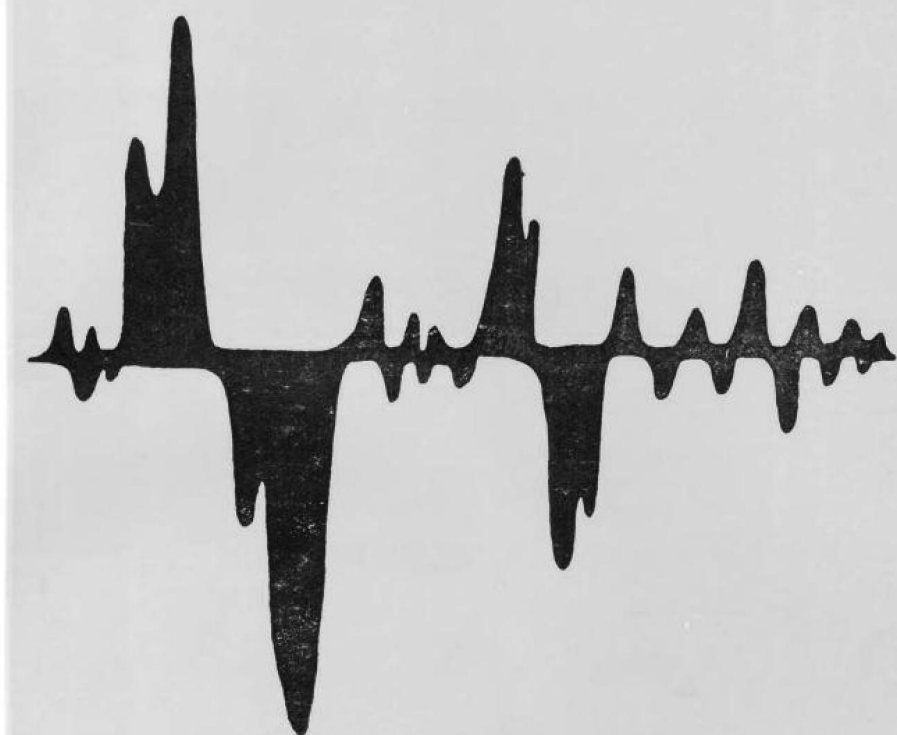




电机电工的微机测试

DIANJI DIANGONG DE
WEIJI CESHU

● 任仲岳 主编

● 上海交通大学出版社

电机电工的微机测试

任仲岳 楼永乐 编著

上海交通大学出版社

内 容 简 介

电工和电机的测试,通常采用传统的电压、电流等指示仪表,由人工目测读数记录,然后计算出性能参数。这种方法生产效率低,劳动强度高。

微型计算机的问世,使电工和电机测试技术发生着深刻的变革。微型计算机可对电压、电流、转矩、转速等物理量进行采样和计算,得出电机的各种参数和性能,从而大大减轻了劳动强度,提高了测试精度和测试效率,使劳动生产率得到成倍增长,测试数据和计算结果能自动打印,克服和消除了人为因素造成的误差。因此,微型计算机技术的应用,对电工和电机测试技术的发展,具有重大的现实意义。

本书主要介绍采用微型计算机进行电机检查试验和型式试验的测试技术。其中包括:微型计算机的基本原理;前置部件的原理和结构;微型计算机测试原理;微型计算机测试电机的软件编制和电机参数测试实例等。

本书可作高等院校电机、电工类专业高年级学生研究生的教材或教学参考书,可供电气工程技术人员及有关读者阅读。

电机电工的微机测试

任仲岳 楼永乐 编著

上海交通大学出版社出版

(淮海中路1984弄19号)

新华书店上海发行所发行

常熟文化印刷厂排版印装

开本 787×1092 毫米 1/16 印张 30.5 字数 755,000

1986年5月第1版 1986年7月第一次印刷

印数 1—30,000

统一书号 15324·103 科技书目: 131—286

定价: 5.00 元

前 言

微型计算机的问世,使国民经济的各个部门发生着巨大的变革。在电机和电工领域的测试技术方面,也同样冲击着古老、传统、落后的测试方法。

电机在出厂时或设计定型时,需要进行项目众多的试验,以获取电机的性能参数。目前,我国大多数厂家的电机试验还是沿用几十年来的手工操作、目测读数的方法,不仅劳动强度大,工作效率低,而且测试误差也较大。人工测试电机参数,无论从数量上,还是从质量上讲,都与电机工业的迅速发展不相适应。要改善劳动强度,提高测试效率,保证测试精度,必须寻找一种可靠性好、效率高的电机测试新方法。用微型计算机来测试电机的参数,就是属于这种新方法。

微型计算机测试电机的优点在于:

1. 精度高 普通测试仪器一般为 0.5 级和 0.2 级,在操作中还会有人为误差。微型计算机配上 8 位 A/D 接口测试模拟电压信号,在满量程时,精度达 0.4%;如使用 12 位 A/D 接口,精度可达 0.025%。
2. 快速 微机能在微秒级到毫秒级的短暂瞬间完成对电机参数的测量。
3. 可靠 微机进行大量测试不易出错,重复性好。
4. 能实现自动化作业,自动输出打印,也可装配在检验流水线上,自动完成测试工作。

在国外,用微型计算机进行电机试验,实现自动测试和数据处理的作法日益增加。如西德的西门子公司为慕尼黑大学电机试验室制造的 300 型过程控制计算机,在电机试验中就大大简化各种参数的测量。日本国际检测器公司生产的 MDP 101、102 型电机性能综合测试机,可以自动测试电压、电流、转矩、转速、功率、效率、功率因数等 12 个项目,并由微机进行快速数据处理。法国的 CEM 公司也试制了自动测试台,主要适用于 0.5 kW—800 kW 三相异步电机的测试。

在国内,也有一些单位开始把微型计算机用于电机测试。但从所接触到的资料看,目前尚无比较完善和成熟的微型计算机测试电机系统。

作者早在一九八〇年开始探索进行这方面的研究。先后为上海交通大学电机专业研究生、高年级学生讲授了电机的微型计算机测试技术课程,同时,为哈尔滨电工学院、沈阳机电学院电机专业的研究生和教师以及为苏州市科委组织的 120 位工程师进修班学员、中国电工技术学会船舶电工专业委员会、微特电机专业委员会等短训班上讲授了该课程。并编写了《微型计算机原理及其在电机测试中的应用》一书作为教材。

作者在一九八三年完成了异步电动机 $M-s$ 特性的微机测试课题的多种方案后,承接了上海微型电机厂、革新电机厂、苏州电机厂、南洋电机厂、无锡县起重机厂、无锡电机厂、汕头微电机厂、福州发电设备厂、咸宁电机厂、南京电机厂、平湖电机厂、镇江电机厂、鹤山电机厂提出的异步电动机、微特电机、直流电机、同步发电机的微型计算机测试台的研制任务。有的是在流水线上测试检查试验项目,有的是批量测试检查试验项目,有的是型式试验。

本书是在上述基础上,把电机、电工的微型计算机测试原理、结构、实例、软件等编写成书,

以促进我国电机、电工领域微型计算机测试技术的推广,改进过去落后的测试技术。

由于工厂测试的实际情况,要求在满足国家标准规定精度的前提下,测试装置必须简单、实用、可靠,而不能一味要求精度的无限提高;装置不允许复杂、庞大,操作力求简便。因此,根据我们的经验,采用八位单板微型计算机及其系统已绰绰有余。

电机、电工的微机测试,可以使用多种微型计算机。本书以八位单板微型计算机及其系统为主进行介绍,如 CJ-801、MIC-85、SYS-8 等机种,没有涉及更高档的微型计算机系统,以求价廉物美,简单实用。对于其它机型的微型计算机,同样可以依据本书介绍的原理,设计相应的测试系统。

本书的主要目的是在电机、电工领域推广微型计算机测试技术,面向广大电机、电工厂,实用为主,以俾广大工厂、研究所中专以上程度的科技人员都能看懂、掌握、应用,也可为大专院校高年级学生及研究生的选修课教材或参考书。

本书将以电机测试项目为例进行逐项介绍,因为电机的测试项目齐全,它几乎包括了所有电工产品的测试要求,对几乎所有电工产品的测试都能适用和具有参考价值。

本书第一、二、三章简要介绍 CJ-801、MIC-85、SYS-8 单板微型计算机系统的基本原理和指令系统。第四章介绍电机测试项目的微型计算机测试方法、原理,以及前置部分元器件的功能、作用原理和结构。第五章介绍测试程序、框图和原理。第六章介绍一些实例。

本书在写作过程中,曾得到周俭、伍建青同志提供的部分素材,全书由哈尔滨工业大学王宗培教授审稿,特此铭谢。

作 者

一九八五年二月 于上海交通大学

目 录

第一章 CJ-801 单板微型计算机.....	(1)
§ 1-1 简介及主要技术特性	(1)
§ 1-2 Z80 CPU 的结构.....	(2)
一、CPU 在微机系统中的作用.....	(3)
二、Z80 CPU 的结构.....	(3)
§ 1-3 Z80 CPU 的指令系统	(10)
一、指令和数据格式.....	(10)
二、寻址方式.....	(11)
三、指令系统分类.....	(12)
四、条件标志.....	(14)
§ 1-4 并行输入/输出接口芯片 Z80 PIO.....	(15)
一、PIO 的结构.....	(16)
二、PIO 的操作说明.....	(19)
三、PIO 的使用.....	(25)
§ 1-5 定时器/计数器芯片 Z80 CTC	(26)
一、CTC 的结构	(27)
二、CTC 的操作说明	(29)
三、CTC 的硬件连接	(32)
第二章 MIC-85 单板微型计算机	(33)
§ 2-1 Intel 8085 A 单片 8 位 NMOS 微处理器	(33)
一、Intel 8085 A 的总体结构	(33)
二、Intel 8085 A 的功能和特性	(37)
三、Intel 8085 A 的引脚功能	(39)
§ 2-2 MIC-85 单板微型计算机的结构.....	(41)
一、系统的总线结构.....	(41)
二、存储器.....	(41)
三、I/O 接口.....	(42)
四、MIC-85 单板机的特点.....	(43)
五、Intel 8085 A 指令系统简介	(44)
§ 2-3 可编程序并行 I/O 接口 8255	(46)
一、8255 的结构	(46)
二、8255 的引脚功能	(47)
§ 2-4 可编程序串行通信接口 8251	(55)
一、8251 简介	(55)

二、8251 的结构	(57)
三、8251 的引脚功能	(57)
§ 2-5 带有 I/O 接口和计时器的静态 RAM 8155	(62)
一、8155 的结构	(62)
二、8155 的引脚功能	(62)
三、8155 的工作原理	(63)
第三章 SYS-8 微型计算机系统	(68)
§ 3-1 概述	(68)
一、系统的主要技术特性	(68)
二、系统的设计特点	(69)
三、系统的结构	(69)
§ 3-2 SYS-8 微型计算机系统的主机使用	(71)
一、进入 BASIC 状态概述	(71)
二、NTBUG 状态下命令的基本使用	(72)
三、EPROM 编程步骤	(73)
§ 3-3 CP/M 操作系统简介	(73)
一、CP/M 操作系统概况	(73)
二、CP/M 操作系统的命令	(74)
三、软盘与 CP/M 操作系统的使用	(74)
第四章 微型计算机测试原理及测试系统的前置电路	(86)
§ 4-1 型式试验和出厂试验项目	(86)
一、异步电机	(86)
二、同步电机	(86)
三、直流电机	(87)
四、驱动微电机	(87)
五、控制微电机	(88)
§ 4-2 主要测试项目的测试原理和测试方法	(89)
一、绕组在实际冷状态下直流电阻的测定	(90)
二、绕组对机壳及其相互间绝缘电阻的测定	(90)
三、空载试验	(91)
四、短路(堵转)试验	(92)
五、转子开路电压测定	(92)
六、M-s 特性测定	(92)
七、低压起动、超压、超速试验	(93)
八、绕组对机壳及其相互间绝缘介电强度的试验	(93)
九、效率、功率因数和转差率测定	(93)
十、发电机的空载特性试验	(94)
十一、发电机的短路特性试验	(95)
十二、温升试验	(95)

§ 4-3 前置部分元器件原理和结构	(96)
一、恒流源	(96)
二、500 V 整流电源	(98)
三、放大、整流电路	(98)
四、交-直流电量变送器	(100)
五、采样系统	(112)
六、模/数转换器	(118)
七、前置电路的数字部分	(122)
第五章 微型计算机测试电机的软件编制	(126)
§ 5-1 汇编语言的程序设计	(126)
一、循环程序设计	(126)
二、算术运算程序	(132)
三、字符处理	(139)
四、子程序	(140)
五、其它应用程序	(143)
§ 5-2 采样电路和采样程序	(151)
一、A/D 转换电路	(151)
二、微机控制 ADC 0808	(153)
三、采样电路	(155)
四、采样程序	(155)
§ 5-3 直流量的测量	(158)
§ 5-4 交流量的测量和有效值的计算	(163)
一、计算交流量有效值的基本公式	(163)
二、测量交流电压信号有效值的程序	(165)
三、各子程序介绍	(167)
§ 5-5 交流功率的计算	(182)
一、计算交流电路平均功率的方法	(182)
二、计算交流平均功率的 CALPOW 程序	(184)
三、SEARCH 子程序	(188)
四、计算实例	(191)
五、用动态法测试堵转功率和堵转电流	(193)
六、计算电流转换系数 K_i 和功率转换系数 K_p	(195)
§ 5-6 微型计算机系统对采样信号的分析处理	(199)
一、数据采样	(199)
二、数据分析	(202)
三、实例	(204)
四、电机测试常用的 CBASIC 语句	(205)
§ 5-7 打印输出程序	(206)
一、温度显示程序 TEMP	(207)

二、多字节十六进制数转化为二十进制数程序 BCD	(210)
三、电机测试打印程序 PRN	(216)
第六章 电机测试实例	(224)
§ 6-1 锥形转子异步电动机出厂试验的批量测试	(224)
一、试验项目及测试装置	(224)
二、定子绕组冷态直流电阻的测试	(227)
三、测试程序流程图	(237)
§ 6-2 在校验流水线上的异步电动机出厂试验	(240)
一、电机试验项目	(240)
二、电机测试流程图	(241)
三、12 位 A/D 转换器的工作原理	(247)
四、12 位二进制数数据处理 PUI 12 程序	(248)
五、异步电机出厂试验流水线微机测试 JVL-I 程序	(269)
§ 6-3 分马力电机出厂试验流水线	(275)
一、测试系统简解	(275)
二、前置接口线路	(276)
三、软件程序	(278)
§ 6-4 异步电动机 M-s 特性测试	(279)
一、PY1A 测试系统	(280)
二、M-s 特性的全自动测试	(304)
三、动态微分法	(308)
附 录	(313)
附录 1 Z80 指令的寻址方式和操作码	(313)
附录 2 Z80 指令系统的功能表	(325)
附录 3 Intel 8080 A/8085 A 十六进制指令码	(336)
附录 4 Intel 8080 A/8085 A 和 Z80 微处理器通用的基本指令表	(337)
附录 5 ZC-42 兆欧表	(340)
附录 6 批量测试总程序 JVP-I	(341)
附录 7 异步电机出厂试验流水线测试总程序 JVL-I	(377)
附录 8 分马力电机流水线测试总程序 JVL-II	(430)
附录 9 电机测试常用程序索引	(477)

第一章 CJ-801 单板微型计算机

本章主要介绍 CJ-801 单板微型计算机(下简称单板机)的结构原理和指令系统。由于这方面的专门书籍较多,本书对此只作一些最基本的介绍,详细内容请参阅有关专门书籍。

§ 1-1 简介及主要技术特性

CJ-801 单板机是使用 Z80 系列器件,组装在一块印刷电路板上的完整的微型计算机(下简称微机)。它结构简单,布局合理,功能齐全,用途广泛。

CJ-801 单板机可作为“智能”部件,用于生产过程控制、试验检测、各种仪器仪表的单机控制、数据处理等。

CJ-801 单板机具有简易的开发功能,如在用户程序内可设置多至五个断点,可单步执行存于 RAM 或 PROM 中的程序,可对 2716/2758 EPROM 进行编程等,因而可以用作调试样机。

CJ-801 作为 CJ-80 系列的基础部件,在 S-100 总线插座上附加元、器件扩展板后,内存容量将增大 16 K,可使用 BASIC 语言,如果再增配字符键盘、显示器(或以黑白电视机代用)、微型打印机等外围设备,就可构成一种经济实用、功能更为完善的简易微机系统。

CJ-801 单板机的主要技术特性有:

1. 中央处理单元为 Z80 CPU。
2. 时钟(ϕ)频率为 1.9968 MHz,便于使用 8080 A 的接口电路。晶体振荡器的频率为 8.9936 MHz。
3. RAM 为 4 K 字节的 2114 静态读写存储器。
4. ROM 为 2 K 字节,编入监控程序 CJBUG。
5. 两个 PROM 插座,可插入 4 K 字节的 PROM 或 EPROM。
6. 一个 Z80 PIO 并行 I/O 接口芯片,有两个 8 位可编程的 I/O 口,均可供用户使用。
7. 一个 Z80 CTC 计数器/定时器芯片,它有四个通道:通道 0 供用户使用,其余由 CJBUG 使用。
8. 按键有 28 个,其中 16 个为十六进制数字键;12 个为命令键,具有 20 种功能。这 12 个命令键是:

MON' (换上档的监控)键

MON (换下档的监控)键

STEP (单步执行程序)键

EXEC (连续执行程序)键

TROM/MEM (EPROM 预检功能/存储单元检查)键

REG'/REG (辅助寄存器检查/寄存器检查)键

2 FBE/LAST (用户定义功能/查上一个存储单元)键

2 FBC/NEXT (用户定义功能/增量检查)键

PROM/SI(EPROM 写入/单指令调试)键

DELETE/INSERT (删除指令/插入指令)键

DUMP/PORT(信息转贮磁带/口检查)键

LOAD/BP (磁带输入/设置断点)键

9. 显示器有 6 位 LED 数字显示,通常左 4 位显示地址,右 2 位显示数据。

10. 配有音频磁带录音机接口。

11. 布线区为 2.5×7 英寸。

12. S-100 总线插脚。

13. 电源为 $+5V \pm 5\%$ 、1A。可增配 7805 稳压器件,这时输入须接 $\geq +7V$ 直流电源。若要对 EPROM 进行写入,尚须接入 $+25V \pm 1V$ 、30mA 的电源。

CJ-801 单板机的 Z80 CPU 指令系统有 158 条指令,除了包含 8080 A 的全部指令外,还增加了数据块传送与查找、位操作(置 1、置 0、测试)、相对转移、变址寻址、16 位的算术运算等指令。Z80 CPU 的内部寄存器也较 8080 A 多,增设了 IX、IY,以及一套辅助寄存器。这些特点大大增强了 Z80 CPU 的处理功能,并便于使设计的 Z80 软件与 8080 A 的软件相容。

CJ-801 单板机使用八个廉价的静态读写存贮器 2114,容量为 4K 字节。板上配的 2K 字节的 ROM,存放监控程序 CJBUG,便于用户利用键盘或磁带录音机输入程序和数据,以及提供其它的软件功能。板上设有一个完整的 EPROM 编程器,允许用户将信息写入 2716/2758 EPROM 中。编程所需电压为直流 $+25V \pm 1V$,最大电流为 30mA。

CJ-801 单板机配有磁带录音机接口。用转录线将磁带录音机与机上插口相连,就可以方便地存取数据和程序。单板机上红色发光二极管能指示磁带上有无信息。利用这只二极管的指示,可在一盘磁带上录制几个文件。RAM 和磁带之间的存取由命令键控制。板上配有 6 位数码显示,供显示地址和数据。通常左 4 位显示地址,右 2 位显示数据(对于 PG、SP、IX、IY 等 16 位寄存器而言,右 4 位显示数据)。

CJ-801 单板机配有三个按钮开关;按钮 S_1 用来对整机提供 RESET (复位)信号。开关 S_2 有两个位置可供选择,如果置向 MONRST 位置,则在复位后,显示器上出现“—”,并扫描键盘的输入;如取指向 PRGM₁ RST 位置,则在初始化后进入 PROM₁ 中的用户程序。开关 S_3 用来选择 PROM₂ 中的 EPROM 处于“写入(PGM)”或“读出(READ)”状态。

此外,在 CJ-801 单板机面板的左上方,有 S-100 总线的插脚。Z80 CPU 的地址、数据、控制总线都接到 S-100 总线,以便在插座上扩充存贮器板及 I/O 接口板。面板的左方有 2.5×7 英寸的布线区,约可装 25 片 IC (集成电路)。连接到布线区的信号有:

Z80 CPU 的数据、地址和控制信息。

Z80 PIO 两个口(port)的数据和联络信息。

Z80 CTC 通道 0 的输入和输出。

§ 1-2 Z80 CPU 的结构

CPU 是微机的中央处理单元 (Central Processing Unit),它是微机的中枢,担负着处理数据和控制整个微机系统的使命。根据厂家所制定的指令系统,在使用 Z80 CPU 时,用户可

以直接编写程序。

为了研究 Z80 CPU 的结构,我们需要建立它的程序模型,从编写程序的角度来观察组成 CPU 的各部分。为了掌握 Z80 程序的编写方法,我们将指令系统视为 CPU 的传递函数,逐条理解执行每条指令时在 CPU 硬件结构中所引起的变化。

一、CPU 在微机系统中的作用

因为 CPU 的结构和指令系统是根据对 CPU 功能的要求来设计的,所以在具体介绍 Z80 CPU 之前,有必要先来考察它在整个微机系统中的作用。

如图 1-1 所示的微机系统,除 CPU 外,还有存储器 and 输入/输出器件 (I/O device),都由 $\overline{\text{CPU}}$ 表示。CPU 的功能可以概括为对数据的读写 (CPU 与 $\overline{\text{CPU}}$ 之间)、运算 (CPU 内部)、传送 (CPU 内部以及与 $\overline{\text{CPU}}$ 之间),以及 CPU 对它本身及对 $\overline{\text{CPU}}$ 的控制。所有上述四种功能,都是通过对于数据字或控制字的传送和处理来实现的。这些字的每 8 位 (bit) 字长叫做一字节 (byte),每 4 位字叫做半字节 (nibble)。

在 CPU 内部的数据传送主要是通过软件设计 (程序编写) 来操纵的,在 CPU 和 $\overline{\text{CPU}}$ 之间的数据传送主要是通过硬件设计 (电路接口) 来实现的。

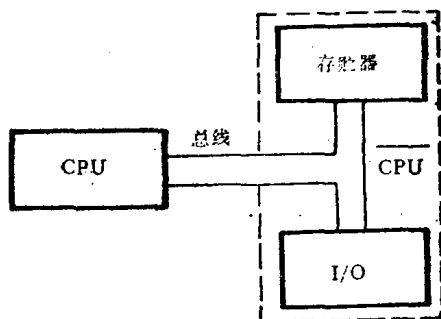


图 1-1 微机系统的简化框图

二、Z80 CPU 的结构

Z80 CPU 的结构如图 1-2 所示,主要由寄存器、运算器、指令译码及定时和控制几部分组成。

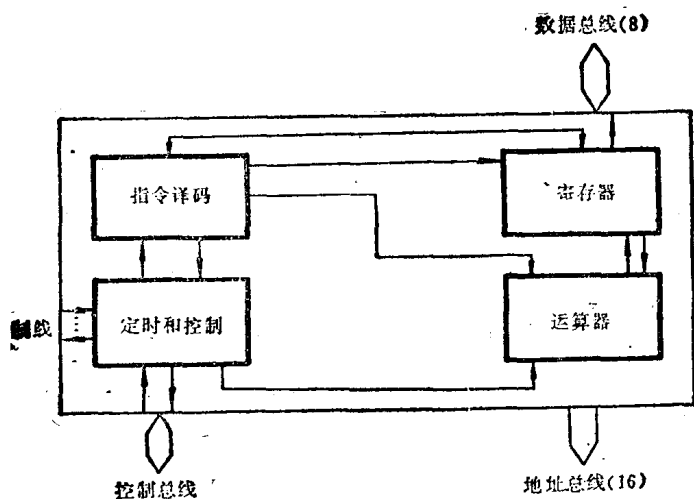


图 1-2 Z80 CPU 的简化框图

1. 寄存器

在 CPU 中,寄存器实质上是一个存储器,它由一定数目的二态存储单元组成,本身具有一定的地址以便加以识别。

在 Z80 CPU 中,包含有 208 位可供用户使用的寄存器,全部用静态 RAM 实现。它们分为专用和通用两类,其程序模型如图 1-3 所示。

Z80 CPU 中的专用寄存器包括:

(1) 程序计数器 (Program Counter, 简称 PC)

程序计数器用来存放现行指令的 16 位地址。每当指令取出后,它就自动加 1 (除转移指令外),以便指向下一条指令的存储地址。在执行转移指令时,为了改变程序的执行次序,处理器要把新指令的地址 (或入口地址) 自动置入程序计数器,取代已递增的数值,即将程序计数器

的内容指向子程序的首地址。子程序执行完时,由一条返回指令,使控制返回主程序。}

(2) 堆栈指示器(Stack Pointer, 简称 SP)

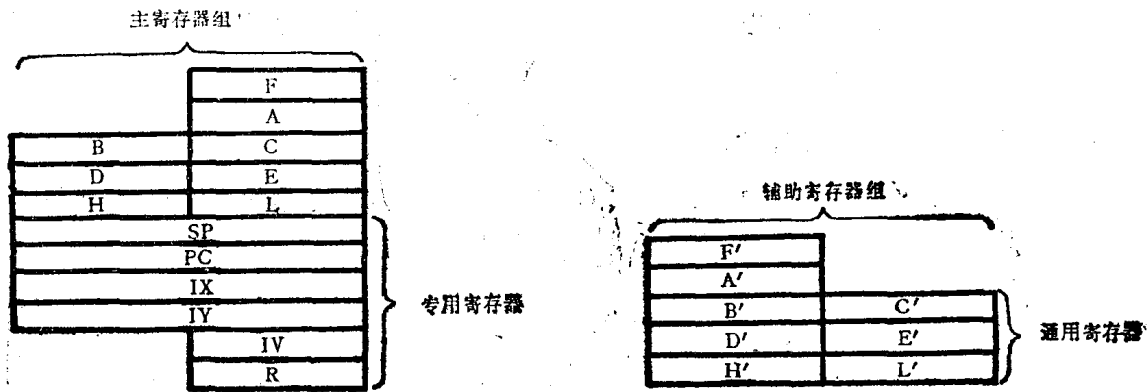


图 1-3 Z80 CPU 的寄存器

用于贮存某一指向特定存贮单元地址的寄存器叫做指示器。指示器的内容,即所贮存的地 址叫做指针。

堆栈是一种暂存数据(或地址)的存贮区,组成堆栈的一些存贮单元,象一沓碟子,有规律地排列着。最先进栈的数据字构成栈底,堆栈的另一端为栈顶(Top of Stack, 简称 TOS)。对于用户来说,只有最后进栈的一个数据字,即处于栈顶的一个数据字才能被存取(见图 1-4)。

处理器设有堆栈指令,能把累加器和其它通用寄存器的内容压入堆栈;或从堆栈中弹出,把数据送回累加器和寄存器。故用堆栈可以简单地实现多级中断的现场保护与恢复,子程序多重嵌套和简化数据操作。当执行转子指令时,下一条指令的地址(即返回地址——PC 的内容)被压入栈中,程序计数器(PC)的内容被调用的子程序的首址所代替。子程序结束时,返主指令取出上述压入栈中的返回地址,并把它送回指令计数器。

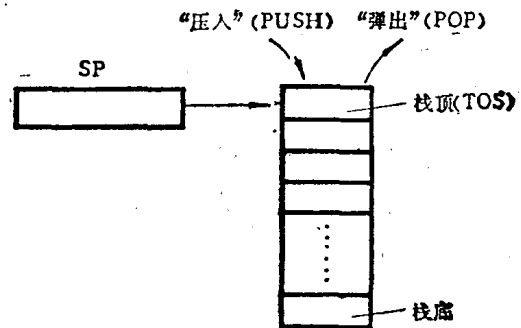


图 1-4 堆栈与栈操作

从软件的角度看,堆栈是一种数据结构。其中各元素的内容和数目都在动态地变化着,但变化方式不是随意的,而是先进后出(First In Last Out, 简称 FILO)的。所有的元素进栈或出栈都要通过栈顶。中间的元素不能跳出这个序列。因此堆栈也叫做下推表。

在堆栈操作中,堆栈中的各元素实际上并未变动。唯一的变化发生在堆栈指示器 SP 中。SP 是一个专用 16 位寄存器,用于贮存栈顶地址。当一个数据字进栈时,需将 SP 增 1 (或减 1)。栈顶上升,数据字存放在 SP 增量后指向的新栈顶。这种操作叫做“压入”(PUSH)。如果要 从栈中取出数据,则最先取出已经处于栈顶的数据字,然后将 SP 减 1 (或增 1),降下栈顶,依此类推。这种操作叫做“弹出”(POP)。

(3) 变址寄存器(Index Register)

在 Z80 CPU 中,设有两个完全相同的变址寄存器 IX 和 IY。这是一种贮存 16 位地址的寄存器。其内容不仅可以在程序控制下循环增 1 或减 1,而且能与指令包含的一个操作数(叫

做偏移, offset) 相加, 形成一个新的有效地址, 指向所要访问的单元(见图 1-5)。在处理数组和表时, 使用这种寄存器尤为便利。

(4) 中断矢量寄存器(Interrupt Vector register, 简称 IV 或 I)

使微机暂停正常程序流程, 接受输入信号的请求去执行一定的服务子程序(称为中断服务程序), 然后再返回原来的程序流程, 这种工作方式叫做中断(见图 1-6)。

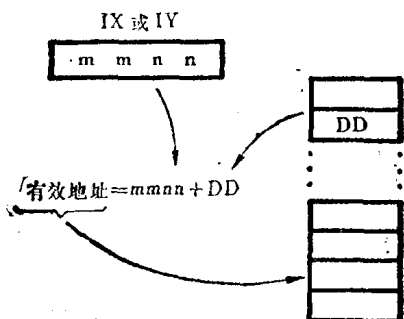


图 1-5 变址操作示意图

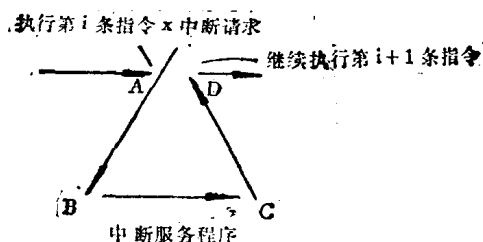


图 1-6 中断操作示意图

为了能够准确返回原来的程序流程, 需要自动将中断前时刻各 CPU 寄存器的内容保护起来, 称为保护现场; 而在执行完中断服务程序之后再恢复这一现场。此外, 为了使 CPU 在响应中断后能自动转向中断服务程序, 需要形成中断服务程序的入口地址, 也叫做中断矢量。Z80 CPU 为用户提供了形成中断矢量的三种方式: 方式 0、方式 1 和方式 2。在方式 2 中, 借助于 8 位的中断矢量寄存器 IV 来形成中断矢量是最完备的一种。如图 1-7 所示, 当 Z80 CPU 工作在这种方式时, 需要制定一个 16 位中断矢量地址表。这个表可以位于可寻址的存储器中的任何部位。其中各登记项所组成的 16 位地址用来标识各中断服务程序中第一条可执行的指令, 即入口地址。当 CPU 在方式 2 的条件下响应中断时, 请求中断的外部设备应将一个中断矢量的低 8 位放在数据总线上。Z80 CPU 将 IV 的内容与这个矢量结合起来, 形成一个 16 位地址。用这个地址, 就能够访问中断地址矢量表, 进入所需要的中断服务程序。

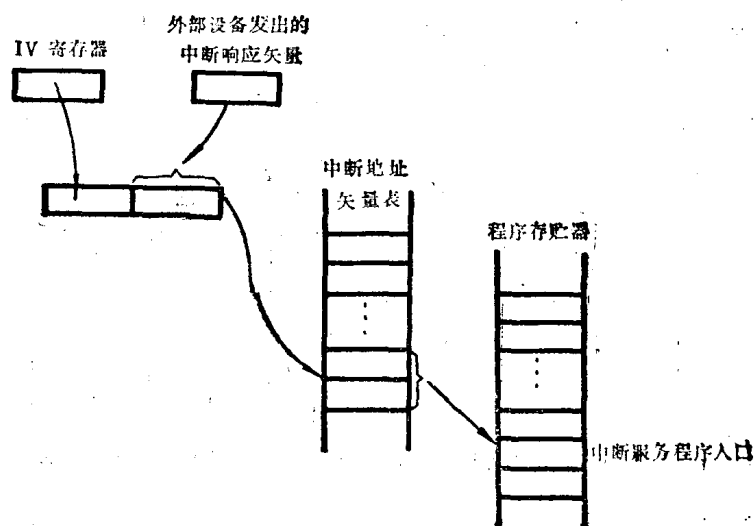


图 1-7 中断矢量的形成

(5) 存储器刷新计数器(memory Refresh counter, 简称 R)

在 Z80 微机系统中, 允许采用十分廉价的动态 RAM。但是, 除非大约每 2ms 进行一次刷新, 否则在这种存储器中所贮存的信息就会消失。为了解决这个问题, 需要设置动态刷新电路, 以便能够周期性地访问动态 RAM, 并将各存储单元的内容重新写入动态 RAM。在 Z80 CPU 中的存储器刷新计数器就是为此而设置的。

(6) 累加器(Accumulator, 简称 A)和标志寄存器(Flag register)

在 CPU 中,累加器是使用最频繁的一种寄存器。在 CPU 完成各种运算期间,累加器作为中间结果的暂存器;在各种 8 位算术运算中,总有一个操作数存放在累加器中。CPU 还利用累加器来实现逻辑操作、移位及某些指令所要求的特殊操作。

标志寄存器由六个 1 位标志触发器组成(图 1-8)。它们反映 CPU 内部的某种操作状态,保存或反映运算的部分结果;有的标志还能输送补充加数(如 CY 标志)。在 Z80 CPU 中设有两类标志:一类主要为 CPU 形成判定操作提供测试条件,叫做测试标志;另一类主要用于 BCD 运算,叫做非测试标志。测试标志用作执行转移、调用或返回指令的条件。根据对某个标志位状态测试的结果是 1 还是 0,就能够决定是否执行相应的操作。测试标志包括:

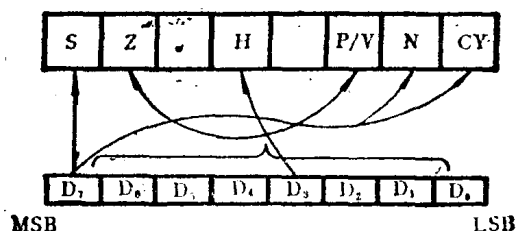


图 1-8 Z80 CPU 的标志寄存器

① 进位标志(Carry,简称 CY)。如果最后一次运算从最高位(MSB)产生进位,则将其存入此标志。在做减法时,最高位产生借位时此标志也能置 1。此外,各种移位指令也能影响此标志。

② 零标志(Zero,简称 Z)。如果最后一次运算结果为零,则此标志置 1,否则置 0。

③ 符号标志(Sign,简称 S)。在有正负号的运算中,用数据字的最高位表示正负号。如果最高一次运算的结果为负数,S 标志置 1,否则置 0。

④ 奇偶/溢出标志(Parity/Overflow,简称 P/V 或 P/O)。这个标志标识运算结果的奇偶性,如果结果中有偶数个 1,则 P/V 标志置 1,否则置 0。这个标志还标识是否发生二的补码溢出。如果发生溢出,则此标志置 1,否则置 0。

对于一个数据字节,如果最高位表示正负号则有 7 个有效位,能表示 $-128 \sim +127$ 范围内的数。如果运算的结果超出了这个数值范围,就会发生溢出。以两个数 M、N 相加为例,设其最高位分别为 M_7 、 N_7 ,结果为 R,其最高位为 R_7 ,其真值表见表 1-1。

表 1-1 求和运算溢出真值表

	1	2	3	4	5	6	7	8
M_7	0	0	0	0	1	1	1	1
N_7	0	0	1	1	0	0	1	1
R_7	0	1	0	1	0	1	0	1
P/V	0	1	0	0	0	0	1	0

由表 1-1 可以看到:在第二和第七种情况下,两个数同号,和的符号却不同,说明发生了溢出。

在第三和第六种情况下,M、N 具有不同符号,不会发生溢出。

在第一和第八种情况下,两个数同号,和的符号相同,说明没有发生溢出。

判别减法是否发生溢出,由参加运算数的符号决定,其真值表见表 1-2。

由表 1-2 可以看到:若 M、N 具有不同符号,差的符号与被减数相反,说明已发生溢出。

所以,产生溢出条件为:第 6 位有进位送入最高(符号)位,从最高位无进位输出;上一位无进位送入最高(符号)位,从最高位有进位输出。

将此条件写成表达式:

表 1-2 减法运算溢出真值表

	1	2	3	4	5	6	7	8
M_7	0	0	0	0	1	1	1	1
N_7	0	0	1	1	0	0	1	1
R_7	0	1	0	1	0	1	0	1
P/V	0	1	0	0	0	0	1	0

$$P/V = d_{bcy} + CY$$

式中 d_{bcy} 表示从和的第 6 位向最高(第 7)位的进位。

由上述真值表可见

$M_7 \cdots \cdots 1 \leftarrow$ 第 6 位无进位

$N_7 \cdots \cdots 1$

(N_7)

0
↓
CY = 1

$M_7 \cdots \cdots 0 \leftarrow$ 第 6 位有进位

$N_7 \cdots \cdots 0$

($\bar{N}_7$)

1
↓
CY = 0

Z80 CPU 的非测试标志有两个:

一个是半进位标志(Half carry, 简称 H)。当低位半字节向高位半字节产生 BCD 进位或借位时,此标志置 1。这个标志用于校正 BCD 加法或减法的结果。

另一个是减标志 (Negate, 简称 N, 此标志也叫做加/减标志, Add/Subtract flag)。在 BCD 运算时,十进制调整指令 DAA 利用此标志来区别加法和减法。如果(前面进行的一次)操作是加法, N 置 0; (前面进行的一次)操作是减法, N 置 1。

累加器和标志位都可以由一些特定的指令来直接访问或受到影响。对于某几种涉及寄存器对指令(如 PUSH, POP 指令),累加器和标志寄存器被视为一个寄存器对,叫做程序状态字(Program Status Word)或处理状态字(Processor Status Word, 简称 PSW)。在 Z80 CPU 中,有两组独立的累加器 A、A' 和标志寄存器 F、F'; 组成 PSW 和 PSW'。

(7) 通用寄存器(Working registers)B、C、D、E、H、L 和 B'、C'、D'、E'、H'、L'。通用寄存器具有多种功能,并能在程序的直接操纵下工作。利用通用寄存器,可以暂存参加运算的操作数和中间结果,避免中间结果在存储器 and 累加器之间来回传送。从这个意义上讲,它们也叫做次级累加器。另外,通用寄存器也可以存放数据地址,作为数据计数器使用。这时就需组成寄存器对,作为专门指向数据存储区的指示器。

在 Z80 CPU 中有两组独立的 8 位通用寄存器,主寄存器组(main register set) B、C、D、E、H、L 可以进行 8 位操作,也可以组成对进行某些 16 位操作。辅助寄存器组(alternative register set) B'、C'、D'、E'、H'、L' 的排列与主寄存器组一一对应。如前所述,当 Z80 CPU 响

应中断后需要保护现场;待中断服务程序结束后需要恢复现场。如果只有一级中断申请的话,就不必再设栈区,只用一条指令就能将主寄存器内容送入辅助寄存器,然后在需要时再交换回来。这样就能大大简化中断操作。因此,辅助寄存器也可以看成是设置在内部的栈区。

2. 运算器

运算器也叫做算术和逻辑单元 ALU (Arithmetic & Logic Unit)。其主要功能是完成各种算术和逻辑运算。Z80 CPU 除具有基本加、减、比较、增 1、减 1、“与”、“或”、“异”等运算功能外,还具有丰富的移位功能以及对单个位的处理(对于各寄存器的任一位的置、复位、测试)功能。

3. 指令译码及定时和控制

这是整个 CPU 的控制中枢。它把程序存储器送来的指令翻译成控制信号,以产生所需要的动作,它使 CPU 能够找到为指令所要求的贮存在存储器中的地址或数据,它向 ALU 提供控制输入,使之执行指令所规定的运算,它监视外部控制信号,并产生适当的响应,它为存储器和 I/O 器件提供状态、控制和定时信号等。总之,它的功能是在正确的时刻,将信息准确地送往某个目的地。它动作的依据是该时刻执行的指令,它的作用是产生发往 CPU 和 CPU 各部分的控制和定时信号(见图 1-9)。

Z80 CPU 的动作是周期性的,需要精确定时。基本定时脉冲由外部振荡器产生,接至 CPU 的 ϕ 输入端。如图 1-10 所示,在两个定时脉冲上升沿之间的持续时间是一个时钟周期(Clock Period),它等于机器处于一种状态的持续时间,因此叫做 T 状态 (T State) 或 T 周期 (T Cycle)。CPU 实现某种规定的基本操作所需时间叫做机器周期(Machine cycle)或 M 周期,一般为 3 或 4 个 T 状态。有些指令在执行自动插入或 CPU 通过 $\overline{\text{WAIT}}$ 信号/插入等待 (WAIT) 状态,这时就可能有 5~6 个 T 状态。一条指令从取出到执行完毕的持续时间叫做指令周期 (Instruction cycle)。根据指令内容的不同, Z80 CPU 执行一条指令可能需要 1~6 个机器周期。

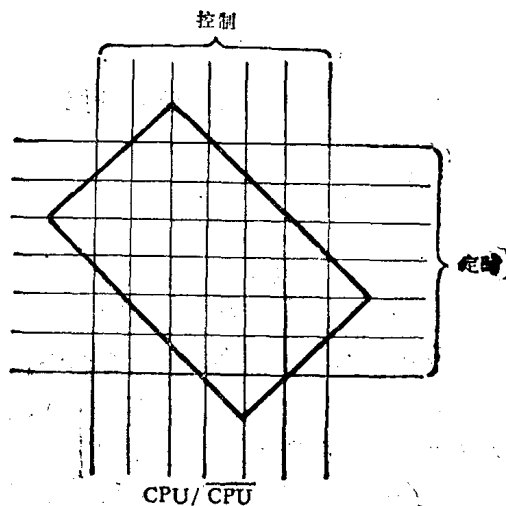


图 1-9 控制器的功能

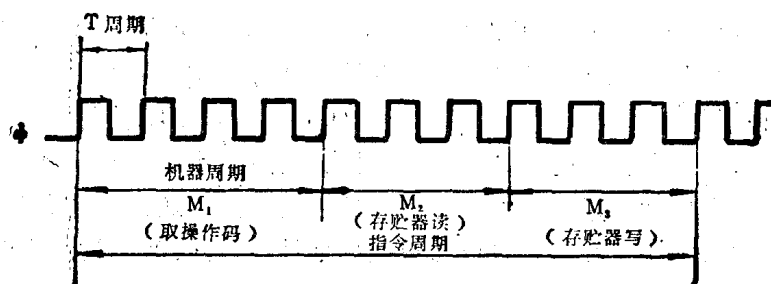


图 1-10 Z80 的定时波形

4. 总线

在 CPU 内部各单元之间的数据传送,是通过内部总线实现的。对于用户来说,有三条总线