAS^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \\ &= \overline{RAS^*} + \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}} + \overline{A_{12}} + \overline{A_{11}} + \overline{A_{10}} \end{aligned}$$

$KYBD^*$ 低电平为有效信号, 所以键盘的地址选择范围为 $3800H \sim 38FFH$ 。

$KYBD^*$ 信号正是键盘输出缓冲器的门控信号。当它处于低电平时, 就使这些输出缓冲器的三态门打开, 这时如果按下键盘某一功能键的话, 就把 $A_0 \sim A_7$ 的地址线及 $D_0 \sim D_7$ 的数据线接通。键盘输出情况如同 ROM 一样, 都是经过 MPU 读入缓冲器把数据传送到 MPU 的指令寄存器内。

§ 1-3-3 视频RAM地址选择

视频显示用RAM的地址选择情况，类似于键盘地址选择，只是信号A₁₀经过Z52的反相门之后，信号变反。因此，其译码逻辑为

$$\begin{aligned}\text{VID}^* &= \overline{\text{RAS}^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \\ &= \text{RAS}^* + A_{15} + A_{14} + \overline{A_{13}} + \overline{A_{12}} + \overline{A_{11}} + \overline{A_{10}}\end{aligned}$$

同理，VID*信号为低电平有效，其地址选择范围为3COOH~3FFFH。

§ 1-3-4 RAM地址选择

图1-10所示插座X3的接点2与接点15之间的连接情况，就表示选定4K RAM的连线方式。因此，其译码逻辑关系式为

$$\begin{aligned}4\text{K RAM}^* &= \overline{\text{RAS}^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \\ &= \text{RAS}^* + A_{15} + \overline{A_{14}} + A_{13} + A_{12}\end{aligned}$$

4K RAM地址选择范围为4000H~4FFFH。若将插座X3的接点3与14连接起来，则形成译码逻辑关系式为

$$\begin{aligned}8\text{K RAM}^* &= \overline{\text{RAS}^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \\ &= \text{RAS}^* + A_{15} + \overline{A_{14}} + A_{13}\end{aligned}$$

其选址范围为4000H~5FFFH。

同理，若将插座X3的接点5与12连接起来，则形成的译码逻辑关系式为

$$\begin{aligned}16\text{K RAM}^* &= \overline{\text{RAS}^*} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \\ &= \text{RAS}^* + A_{15} + \overline{A_{14}}\end{aligned}$$

其选址范围为4000H~7FFFH。

综上所述，我们分析了随机存储器RAM，只读存储器ROM，键盘以及视频RAM等部件的选址情况。据此，可以认识微处理器是怎样通过地址总线同这些部件取得联系的。

现在可以把上述全部选址范围列成表格，因而得到如表1-1所列出的BASIC I存储器分配地址。这是TRS-80微型计算机所用的一种存储器分配表。它表明哪些地址区已经被占用，还有哪些地址区可供用户使用，以及还有哪些空闲区可以进一步开发。

RAM和ROM都是可扩充的。关于ROM的扩展方法，以后还要介绍。实际上，经过扩充以后的微型计算机功能有明显增强。

由TRS-80机的BASIC II存储表（表1-2）可以清楚地看出，与BASIC I相比，BASIC II在存储地址分配上主要有以下两点区别：

(1) ROM扩展使驻存BASIC程序内容增加、功能增强。同时，也使ROM相邻的空闲区大为减少。

(2) RAM从16K增加到48K；供BASIC使用区增加。另外，供用户使用区也相应增加。这样一来，用户就可处理更长的程序。

表1-1 BASIC I存储器分配表

地址（十进制）	地址（十六进制）	内 容
0	0000	BASIC I ROM
4095	0FFF	

续表

4096 14335	1000 37FF	空 闲 区
14336 14591	3800 38FF	键 盘
14592 15359	3900 3BFF	空 闲 区
15360 16383	3C00 3FFF	视 频 RAM
16384 16894	4000 41FF	供BASIC I使用
16895 20479	4200 4FFF	供用户使用 4K RAM
20480 24575	5000 5FFF	8K RAM
24576 32767	6000 7FFF	16K RAM
32768 65535	8000 FFFF	空 闲 区

表1-2 BASIC II 存储器分配表

地址 (十进制)	地址 (十六进制)	内 容
0	0000	BASIC II ROM
12288	3000	空 闲 区
14302	37DE	通讯状态地址
14303	37DF	通讯数据地址
14304	37E0	中断锁存器地址
14305	37E1	磁盘驱动选择锁存器地址
14308	37E4	盒式磁带机选择锁存器地址
14312	37E8	行式打印机地址
14316	37EC	软磁盘控制器地址
14336	3800	键 盘

续表

15360	3C00	视 频 RAM
16384	4000	BASIC II 固定占用的 RAM 中断矢量 (RST ₁ ~7)
16405	4015	键盘装置的控制单元
16413	4010	显示器的控制单元
16421	4025	行式打印机的控制单元
16429	4020	空 闲 区
16464	4050	FDC 中断矢量
16466	4052	通讯中断矢量
16468	4054	空 闲 区
16478	405E	25ms时钟中断
16512	4080	BASIC II RAM
16870	41E6	I/O 缓冲器
32767	7FFF	16K RAM终点
49151	BFFF	32K RAM终点
65535	FFFF	48K RAM终点

§ 1-4 系统RAM及ROM

系统RAM及ROM存贮器是计算机的主要记忆部件。微型计算机有了它和MPU的相互配合,才能把要计算和处理的数据以及程序存入计算机,使计算机脱离人的直接干预,自动地进行工作。

早期计算机主要用磁芯做内存贮器。直到70年代以后,随着大规模集成电路工艺技术的发展,半导体存贮器的集成度提高,体积显著减小,运算速度加快,成本降低。因此,微型计算机的系统RAM及ROM几乎都采用半导体存贮器。

微型机中的系统RAM,通常采用MOS型动态RAM,它由MOS器件构成;而ROM通常采用EPROM,它能改写多次,便于用户自己根据需要编程。

§ 1-4-1 系统RAM

系统RAM能随时进行数据改写。用它来暂时地存贮指令和数据,在执行程序时存放中间