微型计算机丛书

硬件

微型计算机系统

内容简介

这本丛书取材于国内外有关资料，比较详细地介绍了微型计算机软、硬件系统。内容包括宏汇编、BASIC、微型机操作系统和微型计算机系统四大部分。宏汇编部分有宏汇编程序原理和程序编制法、编辑程序的输入和修改以及调试程序的原理和使用；BASIC部分介绍了扩展BASIC、编译BASIC和控制用BASIC；微型机操作系统部分主要叙述了微型机操作系统以及CDOS等操作系统实例；微型计算机系统部分包括典型的8位、16位微处理器、微型机结构系统、微型机的拓展与开发系统、微型机外部设备以及微型机在各领域中的应用。

本书用例丰富，实用与理论并重，讲述由浅入深，具体易懂，是目前国内微型机资料中较新、较全面、较系统的一种参考书。

本书适于作微型机培训教材或自学入门书。

目　　录

第一章　微处理器和微型计算机概述

第二章 微计算机系统

第三章 系统的扩充

第四章 微型机专用系统

第五章 微处理机开发系统(MDS)

第六章 微型计算机的外部设备

第七章 安 装 調 试

第一章　微处理器和微型计算机概述

1.1　概　　况

1.1.1　什么是微型计算机

微型计算机是由一片或几片大规模集成电路组成的计算机，工作原理与一般小型计算机相同。通常把中央处理器(CPU)部分制在一两片集成电路上，称为微处理器(Microprocessor)，再接上存储器（ROM和RAM)和I/O接口，构成计算机系统称为微型计算机(Microcomputer)，如图1-1所示。以前一般是这样分法，但目前已出现了单片的微型计算机，在一片集成电路上包括全部计算机内容，即CPU、RAM、ROM和I/O接口等。

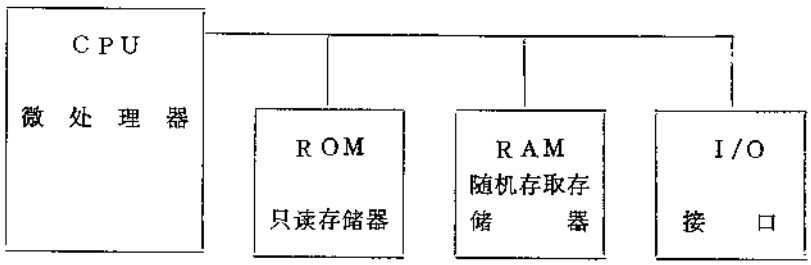


图1-1　微计算机系统示意图

由于大规模集成电路的发展，微处理器和微型计算机大量生产，价格日益下降。“价钱便宜”是微型计算机的主要特点之一。故又称微型计算机为“廉价的计算机”。它为计算机的应用，开辟了广阔的前景。

1971年11月Intel公司制造的4004微处理器，是第一个微处理器，它是4位的，采用PMOS技术，16个引脚，双列直插式封装。可作为一般计算器，功能较低，未推广使用。

1972年初Intel公司制出了8位的微处理器8008，18个引脚，采用PMOS技术，1973年秋天，Intel 8080问世。此后，许多公司争相制造微处理器和各种配套片子，型号和数量繁多，发展甚快。

1.1.2　8位微处理器

自从Intel 8080问世以来，相继出现多种8位微处理器，几年来8位微型机一直占主要地位，许多制造厂，自成系列，互相竞争，产品性能逐步改进。目前销售量较多的8位微处理器是8080、6800、6502、Z-80、8085等，Motorola公司围绕6800微处理器制造了一套存储器和接口片子，即6810(RAM)、6820(PIA)、6830(ROM)和6850(ACIA)，采用了NMOS技术和5伏单一电源。MOS Technology公司的6501、6502、性能与6800类似，但每片的售价都降到25美元，比以前的微处理器便宜四倍，从而整个微处理机市场价格降低。Zilog公司在8080基础上制造了Z-80微处理器，性能有许多改进。以后，Intel公司制造了8085，投入市场竞争。1980年National Semiconductor公司的产品NSC800是最新的8位微处理器，采用了CMOS技术，40个引脚，综合了Z-80

和8085的优点。8位微处理机仍在发展，但不像前几年那样快了。

各厂家在研制新的微处理器片子的同时，都注意了片子配套和系统的发展。因此8位微型机的片子和系统的发展比较完善，以满足更高运算能力和运算速度的需要。

微型机硬件的发展可分成五个等级：

（1）片子：包括各种RAM、ROM存储器片子、通用异步I/O接口片子，磁盘驱动器接口片子，模/数转换片子和其它专用片子。这些片子多是大规模集成电路，复杂程度不亚于CPU。

（2）模块板：由片子组成各种功能的模块板，如CPU板，存储器板，I/O接口板等。包括单板微型计算机。这些板多是以一块或几块大规模集成电路为核心，配一些中小规模集成电路，以适应各种总线、系统的需要。使用比较灵活，只须把板子插在总线上，就可进行系统扩充。

（3）小的微型机系统：包括CPU，RAM，ROM，I/O接口和必要的输入输出设备。

（4）微型机开发系统：功能齐全的微型计算机系统，可作为研制微型机产品的工具。

（5）多处理机系统：分布式系统

软件的发展也可相应的分为五个等级。从低级语言到高级语言，系统软件已相当完备，基本继承了小型计算机的软件技术，但还赶不上小型机。

1.1.3 8位单片微计算机

为了适应大量的控制方面的需要，几个主要的半导体制造厂都生产了8位的单片微计算机，例如Intel的MCS-48系列，Mostek的3870、Motorola的6801、Zilog的Z-8等都是8位单片微计算机。这些单片计算机的特点是在一个片子上做出了功能齐全的计算机系统，功能与8位机接近，但价钱便宜的多。有的单片机上包括计时器，通用异步接口(UART)、模/数转换等，作控制机用，比较方便，存储容量不大，但能满足一般控制的需要。存储器包括程序存储器ROM或PROM和数据存储器RAM。为了便于新系统的研制，有的配有可改写的EPROM。

由于控制用的计算机需要量非常大，8位单片计算机是微型计算机一个发展方向。微型计算机的发展趋势是片子内部包括的元件数越来越多，组成一个计算机系统所用的片子数则越来越少。

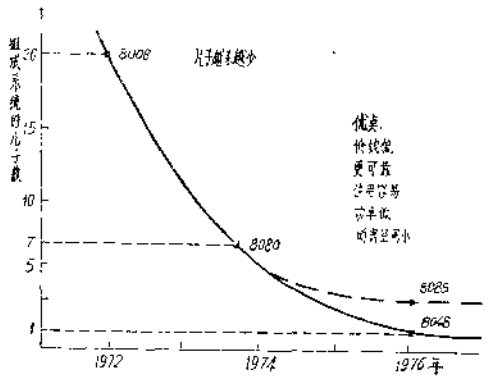


图1-2 微计算机系统发展趋势
（1）片子越来越少

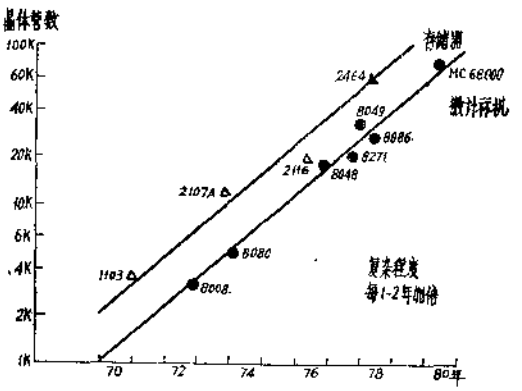


图1-3 线路复杂程度发展趋势

1.1.4 16位微处理器

微型计算机的另一个发展趋势，是向着高功能的高档机发展，与小型计算机直接竞争。16位的微处理器早已出现，如National

Semiconductor的IMP16，以后改为PAGE，由于本身缺陷及价格昂贵，一直未能发展。最近制出的8086、Z8000、MC68000是16位的高档微处理器。可在智能终端，小的商业系统、数据处理、科学计算、过程控制等方面应用。这些新的微处理器，可配大容量的内存，注意了多处理器系统工作时的通讯联络问题，有更快的取指令和存储器存取速度，有外接存储器管理和I/O控制片子，总之，具有高档小型机甚至大中型机的某些特点。估计几年之内可与小型机比美。目前软件问题是发展的主要障碍。16位机刚刚发展，目前还不能代替8位机。

1.2 8080微处理器

1.2.1 8080A CPU结构

Intel 8080微处理器是1973年问世的，以后，改进为增强型8080A CPU。8080A是8位微处理器，地址总线16位，数据总线8位，寻址范围为64K，能选择256个输入口和256个输出口，有78条基本指令，四种寻址方式(立即、直接、寄存器及间接)，采用双相非重叠时钟，时钟频率2 MHz，加法指令执行时间为2 μS。

8080A由四部分组成：寄存器阵列与地址控制逻辑，算术及逻辑运算单元，指令寄存器译码器及控制部分，总线缓冲器等，它们由内部总线连接起来，如图1-4所示。

（1）寄存器阵列与地址控制逻辑

寄存器阵列是一个读写存储器阵列，由六个16位的寄存器构成。其中可供程序员使用的寄存器有：

程序计数器(PC)，它指示程序中下一个要执行的指令的地址，每当取出指令以后，它就自动加1。

堆栈指示器(SP)，它保存下一个可供使用的存储器中栈顶单元的地址，数据压入堆栈后，(SP)－1，(SP)＋1后数据弹出堆栈。

六个通用寄存器的名字为，B，C，D，E，H，L它们与累加器A一起，可用来保存数据或地址，它们既可单个使用，也可作寄存器对(16位)使用，组成BC、DE、HL三对寄存器，用寄存器对的第一个字符代表寄存器对的名称，例如寄存器对B即表示BC。用寄存器对存放16位的数据或地址时，高8位存在B、D、H中，低8位存在C、E、L中。

由寄存器选择电路来确定6个8位寄存器中的一个，通过多路转换开关将选中的寄存器与内部数据总线连接，进行相互之间的8位数据的传送。

16位的地址锁存器，可从三个寄存器对中的任一对接收16位的数据，并由它送到16位的地址缓冲器，或送到16位的加1减1器上，加1减1器从地址锁存器接收了数据加1减1后，将数据送回到寄存器阵列。因此，16位的数据能在寄存器对、地址锁存器、加1减1器之间进行传送。从地址缓冲器输出到外部地址总线上。

暂存寄存器对WZ，不能由程序来选择，只在CPU内部执行指令时使用。

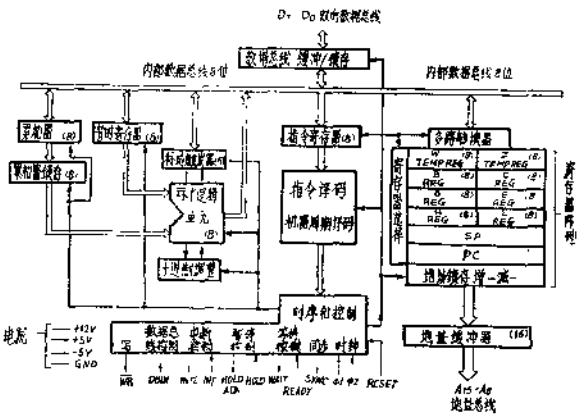


图1-4 8080ACPU功能框图

（2）算术及逻辑运算单元(ALU)

ALU包括下面一些单元：一个8位的累加器A，一个8位的累加锁存器(ACL)，一

个8位暂存寄存器(TMP)，一个8位的运算器(ALU)，一个5位的标志寄存器(F)及十进制调整线路。5个标志位是进位(CY)、零(Z)、符号(S)，辅助进位(AC)及奇偶(P)。

算术、逻辑、移位或循环等操作由ALU执行。ALU的运算数据由TMP、ACL及进位触发器输入，运算结果送内部数据总线或累加器A，同时也将运算结果的状态送到标志寄存器，以便作转移测试的条件。

累加器A从内部数据总线接收数据，并输出数据到ACL及内部数据总线。在进行十进制调整时，通过检测条件标志寄存器的进位(CY)，辅助进位(AC)，由ALU完成对累加器A上的数的十进制调整。

暂存寄存器TMP与内部数据总线交换信息。

(3) 指令寄存器译码器及控制部分

在取指令周期，指令的操作码从内部数据总线传送到指令寄存器，指令寄存器又将操作码送指令译码器。译码器的部分输出结合各种时序信号，用组合逻辑(寄存器选择与控制，地址总线控制电路)形成寄存器阵列所需要的控制信号。译码器的另一部分输出，结合时序信号，用组合逻辑(运算器控制电路)形成算术及逻辑运算单元所需要的控制信号。

时序及控制线路根据指令译码器的输出和从外部来的控制信号产生机器的状态及周期的时序信号并向外部输出控制信号或应答信号。

(4) 总线缓冲器

8位双向数据总线缓冲器用于隔离CPU内部数据总线和外部数据总线D_0-D_7。输出时，内部数据总线的内容，先送至锁存器，再送缓冲器。输入时，数据从外部数据总线送到内部数据总线。不传送数据时，缓冲器关闭。

1.2.2 8080A的机器周期和状态

CPU的工作是周期性的，它由时钟提供时序信号，使CPU按一定时间顺序进行取指令，取数据等操作。一条指令的取出和执行所需的时间称为"指令周期"，完成某一确定的操作(如取指令、读存储器、写存储器等)所需的时间称为机器周期。每个机器周期又分为3—5个状态。状态的时间宽度即时钟脉冲的间隔称为时钟周期，它是基本时间标准。

完成一条指令需要一到五个机器周期，机器周期用M_1、M_2、……M_5表示，而时钟周期(状态)则用T_1、T_2、……T_5表示。

一条指令需要多少机器周期，主要取决于CPU访问存储器或可寻址的外部设备的次数，访问次数愈多，需要机器周期数愈多。

8080A的每个机器周期对应于一个基本操作。共有10种基本操作。任何一条指令的第一个机器周期一定是"取指令"操作。十种基本操作由每个机器周期的第一个时钟周期期间发出的8位状态信息来区分，如表1-1所示。状态信息在SYNC(同步)信号期间(第一个时钟周期)出现在数据总线上。这个信息在整个机器周期内保存在外部锁存中，用它区分操作类型，产生对外部电路的控制信号。

除暂停、中断、保持三种基本操作外，一个机器周期可分为3～5个状态(T_1、T_2……T_5)。

T_1：用Φ_2的上跳沿产生同步信号SYNC，它表示每个机器周期的开始，并送外部电路作选通信号用。同步信号SYNC在T_2的Φ_2上跳沿时结束宽度为一个时钟周期。在T_2的开始，同步信号作用期间，在数据总线上出现本机器周期的状态信息码，并送到外部的锁存器中，以产生外电路的控制信号。同样，在T_1、Φ_2的上跳沿，地址总线接收程序计数器PC来的指令地址，这一信息一直保持到T_3。

表 1-1　8 位 周 期 信 息

数据总线位	周期信息	取指令	存储器读	存储器写	堆栈读	堆栈写	输入读	输出写	中断响应	暂停响应	暂停时中断响应
D_0	INTA	0	0	0	0	0	0	0	1	0	1
D_1	$\overline{WO}$	1	1	0	1	0	1	0	1	1	1
D_2	STACK	0	0	0	1	1	0	0	0	0	0
D_3	HLTA	0	0	0	0	0	0	0	0	1	1
D_4	OUT	0	0	0	0	0	0	1	0	0	0
D_5	MI_1	1	0	0	0	0	0	0	1	0	1
D_6	INP	0	0	0	0	0	1	0	0	0	0
D_7	MEMR	1	1	0	1	0	0	0	0	1	0
控制信息	通过8228产生的对应控制信息	MEMR	MEMR	MEMW	MEMR	MEMW	I/OR	I/OW	INTA	(NONE)	INTA
序　号	十种基本操作	1	2	8	4	5	6	7	8	9	10

T_2：为了使 CPU 能与慢 速存 储器 或 I/O 设备同步工作，当CPU向存储器或I/O发出地址信息后，就要 进行 查询。确认 有 无 READY和HOLD信号，并检 测是否为暂停指令。

T_W：如READY为低 电平，或已 检测到是执行暂停指令，则CPU 进入等待状态。此时，通过WAIT线输出高电平，说明CPU处于等待 状态，一 直到 READY 变为 高电平。

T_3：当READY 是高 电平，或者 没有 HOLD信号或暂停指令时，即进入 T_3状态。在此时钟周期内，将指令字节（当是“取指令”周期时）或数据字节（当是“存储器读”、“堆栈读”，或“输入读”周期时）或中断指令（当是“中断响应”周期时）从数据总线输入 CPU。当为“存储器写”、“堆栈写”、或“输入写”周期时，CPU 把数据字节输出到数据总 线上去。

T_4、T_5：如果执行一条特定指令，则需要有 T_4、T_5状态。如果 不需 要，CPU 可以跳过它们中的一个或两个。T_4与T_5仅用于CPU的内部操作。例如完成 对寄 存器对的内容加 1 后，又送回寄存器对中去。

并不是所有的机器周期都有这些状态，它取决于被执行的指令类型和在这个指令周期内具体的机器周期。CPU 的任 何一个 机器周期，只要其处理动作完了，即直接进行到下一个机器周期，而不需要每次都进入到 T_4及 T_5。

上述五个状态的时间图如图1-5所示。

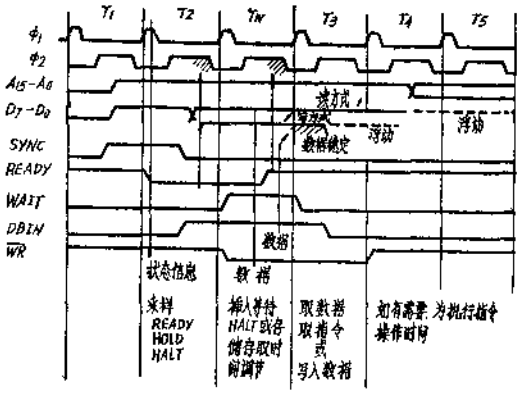


图 1-5　8080A CPU时序图

1.2.3　引脚功能

8080A有40条引脚，如图1-6所示，其中电源 及地 线占 4 条，即 + 5 V，－ 5 V，+12 V及地，另有输入输出线36条。每条线

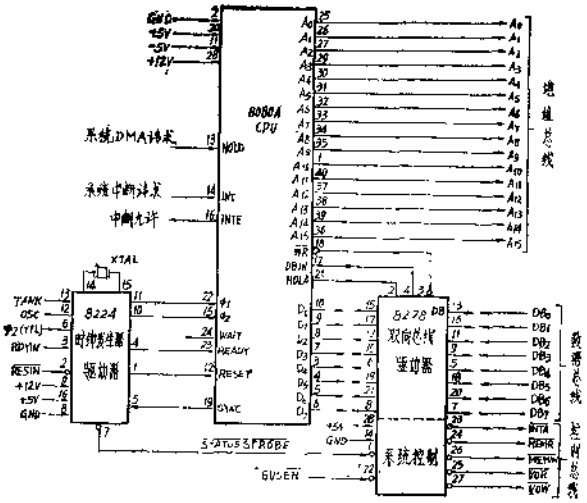


图 1-6　8080A CPU引脚图

的功能，说明如下：

A_{15}-A_0　地址总线(输出，三态、高为有效)提供64K字节的存储器地址（16位）或256个输入或输出设备的代码(8位)。

D_7-D_0　数据总线(输入输出、三态、高为有效)提供CPU与存储器和I/O设备之间进行指令与数据传送的双向通路在每个机器周期的第一个时钟周期SYNC期间，在数据总线上输出状态信息码。

SYNC　同步信号(输出，高为有效)表示一个机器周期开始的信息。

READY　准备就绪(输入，高为有效)

WAIT　等待(输出，高为有效)用来使CPU与慢速的存储器或I/O设备协调工作。当CPU送出一地址后，检测READY线，如是低电位，则自动插入一个等待状态T_W CPU进入等待状态，WAIT变高位，暂不读写；若READY为高位，则不等待。因此WAIT线可用来识别CPU是否处于"等待"状态。

HOLD　保持(输入，高为有效)

HLDA　保持响应(输出，高为有效)当外部设备使HOLD为高位时，则向CPU申请"保持"，CPU在T_3状态后，完成了对地址及数据总线的使用，即响应此"保持"申请，CPU的地址总线及数据总线则处于高阻态，同时，通过HLDA输出高位表示已响应"保持"的申请。在"保持"期间，地址总线与数据总线处于发出申请的那个外部设备的控制之下，进行与存储器的信息传送。当外部设备完成它的数据传送后，HOLD申请结束，HLDA恢复低位，CPU恢复到上一个已执行了的周期后面的一个机器周期。

RESET　复位（输入，高为有效）当此线为高位时，使程序计数器PC清零，但不清除其它寄存器，因此，复位后CPU从零单元起开始执行程序。

DBIN　数据总线允许输入(输出，高为有效)此信号送给外部电路，说明数据总线处于输入工作状态，用此信号来开放从存储器或I/O向CPU传送数据的门控电路。

$\overline{WR}$　写(输出，低为有效)此线为低位时CPU为写状态。

INTE　中断允许(输出，高为有效)当INTE为高位时，允许外部来的中断申请，一旦接受中断申请后，INTE则变低位，不许立即接受下一个中断。在复位或执行DI（关中断)指令后，此线为低位，禁止中断。当执行EI(开中断)指令后，INTE变高位，允许中断申请。

INT　中断申请(输入，高为有效)CPU在执行完一条指令后，或在暂停时，检查此线是否有中断申请。如果有申请，而中断允许又处于高位，则CPU响应中断。如果CPU处于HOLD状态或INTE为低位，则CPU不能响应中断申请。

1.2.4　中断

8080A CPU是这样来处理中断的：当外部设备发出一中断请求时，即把高电位输入到CPU的INT端，当CPU的中断允许端INTE为高电位时，在现行指令执行完以后，就可响应中断申请，完成下述工作：

（1）将INTE端变为低电位，禁止再次中断申请。

（2）由中断设备给出一条适应中断处理的特殊的单字节调用子程序指令，即重新启动指令RST。

（3）将PC中所存的(返回）地址压入堆栈，作为以后返回指令用的返回地址。

RST指令的格式如下：

I	I	N	N	N	I	I	I
7	6	5	4	3	2	1	0

其中NNN为000～111范围内的数，每一个数对应一个中断设备。在执行RST指令时，除将PC的内容压入堆栈外，且使PC的内容为0000000000NNN000。因此，给出了相应中断的服务子程序的入口地址，以保证CPU响应中断后能自动指向相应的服务子程序。显然，对应于每一中断服务子程序的首地址分别为十进制的0、8、16、24、32、40、48、56(十六进制的00、08、10、18、20、28、30、38)。对不同的中断设备可预先编码。这样，每一个子程序只有8个存储单元。如果8个单元不够用，可以在其中安排一条调用更长的子程序的指令来处理中断或转移到其它地址。

1.2.5 保持

微型机可以执行直接存储器存取(DMA)的操作。一般情况下是CPU控制全部数据的传送，数据一定要经过CPU。如果有大批数据须在外部设备与存储器间传送，如都经过CPU，时间很不经济。因而8080A配有直接存取(DMA)控制。当外部设备发出请求"保持"状态，CPU响应请求，使其HLDA(保持响应)端为高电平，CPU停止操作，退出对地址总线和数据总线的控制。在此期间，总线将由请求保持的设备控制使用，直接与存储器进行数据传送。当完成数据传送后，HOLD结束，CPU回到原执行的机器周期的下一个机器周期。

从CPU外部来看，一旦CPU响应保持，地址和数据总线变为高阻态，就停止操作。然而从CPU内部来看，CPU的有些功能仍可继续进行。如果一个HOLD申请在T_3被响应，而CPU又处在需要用T_4、T_5状态才完成的机器周期中，则CPU进入保持前，仍继续执行T_4、T_5状态。这样CPU的内部处理和外部的DMA传送相重叠，节约了CPU的时间。

1.2.6 暂停

当执行一条暂停指令时(HALT)，CPU在下个机器周期的T_2之后进入暂停状态(TWH)。8080A有三种方法脱离暂停状态。

（1）RESET(复位）端上输入一个高电平，使8080A恢复到T_1状态，同时，PC清零。CPU开始从"0"存储单元执行。

（2）HOLD状态的输入会使8080A进入保持状态。当HOLD转为低电平，8080A在下一个Φ_1时钟脉冲的上升沿重新进入暂停状态。

（3）中断响应将使8080A脱离暂停状态，并在下一个Φ_1时钟脉冲上升沿进入T_1，CPU执行由外部进入数据总线的指令。注

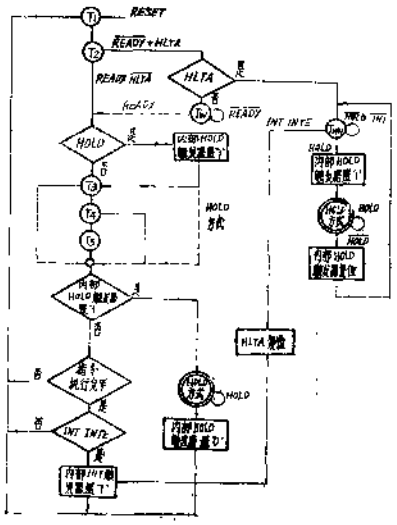
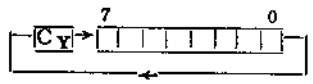
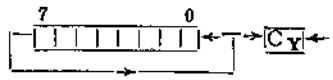
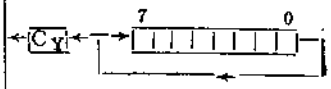


图1-7 状态转换图

127 指令系统简表

表 1-2 8080A 指 令 系 统 表

指令符号	字节	周期	功能
MOVr_1,r_2	1	1	$(r_1)\leftarrow(r_2)$
MOVM,r	1	2	$M\leftarrow(r)$
MOVr,M	1	2	$r\leftarrow(M)$
HLT		2	保 持
MVIr	2	2	$r\leftarrow(B_2)$
MVIM	2	3	$M\leftarrow(B_2)$
INRr	1	1	$r\leftarrow(r)+1$
DCRr	1	1	$r\leftarrow(r)-1$
INRM	1	3	$M\leftarrow(M)+1$
DCRM	1	3	$M\leftarrow(M)-1$
ADDr	1	1	$A\leftarrow(A)+r$
ADCr	1	1	$A\leftarrow(A)+(r)+(CY)$
SUBr	1	1	$A\leftarrow(A)-(r)$
SBBr	1	1	$A\leftarrow(A)-(r)-(CY)$
ANAr	1	1	$A\leftarrow(A)\wedge(r)$
XRAr	1	1	$A\leftarrow(A)\forall(r)$
ORAr	1	1	$A\leftarrow(A)\vee(r)$
CMPr	1	1	$(A)-(r)$ (A)不变
CMP M	1	2	$(A)-(M)$ 若(A)=(r),则Z=1；若(A)<(r),则C_Y=1
CPI	2	2	$(A)-(B_2)$
ADD M	1	2	$A\leftarrow(A)+(M)$
ADC M	1	2	$A\leftarrow(A)+(M)+(C_Y)$
SUB M	1	2	$A\leftarrow(A)-(M)$
SBB M	1	2	$A\leftarrow(A)-(M)-(C_Y)$
ANA M	1	2	$A\leftarrow(A)\wedge(M)$
XRA M	1	2	$A\leftarrow(A)\forall(M)$
ORA M	1	2	$A\leftarrow(A)\vee(M)$
ADI	2	2	$A\leftarrow(A)+(B_2)$
ACI	2	2	$A\leftarrow(A)+(B_2)+(C_Y)$
SUI	2	2	$A\leftarrow(A)-(B_2)$
SBI	2	2	$A\leftarrow(A)-(B_2)-(C_Y)$
ANI	2	2	$A\leftarrow(A)\wedge(B_2)$
XRI	2	2	$A\leftarrow(A)\forall(B_2)$
ORI	2	2	$A\leftarrow(A)\vee(B_2)$
RAL	1	1	$A_{m+1}\leftarrow A_m$,$A_0\leftarrow C_Y$,$C_Y\leftarrow A_7$  循环左移

指令符号	字节	周期	功能
RAR	1	1	$A_m\leftarrow A_{m+1}$,$C_Y\leftarrow A_0$,$A_7\leftarrow C_Y$ 
RLC	1	1	$A_{m+1}\leftarrow A_m$,$A_0\leftarrow A7$,$C_Y\leftarrow A_7$ 
RRC			$A_m\leftarrow A_{m+1}$,$A_7\leftarrow A_0$,$C_Y\leftarrow A_0$ 
JMP	3	3	$PC\leftarrow(B_3),(B_2)$
JC	3	3	若C=1则$PC\leftarrow(B_3),(B_2)$
JNC	3	3	若C=0则$PC\leftarrow(B_3),(B_2)$
JZ	3	3	若Z=1则$PC\leftarrow(B_3),(B_2)$
JNZ	3	3	若Z=0则$PC\leftarrow(B_3),(B_2)$
JP	3	3	若S=1则$PC\leftarrow(B_3),(B_2)$
JM	3	3	若S=0则$PC\leftarrow(B_3),(B_2)$
JPE	3	3	若P=1则$PC\leftarrow(B_3),(B_2)$
JPO	3	3	若P=0则$PC\leftarrow(B_3),(B_2)$
CALL	3	5	无条件调用子程序 $(SP-1),(SP-2)\leftarrow(PC),(SP)-2$ $PC\leftarrow(B_3),(B_2)$
CC	3	5/3	若C=1则调转
CNC	3	5/3	若C=0则调转
CZ	3	5/3	若Z=1则调转
CNZ	3	5/3	若Z=0则调转
CP	3	5/3	若S=1则调转
CM	3	5/3	若S=0则调转
CPE	3	5/3	若P=1则调转
CPO	3	5/3	若P=0则调转
RET	1	3	无条件返回主程序 $PC\leftarrow(SP),(SP-1)$
RC	1	3/1	若C=1则返回
RNC	1	3/1	若C=0则返回
RZ	1	3/1	若Z=1则返回
RNZ	1	3/1	若Z=0则返回
RP	1	3/1	若S=1则返回
RM	1	3/1	若S=0则返回
RPE	1	3/1	若P=1则返回
RPO	1	3/1	若P=0则返回

此八条为有条件调子程序

此八条为有条件返回主程序

指令符号	字节	周期	功能	指令符号	字节	周期	功能
RST	1	8	再起动指令	DAD D	1	8	HL←(H)(L)+(D)(E)
			(SP−1),(SP−2)←(PC)	DAD H	1	8	HL←(H)(L)+(H)(L)
			PC←000000000AAA000	DAD SP	1	8	HL←(H)(L)+(SP)
IN	2	3	A←(B_2)	STAX B	1	2	[(B)(C)]←(A)
OUT	2	3	(B_2)←(A)	STAX D	1	2	[(D)(E)]←(A)
LXI B	3	3	C←(B_2),B←(B_3)	LDAX D	1	2	A←[(D)(E)]
LXI D	3	3	E←(B_2),D←(B_3)	LDAX B	1	2	A←[(B)(C)]
LXI H	3	3	L←(B_2),H←(B_3)	INX B	1	1	BC←(B)(C)+1
LXI SP	3	3	SP_L←(B_2),SP_H←(B_3)	INX D	1	1	DE←(D)(E)+1
PUSH B	1	3	(SP−1)←(B),(SP−2)←(C)	INX H	1	1	HL←(H)(L)+1
PUSH D	1	3	(SP−1)←(D),(SP−2)←(E)	INX SP	1	1	SP←(SP)+1
PUSH H	1	3	(SP−1)←(H),(SP−2)←(L)	DCX B	1	1	BC←(B)(C)−1
PUSH PSW	1	3	条件标志寄存器与累加器A组成一对寄存器	DCX D	1	1	DE←(D)(E)−1
				DCX H	1	1	HL←(H)(L)−1
			(SP−1)←条件标志(SP−2)←(A)	DCX SP	1	1	SP←(SP)−1
POB B	1	3	C←(SP),B←(SP+1)	CMA	1	1	A←$(\overline{A})$
POB D	1	3	E←(SP),D←(SP+1)	STC	1	1	CY←"1"
POB H	1	3	L←(SP),H←(SP+1)	CMC	1	1	CY←$(\overline{CY})$
POB PSW	1	3	A←(SP)	DAA	1	1	十进制调整
			标志寄存器←(SP+1)	SHLD	3	5	[(B_2)(B_3)]←(L)
SDA	3	4	(B_2),(B_3)←(A)				[(B_2)(B_3)+1]←(H)
LDA	3	4	A←(B_2),(B_3)	LHLD	3	5	L←[(B_2)(B_3)]
XCHG	1	1	(H)⇄(D),(E)⇄(L)				H←[(B_2)(B_3)+1]
XTHG	1	5	(L)⇄(SP),(H)⇄(SP+1)	EI	1	1	允许中断
SPHL	1	1	SP⇄(H),(L)	DI	1	1	屏蔽中断
PCHL	1	1	PC⇄(H),(H)	NOP	1	1	空操作
DAD B	1	3	HL←(H)(L)+(B)(C)				

意，当进入暂停状态，允许中断（INTE）标志必须置"1"，否则8080A不能响应中断，要脱离暂停状态就只有靠RESET信号。

整个CPU的状态转换可以用 CPU 的状态转换图来说明。见图1-7。

1.2.8 MCS-80 微计算机系统

图1-8是一个典型的用8080A组成的微计算机系统。它由三部分组成：中央处理器单元(CPU Module)、存储器单元和I/O单元。每个单元包括几个片子。通过8位数据总线、16位地址总线和6位控制总线把三个单元连接起来。

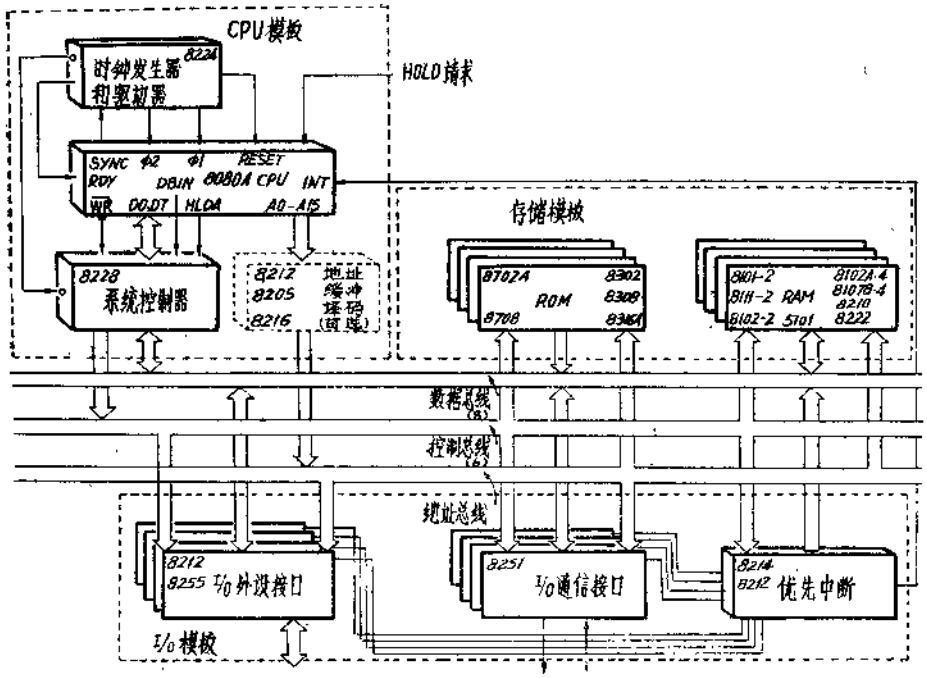


图 1-8　微计算机系统

1.3　Z-80微处理器

Z-80微处理器是Zilog公司1974年研制的8位微处理器。Z-80的主要设计人员参加过Intel8080的设计工作，因此Z-80基本上保持了8080的特点。但它比8080指令更多，速度更快并有许多改进。

1.3.1　Z-80特点

Z-80与8080虽有许多相似之处，如指令系统包括8080的78条指令，寄存器结构相似，但Z-80与8080相比，还有以下一些特点：

（1）N沟道硅栅E/D MOS微处理器

Z-80采用了N沟道E/D MOS工艺，集成度高、速度快、功耗低。例如，8080A的集成度是88个门/mm²，而Z-80则提高到133个门/mm²。

（2）单一电源

8080A的电源需要三种电压，±5V、+12V，而Z-80只需要一个电压+5V。

（3）单相时钟

8080A使用双相时钟，须用8224片作为时钟振荡器，而Z-80只须单相时钟，用简单的TTL电路就能实现，时钟频率提高为4MHz。

（4）控制总线

Z-80有13条独立的控制总线，直接产生控制信号，而8080的系统控制状态，是由数据总线上的状态信息码，经系统控制器8228片译码产生。Z-80的读、写控制信号原则也有所改变，它使用一个公用的读、写信号，然后用存储器请求或I/O请求信号来区分是存储器读、写，还是I/O读、写。

（5）动态存储器自动再生

Z-80CPU内部有7位再生计数器，利用取指令周期中间的空隙，自动进行动态RAM的再生，不必另设再生电路。

（6）中断