

微 机 在 自 动 控 制 系 统 中 应 用

刘 敏 清 编 著
王 敏 元

中 国 铁 道 出 版 社

1987年·北京

内 容 简 介

本书着重介绍了微机在自动控制系统中的应用。它从实用出发,以 Z 80 微机为主,介绍了微机基本系统;输入/输出接口和设备;微机在机车自动控制系统中的应用及故障检测;还列举了微机系统在步进电机的变速系统、小功率直流随动系统、650 轧钢作业线监测管理以及导爆管质量检测等方面的应用。书中的数据、电路和程序可供实际使用时参考。

本书适合具有中等以上文化程度的工程技术人员、科研工作者自学,在校师生亦可参考。

微机在自动控制系统中应用

刘敏清 编著
王元一

中国铁道出版社出版

责任编辑 陈广存 封面设计 安宏

新华书店北京发行所发行

各地新华书店经售

北京顺义燕华营印刷厂印

开本: 850×1168 毫米 1/32 印张: 13.125 字数: 325 千

1987 年 11 月第 1 版第 1 次印刷

印数: 0001—7,000 册 定价: 4.05 元

前 言

微型计算机已经在生产过程自动控制领域中得到了广泛的应用，并日趋成熟。它对自动控制系统中的常规控制设备进行了革新，形成了微机系统。它通过周密的程序设计，不仅能够完成“故障—安全”原则，而且能够通过过程控制，实现动力设备的自动化、半自动化；燃料、能源以及产品质量的最优化控制。它通过信息管理可逐步实现管理现代化。它还能够实现对复杂过程控制的监控及监测。因此，微机已渗透到生产过程的各个领域，而且有明显的经济效益。例如，宝钢从日本引进的DD80型无线遥控内燃机车，由于无线遥控系统中，采用Z80微机作为过程中间控制机，从而实现了比人工操作更为科学的20多种功能的状态自动控制。不仅确保了设备的安全可靠运行；而且可以节省燃料消耗；同时也改变了传统的机车操作方式，司机除驾驶机车外，还兼做摘钩联挂车辆，操作车上转辙装置转动道岔，选择进路等多项工作，从而提高了工作效率，节省了定员。

本书涉及的内容较广。第一和第二章的目的是为了进一步明确微机系统各部分之间关系，介绍了微机基本系统及输入/输出接口和设备；第三、四、五和第六章着重介绍了微机在机车自动控制系统中的应用，目的是给读者有一个微机系统的整体概念；第七章列举了微机系统在步进电机的变速系统、小功率直流随动系统、650轧钢作业线监测管理以及导爆管质量检测等方面的应用。因此，本书不仅对Z80微机的硬件和软件作了介绍，另外对微机的外围传感器，微机系统的接口、运用和故障检测装置等亦作了介绍。

承蒙邓永林同志参加了全书的编著工作。另外，龚铁成、傅新福同志也参与了部分编写工作。在编著过程中还得到了上海宝

山钢铁总厂有关领导以及关心此书编著的一些同志的支持和幫助，在此，表示诚挚的感谢。

本书初稿写出后，曾在马鞍山钢铁公司职工大学中进行了讲解。并收集了教师和同学们的反映，吸取了他们提出的宝贵意见。

由于编著者水平有限，时间又很仓促，书中一定存在不少缺点和错误，敬请读者批评指正。

编著者 1986.10.

目 录

第一章 微机基本系统及主要输入/输出接口	1
第一节 Z80-CPU 简介	3
第二节 半导体存储器	22
第三节 微机的输入/输出接口概述	38
第四节 Z80-PIO 简介	42
第五节 Z80-CTC 简介	60
第六节 Z80-SIO 简介	73
第七节 可编程序接口8255简介	90
第八节 通用接口8212简介	98
第二章 微机的数-模和模-数转换器及主要输入/输出设备	107
第一节 数-模转换器 (D/A)	107
第二节 模-数转换器 (A/D)	119
第三节 电气传感器	137
第四节 调节器	145
第三章 机车自动驾驶遥控控制装置和接收装置的电路及原理	152
第一节 时钟脉冲和时序脉冲的产生	153
第二节 同步字码的规定和形成	156
第三节 机器位 (又称识别位) 和地址字码、指令字码的规定和形成方法	161
第四节 反转连送的实现	170
第五节 奇 (偶) 校验位功能的实现	171
第六节 FS调制的实现	173
第七节 电压检测功能的实现	176
第八节 接收机电路	176
第九节 解调电路	186
第十节 接收控制和串行/并行转换功能的实现	186

第十一节	校验功能的实现	194
第十二节	错误检查判断电路	208
第十三节	传送顺序控制电路	214
第十四节	接收装置继电器电路	217
第四章	自动运转控制装置(简称ATO)电路及原理	221
第一节	微处理机	223
第二节	微机的输入/输出接口	234
第三节	模/数转换	239
第四节	显示电路	246
第五章	机车自动驾驶微机指令系统汇编程序	248
第一节	微机系统的使用	248
第二节	微机信息的数和编码系统	250
第三节	微机指令系统	256
第四节	微机控制程序流程图	259
第五节	微机的汇编程序	267
第六节	燃料的最优化控制	273
第六章	机车自动驾驶微机系统的维护检修及故障处理	277
第一节	运 用	277
第二节	维护检修及故障处理	277
第三节	微机系统的特殊检测装置	291
第四节	列车自动驾驶遥控机车的地面辅助设施 (车上转换装置)	299
第七章	微机在其它自动控制系统中的应用	301
第一节	微机控制步进电机的变速系统	301
第二节	小功率直流随动系统的双模控制	315
第三节	650轧钢作业线微机监测管理系统	330
第四节	导爆管产品质量的微机检测方法	346
附录一	ASCII码表(美国标准信息交换码)	363
附录二	Z80的指令分类表(按字母顺序排列)	364
附录三	Z80的指令机器码表	396
主要参考文献		406

第一章 微机基本系统及 主要输入/输出接口

微机是在小型电子计算机的基础上,用集成电路技术发展起来的一种具有许多优点的电子计算机。它不但具有一般电子计算机的快速、精确和通用等许多特点,还具有体积小、价格便宜、可靠性高、容易掌握、适应性强以及功耗低等优点。因而微机的发展速度很快,其应用范围亦在不断扩大。

我国在自动控制单体系统中使用的微机,目前以Z80元件系列为最多。因为Z80元件系列可以和许多类型的标准半导体存储器相结合,组成具有不同功能的微机系统。其硬件系统基本上由三部分组成:微处理器(CPU),存储器(ROM, RAM)和外设接口电路(I/O)。本书用Z80来简要介绍微机基本系统及主要输入输出接口。

在微机系统中,外部信息的传送是通过总线进行的,Z80微机与其它微机一样有三组总线。

1. 数据总线(DB)

Z80-CPU数据总线为8位(即字长为8位)并且有三态电路。Z80数据总线是一种双向传送的总线,共有8根($D_7 \sim D_0$),用于和存储器、I/O设备进行数据交换。

2. 地址总线(AB)

Z80-CPU具有16位(即 $A_{15} \sim A_0$)地址总线,共有16根,是单向总线,用来传送地址码。微机总线上各器件之间的联系,主要依靠地址码来勾通。因此,可寻址的内存单元为 $2^{16} = 64K$ ($1K = 1024$)。I/O接口也是通过地址总线的低八位来寻址的,故可寻址256个外设接口。地址总线同样具有三态电路,CPU内部的地址信息必须经过地址锁存器和三态缓冲器后方能送入外部

地址总线。

3. 控制总线 (CB)

Z80-CPU共有13根控制总线。它又可分为三条更小的总线, 分别用于系统控制(6根); CPU控制(5根); 总线控制(2根)。其中只有一部分控制总线(如 $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, $\overline{WR}$)具有三态电路。CPU通过控制总线随时掌握各器件的状态, 并根据需要随时向有关器件发出控制命令。

微机硬件方框图如图1—1所示。

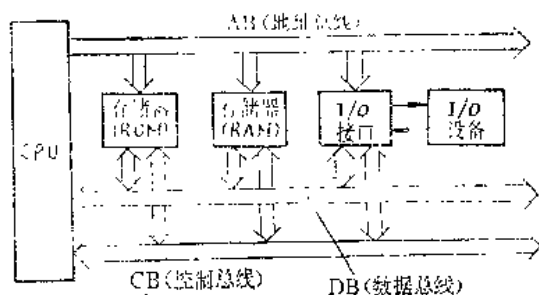


图1—1 微机硬件方框图

Z80微机是一种八位机, 内存每个单元存放一个8位二进制位的信息, 即为一个字节。Z80与基本微机结构相比, 有下列特点

1. 设有两组通用寄存器, 可由指令相互切换工作, 提高了程序转换及数据处理速度。

2. 增加了一些专用寄存器, 扩大了功能。

3. 有较强的中断处理能力。

4. 时钟、电源种类单一。它使用单相时钟 ϕ 和+5V电源, 使用方便。

5. 指令功能比较齐全。Z80有158条指令兼容了8080的指令系统。它还设立了位操作和一些高效指令(如数据块传送、检索功能指令等), 方便了程序设计。

Z80微机基本组成有: Z80-CPU; 半导体存储器(ROM, RAM)以及输入输出接口电路(I/O)。Z80微机常用的接口电

路有：并行的输入输出接口Z80-PIO；计数/定时电路Z80-CTC；串行输入输出接口Z80-SIO；可编程序接口8255；通用接口8212等。

第一节 Z80-CPU 简介

一、Z80-CPU的方框图

Z80-CPU是一种八位的40条引脚的内部单总线结构的微处理器。它的内部结构方框图如图1-2所示。

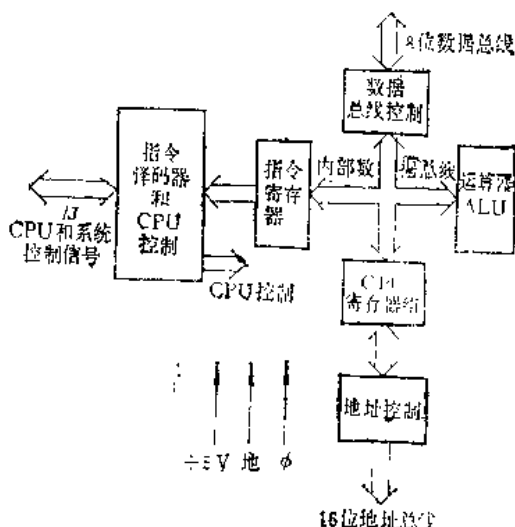


图1-2 Z80-CPU方框图

Z80-CPU在执行一条指令时，三条总线使用过程及顺序如下：

在取指时，地址总线决定待取指令的地址；控制总线决定存储器读的时序；数据总线进行传送指令。

在执行时，地址总线决定I/O接口或存储器，取决于所执行的指令；控制总线决定对执行过程进行同步；数据总线进行传送指令。地址总线、控制总线 and 数据总线尚需根据正在执行的指令要求被重复使用。

二、Z80-CPU寄存器

Z80-CPU 中包含了208位可供使用的读/写存储器。全部Z80-CPU寄存器采用静态RAM实现。图1-3所示Z 08位存储器组成18个8位寄存器以及4个16位寄存器可供使用。它们是累加器A, 状态标志寄存器F, 6个8位通用寄存器B、C、D、E、H、L, 8个8位交换寄存器A'、F'、B'、C'、D'、E'、H'、L', 2个8位专用寄存器(中断页面地址I与动态存储器刷新R), 4个16位专用寄存器PC、SP、IX和IY, 此外, Z80-CPU还有一些仅供内部使用的寄存器, 例如2个8位暂存器W和Z。

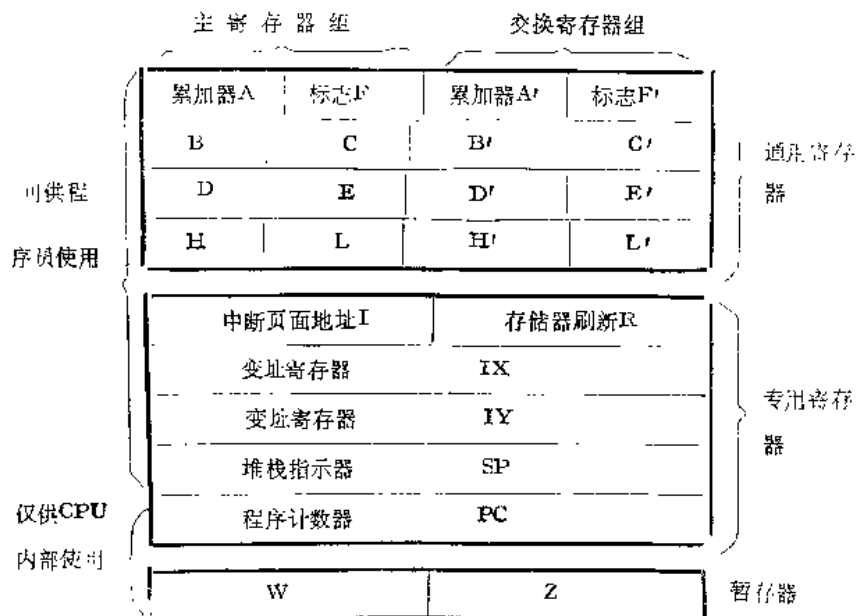


图 1-3 Z80-CPU寄存器结构

下面仅对专用寄存器, 累加器和状态标志寄存器, 通用寄存器, 运算器等简述如下:

(一) 专用寄存器

1. 16位程序计数器PC

PC用来记录程序运行的运态轨迹，它保存着有待从存储器取出的现指令地址，即作为程序运行时，指示要取指令的地址。通常程序中的指令是按顺序执行的，当需要将欲执行的指令码取出时，首先把PC的内容送地址锁存器，再由地址锁存器经三态缓冲器送地址总线，从而选中存储器某一存储单元。然后再从此存储单元中取出欲执行的指令码。与此同时，PC的内容送地址锁存器后，便被加1器自动加1，以便为取出下一条指令做好准备。因此，CPU总是把PC中的内容作为地址，PC中总是包含着一条要执行的指令的16位地址。当程序出现分支时，新的地址码由转移指令置入PC，此时PC不自动增量。当调用子程序时，在转子指令的控制下将原来PC中的内容（主程序的断点）保护进栈之后，再打入子程序的入口地址。当子程序执行到返回指令时，该指令将使保存在堆栈中的主程序断点退回PC。

2. 16位堆栈指示器SP

数据从堆栈那里出入的方式有一个特点，就是“先进后出(FILO)”“后进先出(LIFO)”。数据可通过执行进栈（称为压入）和退栈（称为弹出）指令使CPU“寄存器对”及IX，IY的内容进栈或退栈。用堆栈可以简单地实现多级中断、多个子程序嵌套和简化许多类型的数据操作。

3. 2个16位的变址寄存器IX与IY

2个独立的变址寄存器，主要用来存放过渡性的地址，以便达到自动变换地址的目的。变址寄存器可以自身加1或减1来改变地址，作为地址计数，也可以和外来的地址相加，以形成新的地址。这样就增加了寻址的灵活性。

4. 8位中断页面地址寄存器I

I寄存器是用来存放中断程序地址的高8位的地方，所以称之为页面地址。Z80-CPU在响应屏蔽中断方式2时，能够很好地完成对任何存储器位置的间接访问，为达此目的，I寄存器存放间接地址的高8位，请求中断的外设送来的中断向量作为间接地址的低8位，构成一个16位地址。这个地址指向一个中断起始地

址表，表中给出中断服务程序首址，这个特点允许把中断子程序动态地安置在存储器的任何位置上，且对子程序的访问时间是很短的。

Z80 的中断方式 2 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）。

5. 8 位存储器刷新寄存器 R

Z80 可以使用动态存储器 RAM，它利用寄生电容上的电荷多少来表达存储的信息。由于 MOS 电路的漏电，电容上电荷的泄放，将使信息丢失。因此每隔一段时间（在 $+70^{\circ}\text{C}$ 以下为 2ms，温度上升时则小于 2ms）要进行一次刷新。Z80 中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，不使用存储器）来刷新的。是按次序一行一行进行刷新，一行刷新后 R 自动加 1，以便对下一行动态存储单元进行刷新。实质上 R 是在刷新时进行计数的。刷新由 CPU 自动管理，保证在 2ms 内对整个 64K 内存都刷新一遍。每次刷新的内存单元的地址由 R 寄存器提供。

（二）累加器 A 和状态标志寄存器 F

累加器是基本的源寄存器或目标寄存器，算术或逻辑运算时，都从累加器取得一个操作数，并将结果送回累加器。A' 是 A 的交换（备用）寄存器，只要用交换指令就能实现 AF 与 AF' 交换。

状态标志寄存器 F，表示 CPU 执行指令后的有关的状态标志。如是否有进位、辅助进位、运算结果是否为 0 等，供程序测试、判断、实现分支和转移。因此，F 寄存器使用的好坏关系到程序的质量。F 寄存器有 8 个二进制位，但只用了其中的 6 位作为标志位。这 6 位是进位标志 C，减法运算标志 N，奇偶/溢出标志 P/V，半进位标志 H，零标志 Z，符号标志 S。各位分配如图 1—4 所示，6 个标志中仅有 4 个是可测试的（C、S、Z 和 P/V），另 2 个标志（N 和 H）仅供 CPU 内部使用。

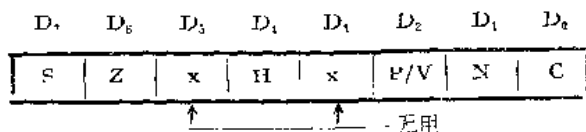


图1-4 状态标志寄存器F

各位标志的意义如下：

1. 进位标志C

进行算术运算时，进位标志C反映运算结果最高位 D_7 的状态（累加器 D_7 ）。作加法时，产生进位则 $C=1$ ，否则 $C=0$ ；作减法时，产生借位则 $C=1$ ，否则 $C=0$ 。

进行BCD码运算时，进位标志C反映两个2位十进制正数相加的进位情况（ $C=1$ ，表示十进制和有进位）和相减的借位情况（ $C=1$ ，表示有借位十进制差为负数）。

进行逻辑运算时，进位标志 $C=0$ 。此外，当Z80-CPU进行循环和移位操作时也会影响该进位标志。

2. 加减标志N

进行加法运算时 $N=0$ ，进行减法运算时 $N=1$ ，以此来区别加法和减法的不同操作。累加器十进制调整指令DAA，就是根据N位判别前一次运算是加法还是减法来进行十进制调整的，因而减法标志N可对十进制调整电路进行控制。

3. 奇偶/溢出标志P/V（或P/O）

该标志是一个多用途标志。在逻辑运算、个别输入操作以及部分循环和移位操作时，当运算结果（累加器的 $D_0 \sim D_7$ ）中1的个数为偶时则置位 $P/V=1$ ，否则复位 $P/V=0$ ；在算术运算时，如运算结果使累加器有溢出时则置位 $P/V=1$ ，否则复位 $P/V=0$ ，所谓溢出是指当累加器值超过了能够表示2的补码数的上限（+127）或者小于下限（-128）时；在数据块传送和搜索操作过程中 $P/V=1$ 不变，直到字节计数器减到零才使 $P/V=0$ ；当执行<D, A, I或<D, A, R指令时，此标志被置为中断允许触发器 IF_2 的现行状态。

4. 半进位标志H (或AC)

半进位标志H, 又称辅助进位标志。运算后如累加器的低四位中的最高位 D_3 有进位(加法)或借位(减法)时, 则 $H=1$, 否则 $H=0$ 。当执行DAA指令时, 此标志与进位标志C一起对十进制调整电路进行控制, 使运算结果被修正为BCD码。

5. 零标志Z

如运算结果累加器的 $D_0 \sim D_7 = 0$ 时, 则 $Z=1$, 否则 $Z=0$ 。零标志Z的功能也很多, 例如在比较操作和异或操作后, 使用该标志可判断两数是否全等或输入输出数据传送中是否出错等。

6. 符号标志S

符号标志S又称负数标志。运算结果如累加器的最高位 $D_7=1$ 时, 则 $S=1$, 否则 $S=0$ 。这对于算术、逻辑以及大部分循环和移位操作等都是适用的。

(三) 通用寄存器

Z80-CPU中有两套配好的通用寄存器组, 都可以用来存放数据, 每组有6个8位寄存器, 即B、C、D、E、H、L与 B' 、 C' 、 D' 、 E' 、 H' 、 L' 。如图1—3所示。它们可以单独使用, 也可以组成16位的寄存器对, 一组是BC、DE、HL; 另一组是 BC' 、 DE' 、 HL' 。通过交换指令EXX, 可以把这一组的内容与另一组的内容成组一一对应交换。

由于Z80-CPU具有较多的通用寄存器, CPU可以直接处理其中存放的数据和地址, 故可以减少访问存储器的次数, 节省访问时间, 提高运算速度。

(四) 运算器ALU

Z80内部的ALU是通过内部数据总线与寄存器和外部数据总线相联进行信息交换, 它有两个数据输入端, 分别与暂时寄存器TMP和ACT相连, 它还有两个输出端, 一个用来输出操作结果, 并将操作结果经数据总线再送回累加器A, 另一个输出端用来输出操作结果的状态标志到F寄存器。ALU所能完成的功能有:

加、减、与、或、异或、求反、求补、增量(加1)、减量

(减1)、移位、循环移位、位操作以及二—十进制运算等多种算术逻辑操作。其中增量、减量在“加1/减1器”中进行；移位和循环移位在总线间进行；二—十进制运算是先将BCD码进行二进制运算，再利用二—十进制调整电路进行BCD码修正，得到二—十进制运算结果。

(五) 指令寄存器和CPU控制

这部分相当于控制器的功能，包括指令寄存器，指令译码器以及CPU控制部件。每条指令从存储器取出后便放到指令寄存器进行译码，经译码产生操作码译码信号。CPU控制部件按照一定的时间顺序，发出一系列控制信号，以使程序中每条指令的各种微操作按顺序正确执行。

三、Z80-CPU引脚说明

Z80-CPU有40条引脚，如图1-5所示

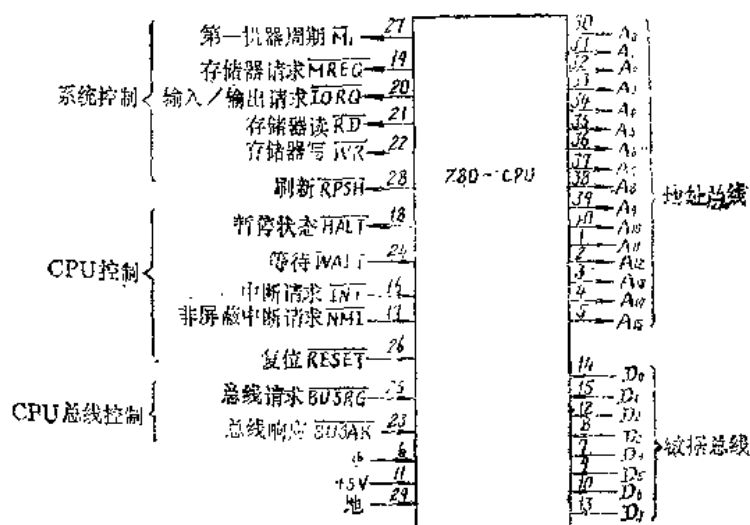


图1-5 Z80-CPU引脚图

Z80-CPU有16条 ($A_{15} \sim A_0$) 为地址总线；有8条 ($D_7 \sim D_0$) 为双向数据总线；3条作为电源、地和时钟信号线；另外 13

条为控制信号线。Z80-CPU各引脚的符号上有一横的为低电平有效，没有一横的为高电平有效。各引脚的功用如下：

(一) $A_{15}-A_0$ 地址总线 (单向输出线，三态)

$A_{15}-A_0$ 组成 16 位地址总线 (A_{15} 为最高位, A_0 为最低位)，它是三态输出，高电平有效。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。当采用动态 RAM 时，在刷新期间，低 7 位 ($A_6 \sim A_0$) 由 R 寄存器提供，所指出的是有效的刷新地址，

(二) D_7-D_0 数据总线 (双向，三态)

D_7-D_0 是双向三态的数据线 (D_7 为最高位, D_0 为最低位)，高电平有效。用于 CPU 和存储器、输入输出装置之间进行数据交换。

(三) 控制信号

Z80-CPU 有 13 条控制信号线，按其功用可分为三部分：系统、CPU 控制和总线控制信号。

1. 系统控制信号

(1) $\overline{M}_1$ 机器周期 (输出、低电平有效)

该信号有效时, $\overline{M}_1$ 指明现行机器周期是取指令操作码周期。如果指令操作码有二个字节 (这些双字节操作码的第一字节为十六进制数 CBH、DDH、EDH 或 FDH)，则每取一个操作码字节，都发生一次 $\overline{M}_1$ 信号，IO 所以有两个 $\overline{M}_1$ 有效信号。此外, $\overline{M}_1$ 和 $\overline{IO}/\overline{RQ}$ 同时为低电平时，表示现行周期是中断响应周期；在 $\overline{M}_1$ 周期内与 $\overline{RFSH}$ 刷新信号相配合。表明可刷新动态 RAM。

(2) $\overline{MREQ}$ 存储器请求 (三态输出，低电平有效)

此信号有效时，表示存储器读或写操作所需的有效地址，已经送到地址总线上了。有时也可将此信号作为存储器的片选信号使用。

(3) $\overline{IORQ}$ 输入/输出请求 (三态输出，低电平有效)

此信号有效时，表示输入或输出的读或写操作所需的有效输入或输出地址，已经送到地址总线上了。当中断被响应时，也产

生一个 $\overline{IORQ}$ 信号,它与 $\overline{M_1}$ 同时为低电平时,表示中断向量(中断服务子程序的首地址)已送到数据总线上。在 M_1 周期可以发生中断响应操作,但在 M_1 周期决不发生输入/输出操作。

该信号与 $\overline{RD}$ 同时为低电平有效时,表示要从I/O口中读入信息给CPU。该信号与 $\overline{WR}$ 同时为低电平有效时,表示要从CPU中输出信息到I/O口。

(4) $\overline{RD}$ 存储器或I/O读(三态输出,低电平有效)

该信号低电平有效时,表示CPU要从存储器或I/O接口读入数据。 $\overline{RD}$ 信号一般与存储器或I/O接口的允许读入端($\overline{RD}$ 端)相连。被寻址的存储器或I/O接口,应用 $\overline{RD}$ 信号把它们的数据门打开(即作为三态的选通信号),使数据进入CPU的数据总线上。

(5) $\overline{WR}$ 存储器或I/O写(三态输出,低电平有效)

该信号低电平有效时,表示CPU送给存储器或I/O接口的数据已送到数据总线上。被寻址的存储器或I/O接口可以到数据总线上取得有效数据,通常可将此信号作为将数据写入到存储器或I/O接口的选通信号。

(6) $\overline{RFSH}$ 动态存储器刷新(输出,低电平有效)

当系统中使用动态存储器时,才使用此信号。它用来对动态存储器周期性地刷新,以防止内存信息丢失。当 $\overline{RFSH}$ 与 $\overline{MREQ}$ 都为低电平时,表明此时是刷新周期,地址总线的低7位($A_6 \sim A_0$)状态是存储器的刷新地址。地址总线的高8位包含I寄存器内容, A_7 是逻辑零。

2. CPU控制信号

(1) $\overline{HALT}$ 暂停(输出、低电平有效)

此信号有效时,表示CPU已经执行了一条 $\overline{HALT}$ 指令,处于暂停状态,此时CPU将不执行任何操作。程序计数器PC不加1,但仍产生 $\overline{RFSH}$ 信号,存储器的刷新操作继续进行,直到CPU接受到非屏蔽中断或者接受到屏蔽中断时,操作才能重新开始。

(2) $\overline{WAIT}$ 请求等待(输入,低电平有效)

此信号有效时,表示CPU所寻址的存储器或I/O接口,尚未