微型计算机的接口技术和应用

上册

戴明桢 编

南京航空学院

1985.12.

微型计算机的接口技术和应用

目前，微型计算机的应用已经深入到国民经济的各个领域，就其应用的范围来说，大致可以分成以下四个方面。

(1) 科学计算；

(2) 企业事业单位的管理；

(3) 工业控制；

(4) 智能化产品。

在前两个方面的应用中，一般采用微型计算机系统。所需要的接口主要用于人机交换信息，例如键盘接口、显示器接口、打印机接口以及软磁盘接口等，通常采用高级语言编制软件。而在后两个方面的应用中，除了人机之间交换信息以外，还需要在被控制对象和计算机之间交换信息。所需要的接口常常要按特定的被控对象或产品专门设计，较多地采用汇编语言或高级语言与汇编语言交叉应用。

接口技术是微型计算机推广应用中的一项很为重要的工作。为了使得已经掌握了微型计算机基本原理的同志，尽快地掌握接口技术，把微机应用于生产实践、科学研究和技术改造等方面去，特编写了这本《微型计算机的接口技术和应用》教材。应当说明的是，教材中的一部分内容是根据编者近几年来的教学、科研的心得体会编写的；而有些内容则引自国内外的文献资料和兄弟院校的教材或刊物。

本教材共分九章。第一章是概论。简要地回顾一下微型计算机系统的基本组成，着重说明微型机的输入／输出工作方式。第二章介绍CPU与各类存贮器的接口技术。第三章收集了几种典型的接口芯片

（包括通用的和专用的接口芯片）资料，以便对接口芯片有个基本的了解。第四章和第五章是本教材的重点。第四章中介绍常用的各种输入／输出设备及其接口方法。第五章专门介绍模拟通道接口技术，即A／D 和 D／A 接口电路。在第六章中，简要地说明一下，总线接口技术。第七、八、九三章是有关微型机应用方面的章节。考虑到单片机的发展和日益广泛的应用，在第七章中以一定的篇幅介绍单片微型计算机。第八章是微型计算机的应用系统设计导论，也是本教材的一个重点。主要讲两个问题：一是TP801监控程序的分析；二是有关专用单板机的设计问题。最后，在第九章中适当介绍几个微型计算机应用的实例。

由于编者水平有限，加上编写时间仓促，教材中定有许多错误和不妥之处，请读者批评指正。

编者　　1985 年 12 月

《微型计算机的接口技术和应用》

上 册 目 录

第一章　微型计算机系统的组成

§1.1 微型计算机系统的组成

(一)完整的微型计算机系统

微型计算机系统，由微型计算机的硬件和软件两大部分组成。所谓硬件，是指组成计算机的物质设备，例如运算器、控制器、存贮器、输入／输出设备等；而软件，指的是使用计算机和发挥计算机效率功能的各种各样的程序。一个完整的微型计算机系统的组成如图1.1－1所示。

(二)微型计算机的硬件

微型计算机的硬件，由微处理器（CPU）、存贮器以及输入／输出（I／O）接口电路等部分组成。CPU通过I／O接口电路与外部设备（I／O设备）相连接。微型计算机内部通过三组总线——地址总线（A.BUS）、数据总线（D.BUS）和控制总线（C.BUS）来连接。微型计算机的硬件组成框图如图1.1－2所示。由于大家对硬件比较熟悉，这里就不多作介绍了。

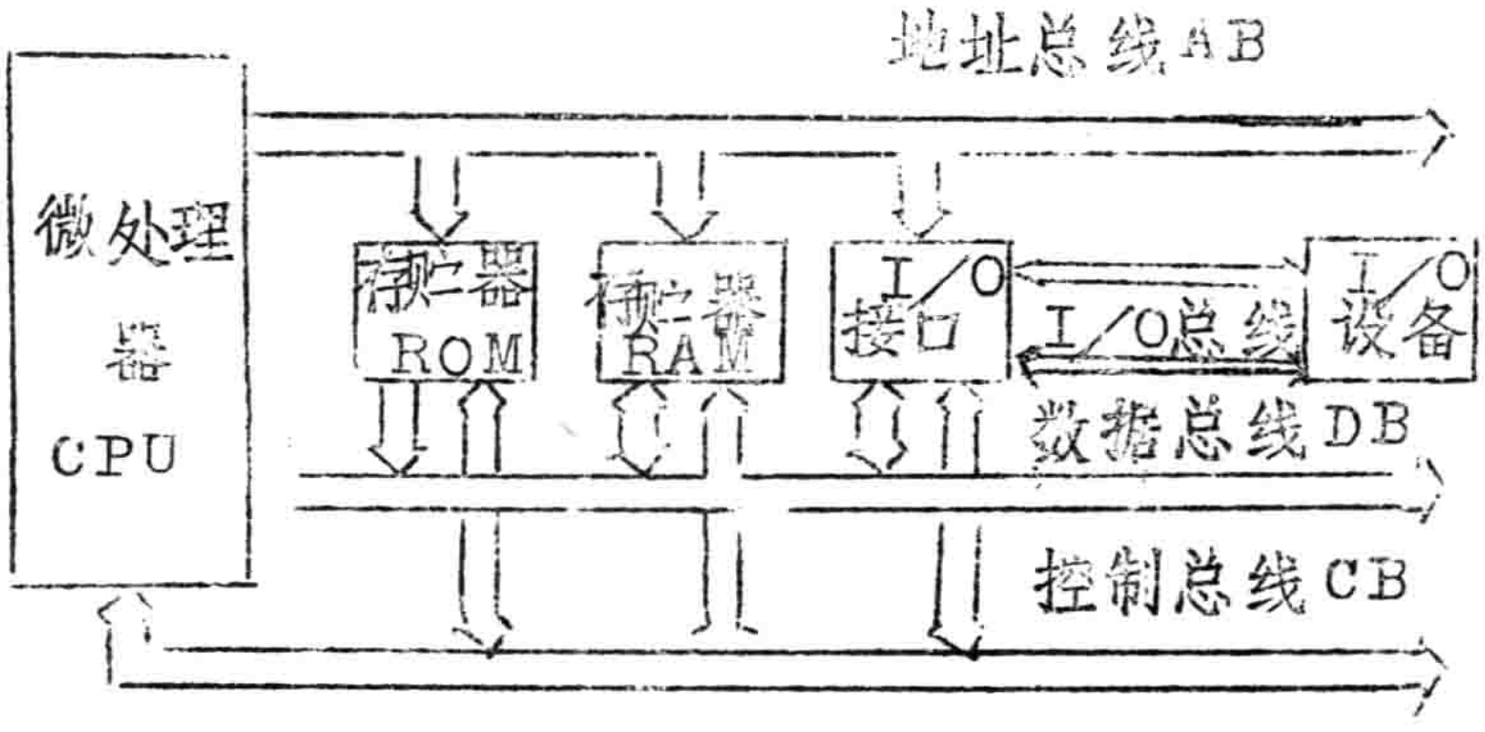


图1.1－2 微型计算机的硬件组成框图

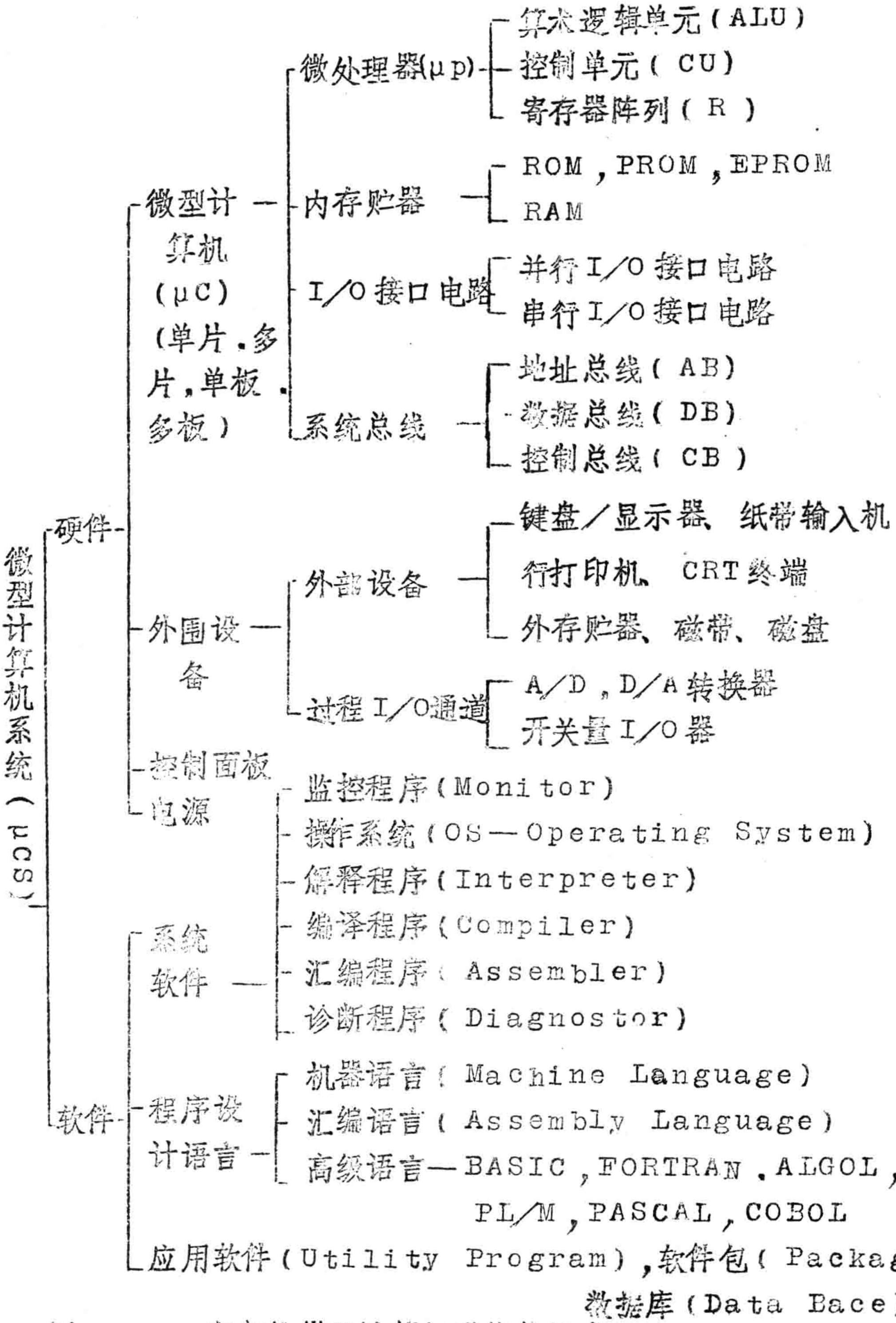


图1.1—1 完整的微型计算机系统的组成

(三) 微处理器的结构

由于本教材所介绍的接口电路以及应用实例以Z80—CPU为主，故先在这里简要地叙述一下Z80—CPU的内部结构及其引脚的功能，以便于阅读后面的内容。附录(一)给出了常用的8位和16位微处理器性能一览表供读者查阅。

Z80CPU是一种八位的40条引脚单总线结构的微处理器。它的内部结构如图1.1—3所示。

(A) Z80的内部结构

(1) CPU内部寄存器组

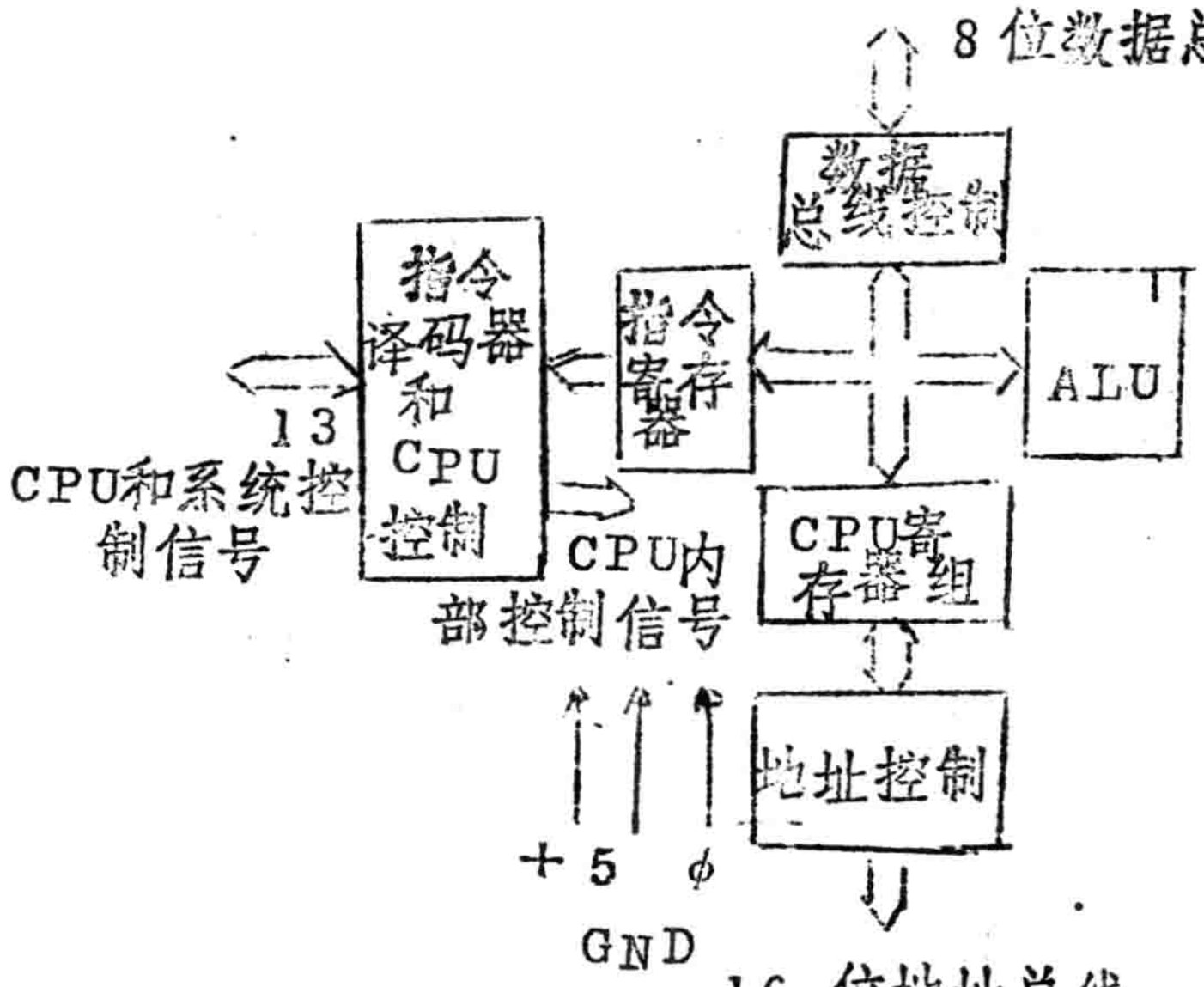
Z—80的内部寄存器组也分成两种类型，一种是专用寄存器，另一种是通用寄存器。现分别把它们的功用说明如下。

1.程序计数器PC

是16位计数器，CPU总是把PC中的内容作为地址，从内存中取出一条指令，加以译码和执行。所以，PC中总是包含着下一条要执行的指令的16位地址。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（更确切地说为一个指令字节）后，PC就加1，只有在执行转移指令或子程序调用时（或中断时），把要转向的地址赋给PC。

2.堆栈指针SP (Stack Pointer)

堆栈是在外部（CPU的外部）存贮器中的一个按照后进先出（Last — in First — out）的原则组织的存贮区域。堆栈指针包含着16位地址，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所指的单元）或把堆栈的内容（SP所指的单元）弹出到内部寄存器中去。利用堆栈，可实现多级中断，子程序嵌套以及其他操作。



主寄存器组		备用寄存器组		
累加器 A	标志 F	累加器 A'	标志 F'	通用寄存器
B	C	B'	C'	
D	E	D'	E'	
H	L	H'	L'	

中断向量 I	存贮器刷新 R
变址寄存器 IX	
变址寄存器 IY	
堆栈指针 SP	
程序计数器 PC	

专用寄存器

图 1.1 — 3　Z80 — CPU 的内部结构

3．两个变址寄存器(Index Register)IX 和 IY

这是两个独立的16 位的寄存器，通常它们各自包含着一个16位的基地址，由它加上指令中给定的偏移量以形成操作数的有效地址。这种寻址方式使许多类型的程序大大简化，特别在用到数据表格的场合。

4．中断页地址寄存器I

这是一个8 位的寄存器。当CPU用中断方式与外设交换信息时，若外设有中断请求，而CPU 也允许和响应了中断，则程序就要转向中断服务程序。在Z80的方式0中断时，当CPU响应中断时，外设提供RST00H — RST38H指令中的任一条，程序就固定地转向0000H — 0038H八个入口地址的任一个。为了扩大中断功能，Z—80有中断方式2。它允许有128个中断服务程序的入口，这些入口地址形成了一个表格，这个表格存放在内存的哪一页（若以256个字节为一页，则64K内存就分为256页），就由I寄存器中的内容确定（由用户预先给定），所以，I寄存器中的内容，是中断服务程序入口地址表的页地址。

5．存贮器刷新寄存器R

为了增加半导体存贮器的集成度，在大容量的存贮器中广泛采用了MOS动态存贮器，它利用寄生电容来存贮信息，为了防止信息被泄漏走，所以要定期（一般为2ms）对动态存贮器进行刷新（Refresh) Z80是利用取指周期的后两个T状态（此时CPU对指令进行译码和执行内部操作，不使用存贮器）来刷新的，每一个取指周期对一部分存贮器进行刷新，以保证在2ms内对整个64K内都刷新一遍。每次刷新的内存单元（一次刷新内存中的一行，而不是一个单元）的地址

由R寄存器（7位）提供，而在每刷新一行后，R的内容自动加1（实质上R为一个计数器），以指向下一行。

6．累加器和状态标志寄存器

Z80中有两个累加器和与它相连的状态标志（Flag）寄存器。在进行算术和逻辑操作时，累加器A中的内容必为一个操作数，且操作的结果放在累加器中。另外，算术和逻辑操作的结果的一些特征，如操作结果是否为0，有没有进位等等都要寄存在标志寄存器中。

程序员可用简单的交换指令来选择两个累加器中的任一个（在复位状态后是累加器A工作）。

7．通用寄存器组

Z80中有两组一样的通用寄存器，每组都有6个8位的寄存器，它们可以分别为6个8位寄存器使用，也可以两个连起来形成BC、DE、HL三对16位的寄存器对。它们主要用于寄存参与运算的8位数据，或操作数的16位地址。

在工作时，只有一组寄存器参与操作，但可以用一个简单的交换指令，来选用另一组寄存器。

(2) 算术和逻辑单元ALU

CPU的8位算术和逻辑指令在ALU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；减；逻辑“与”逻辑“或”；“异或”；比较；左或右的移位或循环；加1；减1；位操作等。

(3) 指令寄存器和CPU控制

这部分相当于控制器的功能。从存贮器中取出来的指令，要输入到指令寄存器IR中，然后由指令译码器译码，通过定时和控制电路，

在规定的时刻发出例如寄存器传送，存贮器读写，执行加或减或各种逻辑运算等所需要的全部内部控制信息；以及发出所需要的外部（CPU的外部）控制信号。

(B) Z80引脚及其功能

Z—80用40条引脚，如图1.1—4所示

其中有16条为地址总线A_{15}—A_0；有8条为双向数据总线D_7—D_0；三条作为电源、地和时钟信号线，另外13条为控制信号线。这些引线的功能如下：

A_0—A_{15}，地址总线，它是三态输出，高电平有效。A_0—A_{15}，组成16位地址信息，用于与内存（最大到64K字节）以及I/O装置交换数据。I/O寻址用地址总线中的低8位，使用户可直接选择256个I/O端口。A_0是最低有效位。在刷新期间，低7位由R寄存器提供，包含一个有效的刷新地址。

D_0—D_7，三态输入/输出，高电平有效。D_0—D_7组成了一个8位的双向数据总线。系统内的所有数据的传送CPU ⇄ 存贮器或

CPU → 外设端口，以及存贮器 → 外设都是通过这个数据总线传送的。

$\overline{M_1}$ 输出，低电平有效。$\overline{M_1}$ 指示现行的机器周期是取操作码周期。当操作码是两字节操作码时，则每一取操作码周期，都发出 $\overline{M_1}$

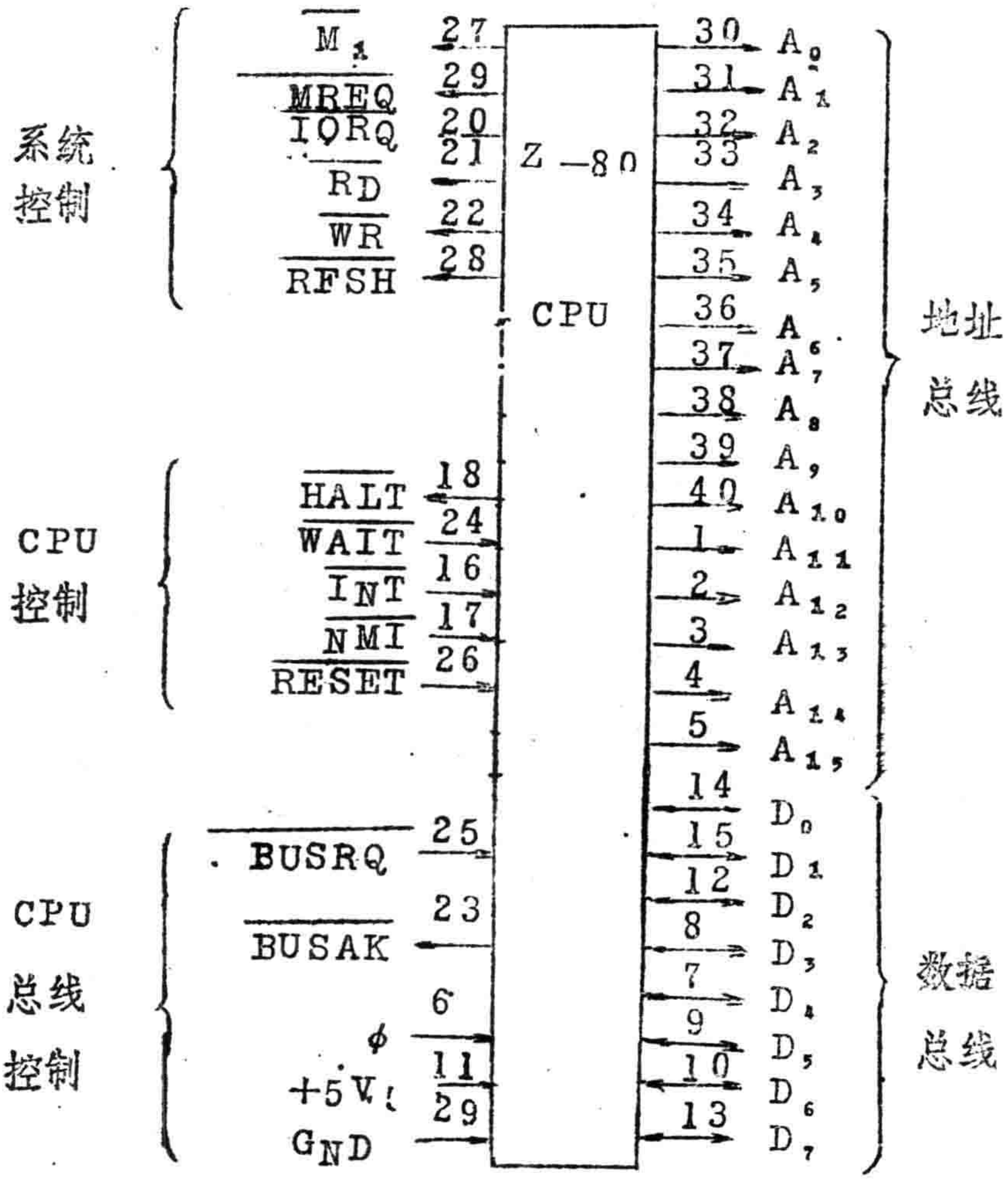


图1．1－4　Z80 CPU引脚图

信号，这样的两个字节的操作码经常用 CBH，DDH，EDH 或 FDH 开始。$\overline{M_1}$ 也与 $\overline{IORQ}$ 信号一起指示为一中断响应周期。

$\overline{MREQ}$ 三态输出，低电平有效。这个存贮器请求（Memory Request）信号指示地址总线上保持有一个供存贮器读或写操作的有效地址。

$\overline{IORQ}$ 三态输出，低电平有效。这个输入／输出请求（Input/Output Request）信号指示地址总线的低八位中保持有一个供 I/O 读或写的有效的 I/O 端口地址。当中断被响应时，也产生一个 $\overline{IORQ}$ 信号，它与 $\overline{M_1}$ 一起表示中断响应，通知外设应把中断矢量放到数据总线上。在 M_1 期间可以发生中断响应操作，而在 M_1 期间绝不发生 I/O 操作。

$\overline{RD}$ 三态输出，低电平有效。这个读信号（Read）指示 CPU 要求从存贮器或一个 I/O 装置读入数据。所寻址的 I/O 装置或存贮器，应用这个信号把它们的数据门打开（即作为三态门的选通信号）使数据进入数据总线。

$\overline{WR}$ 三态输出，低电平有效。这个写（Write）信号指示 CPU 的数据总线上保持有有效的数据要存入（写入）所寻址的存贮器或 I/O 装置。

$\overline{RFSH}$ 输出，低电平有效。这个刷新信号（Refresh）指示地址总线的低七位保持有动态存贮器的刷新地址。这个信号应与 $\overline{MREQ}$ 信号一起用于刷新动态存贮器。A，是逻辑0，地址总线的高八位包含 I 寄存器内容。

$\overline{HALT}$ 输出，低电平有效。$\overline{HALT}$ 指示 CPU 已经执行了一条 $\overline{HALT}$ 软件指令，CPU 就进入了暂停状态。一直要到 CPU 或者接受到非屏蔽中断（NMI — Non Maskable Interrupt）或者接

受到屏蔽中断（ Maskable Interrupt ）操作才能重新开始。在暂停期间CPU执行NOP（无操作）指令，以维持能进行存贮器刷新。

$\overline{\text{WAIT}}$ 输入，低电平有效。$\overline{\text{WAIT}}$ 告诉CPU所寻址的存贮器或I/O装置，尚未准备好数据传送。只要这个信号有效，CPU继续插入等待周期。这个信号能使CPU与任何速度的存贮器或I/O装置同步。

$\overline{\text{INT}}$ 输入，低电平有效。这个中断请求（ Interrupt Request）信号由I/O装置产生。在CPU内部的由软件控制的中断允许触发器IFF为开放中断状态；且在没有总线请求情况下，则在现行指令结束时响应中断。当CPU接受中断，在下一个指令周期的开始，送出一个中断响应信号（$\overline{M_1}$ 周期有 $\overline{\text{IORQ}}$ 信号）。CPU能用三种不同的方式响应中断。

$\overline{\text{NMI}}$ 输入，负边沿触发。这个非屏蔽中断请求信号，具有比 $\overline{\text{INT}}$ 更高的优先权，它不受IFF控制且总是在现行指令的结束时被响应。NMI自动地迫使Z—80CPU转向0066H单元。PC的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。注意：继续等待周期会阻止现行指令的结束，另外，总线请求信号 $\overline{\text{BUSRQ}}$ 优先于 $\overline{\text{NMI}}$ 。

$\overline{\text{RESET}}$ 输入，低电平有效。复位信号 $\overline{\text{RESET}}$ 迫使PC为0且初始化CPU。CPU的初始化包括：

① 清除中断允许触发器；

② 置I寄存器为00H；

③ 置R寄存器为00H ；

④ 置中断方式为0 。

在复位期间，地址总线和数据总线处于高阻状态，且所有控制信号处于无效状态，不发生刷新。

$\overline{BUSRQ}$ 输入，低电平有效。这个总线请求信号（Bus Request）用于要求CPU把地址总线、数据总线和三态的输出控制信号处于高阻状态，使别的装置能控制这些总线。这主要用于存贮器与快速外设之间直接传送信息（DMA）。当$\overline{BUSRQ}$有效，CPU在现行的机器周期结束时立即响应，使这些总线处于高阻状态。

$\overline{BUSAK}$① 输出，低电平有效。这个总线响应（Bus Acknowledge）信号，告诉总线请求装置，CPU的地址总线、数据总线和三态的控制总线已处于高阻状态，外部装置现在可以用这些总线了。

ϕ 单相系统时钟。Z80时钟频率为2.5MHZ，Z80A时钟频率为4.0MHZ 。

① 当Z80 CPU在WAIT状态与总线响应状态，不发生动态RAM的刷新，使用时要注意。