

微型计算机



microcomputer



上海交通大学

1983-1

MIC-80 微型计算机

杜毅仁

金树福

杨辰圆

一九八三年六月

前 言

个人微型计算机和普通的单板计算机之间的区别主要在于前者有固化的高级语言,(一般大都为 BASIC 或 PASCAL),而且配有标准的 ASCII 输入键盘和 CRT 字符显示器,板内随机存储器(RAM)亦足够大(一般在 16KB 以上),个人微型机的主要性能和系统配置大体上与国外流行的个人计算机,诸如 TRS-80、APPLE 等机的主体部份相仿。为了降低成本,特以单板的形式出现,可使用户花很少的费用,即可得到一个灵巧完善的微型计算机系统。

MIC-80 个人微型计算机,中央处理器(CPU)采用 Z-80,板内已有 12KB 的 BASIC 固件,65 个键的 ASCII 输入键盘,利用一般家用 12 吋电视机作为显示设备,板内随机存储器(RAM)最大容量为 48KB(可按用户需要以 16KB 为模增减),只要一开机,功能很强的扩展 BASIC 就准备好为用户服务,板内装有盒式磁带机接口,任何类型的音频盒式录音机均可作为 MIC-80 的外存储器,一盘 60 分钟的标准磁带,可存放 200KB 数据,板内还装有行式打印机接口,可以用来驱动用户所选用的打印机。为了方便用户,进一步扩展计算机的功能,板内还装有三只 50 蕊插座,用户可利用扩充接插印刷电路板来安装欲扩充的接口电路,如 A/D, D/A 转换器,软磁盘控制器,xy 记录仪接口,通讯接口,监控接口,电子游戏接口等,使 MIC-80 能适应各种用户的需要,图一是 MIC-80 的照片。

MIC-80 的设计在硬件上考虑得比较完备,而在软件上又能与 TRS-80 兼容。特别是 Z80(CPU)与 APPLE 中的 CPU(6502)相比,有着很大的优势,因而使 MIC-80 具有更大的潜在能力。可以这样说,MIC-80 在性能/价格比方面完全可以做到优于 TRS-80 和 APPLE 等个人计算机。

为了帮助用户掌握 MIC-80 机,我们编写了这本书。它主要介绍了 MIC-80 机的硬件原理、软件 BASIC 以及机器的使用。

有关编辑/汇编程序,监视、纠错和调试程序将另册介绍。

由于编写时间较短,错误难免,希批评指正。

编 者
1983.春

目 录

第一章	MIC-80 系统构成和设计上的特点	(1)
第二章	MIC-80 的基本工作原理	(2)
(一)	Z80 CPU	(2)
(二)	Z80 接入 MIC-80 系统	(8)
(三)	地址分配及译码	(13)
(四)	动态 RAM	(17)
(五)	显示	(21)
(六)	键盘	(31)
(七)	录音机接口	(33)
(八)	打印机接口	(37)
(九)	机器的使用	(39)
第三章	扩展 BASIC 语言的概述	(41)
第四章	键盘操作命令	(50)
第五章	数据信息的输入和输出	(59)
第六章	分支控制	(73)
第七章	循环控制	(84)
第八章	字符串	(95)
第九章	数组	(105)
第十章	初等函数	(113)
第十一章	特殊函数和逻辑运算	(117)
第十二章	编辑	(127)
第十三章	程序的优化	(132)
第十四章	程序实例	(135)
附 录		
(一)	扩展 BASIC 一览表	(153)
(二)	扩展 BASIC 的错误信息表	(158)
(三)	控制、图形和 ASCII 代码	(159)
(四)	内存分配图	(160)
(五)	显示器屏幕坐标图	(161)
(六)	导出函数	(162)
(七)	一字节十六进制——十进制换算表	(162)
(八)	MIC-80 所用集成电路图	(163)
(九)	元件明细表	(177)

(十)	扩展接插座引脚排列	(180)
(十一)	Z-80 指令系统	(181)

目 录

(1)		第一章
(2)		第二章
(3)		(一)
(8)		(二)
(13)		(三)
(17)		(四)
(21)		(五)
(31)		(六)
(33)		(七)
(37)		(八)
(38)		(九)
(41)		(十)
(50)		第三章
(53)		第四章
(57)		第五章
(61)		第六章
(65)		第七章
(68)		第八章
(101)		第九章
(113)		第十章
(117)		第十一章
(157)		第十二章
(133)		第十三章
(135)		第十四章
		附录
(153)		(一)
(158)		(二)
(159)		(三)
(160)		(四)
(161)		(五)
(163)		(六)
(163)		(七)
(163)		(八)
(177)		(九)
		(十)

第一章 MIC-80 系统构成和设计上的特点

图 1-1 是 MIC-80 的系统框图, Z80CPU 是其中中心, 它通过三组总线缓冲器即 8 位双向数据总线缓冲器、16 位单向地址总线缓冲器以及控制总线缓冲器、与只读存储器 EPROM、随机存储器 RAM、ASCII 码键盘、显示部件(包括时间脉冲发生器, 字符只读存储器和图形处理器以及视频信号合成器等)、打印机接口、磁带机接口等部份相连, 它们均在 Z-80 的控制下有节奏地进行工作, 完成预期的任务。

EPROM 中存放 BASIC 解释程序和操作控制程序。EPROM 是 MIC-80 实际上的主宰, Z-80 只是执行机构而已, EPROM 告诉 CPU 应该做什么, 其结果放到那里去, 有了 EPROM 中的命令, Z-80 才能工作, 否则它将一直空转等待下去, 直到收到 EPROM 来的程序为止, EPROM 共有 12KB, 由 6 片 INTEL 2716 或 3 片 2732 构成, 其地址空间从 0000 开始直到 2FFF, 电源接通以后, Z-80 地址总线输出的地址为 0000, CPU 可以从 EPROM 的 0000 号单元内提取第一条指令, 从而使计算机开始工作。MIC-80 的 E-BASIC 语言与 TRS-80 的 LEVEL II BASIC 兼容, 这是一个扩充的 BASIC 语言, 不但具有人机对话的特点, 而且有比较强的字符串处理功能, 加上它的基本语句类型多, 内容丰富, 表达能力强, 还有较多的特殊函数语句, 故它既可用于科学计算, 也可用于数据处理, 工程管理、资料检索、图表制作以及数控机床, 电子游戏等方面, 是一种学习容易, 使用方便的高级语言。

更换 EPROM 的内容方便的使 MIC-80 微型计算机系统变成功能不同或专用的控制计算机。

值得指出, MIC-80 将高级语言固化, 虽然占用了近 12KB 的内存空间, 但它提高了单板计算机的可靠性和可用性, 为广大用户提供了方便可靠的工具, 计算机的造价由此也大为降低, 这是 MIC-80 设计上的第一个特点。

表 1-1 提供了 MIC-80 存储器地址分配情况。我们可以清楚地看到紧接着 12KB 扩展 BASIC 之后所占用的地址(0000~2FFF)是一个 4KB 的区域, 地址从 3000 到 3FFF, 这个地址区域实际上一分为三。一份备用可为软磁盘, 打印机等接口的地址、(3000 到 37FF), 一份作键盘控制用地址(3800 到 38FF), 另一份给显示 RAM 使用(3C00 到 3FFF)。

这种把主要部件采用存储器映象的作法, 使 MIC-80 的控制非常方便, Z-80 发出的地址直接意味着 Z-80 与那部份打交道, 当 Z-80 需要扫描键盘时, 其地址必然在 3800 到 38FF 范围内变化, 当 Z-80 要与显示 RAM 交换数据时, CPU 发出的地址必然在 3C00 到 3FFF 范围内。这种结构方式可以节省硬件, 也使控制简单直观, 这是 MIC-80 设计上的第二个特点。

内存 RAM 采用 16K 位动态存储器片构成, 并以 16 千字节(KB)为模进行扩充, 主机板内最多可扩至 48KB, 占用地址从 4000 开始一直到 FFFF, 可分为三个区, 每区 16KB。4000 到 7FFF 为基本内存区, 8000 到 BFFF 为第一扩充区, C000 到 FFFF 为

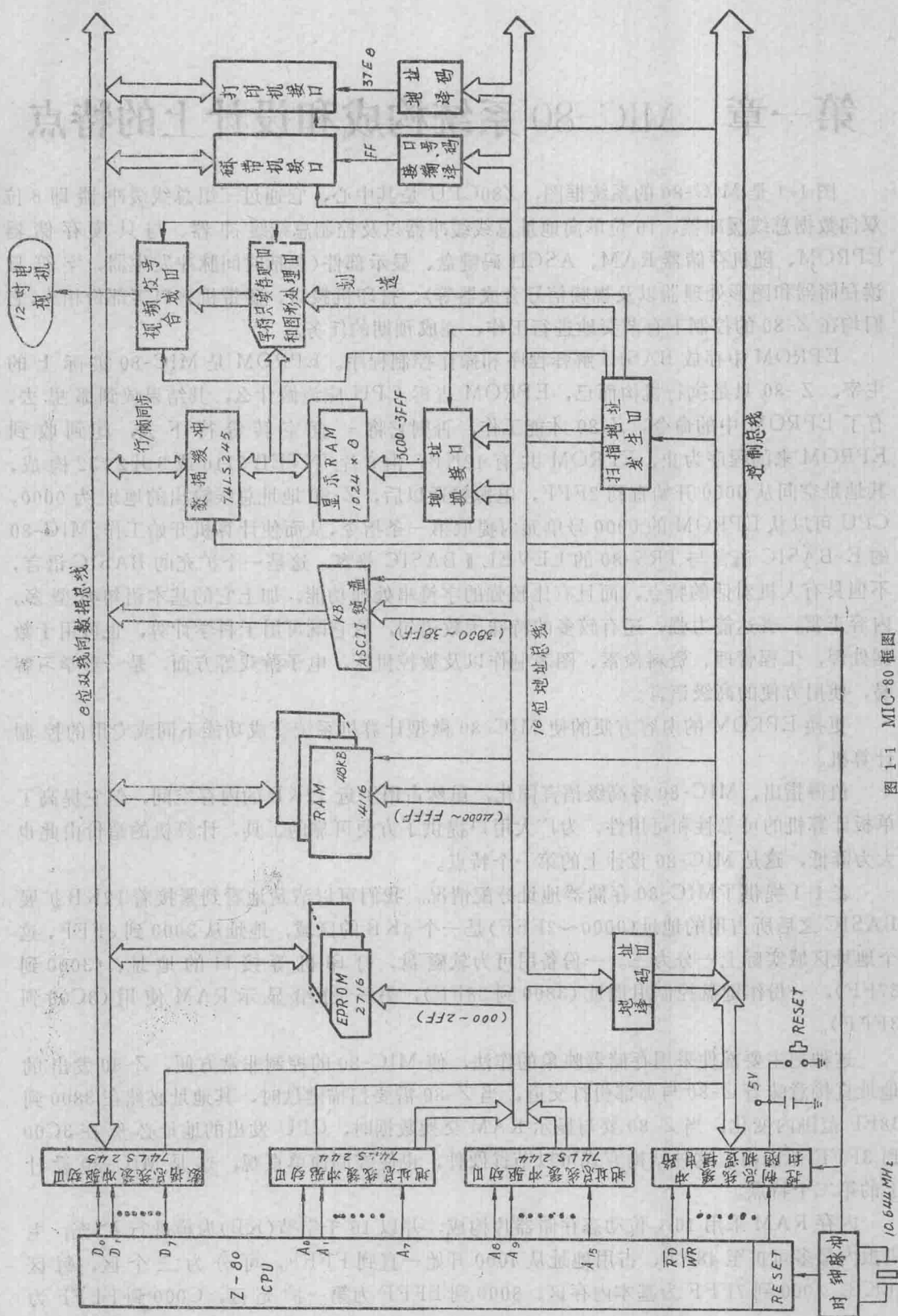


图 1-1 MIC-80 框图

表 1-1 MIC-80 内存分配图

十 进 制	十 六 进 制	内 容	容 量
0	0000 2FFF	E-BASIC ROM	12KB
12288	3000 37FF	备用和外部设备控制地址	2KB
14336	3800 38FF	键 盘	256B
15360	3C00 3FFF	显示 RAM 静态	1KB
16384	4000 7FFF	基本内存区, 动态 RAM	16KB
32768	8000 BFFF	第一扩充区, 动态 RAM	16KB
49152	C000 FFFF	第二扩充区, 动态 RAM	16KB

第二扩充区。16K 动态 RAM 片子的十四位地址采用两次打入的(这样可以节省存储片的引脚数)。先在行选择信号 $\overline{\text{RAS}}$ (低电平有效)的控制下, 打入七位行地址($A_0 \sim A_6$), 然后在列选择信号 $\overline{\text{CAS}}$ 的控制下再打入七位列地址, ($A_7 \sim A_{13}$), 值得注意的是地址转换控制应在 $\overline{\text{CAS}}$ 之前约 100ns 处进行, 以便保证 $\overline{\text{CAS}}$ 低电平到来之时, 行地址电平已经到达稳定值, 三者的时间关系如图 1-2 所示。

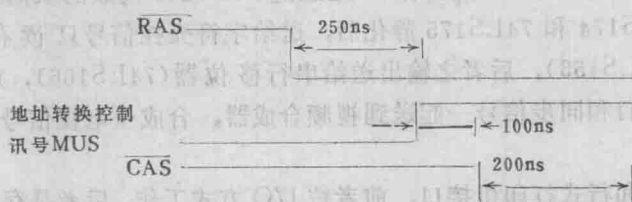


图 1-2 动态 RAM 的刷新控制波形

动态 RAM 的刷新是在 $\overline{\text{RAS}}$ 为低电平, $\overline{\text{CAS}}$ 为高电平时间内进行的, 就是说在此时期内, $\overline{\text{RAS}}$ 所选中行上的每个存储单元被再生一次。刷新动态 RAM 的周期应小于 2ms。刷新地址的具体内容是由 Z-80 来控制的。这种刷新的办法, 节省了不少硬件器材。这是 MIC-80 微型计算机设计的第三个特点。

MIC-80 的 ASCII 键盘共有 65 只键, 其中 53 只基本上按标准打字机位置排列, 另外 12 只键组成 1 个 3×4 的矩阵位于标准键盘的右方, 这 12 个键与右方标准键盘对应的键符功能完全相同(电路上是并联的)。

键盘的读入是通过软硬结合的处理方法, 这再一次显示了它的优越性。它使键盘控制非常简单, 可靠。这是 MIC-80 设计上的第四个特点。

MIC-80 的显示器既可用来显示字符, 亦可显示简单的图形。由于采用电视机扫描方式, 故可用电视机作为显示设备, 这为 MIC-80 进入小单位和家庭创造了条件。值得强调的是, 显示过程的处理亦沿用软硬结合的办法进行设计, 使显示部件所支付的开销大为减少。这是 MIC-80 设计上的第五个特点。

实际上, MIC-80 系统比一台常规的 CRT 字符显示器还便宜。从这一点就可看出 MIC-80 设计上的主要特色了。

为了在电视屏上稳定地显示字符或图形, 必须引入一个显示 RAM, 用来暂时存放被显示内容的编码, 对字符来说就是 ASCII 码, 对图形来说则用 ASCII 码的扩充部分的编

码。每个编码为 7 位二进位。为了在屏上显示 1024 字，显示 RAM 的容量为 1024×7 ，最高位的信息是用来区分显示字符和图形的。RAM 中的地址与屏上显示的字位之间成一一对应的关系。随着电视扫描到达的字位，与其对应的显示 RAM 中的 ASCII 码就被读出，并送给字符显示 ROM 或图形处理器，在划选的控制下，译出字符对应的亮控信号，送入视频合成器，以控制 CRT 的阴极，就可控制该字符点阵中那点该亮，从而显示出字符来。电视机扫描要周而复始地进行，每秒钟重复 50 次，这样才能在屏上看到稳定的字符或图形。为了使电视机的扫描和访问显示 RAM 的地址保持严格的同步，MIC-80 中引入了一个扫描地址发生器，它产生访问显示 RAM 的地址和电视机的行、帧同步信号，从而保证两者的协调一致。

显示 RAM 的地址，一方面由扫描地址发生器提供，另一方面也必须接受 Z-80 的控制。当 Z-80 要与显示 RAM 交换数据时，其地址须由 CPU 提供，并受到 CPU 读、写信号的控制。

由图 2-15 显示接口图可知，4 片 74LS93 和一片 74LS157 组成扫描地址发生器，它既产生扫描地址，亦产生行/帧同步信号。三片 74LS157 构成地址转接器，显示时，把扫描地址送出；CPU 要访问显示 RAM 时， $\overline{VID}$ 为低电平，则把 CPU 送来的地址送至由两片 2114 构成的显示 RAM。显示 RAM 的输出，一是经过 74LS245 构成的双向缓冲器与数据总线相连。二是经 74LS174 和 74LS175 净化后，送给字符亮控信号只读存储器 (MCM6670) 或图形处理器 (74LS153)，后者之输出送给串行移位器 (74LS166)，就变成了字符或图形的亮控信息，它们和同步信号一起送到视频合成器，合成全电视信号后输向电视机，显示字符或图形。

板内装有盒式录音机接口和行式打印机接口，前者按 I/O 方式工作，后者是存储映象方式。录音机 I/O 接口编号为 FF，行式打印机对应的存储器地址编码为 37E8。

第二章 MIC-80 基本工作原理

(一) Z80CPU

Z80 微处理器是美国 ZiLog 公司生产的八位微处理器，也是目前八位微处理器中功能较强的一种。Z80 微处理器包括了 8080 处理器的全部处理功能，软件可与 8080 兼容，指令比 8080 又增加了 80 条，共 158 条，指令周期快，特别是 Z80 具有许多独特的指令，可用于保证多字节数据块操作，这在数据通信和文本处理中是很方便的，Z80 也是用 n 沟硅栅耗尽负载工艺制造，用单一的 5 伏电源，具有和 TTL 兼容的输入和输出电平。

MIC-80 微型计算机就是采用这种 Z80 微处理器作为系统的 CPU(中央运控部件)。

图 2-1 提供了 Z80 内部的结构图。Z80 中包含有十八个 8 位寄存器和四个 16 位寄存器，全部寄存器用静态 RAM 实现。

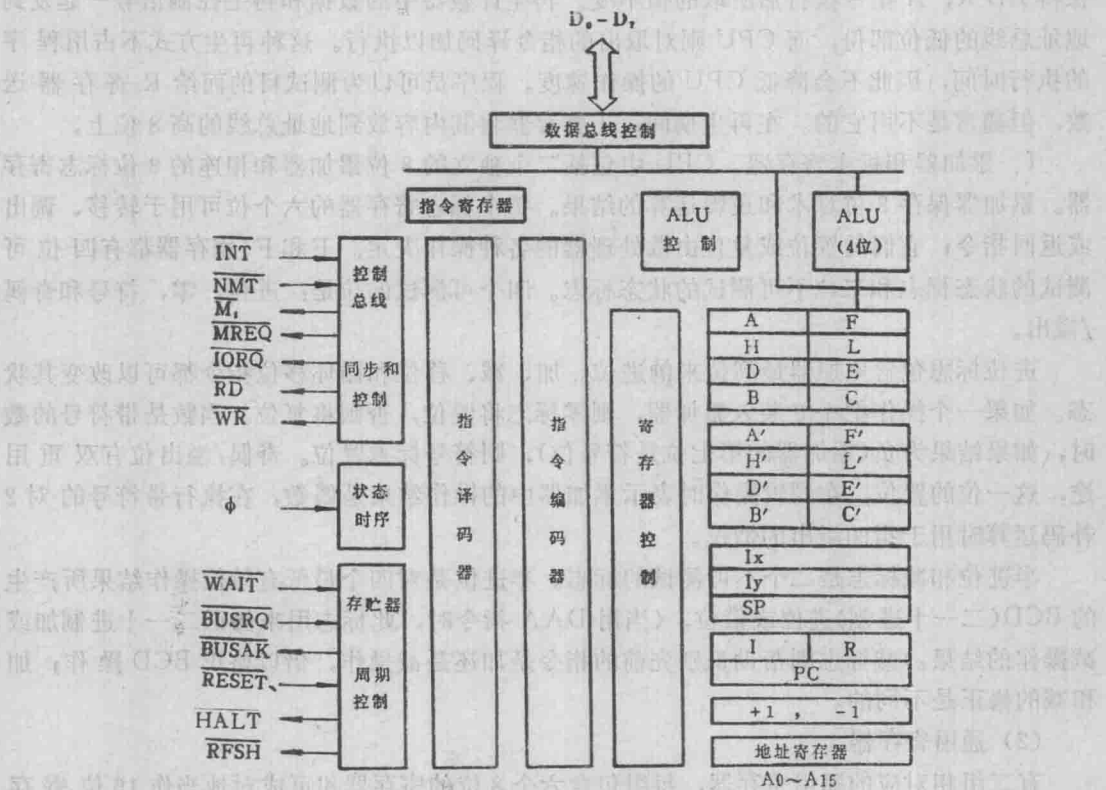


图 2-1 Z80 CPU 内部结构

(1) 专用寄存器

a. 程序计数器(PC) 程序计数器中保存着有待从存储器取出的现指令的 16 位地址。当其内容送到地址线之后便自动加 1。在发生程序转移时，新的值自动置入程序计数器，取代已递增的数值。

b. 堆栈指示器(SP) 堆栈指示器保存着现栈顶的 16 位地址。堆栈可位于系统 RAM 存储器的任何位置上。片外堆栈存储器是按后进先出存储器的形式组织的, 数据可通过执行压入(PUSH)和弹出(POP)指令, 把 CPU 寄存器中内容压入堆栈, 或从堆栈弹进 CPU 寄存器。从堆栈弹出的数据总是最后一个压入的数据, 用堆栈可以简单地实现多级中断, 无限制的子程序嵌套和简化许多类型的数据操作。

c. 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用的 16 位基地址。变址指令中还有一个附加的字节, 用来指明相对此基地址的位移量。规定位移量是对 2 补码的带符号整数。这一寻址方式大大简化了许多类型的程序, 特别是用在数据表的情况里。

d. 中断页面地址寄存器(I) Z80 CPU 在响应中断时可以间接调用任一存储单元。此时 I 寄存器用于存放间接地址的高 8 位, 而中断设备则提供地址的低 8 位。这种特征使中断处理例程可以动态地位于存储器的任何位置, 且使访问例程的绝对时间为最小。

e. 存储器再生寄存器(R) Z80 CPU 中包含存储器再生计数器, 使得用动态存储器和静态存储器一样简单方便。在每次取指令后 8 位寄存器中的七位自动加 1, 而第八位则保持 LD R, A 指令执行后所取的值不变。再生计数器中的数据和再生控制信号一起发到地址总线的低位部份, 而 CPU 则对取出的指令译码加以执行。这种再生方式不占用程序的执行时间, 因此不会降低 CPU 的操作速度。程序员可以为测试目的而给 R 寄存器送数, 但通常是不用它的。在再生期间, I 寄存器将其内容放到地址总线的高 8 位上。

f. 累加器和标志寄存器 CPU 中包括二个独立的 8 位累加器和相连的 8 位标志寄存器。累加器保存 8 位算术和逻辑运算的结果。每个标志寄存器的六个位可用于转移、调出或返回指令; 它们的置位或复位由微处理器的各种操作决定。F 和 F' 寄存器都有四位可测试的状态标志和二位不可测试的状态标志。四个可测试的位是: 进位, 零, 符号和奇偶/溢出。

进位标志包含累加器最高位来的进位。加、减、移位和循环移位指令都可以改变其状态。如果一个操作把全 0 装入累加器, 则零标志将置位, 否则将复位。当数是带符号的数时, 如果结果为负(累加器的第七位是符号位), 则符号标志置位。奇偶/溢出位有双重用途, 这一位的置位, 在逻辑操作时表示累加器中的操作结果是偶数, 在执行带符号的对 2 补码运算时用于指明溢出的情况。

半进位和减标志是二个不可测试的标志。半进位是对四个最低有效位操作结果所产生的 BCD(二一十进制)进位或借位。(当用 DAA 指令时, 此标志用来修正二一十进制加或减操作的结果。)减标志则帮助识别先前的指令是加还是减操作, 借以修正 BCD 操作, 加和减的修正是不同的。

(2) 通用寄存器

有二组相对应的通用寄存器, 每组包含六个 8 位的寄存器和可成对地当作 16 位寄存器使用它们。一组 BC、DE 和 HL, 而另一组叫 BC'、DE' 和 HL'。任何时候程序员都可以通过一条交换指令选择其中一组寄存器进行工作。在要求快速响应中断的系统中, 有一组通用寄存器和累加器/标志寄存器可留作处理这一快速例程之用。为从一个例程转到另一个例程, 只要执行简单的交换指令即行。这样, 在处理中断和子程序时, 不必再将寄存器内容存入外部堆栈或从其中取出, 因而大大减少了中断服务时间。对这些通用寄存

器，程序员可以派广泛的用途。这些寄存器还大大简化了程序设计，特别是在以 ROM 为基础的只有少量读/写存储器系统中。

(3) 运算器(ALU)

CPU 的 8 位算逻辑指令是在 ALU 中执行的，ALU 还控制内部寄存器和外部数据总线之间的数据交换。ALU 执行的功能有：加、减、逻辑与、逻辑或、逻辑异或、比较、左或右移或循环(算术和逻辑)、增量、减量、位置位、位复位、位测试。

(4) 指令寄存器和 CPU 控制

每条指令从存储器取出后便放到指令寄存器进行译码。控制部份在实现这个作用之后，产生和提供从或向寄存器读或写所需的全部控制信号，控制 ALU 和提供全部要求的外部控制信号。

Z80 CPU 引脚图 2-2 提供了 Z80 CPU 双列直插 40 脚输入输出的分布情况。虽然 Z80 在定时和控制信号方面和 8080 是兼容的，但是引脚的分配不同。全部输出脚可在 0.4 伏以下吸收 1.8 毫安电流，这相当于一个标准的 TTL 负载。

从 Z80 出来的三条总线——十六位地址总线，八位双向数据总线和十三根线的控制总线，共占去 Z80 的四十条脚中的三十七条。余下三条用于电源、地和单相时钟。和 8080 不同的是 Z80 不需要状态锁存或时钟，且中断向量处理和动态存储器再生，完全由微处理器片内的逻辑实现。

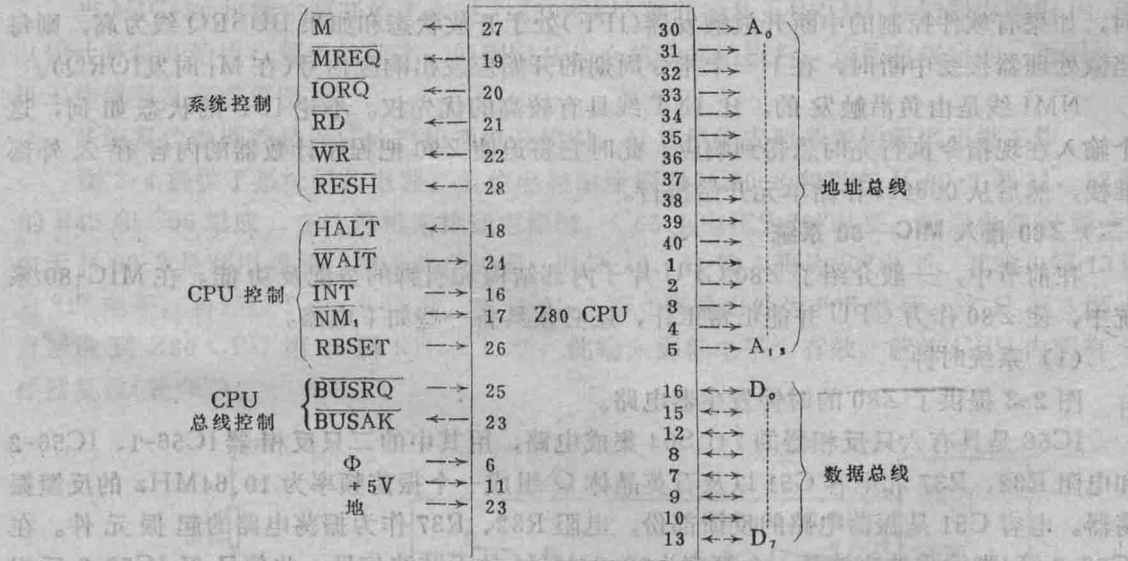


图 2-2 Z-80 引脚分配

十三条控制总线实际上分为三组：系统控制(六条)、微处理器控制(五条)和微处理器总线控制(二条)，一条总线控制线起着总线请求线(BUSRQ)的作用，由这根线输入的信号，不仅要求微处理器的地址和数据总线变成高阻状态，而且也要求系统控制总线中的存储器请求、I/O 请求、读数据和写数据等控制线变为高阻抗状态，使其它器件能占用总线。另一条总线控制线用于输出，称为总线响应(BUSAK)的输出信号，它变为高电平时指示上述各线进入高阻抗的第三态。

六个系统控制信号都是从 Z80 输出。 $\overline{M}_1$ 线上(机器周期 1)电平变低指明 Z80 处于

取指令操作码阶段。存储器请求线(MREQ)在地址总线上保存着存储器读或写操作需要的有效地址时变低。I/O 请求线(IORQ)变低则指明地址总线的低位字节保存着 I/O 读、写操作需要的有效 I/O 口地址。

存储器读和存储器写二条线(RD 和 WR)也是低为有效。RD 指明微处理器要从存储器或 I/O 设备读数据。而 WR 则指明数据总线保存着数据,待存入所寻址的单元。当第六条系统控制线即再生信号(RFSH)变低时,它指明地址总线较低七位含有动态存储器的再生地址,因而现在的 MREQ 信号应该用作全部动态存储器的再生读出。

五条微处理器控制线中,有四条是输入线,一条是输出线。它们全是低为有效。唯一的输出线是暂停线(HALT),它指明 Z80 何时执行完暂停指令和在等待不可屏蔽或可屏蔽中断。在暂停期间,微处理器自动执行空操作指令,以保持动态存储器再生。等待输入(WAIT)给微处理器指明,所寻址的存储器或 I/O 设备尚未作好数据传送的准备(此时微处理器将进入等待状态,直到此控制线为低)。这根控制线使任何速度的存储器或外部设备都能和 Z80 同步。

为了在接通电源之后使微处理器复位或置位于初始状态,RESET 线可以拉到低电位。此时将迫使 Z80 的程序计数器置成 00H,封锁中断开放触发器,把 I 寄存器置为 00H,把 R 寄存器置为 00H,和把中断置成 0 方式。

最后二根控制线是中断请求(INT)输入和不可屏蔽中断(NMI)输入。当 INT 线为低时,如果有软件控制的中断开放触发器(IFF)处于开放状态和如果 BUSRQ 线为高,则每当微处理器接受中断时,在下一个指令周期的开始应发出响应信号(在 M₁ 时发 IORQ)。

NMI 线是由负沿触发的,比 INT 线具有较高的优先权。不论 IFF 的状态如何,这个输入在现指令执行完时总得到响应。此时它将迫使 Z80 把程序计数器的内容存入外部堆栈,然后从 0066H 存储单元开始执行。

(二) Z80 接入 MIC-80 系统

在前节中,一般介绍了 Z80 CPU 片子内部结构和引脚的分配及功能。在 MIC-80 系统中,使 Z80 作为 CPU 并能正常工作,还必须具备一些如下电路。

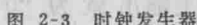
(1) 系统时钟

图 2-3 提供了 Z80 的时钟发生器电路。

IC56 是具有六只反相器的 74LS04 集成电路,用其中的二只反相器 IC56-1、IC56-2 和电阻 R32、R37 和电容 C51 以及石英晶体 Q 组成一个振荡频率为 10.64MHz 的反馈振荡器。电容 C51 是振荡电路的反馈部份,电阻 R32、R37 作为振荡电路的起振元件。在 IC56-2 第 4 脚输出处应该是一个频率为 10.64MHz 的正弦波信号,此信号经 IC56-3 反相器的缓冲,在 IC56-3 第 6 脚输出是一个 10.64MHz 的方波信号,这个方波信号接到集成电路 IC72。

IC72 是一只 12 分频的计数器。此计数器分为除 2 以及除 6 部分,CP₀ 和 Q₀ 是除 2 部份的输入端和输出端,CP₁ 和 Q₃ 为除 6 部份的输入端和输出端。频率为 10.64MHz 的方波在 IC72 的 CP₁ 输入,经过除 6 部分,六分频后在 IC72 的 Q₃ 端得到 1.774MHz 的方波,再经过电路 IC75 的缓冲,直接送到 Z80 CPU 的第 6 脚,作为 Z80 工作的单相时钟脉冲。

IC72 的第 6、7 端是计数器的清除端,输入高电平为有效,现固定接在 IC56-4 的第



(2) 系统复位

系统复位电路就是保证计算机通电开机时,计算机能按照预置的程序正常工作。

图 2-4 提供了系统复位电路。复位电路由连接到 IC90-2 和接在 IC90-2 第 11、12 脚的 R45 和 C65 组成。在计算机未接通电源时, C65 上电压为“0”电平, 接通电源的瞬间, 由于 IC90-2 是利用或非门接成的反相器, 其第 12、11 输入脚为“0”电平, 其输出第 13 脚为“1”电平, 再经过 IC91-3 反相, 在 IC91-3 第 3 脚输出处为“0”电平, 这只第 3 脚是直接连到 Z80 CPU 第 26 脚 $\overline{\text{RESET}}$ 处; 此输入脚低电平为有效, 就使 CPU 内所有寄存器复位(置“0”)。



接通电源之后, +5 伏通过 R45 电阻开始对电容 C65 充电, 经过一定的充电时间(数微秒)后, 电容 C65 两端电压上升到 1.4 伏时, IC92-2 就发生翻转, 其输出第 13 脚变为“0”电平, 再经 IC91-3 的反相, Z80 CPU 的第 26 脚 RESET 变为高电平“1”, 系统复位(数微秒)就告完成。

在系统复位期内, CPU16 根地址总线的值为 0000H, 一旦经数微秒后, 复位结束, CPU 的 PC(指令计数器)就开始从 0000H 处开始计数工作。也即 EPROM 从地址 0000H 单元处开始执行程序。

复位按钮开关 K1 是与接通电源开关时的置“0”(复位)具有同样的功能。在该开关的回路中所加的电容器 C12 要比电源复位电路中的电容 C65 为小。这是为了使它建立逻辑“1”的状态比电源复位电路中的电容快的缘故。

按下复位开关 K1, IC91-1 第 10 脚呈现 0 伏, 于是 IC91-1 第 8 脚电平为“1”, 再经 IC91-2 反相, 在 IC91-2 第 6 脚上输出就为“0”电平, 而第 6 脚是直接至 Z80 CPU 的第 17 脚 NMI 上, 此脚低电平为有效, 迫使 Z80 CPU 内的 PC(指令计算器)从 0066H 处开始工作, 在按下开关 K1 期间, Z80 的 16 根地址总线值为 0066H, 一旦放开开关后, 电容 C12 经 R12 由 +5V 充电后达到 1.4V 时, IC91-1 和 IC91-2 相继发生翻转, NMI 变为“1”电平, Z80 CPU 就从 EPROM 的 0066H 的地址单元开始执行程序。

复位开关 K1 的另一个作用是, 当 Z80 CPU 由于某种原因失去控制, 计算机已不能正确执行程序时, 按下此复位开关, 用它来使计算机返回到重新执行正确的程序。譬如, 在使用盒式磁带机来传送程序的过程中, 由于某种原因使磁带机继续不停的工作, 这时按下开关 K1, 传送程序即可中断, 计算机重新返回到初始的准备状态。

因为某种原因计算机在执行过程中常需要加进停止命令。而这时当你按下开关 K1, 计算机也返回不到初始准备状态, 在发生这样的情况下, 我们就得切断电源开关从头开始。值得注意的是, 通过按动复位开关 K1 使计算机返回到准备状态时, 这不会影响存在计算机存储器内的用户程序以及数据, 而通过重新开启电源开关使计算机回到初始准备状态的方法将使存在计算机存储器中的程序和数据, 会因为存储器断电而全部清除, 用户将不得不从“头”开始。

由于 CPU 的失控, 在不得已的情况下必须切断计算机电源开关时, 要至少停 10 秒钟后方能启动电源开关。这正是在 C65 中所保留的充电电荷要得到完全的泄放的时间。如果计算机断开电源不到 10 秒钟就立刻接通电源时, 则从 Z80 CPU 地址总线上就不能出现初始地址 0000H, 而且 Z80 CPU 又会重新失去控制。

Z80 CPU 的第 18 脚是 HALT 端输出, 低电平为有效, HALT 信号指明, CPU 执行了 HALT 软件指令, 并正在等待不可屏蔽或可屏蔽中断。在暂停时期, CPU 执行空操作指令, 以保证动态存储器的刷新。

IC90-1 第 8 脚、第 9 脚分别和 IC91-1 第 8 脚和 IC90-2 第 13 脚相连, 而 IC90-1 第 10 脚就作为系统复位信号 SYSRES。该信号通过接插座和要扩的外部接口相连。

(3) 总线驱动

Z80 CPU 共有三组总线, 即地址总线、数据总线和控制总线, 其全部的输出脚可在 0.4 伏以下吸收 1.8 毫安的电流, 这相当于一个标准的 TTL 负载, 而现实的情况是 Z80 CPU 一旦接入 MIC-80 系统内, 每条总线还要接上许多负载, 诸如 EPROM、RAM、键

盘、显示 RAM 等。这就必需在 CPU 和这些负载之间加设总线驱动电路。

图 2-5 提供了地址总线驱动电路。我们选 IC82、IC65(74LS244)作为总线驱动器。74LS244 是单向的、具有 8 路输入输出的缓冲驱动器，每路输出端最大可吸收 24mA 电流，具有三态控制功能。用二块 74LS244 共 16 个输入端分别和 Z80 CPU 地址总线 $A_0 \sim A_{15}$ 一一对应连结，它们的输出端分别连接到 EPROM、RAM 的地址转接器、显示 RAM 地址多路转接器和一些地址译码器。为了计算机的测试和维修时的方便，IC82、IC65 第 1 脚、第 19 脚 $\overline{OE}$ 端接至 IC80-1 输出脚第 13 脚，而 IC80-1 输入脚第 14 脚通过电阻 R34 接在 +5V 上(“1”)，所以 IC80-1 输出脚第 13 脚，即连接到 IC82、IC65 第 1 脚和 19 脚的 $\overline{OE}$ 端平时总为低电平(“0”)，这样保证了 Z80 CPU 的地址总线经 IC82、IC65 缓冲驱动而送到 MIC-80 系统地址总线上。

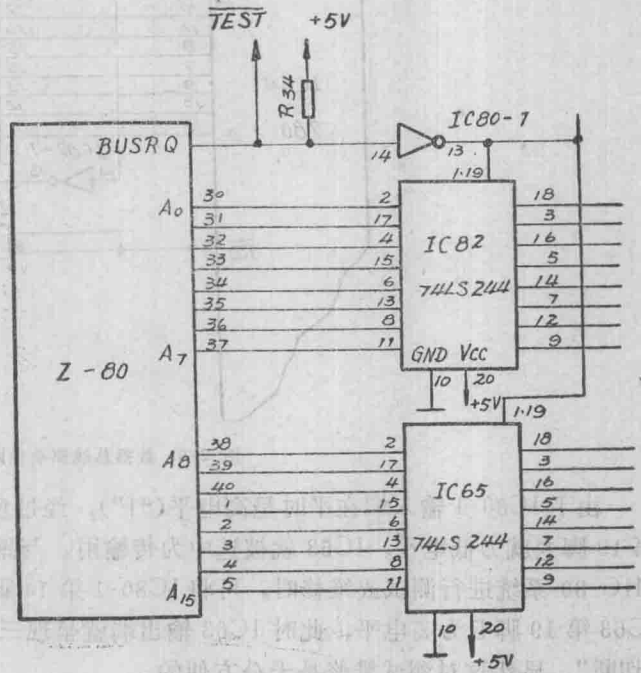


图 2-5 地址总线驱动电路

只有在对计算机进行测试和维修时，可将 IC80-1 输入端第 14 脚(即接插座上的 TEST 端)接地(“0”)，可方便地使 IC82、IC65 输出端呈现第三态，即高阻状态，此时，Z80 CPU 的地址总线和 MIC-80 系统地址总线在缓冲器的位置上被“切断”，用户就可对 MIC-80 进行各种测试和维修。

图 2-6 提供了数据总线的驱动缓冲电路。我们选集成电路 IC63(74LS245)作为数据总线驱动电路，74LS245 是双向 8 路输入输出的总线驱动器并具有三态控制功能。

将 IC63 的 $B_0 \sim B_7$ ，分别和 Z80 CPU 数据总线 $D_0 \sim D_7$ 连接，而 IC63 的 $A_0 \sim A_7$ 与 MIC-80 系统数据总线；即与动态 RAM、显示 RAM、打印接口、盒式录音机接口相连接。地址总线只能从 Z80 CPU 单方向地向 MIC-80 系统地址总线传送，而数据总线却不同，Z80 CPU 数据总线不但能向 MIC-80 系统数据总线传送数据，同时也能由 MIC-80 系统数据总线向 Z80 CPU 的数据总线传送数据，其传输方向由 IC63 第 1 脚 DR 的电平控制，而 DR 是与 IC86-1 的第 8 脚相连，而 IC86-1 的输入脚与 Z80 CPU 的第 21 脚 RD 相连。当 Z80 CPU 要从 MIC-80 系统数据总线上读取(进)数据时，Z80 CPU 第 21 脚 $\overline{RD}$ 发出读脉冲(低电平有效)，经过 IC86 的反相在 IC63 的第 1 脚 DR 处就有高电平，此时 IC63 的传输方向是从 $A_0 \sim A_7$ 传输到 $B_0 \sim B_7$ 。当 CPU 对接在 MIC-80 系统数据总线的存储器等进行写入时， $\overline{RD}$ 发出高电平，经过 IC86-1 反相，在 IC63 第 1 脚 DR 上就为低电平(“0”)，此时 IC63 传输方向是从 $B_0 \sim B_7$ 传输到 $A_0 \sim A_7$ 。

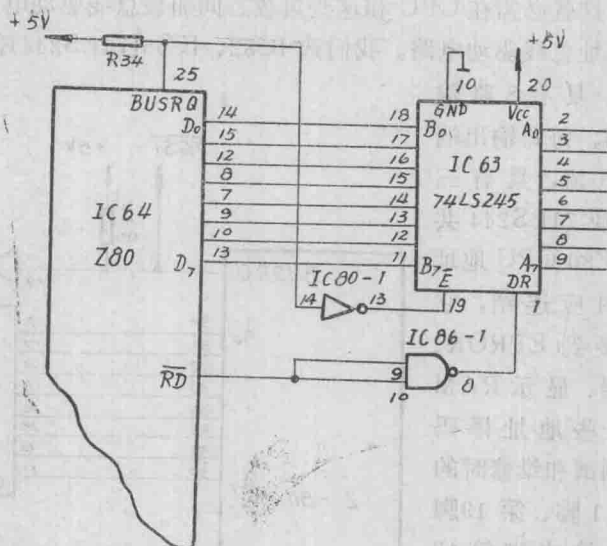


图 2-6 数据总线驱动电路

由于 IC80-1 输入端在平时呈高电平(“1”), 经过反相, 其输出端第 13 脚接到 IC63 第 19 脚 E 就为低电平, IC63 就被选中为传输用。与地址总线驱动器同理, 一旦需要对 MIC-80 系统进行测试或维修时, 可将 IC80-1 第 14 脚即 TEST 脚接低电平, 经过反相 IC63 第 19 脚 E 为高电平, 此时 IC63 输出端就呈现三态(高阻态), 数据总线驱动器就被“切断”, 显然这对测试维修是十分方便的。

图 2-7 提供了控制总线驱动器电路。此电路是由 IC81-2 和 IC80 组成。

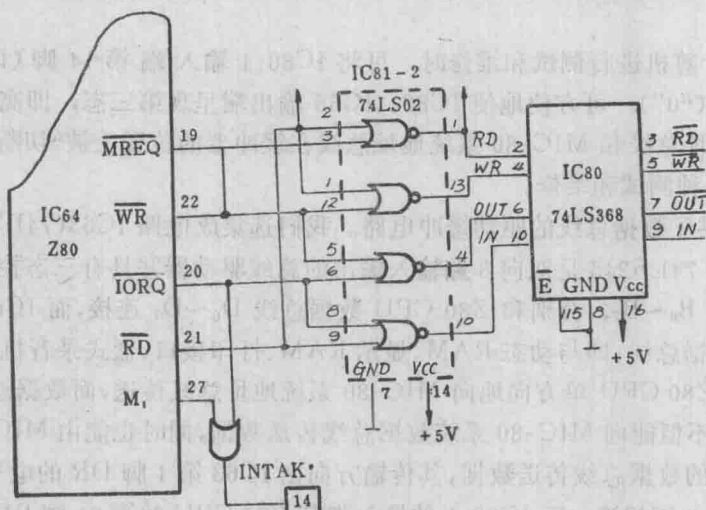


图 2-7 控制总线驱动器

IC81-2 是一片包含有四只或非门的集成电路, 其功能仅是把 Z80 CPU 所输出的控制信号 MREQ、WR、IORQ、RD 信号组合成按 MIC-80 系统控制所要求的 RD、WR、OUT、IN 信号, IC80 仅把上述的四个信号再反相一次, 产生系统所要求的 RD、WR、OUT、IN 信号。