

# 微机化仪表原理及设计

中国仪器仪表学会过程检测控制仪表学会  
中国自动化学会仪表与装置专业委员会

1988.8

## 前 言

微型计算机技术的迅速进展，特别是单片微机的出现和发展引起仪器仪表设计的场巨大的变革。自动化仪表正日趋智能化、系统化、多样化和小型轻量化，其核心微机在仪表中的应用。

近年来，国内各仪表研究所、高等院校，仪表制造厂均在开发带微机的新一代仪表，对传统产品进行更新换代，不少企业单位也开始在生产过程自动化系统中使用这类新型仪表。因此，广大自动化仪表设计、研究人员、生产及使用者迫切希望在原有微机普及知识的基础上，系统了解和掌握微机（智能）化仪表的基本原理、设计方法和具体应用。鉴于目前有关微机（智能）化仪器仪表方面的出版物中大多以叙述智能电工测量仪器为主，而过程检测控制仪表又比电工测量仪器具有更多实际问题。诸如测量控制的实时性，现场抗干扰，仪表的可靠性等。因此，本学会特组织有关人员编写了这份教材，以满足广大工程技术人员和仪表制造、使用单位在设计、制造、维修等方面的需要。

本书首先扼要介绍微机化仪表的功能、组成以及仪表开发的一般过程和设计原则，接着从硬件方面分别介绍常用微处理器（机），输入输出通道和键盘显示器等外围接口电路设计，仪表抗干扰措施以及常用标准接口总线；软件方面阐述软件设计技术，微机化仪表软件的组成和设计方法（包括：监控主程序、键盘管理、中断管理及处理程序），仪表测量算法和控制算法（包括常用控制算法、模糊控制算法、人工智能算法等）。本书还结合所介绍的硬件和软件，阐述微机化仪表的总体设计和设计实例，并介绍仪表的调试方法和开发工具。

全书在内容安排上，注重过程检测控制仪表在微机化上所特有的问题，叙述上力求删繁就简、深入浅出，强调实用性。所涉及的内容都尽可能给出具体线路、程序或软流程。

本书可供从事微机化仪表设计、生产和使用的有关工程技术人员作自学资料，作为专题学习班的教材，也可供高等院校有关专业的师生作教学参考书。

此书出版权属中国仪器仪表学会过程检测控制仪表学会和中国自动化学会仪表与装置专业委员会所有，任何其他单位或个人都不得私自翻印。

编者：华东化工学院（吴勤勤、单青）、上海仪表公司职工大学（郁志杰）、上海第二轻工业局职工大学（朱康余）

组织者：中国仪器仪表学会过程检测控制仪表学会（吴斌昌）

# 目 录

## 第一章 微机化仪表设计概述

§1.1	微机化仪表的功能和组成	(1)
1.1.1	微机化仪表的主要功能	(1)
1.1.2	微机化仪表的基本组成	(2)
§1.2	微机化仪表的设计开发过程	(4)
1.2.1	准备阶段	(4)
1.2.2	研制阶段	(6)
1.2.3	总调阶段	(7)
§1.3	微机化仪表器件的选择	(7)

## 第二章 常用微处理器(机)

§2.1	8085 A 微处理器	(11)
2.1.1	8085 A 结构	(11)
2.1.2	接口信号和功能	(12)
2.1.3	8085 A 最小系统	(15)
§2.2	Z80 微处理器	(16)
2.2.1	Z80 CPU 结构	(16)
2.2.2	接口信号与功能	(18)
2.2.3	Z80 最小系统	(20)
§2.3	MCS—51 单片微型计算机	(21)
2.3.1	概述	(21)
2.3.2	MCS—51 系统结构	(24)
2.3.3	单片机的扩展	(36)
§2.4	MCS—96 单片微型计算机	(38)
2.4.1	概况	(38)
2.4.2	MCS—96 结构	(40)

## 第三章 输入输出通道

§3.1	模拟量输入输出通道	(47)
------	-----------	------

3.1.1	模拟量输入通道	(47)
1.	A/D通道的几种结构形式	(47)
2.	A/D转换电路及其与微处理器的接口	(48)
3.	A/D转换辅助电路	(61)
3.1.2	模拟量输出通道	(64)
1.	D/A通道的几种结构形式	(64)
2.	D/A转换电路及其与微处理器的接口	(65)
3.1.3	模入模出通道设计实例	(69)
3.2	开关量输入输出通道	(74)

#### 第四章 微机化仪表的抗干扰设计

4.1	干扰与噪声源	(78)
4.1.1	外部干扰	(79)
4.1.2	数字电路的干扰	(82)
4.2	抗干扰设计	(84)
4.2.1	串模干扰的抑制	(84)
4.2.2	共模干扰的抑制	(85)
4.2.3	过程通道的抗干扰设计	(87)
4.2.4	电源与电网干扰的抑制	(94)
4.2.5	地线系统干扰的抑制	(96)

#### 第五章 外设接口电路

5.1	LED显示接口	(98)
5.1.1	7段LED显示	(98)
5.1.2	点阵式LED显示	(103)
5.2	键盘接口	(104)
5.2.1	抖动与串键	(104)
5.2.2	非编码键盘	(105)
5.2.3	编码式键盘	(110)
5.3	打印机接口	(121)
5.3.1	打印原理	(121)
5.3.2	打印机接口	(121)

#### 第六章 常用标准接口总线

6.1	IEEE—488并行标准接口总线	(124)
6.1.1	IEEE—488标准接口总线的基本特性	(124)

6.1.2	总线描述	(125)
6.1.3	总线工作过程	(130)
6.2	RS—232C 串行标准接口总线	(133)
6.2.1	RS—232C 传输数据格式和逻辑电平	(133)
6.2.2	RS—232C 接口信号	(135)
6.2.3	单片机通信	(139)
6.3	RS—422、RS—423 标准	(141)

## 第七章 软件设计

7.1	软件设计技术	(143)
7.1.1	结构化设计方法	(143)
7.1.2	由顶向下设计方法	(145)
7.1.3	模块化编程方法	(147)
7.2	微机化仪表软件的组成与设计	(149)
7.2.1	监控主程序	(149)
7.2.2	键盘管理	(150)
7.2.3	中断管理及处理	(153)
7.2.4	子程序模块	(155)
7.2.5	数据表	(156)
7.3	中断管理	(156)
7.3.1	8085 中断系统	(156)
7.3.2	Z80 中断系统	(157)
7.3.3	8051 中断系统	(161)

## 第八章 测量算法

8.1	测量中的误差	(165)
8.1.1	仪表和基本误差	(165)
8.1.2	随机误差的性质与估计	(166)
8.1.3	粗差判别与坏值的舍弃	(170)
8.1.4	系统误差的分类和估计方法	(170)
8.1.5	误差的传递	(172)
8.1.6	误差的合成	(175)
8.2	克服随机误差的方法	(176)
8.2.1	叠加平均法	(176)
8.2.2	数据平滑	(177)
8.3	利用误差模型修正系统误差	(178)
8.4	利用校准数据修正系统误差	(180)

8.5	通过曲线拟合来求得校准方程	(182)
8.5.1	最小二乘原理	(182)
8.5.2	曲线的拟合	(183)
8.6	粗大误差的克服	(187)
8.6.1	统计法	(188)
8.6.2	数字滤波法	(189)
8.7	标度变换(工程量变换)	(190)
8.8	非线性校正	(191)
8.8.1	多项式逼近	(191)
8.8.2	分段逼近	(192)

## 第九章 控制算法

9.1	数字PID算法	(195)
9.1.1	完全微分型算法	(195)
9.1.2	不完全微分型算法	(200)
9.1.3	编制PID算法程序时几个值得注意的问题	(203)
9.2	前馈控制算法	(205)
9.2.1	前馈控制算法原理	(205)
9.2.2	前馈控制算法计算步骤	(206)
9.3	纯滞后补偿算法(Smith预估)	(209)
9.3.1	纯滞后补偿算法原理	(209)
9.3.2	纯滞后补偿算法计算步骤	(211)
9.4	差拍控制与Dahlin算法	(213)
9.4.1	差拍控制	(213)
9.4.2	Dahlin算法	(215)
9.5	解耦算法	(217)
9.5.1	解耦算法原理	(217)
9.5.2	解耦算法计算步骤	(219)
9.6	模糊算法	(220)
9.6.1	模糊控制算法原理	(220)
9.6.2	模糊控制实时算法	(224)
9.7	参数自整定算法	(226)
9.7.1	参数自整定基本原理	(226)
9.7.2	参数自整定修改模型算法	(227)
9.7.3	识别与校正模型算法	(228)
9.7.4	参数自整定算法的计算步骤	(229)
9.8	仿人工智能算法	(231)

## 第十章 微机化仪表的总体设计与实例

{10.1	总体设计	(234)
10.1.1	设计准则	(234)
10.1.2	编写系统设计说明书和绘制总流程图	(236)
10.1.3	系统设计初步评价	(237)
{10.2	微机液相色谱数据处理仪(实例1)	(237)
{10.3	微机温度—时间控制仪(实例2)	(250)
{10.4	智能温度控制仪(实例3)	(257)

## 第十一章 调试方法和开发工具

{11.1	常见故障	(265)
{11.2	调试方法	(266)
11.2.1	静态调试	(266)
11.2.2	动态调试	(267)
{11.3	开发工具	(269)
11.3.1	微型计算机开发系统	(269)
11.3.2	SICE通用单片机在线仿真系统	(272)
11.3.3	简易开发装置	(276)
11.3.4	逻辑分析仪功能简介	(278)

## 附 录

附录1:	8085 A 指令表	(279)
附录2:	Z80 指令表	(284)
附录3:	MCS — 51 指令表	(295)
附录4:	Z80 PIO 可编程并行接口芯片	(304)
附录5:	8255 可编程并行接口芯片	(309)
附录6:	8251 可编程串行接口芯片	(314)
附录7:	8155 RAM/IO 扩展器	(318)
附录8:	集成芯片引脚图	(323)

## 第一章 微机化仪表设计概述

微型计算机技术的迅速发展,引起了仪器仪表设计的根本性变革,即以微机为主体,代替传统仪表的常规电子线路,创制新一代的具有某种智能的灵巧仪表。这类仪表的设计重点,已经从模拟和逻辑电路的设计转向专用的微机系统或微机功能部件,接口与数据通道的设计,以及应用软件的开发。

在仪表中引入微机,不仅能解决常规仪表不能解决或不易解决的问题,而且能简化电路、增加功能、提高精度、降低售价和加快新产品的开发速度。由于这类仪表已能实现人脑的一部分功能,例如四则运算、逻辑判断、命令识别等,有的还能够进行自校正、自诊断,并具有自适应、自学习的能力。因此,人们习惯上称它们为智能仪表。但智能的水平有高有低,目前所见的大部分这类产品,智能化的程度还较低,故将具有不同智能水平的仪表通称为微机化仪表或由微机控制的仪表更为确切。当然,随着科学技术的进一步发展,微机化仪表所具有的智能水平将会越来越高。

### 1.1 微机化仪表的功能和组成

#### 1.1.1 微机化仪表的主要功能

将微机引入仪表中,能解决的问题是多方面的,大体上可实现如下一些功能:

##### 1、自动、快速检测参数

可在几个毫秒、甚至更短的时间内自动地对多个参数进行巡回检测,这对于及时了解和记录被控对象的瞬变工况是十分重要的。传感器的输出信号可以是模拟电压值,经模数转换器转换成数字信号后输入微机。若是脉冲频率信号,则可直接采入,进行处理。

##### 2、自动地修正各类测量误差

在仪表的误差源中,最主要的是传感器和模拟测量电路所造成的误差,诸如零点漂移误差、温度误差、非线性误差等。在微机化仪表中,只要事先测出这些误差规律,就能靠软件进行修正,这可大大提高仪表的测量精度。常见的有热电偶冷端温度补偿、蒸汽流量的温压补偿等。传感器的非线性特性也可由相应的线性化程序进行校正。

##### 3、自动切换量程

量程的自动切换给使用带来方便,并可提高读数的分辨力。

##### 4、数字滤波功能

通过对主要干扰信号特性的分析,确定适当的数字滤波算法,编制相应的滤波程序,可解决一些模拟滤波器难以解决的问题,如低频干扰、脉冲干扰、随机干扰等。

##### 5、控制和报警功能



能实现PID及各种复杂调节规律,例如可进行串级、前馈、非线性、纯滞后、自适应等控制。当超出规定范围时,立即报警。

### 6、多种形式输出数据

最常用的输出形式是数字显示和打印。也可输出模拟量信号,并可用数字通讯方式传递信息,以便与其它仪表或计算机系统相连。

### 7、自诊断功能

在运行过程中,可对仪表本身各组成部分进行一系列测试。一旦发现故障即能告警,并显示出故障部分,以便于处理。

### 8、掉电保护功能

仪表内装有后备电池及电源自动切换电路。掉电时,能自动将电池接向RAM,保护数据不丢失。

在一些不带微机的常规仪表中,也可以或多或少地具有上述的某种功能,但往往要付出较大的代价,性能上的些许提高,会使仪表的成本大大增加,而在微机化仪表中,性能的提高,功能的扩大是比较容易实现的。低廉的微机芯片使这类仪表具有较高的性能价格比。

## 1.1.2 微机化仪表的基本组成

微机化仪表包括:信号预处理、模/数(A/D)转换、微机系统( $\mu C$ )、数/模(D/A)转换、数字显示或打印、串、并行标准接口部分,如图1-1所示。其中微机系统是仪表的核心部分,它由CPU、RAM、EPROM、PIO、SIO、CTC等芯片组成,或者它本身就是一个单片微型计算机。

图1-2表示了微机化仪表的工作流程。输入信号先在预处理电路中进行变换、放大、补偿等处理。若是模拟量则需经A/D转换变成数字信号,再经接口送入缓冲寄存器,以保存输入数据。然后由CPU进行数据处理、计算分析等一系列工作。运算结果可存贮在RAM中,也可通过输出缓冲器送至显示器和打印机,或经D/A转换得到模拟量的输出信号。微机化仪表还可备有RS-232C和HP-IB(即IEEE-488)标准接口总线,以实现数据通信,完成更复杂的测控任务。仪表的整个工作是在程序控制之下进行的,工作程序应预先编好,写入EPROM中,必要的参数、命令可由键盘键入,存于RAM中。

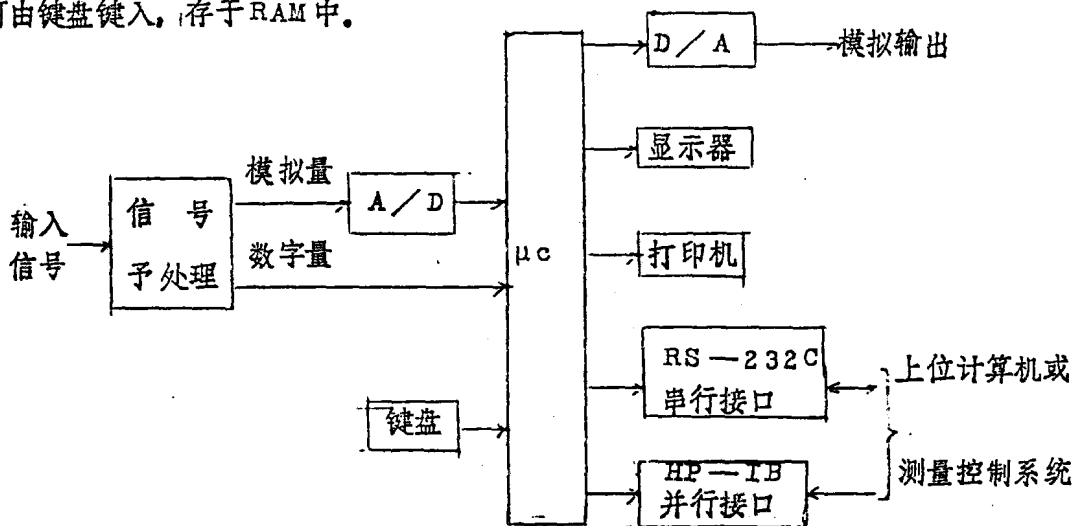


图1-1 微机化仪表组成框图

以上只是微机化仪表的大致组成，至于仪表内部的具体线路和设计方法，将在以后各章节详细阐述。

微机化的自动化仪表近年来发展很快。在国内，已开发出带有微机的各种测控仪表。例如，能自动进行温、压补偿的节流式流量计，能快速、精确进行校核的水流量校验装置，能对各种谱图进行分析和数据处理的色谱数据处理仪，能进行程序控温的多段温度程控仪以及能实现复杂控制规律的数字式调节器智能式控制器等等。

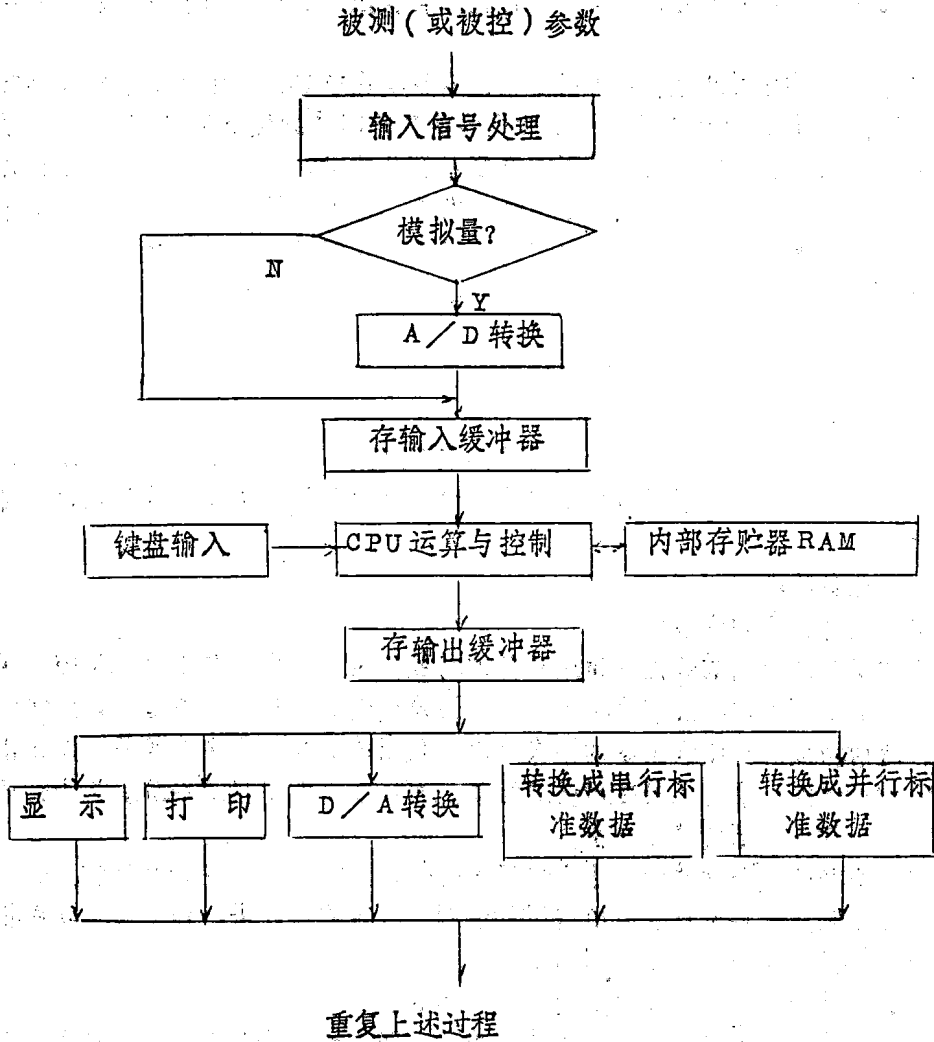


图 1—2 微机化仪表工作流程图

国外一些厂商也不断推出新的产品。例如美国霍尼韦尔公司开发的 DSTJ — 3000 系列变送器，是具有复合功能的智能式变送器。它采用半导体硅单晶片上配置的差压、静压和温度三种传感元件，做到差压值状态的复合测量，对变送器本体的温度、静压特性进行自动补偿，从而获得较高的测量精度 ( $\pm 0.1\%$ )，该变送器还可用遥控操作器远距离地进行零位较准、阻尼调整，测量范围的变更，以及线性或平方根输出的

选择,它不需要到现场进行校验,一切均可在仪表控制室内进行远方设定,使用和维护十分方便。

日本横河公司研制了模拟数字混合式记录仪,它不用传统的闭环随动测量原理,而用开环扫描测量方法,所以没有伺服放大器、平衡电机、滑线电阻等部件。测量信号经多路开关扫描输入后,进行前置放大和A/D转换,再在微机控制下,发出相应的脉冲数驱动步进电机。由步进电机带动打印头作横向移动而画出模拟曲线,也可操作打印头针点的动作,打出数据和表格。这种记录仪的测量精度比传统记录仪高,它能进行模拟或数字显示和记录,还具有自检査、自校正、求差、报警等功能,并带有通信接口。

美国福克斯波罗公司推出了一种数字化的自整定调节器。它除具有模拟式调节器的特点外,能自行计算PID参数,并使过程的恢复时间减到最小值。该调节器所具有的控制方法是人工智能式的,它采用了“专家系统”技术,象有经验的控制工程师那样能应用操作经验来整定调节器,工作迅速、准确。自整定调节器组态灵活,操作方便,节省了控制系统的投入时间,特别当对象特性变化频繁或在非线性系统中,由于它能自动改变参数,始终保持系统品质最佳,因此大大提高了系统运行的经济性。

## §1.2 微机化仪表的设计开发过程

微机化仪表设计和开发过程大致上可分为三个阶段:准备阶段、研制阶段和总调阶段,如图1—3所示。以下对微机化仪表开发过程的各阶段工作内容和设计原则作一简要的叙述。

### 1.2.1 准备阶段

#### 1、确定设计任务和仪表功能

首先确定仪表所要完成的任务和应具备的功能。例如,仪表是用于工业控制呢还是数据处理,其精度如何;仪表的输入信号范围及通道数是多少,通道型式如何;仪表是否具有通信功能;打印和显示的格式如何;仪表的成本约是多少等等。以此作为整个装置的设计依据。设计人员在対仪表性能、可靠性可维护性及经济效益进行综合考虑的基础上,提出仪表的总体方案。

在开发过程中,硬件和软件工作应该同时进行,如图所示。在设计硬件、研制功能部件的同时,完成软件设计和应用程序的编制。两者同时并进,能使硬、软件相互配合,充分发挥微机特长,缩短研制周期。

#### 2、完成总体设计,选择、确定硬件类型和数量

在调研之后着手进行微机化仪表的总体设计,即描绘硬件框图或系统总图,以及编制软件框图或工作流程图,先将仪表功能分成几个大的模块,然后按仪表工作过程和时间关系将其连接起来。

选择微处理器时,需考虑字长、寻址范围与寻址方式、指令的功能、中断能力、DMA能力,以及微处理器的支持器件——LSI外围电路是否齐全等等。

在确定存贮容量时,主要依据整个装置所需要的存贮量、数据量以及堆栈的大小。

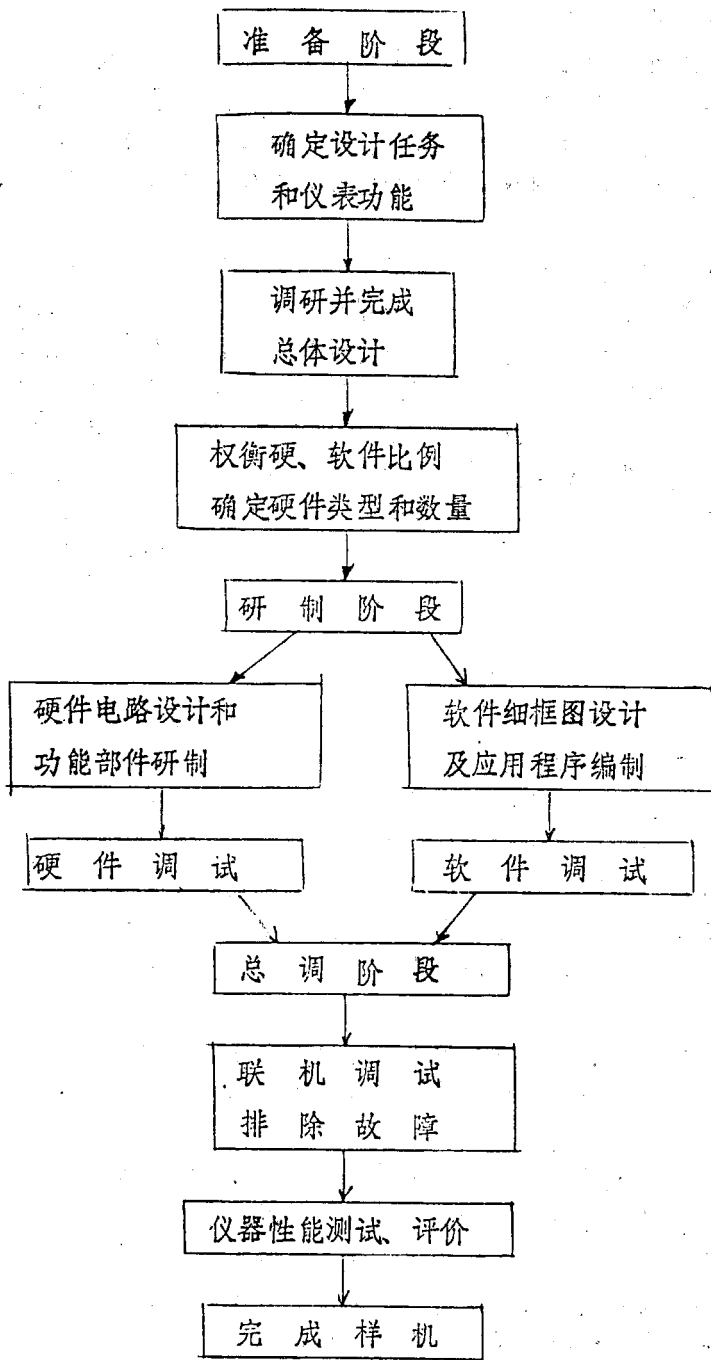


图 1—3 微机化仪表开发过程

在仪表所需的硬件中，输入、输出数据通道往往占有很大的比重，因此在估计和选择输入、输出所需的硬件时，应考虑输入输出通道数，串行操作或是并行操作，数据的字长、传输率以及数据传输方式等等。

由于硬件和软件具有互换性，设计人员要反复权衡仪表硬件和软件的比例。多用硬件可以简化软件设计工作，并使装置的性能得到改善。然而，这样会增加元器件数，成本相应提高。若采用软件来代替一部分硬件功能，虽可减少元器件数，但将增加编程的复杂性，并使系统的速度相应降低。所以应当从仪表性能、器件成本、软件费用、研制周期等各方面考虑，对硬、软件比例作出合理的安排，从而确定硬件的类型和数量。

近几年逐渐为人们所重视的单片微机，内部具有 CPU、RAM、I/O 接口、定时计数器等电路，在有的单片机上，还集成有 A/D 转换器 D/A 转换器、模拟多路开关，因而它本身就是一个完整的微型计算机。而且与多芯片组成的微机相比，单片机的体积小、功耗低，价格也较便宜，故在微机系统和仪表中的应用已日趋增多，但目前的单片微机，RAM 容量较小，I/O 口不多。所以在内存容量要求不大，外部设备要求不多的微机化仪表中采用单片机较为适宜。如果在单片机的基础上外接一些存储器 and 接口电路，则可扩大其应用范围，研制出功能完善、性能优良的微机化测控仪表来。

## 1.2.2 研制阶段

### 1、硬件电路设计、功能部件研制和调试

微机功能部件是微机化仪表的核心部分。伏表的数据处理或控制的主要功能均由其完成。它包括微处理器、存储器、输入输出接口及其它逻辑电路。功能部件一般应分板研制也可集中在一块板上。设计、研制要考虑技术上可行、布局合理及连接方便。先描制逻辑电路图，经反复查对，线路无差错，才绘制成布线图。制成印刷电路板（或实验电路板）以后仍须仔细校核，以免发生差错，损坏器件。

由于微处理器是通过各种接口电路与数据通道以及仪器各个部分相连接的，而数据道又通过测量元件和执行器直接连至被测或被控对象。因此，输入、输出接口和数据通道的设计是研制仪表硬件的重要环节，力求可靠实用。以下各章将对数据通道和各种外围接口电路的设计予以详细介绍。

在研制硬件时，若采用单片微机或者采用商品化的微机模块或单板微机，则研制工作可以简化。究竟采用何种器件，应当由如上所述的仪表性能、成本、研制期限等各种因素决定。

如果逻辑线路设计正确无误，印刷电路板加工完好，那么功能部件板的调试一般来说是比较方便的。部件板运行是否正常，可通过测定一些重要的波形来确定。例如可检查微处理器几个控制信号的波形与硬件手册所规定的指标是否相符，由此断定其工作正常与否。为加快硬件研制进程，迅速查出故障，可采用简易开发装置，例如具有调试、开发手段的单板机或其它微机，将其与功能部件板相连，再编制一些调试检查程序，即可较方便地完成硬件部分的查错和调试任务。

### 2、软件框图设计、程序编制和调试

首先将软件总框图中各个功能具体化，逐级画出详细的框图，作为编制程序的依

据。编写程序可用机器语言、汇编语言或各种高级语言。究竟采用何种语言则由程序长度及所具备的研制工具而定。对于规模不大的软件系统，大多用汇编语言来编制应用程序。因为应用程序仅编一次，从减少存贮容量、降低器件成本和节省机器时间的观点来看，这样做比较合适。程序编制后，再通过具有汇编能力的计算机生成目标程序，经模拟试验通过后，可直接写入可编程只读存贮器EPROM中。

在程序设计过程中还必须进行优化工作，即仔细推敲、合理安排，利用各种程序设计技巧使编出的程序所占内存空间较小，而执行时间又短。

为了更方便地研制各种微机系统，缩短研制周期，目前已广泛使用“微型计算机研制(开发)系统”(Microcomputer Development System, 简称MDS)，它是研制开发微机应用系统、微机化仪表硬件和软件的有效工具。例如Intel公司的MDS-210、220、230、240、287，Tektronix的8500系列，HP公司的HP6400等。这些开发系统的软件和硬件系统很丰富，用它来研制微机系统和测控仪表，可大大减轻设计人员的工作强度。它可帮助设计者积累研制各种应用系统的经验。不仅可加快研制应用系统和仪表的进度，而且可进行全面测试，这有助于提高整个应用系统和微机化仪表的质量和可靠性。

如前所述，在开发微机仪表过程中，硬、软件工作是同时并进的，两者相辅相成，联系十分紧密。因此，设计人员不仅要懂得系统的硬件，而且还要熟悉软件，这样才能设计出理想的仪表来。

### 1.2.3 总调阶段

研制阶段只是对硬件和软件分别进行了初步调试和模拟试验。整机装配好后，必须对整套装置进行联机试验，识别和排除装置中硬件和软件两方面的故障，使其能正常运行。待工作正常后，便可投入现场进行调试，使系统处于实际应用环境中，以考验其可靠性。在调试中还必须对设计所要求的全部功能进行测试和评价，以确定仪器是否符合预定性能指标。若发现某一项功能或指标达不到要求时，则应变动硬件或修改软件，重行调试，直至满足要求为止。最后将调试好的应用程序写入EPROM中。

研制一台微机化仪表大致上需经历上述几个阶段。研制周期的长短与硬件的选择、总框图设计是否合理、研制工具是否完善，设计人员对微机硬件及软件的熟悉程度等等有关。准备阶段所化时间尽管不长，但元、器件的选择、程序框图结构的好坏将直接影响硬、软件功能和质量。在研制阶段，软件设计的工作量往往比较大，而且容易发生差错(指用汇编语言编写的程序)，建议尽可能采用结构程序和模块化方法编制应用程序，这对查错、调试、增删程序十分有利。实践证明，如能在研制阶段把住硬软件的质量关，则总调阶段将能顺利进行，及早制成符合设计要求的样机。

## 1.3 微机化仪表器件的选择

器件的选择主要是指微处理器(或单片微机)的选择，因为微处理器是微机的核心部件，它的特性对微机功能部件及软件的设计，对所研制的仪表性能有很大的影响。要成功地研制一台微机化仪表，必须选择合适的微处理器。选定微处理器(或单片微

机)后,再按设计要求确定与其配套的外围芯片。在选择微处理器时应考虑如下主要特性:

### 1、用途

微处理器是一种通用器件,如果给予足够的外部支持电路和处理时间,它几乎可以完成任何任务。数据处理和控制是微处理器两个主要用途。数据处理要求微机有较强的算术运算能力。一般兼有数据处理任务的控制类微机化仪表大多采用数据处理型的微处理器。微处理器的用途可以根据对微处理器的字长、指令系统、支持硬件和支持软件等进行考察后作出判断。单片微机既适用于控制,也可进行数据处理。

### 2、字长

微处理器的字长取决于并行数据总线的数目。字长较长的微机能处理较宽范围的算术值,适合于数据处理工作。字长较短的微机一般用于控制。在计算量不大的情况下,只要能够满足仪器功能的要求,应采用字长较窄的微处理器。4位字长的微处理器一般设计成简单的控制器。8位微处理器则设计成既可用于数据处理,也可用于控制。用于数据处理时,可进行双倍精度或三倍精度运算。16位的运算精度适合于大多数的数据处理工作。因此这类微处理器大多用于数据处理。由于8位微处理器或单片机适用范围广,价格也不高,故为目前多数微机化仪表所采用。随着微机技术的发展,已出现带有16位微处理器或单片机的高性能测控仪表,并将越来越多地用于生产过程中。

不同字长微处理器的成本、特点和应用范围如表1—1所示。

表1—1 不同字长微处理器的特性

	4 位	8 位	16 位
成本	低	中	高
特点	适宜于十进制运算,指令功能差,速度慢。	适宜于字符、数据处理,双倍字长精度运算时速度要降低。	具有多种指令,功能强,速度快。
应用范围	适合于计算精度低,对处理时间要求不高的地方,如计算器、家用电器、简单控制器,大型玩具等。	用于计算机外围设备、终端设备,测量,监视仪器、数据处理、通信和实时控制。	与一般小型计算机用途相同,可用于数值计算、较复杂的数据处理和实时控制。

### 3、寻址范围和寻址方式

微处理器的地址长度反映了微机可寻地址的范围,它表示系统中可存放的程序和数据量。例如,8位标准微处理器,其地址长度为16位,可寻址的范围为64K。设计人员应根据仪器要求确定合理的容量。一般微处理器都有多种寻址方式,如直接、间接、相对、变址寻址等。选择恰当的寻址方式,能使程序量大为减少,从而可节省

存贮空间和加快程序的执行速度。

#### 4、指令的种类和数量

一般说来，指令条数多的微处理器，操作功能要强些，这可使编程灵活。字长比较短的微处理器，如4位微处理器，指令条数就少，操作功能也差些。但是一个微处理器的功能究竟丰富与否，不能单由指令的数量确定，而要看每一条指令的具体内容。因为每一个制造厂家都有它自己计算指令的方式。例如，Intel公司将立即数送寄存器和存贮器的数送寄存器的指令算作8085微处理器的两条指令，而Motorla公司将这两条指令算作同一条指令的两种寻址方式。虽然8085比6800的指令要多，但是，如果将寻址方式也计入，则6800的实际指令不比8085少。

所选取的微处理器，其指令功能应该面向所要处理的问题。用于控制的仪器，要特别注意I/O指令的功能。用于数据处理的仪器在选择时还应注意数据操作指令的功能，例如算术移位、十进制调整、位操作指令、12的补码指令等等。

八十年代推出的单片微机，由于吸收了各类微处理器的长处，使其指令功能较为完善，它既可适用于数据处理，也可作为性能良好的控制器用。例如8051，具有较强的算术和逻辑运算能力，且擅长位处理操作，它还具有乘法和除法指令，编程较方便。

#### 5、执行速度

微处理器的执行速度可用时钟周期（或微周期）数或机器周期数来表示。大多数微处理器要多个乃至十个时钟周期才能执行一条指令。不同类型的微处理器以不同的方法执行指令，有些微处理器采用高速时钟和许多微操作（例如8085和Z80），另一些微处理器则采用低速时钟和少量强有力的操作（例如6800和6502）。对于两个相邻存贮器中的数送寄存器，这一条指令8085要16个时钟周期，而6800只要5个时钟周期。6800用1MHz时钟速率执行该指令，相当于8085用2MHz时钟执行该指令三分之二的時間。

执行速度的选择也要区别不同的对象而选定。对于采样周期较短而有大量实时计算的数据处理或过程控制系统，应选择速度高的微处理器。

#### 6、功耗

功耗由器件工艺、器件的复杂性以及时钟速率所支配。字长较长的微处理器，因器件电路复杂，其功耗比字长较短而工艺相同的微处理器要大。从器件工艺来说，高速双极性微处理器要消耗更多的功率，NMOS和PMOS的微处理器消耗中等的功率，而CMOS的微处理器所消耗的功率最少。时钟速率本身也影响着一些微处理器的功耗，较慢的时钟速率，微处理器消耗的功率较小。例CMOS工艺的RCA1802微处理器，其功耗随时钟速率而不同，从1MHz的60mW到10KHz的5mW。

#### 7、中断能力和DMA能力

在实际应用中，外部设备常要求微处理器暂时转去执行一个为其服务的子程序。为满足这一要求，微处理器必须具有较好的中断能力。对于快速、多通道实时处理的对象，应选择中断功能丰富的微处理器。直接存贮取（DMA）是一种数据传输方式。这种数据传输方式不是由微处理器进行控制，而是由DMA控制部件暂时“接管”微机CPU，通过总线对存贮器的直接访问。DMA能力对于大块数据传输很有用，它解脱



了由微处理器控制传输数据时必须执行一个数据传输程序的负担。因此DMA传输比程序控制数据传输快得多。如果要求大量的高速数据传输，则必须选择一个具有DMA能力的微处理器。

#### 8、硬、软件支持

选择微处理器时，应考虑该器件有否足够的硬软件支持。从硬件来说，构成一个微机化仪表要有足够的LSI外围芯片，例如串行接口、并行接口、定时计数器、A/D和D/A转换器等。对于单片微机要考虑应有配套的扩展芯片供应。

从软件来说，应用软件研制的费用往往超过元器件成本。为此，应选择那些具有大量基本软件（编辑程序、汇编程序、高级语言等）支持的微处理器，以便采用“微机开发系统”，缩短研制周期。当然，对于较小的微机系统，就不一定需要支持软件。

#### 9、成本

在设计微机化仪表时，成本是优先考虑的项目之一。特别对消费者市场更是如此。当然，估计成本应从整个装置考虑，而不仅仅是微处理器的成本。但是，是否正确地选择微处理器或单片微机，又直接影响到整个仪表的成本。因此，必须仔细权衡，全盘考虑。特别是由于微机技术发展得非常快，必须经常关心微处理器、单片微机及其它芯片和有关外部设备的现行价格，合理地进行选择。