

# 智能仪器中的计算机系统设计

邢 忠 宝

**摘要:** 本文概要介绍在智能仪器中计算机系统设计方面的一些具体问题及在实际工作中对所遇到的一些具体问题的处理方法。探讨一下计算机软件、硬件及接口的基本设计方法。

## 一、前 言

随着微型计算机技术的开发和广泛应用,使得科学仪器产生了深刻的改变,科学仪器由自动化逐步地向智能化发展。一般说来,智能的程度决定于计算机系统的设计技术,涉及到计算机硬件、软件及设计技巧等方面的问题。本文主要结合在科研题目实施过程中,所遇到的一些具体问题,探讨一下仪器中的微机系统总体结构、接口电路及软件设计的一些有关问题,以及一些具体问题的处理方法。

## 二、系统总体布局

智能仪器中微机系统设计所遇到的第一个关键问题,就是系统总体布局。总体布局设计是否合理,往往在整个仪器中起着很重要的作用,它直接关系到仪器基本功能是否能实现,也影响到仪器的稳定性、可靠性、主要技术指标和性能价格比。在确保完成仪器基本功能和主要技术指标前提下,系统总体布局设计时,下面的一些问题是值得慎重考虑的。

### 1. 模块化设计

所谓模块,就是能够完成某个或某几个功能的单独的有机组合体。一台完整的仪器计算机系统包括多个这样的有机组合体。

软件模块就是一个完整的指令集,这一指令集完成某个或某几个基本功能,或者完成某一个特定的功能。如:键盘扫描程序模块、步进电机驱动控制程序模块、扫描测量程序模块等等。在软件设计中,采用模块化,对设计者来说,可以把较为复杂的问题简单化,使程序的编制和调试都易于实现,出现问题,容易查找。对设计者来说,也是一劳永逸的,在不同情况下针对不同仪器,只需改变模块的一些基本参数,而模块的基本结构是不改变的。对使用者,对他人也是容易看懂,易于掌握,便于推广。

硬件模块一般分为两种,一种是主控模块,另一种是从控模块。主控模块能够发出命令,对系统总线有占有和控制功能。如CPU模块、DMA模块等。从控模块大多数是由一个或多个接口电路组成,完成某些特定功能。如:A/D、D/A模块,串并行输入输出模块等。硬件模块化,便于设计、调试、移植,在不同情况下,选用不同模块的组合,可容易的组成新的系统,也有利于系统维护,提高工作效率。

## 2. 标准化与通用性

在软件硬件模块设计过程中，标准化和通用性是值得考虑的问题，这有助于系统的开发，也有利于后人所用。

对于软件模块，它的适应范围越宽越好，效率越高越好。一个成形的模块都依附于某一个指令系统，将某一个模块所依附的指令系统变换成另外一个指令系统，是不困难的。为了便于移植，从标准化、通用性考虑，在指令的使用上要简洁明了，层次清楚。

在硬件模块设计中，对于标准化通用性来说，涉及到的问题是较多的，诸如功能设置，总线信号对接关系，总线引脚定义，模块物理尺寸等等。目前市场上，各厂家都有其不同型号，不同功能的模块出售，使用是很方便的，能直接采用的就直接采用，不能直接采用的，就稍加修改后采用。在单独设计模块时，一般应依附于某厂家的产品进行设计，这对于简化设计过程，增强产品的生命力是有好处的。

## 3. 总线驱动

总线驱动问题应该全盘考虑，在一般的厂家系列产品中，大多数采用分布式总线驱动方式，导致器件数量相应要多一些，模块间有一些重复性的总线驱动器件。在智能仪器中，从提高性能价格比考虑，要求器件数量越少越好，只要能够完成仪器的基本功能，达到仪器各项技术指标要求，应该将器件降低到最低数量，所以一般采用综合式驱动方式。在主控模块设计中，充分考虑总线的驱动能力，把总线的驱动能力提高到能够满足组合成最大系统时的要求。设计从控模块时，不需要再考虑总线驱动问题，使从控模块设计简单化，不但节省器件，也减少组成系统的硬件模块数量。

## 4. 软件硬件的搭配关系

软件硬件的搭配关系，直接影响到系统的总体布局。这种搭配关系，往往是从仪器的基本功能出发，对各种功能进行具体的分析，不同的功能，采用不同的处理方法，获取最佳设计。

计算机完成的功能，大致有三个：一是管理功能；二是控制功能；三是数据采集和处理功能。三个功能既是独立的，但又是相互关联的。

在管理功能中，软硬件的搭配关系一般是固定的。但在不同的仪器中，则有不同的处理方法和表现手段。比如状态信息的提供，可以规纳处理，能用一个状态表示的就不用两个状态，这样不但可以节省器件，也给软件设计带来方便，增强对仪器的管理效率。计算机完成相应的控制功能，是系统功能中的重要环节，也是软硬件搭配关系最复杂的一个部份。一般情况下，只要能够用软件完成的工作，就尽量用软件去完成，而不用硬件，充分发挥软件的效力，把软件的功能扩大到最大限度。在数据采集和处理过程中，则主要是软件的工作。但是所获得的数据质量同软硬件的搭配关系是密切相关的，只有提高数据的质量，诸如数据的准确度、信噪比等等，才能更好的利用数学手段，对数据进行相应的处理。

# 三、接口电路的编址方法

接口电路的编址方法分为两种，一种就是 I/O 编址的 I/O，另一种就是存储器编址的

I/O。后一种编址方法的优点是，CPU对外设的操作可使用全部的存贮器指令，不需要专门的输入输出指令，其缺点是，这种编址方法占用了内存单元，减少了内存容量。

I/O 设备的编址方法可根据具体情况进行选择。在一些小的系统中，往往内存有足够的空间，选择存贮器编址的方法是比较适宜的。

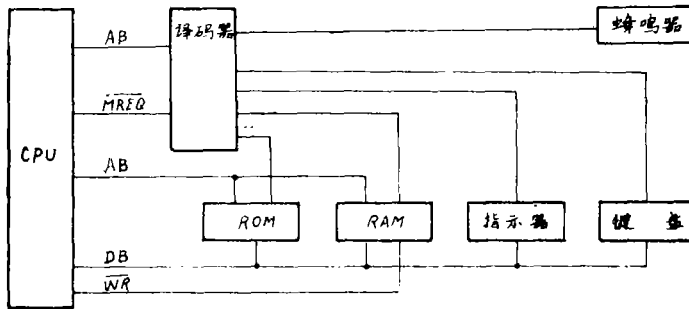


图1 游戏机逻辑框图

图1所示为某游戏机逻辑框图。该系统主机部份由CPU、ROM、RAM等芯片组成，接口电路共有三个。在这些接口中，数据传送都采用直接控制方式，采用存贮器编址方法，不但节省译码器，而且CPU对外的连线数量也减少，CPU工作在单一的存贮器数据交换状态，没有外围接口数据传送状态，对提高CPU的工作效率，是大有益处的。这种系统布局也给游戏机的进一步改进设计打下了良好的基础。

在上述例子中，I/O的编址可以选择两种方法中的任意一种。但在某些仪器中，I/O的编址方法，是不能随意选择的，是受系统总体布局和数据传送控制方式所制约的。在一些较大的系统中，一般采用I/O编址的I/O，以防挤压内存空间。

#### 四、步进电机的驱动与控制

在一些随动控制系统和光学仪器中，越来越多的采用步进电机，有时是多台。步进电机控制驱动接口的设计，按照仪器的工作状态和控制过程，加之软件硬件的相互搭配，可有不同的方案。

##### 1. 单电机驱动控制

图2为三相步进电机的驱动控制接口，图4为细分八相电机的驱动控制接口。由图中可以看出，整个接口电路，包括驱动和控制在内，已经简化到了最简程度。

步进电机的控制量为三个：方向、速度、转角。这三个量通过编程是很容易实现的。以图2为例，若采用六拍驱动控制的话，步进电机三相的通电顺序为：

正转：A—AB—B—BC—C—CA

反转：A—AC—C—CB—B—BA

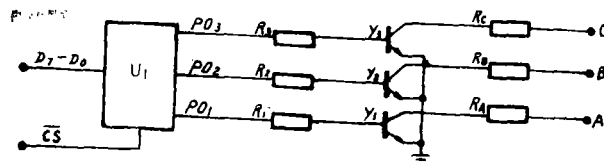


图2 三相步进电机控制驱动接口

由上述通电顺序可对应写出计算机每拍向锁存器送入的驱动码，应为：

正转：01—03—02—06—04—05

反转：01—05—04—06—02—03

据此写出最简程序框图如图3所示。

电机转动角度由驱动拍数 $N$ 决定。上述所列驱动码放在内存某存贮区。在方向调整中，用编程来实现，方法是比较多的。每输出一个驱动码，用程序延时环，决定两拍之间的时间间隔，实现电机转动速度的控制。

图4控制接口，其原理与图2所示相同，驱动控制程序框图也是一样，在各种方式下，驱动码分别如表1、2、3、4所示。

由表1至表4不难看出，在每一种驱动方式中，后一半驱动码与前一半驱动码比较，将高字节与低字节一交换，则两半的驱动码是相同的。因此在内存中只列出驱动码表的一半就可以了。

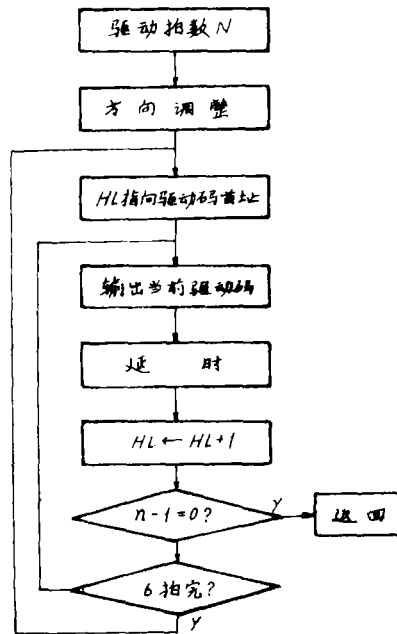


图3 步进电机驱动控制程序框图

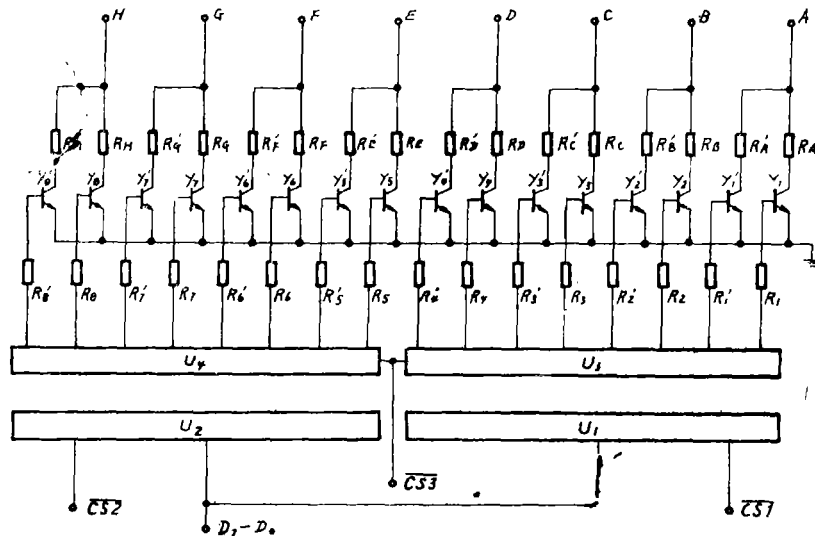


图4 二细分八相步进电机驱动控制接口

表1 双相八拍驱动码表

拍数次序		1	2	3	4	5	6	7	8
驱动码	高字节	00	00	00	03	0F	3C	F0	C0
	低字节	0F	3C	F0	C0	00	00	00	03

表 2 三相八拍驱动码表

拍数次序		1	2	3	4	5	6	7	8
驱动码	高字节	00	00	03	0F	3F	FC	F0	C0
	低字节	3F	FC	F0	C0	00	00	03	0F

表 3 二细分三相 16 拍驱动码表

拍数次序		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
驱动码	高字节	00	00	00	01	03	07	0F	1F	3F	7F	FC	F8	F0	E0	C0	80
	低字节	3F	7E	FC	F8	F0	E0	C0	80	00	00	00	01	03	07	0F	1F

表 4 双向 16 拍驱动码表 (二细分)

拍数次序		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
驱动码	高字节	00	00	00	00	00	01	03	07	0F	1E	3C	78	F0	E0	C0	80
	低字节	0F	1E	3C	78	F0	E0	C0	80	00	00	00	00	00	01	03	07

上述接口电路的优点在于所使用器件数量少, 用软件实现了对电机的控制, 通过改变程序的一些参数, 实现驱动方式的改变。其缺点在于占用 CPU 的时间较长, 补救的办法是, 可以在两拍之间不插入延时环, 而是用这段时间去做所需要做的工作, 如数码显示、数据采集等。

## 2. 群控技术

一台仪器若有多台步进电机, 这时就不是单电机的驱动控制, 而是群控。若多台电机中, 有两台或是几台电机的工作是同步的, 即速度是相同的或者成固定比例, 转角是相同的或者成固定比例, 在这种情况下, 完全可以把它们当成一台电机来对待。

若两台电机的工作不同步, 这时的状态是多种多样的, 在此我们仅讨论一种: 在任何时刻两台电机中只有一台工作的情况。如图 5 所示为其接口电路。

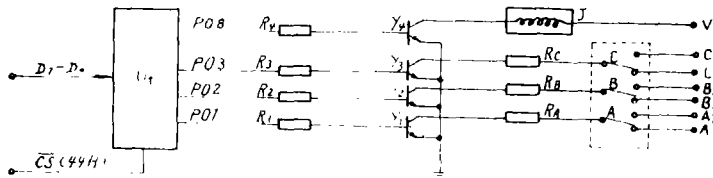


图 5 两台三相步进电机驱动控制接口

图中  $J$  为电机切换继电器,  $A_1, B_1, C_1$  为 1# 电机三相绕组,  $A_2, B_2, C_2$  为 2# 电机三相绕组, 图示为 1# 电机接通。与图 2 比较, 其差别只是多了一套继电器电路, 由继电器电路实现电机的切换。该接口实际上是完成了由一套电机驱动器控制两台电机, 节省了一套驱动器。

在电机切换时应注意以下几个问题:

(1) 保持原电机状态

无论是哪一台电机被接通时,一定要使它的各相通电状态与它被切断时的通电状态一样,防止切换时产生不应该的转动,引起转角误差。

(2) 无电流切换

电机驱动器中的功率管  $Y_1-Y_3$  的导通电流是比较大的,如果带电切换,必然引起较大的电火花,这时系统的稳定性是极为不利的。在继电器动作前切断电机的相电流,实现无电流切换,克服前述毛病。

(3) 切换时间延时

切换电机的顺序一般为:断电机相电流—驱动继电器动作—接通电机送初始驱动码。在每进行一步前都要有时间延时,这主要是考虑到大功率管电流较大,管子的转换时间、继电器的动作时间等都远远大于程序的执行时间。考虑到上述几个问题,以图5为例,设  $U_1$  为八  $D$  锁存器,接口的设备号为  $44H$ , 2#电机上次驱动最后一拍的驱动码为  $83H$ , 那么由1#电机切换到2#电机的切换程序如下:

```

RSS; SUB  A      ; 清累加器
      OUT  44H, A; 切断相电流
      CALL TT     ; 调延时程序
      LD   A, 80H
      OUT  44H, A; 使继电器动作
      CALL TT
      LD   A, 83H
      OUT  44H, A; 送初始驱动码
      HALT

```

上述接口电路可较容易的实现驱动多台步进电机。如果  $U_1$  采用八  $D$  锁存器,继电器电路采用线选法,则该接口可驱动六台三相步进电机,只使用一套电机驱动器。

## 五、诊断软件设计

智能仪器的诊断功能主要靠软件实现。在此我们仅对在单板机或专用性很强的小系统中如何实现这些功能的方式方法进行讨论。

### 1. 人一机信息交换

人一机信息交换是指仪器在测量过程中的不同时刻,仪器自动提出必要的问题,操作者给予回答后,仪器再去自动的实现操作者的意图。

如在单色仪中,开机后,要通过人一机信息交换,设置各种初始测量参数。仪器自动提出问题,首先就要造句,我们以起始波长为例,机器输出语句为:

```
START W AVEL = "nnnn"?
```

该语句对应的  $ASCII$  码字符串为:

53, 54, 41, 52, 54, 20, 57, 41, 56, 45, 4C, 3D, 27, 6E, 6E, 6E, 6E, 27, 3F;

将此字符串依次存贮于内存  $1700H$  开始的地址内,且  $1720H-1724H$  单元存贮内容为:

2E, AO, 2E, B2, OCH, 则机器自动输出该语句的程序如下:

```

II: LD    I, 1720H; 打印机初始化
      LD    BC, 5
      LD    DE, 2FOOH
      LDIR
PRINT: LD    DE, 2EAOH; 字符串送缓冲区
      LD    BC, 19
      LD    HL, 1700H
      LDIR
      CALL 1812H    ; 打印输出语句
      CALL DECKY   ; 等待输入

```

造句时, 应尽量把信息意思表达完整, 清楚易懂。该句中前两个单词指示当前机器需要输入起始波长, 等号后面的“nnnn”则指示起始波长用四位数字表示。其它的参数输入和起始波长是类似的。

## 2. 诊断功能设计

对输入参数要进行诊断。我们以光栅编号输入为例。在光栅衍射效率测试仪中, 光栅编号由1—11共11块光栅, 计算机输出的提示语句为:

```
GRAT# = "nn"?
```

之后程序调用键扫描显示程序DECKY。

### (1) 立即修改功能

在键扫描显示程序DECKY中, 程序等待输入两位数。如操作者输入“09”, 此时若确认该数正确, 则按数据输入结束键, “09”被接收。若输入“09”, 操作者认为该数是错误的, 应该输入“08”, 则在按结束键之前, 输入任一个数码, 程序自动抹除“09”, 重新等待输入, 直至数码输入正确为止。实现立即修改功能的程序如下:

```

LD    A, B          ; 取键值
CP    0AH          ; 是数字键吗?
JR    NC, TEXTD    ; 否, 转出
LD    (IX+0), A    ; 数码存缓冲区
INC   IX
DEC   (IY+1)       ; 取数计数
JR    NZ, LDATA    ; 未够, 返回取下一个数
LD    IX, DISMEM   ; 恢复IX指针
LD    A, 10H       ; 空白码送累加器
LD    B, 6
LD    HL, DISMEM   ; HL指向缓冲区首址
LOOP: LD    (HL), A ; 开始抹除前面输入数码
      INC   HL
      DJNZ LOOP
      LD    A, (IY+0) ; 恢复取数计数

```

```

LD      (JY + 1), A
JR      LDATA ; 返回重取
TEXTD: CP   ODH      ; 是结束键吗?
        TR   HZ, LDATA; 否, 返回取下一个数
        RET          ; 是, 返回主程序

```

注: LDATA是取数程序段入口

由上述程序可看出, 当数据没送够就按了结束键的话, 程序照常返回主程序, 此时没送够的数码, 程序自动赋“0”这对人一机信息交换来说是很方便的。

### (2) 自动诊断功能

程序返回到主程序后, 进入自动诊断, 诊断程序如下:

```

LD      A, (KK) ; 光栅号进入累加器
CP      12      ; 光栅号大于11吗?
JP      NC, PRINT; 是, 返回重取光栅号
DEC     A
JP      M, PRINT ; 光栅号等于 0, 返回重取
TR      NZ, TEXT2; 光栅号不等于 1, 转出
      :          ; 光栅号等于 1, 送相应参数
TR      GOTO    ; 处理完转继续
TEXT2: DEC  A   ; 光栅号等于2吗?
      :
GOTO:   :      ; 程序继续

```

由于程序设定光栅号为1—11, 经诊断后, 如输入的特征数大于11或者小于1, 则该数是非法的, 程序重新返回到输出提示语句处。

### (3) 人工中断功能

输入测量参数后, 经立即修改和自动诊断没有发现错误, 则程序进入扫描测量阶段。如上所述, 光栅号应为“09”, 但输入了“08”, 一旦进入扫描测量后, 操作者会立即发现错误的, 此时用人工中断方式, 将程序返回到输入测量参数处, 重新进行人一机信息交换。该功能的实现方法是, 在按了人工中断键后, 重新起动人一机信息交换程序入口。

经上述三道“防线”, 一般能确保排除错误, 正确进行测量。

## 六、高效率程序模块设计

提高程序模块的效率, 增强模块的功能, 对仪器是大有好处的, 特别对一些小的系统。在此, 我们仅以光栅单色仪键扫描显示程序DECKY模块为例进行讨论。

### 1. 键扫描显示程序的功能

#### (1) 显示功能

在人一机信息交换时显示操作者输入的初始参数。

在扫描测量阶段显示波长现行位置和单位。

在扫描测量结束后, 显示仪器终止状态。

(2) 键扫描功能

键扫描功能包括：检查是否有键按下；求被按下键的键值；区分真伪按键。

(3) 立即修改功能

人一机信息交换时，可对输入的参数进行立即修改。

2. 特性“软件开关”的设置

由于 DECKY 模块在程序执行的不同时刻，其功用有所差别，使用特别频繁，进入该模块的方式也有所不同，一种是调用，再一种是跳入，同时具有子程序和主程序的特性。为使模块具有双重性，在程序中设置了“软件开关”——CALL 标志。当 CALL = “1”时，则为调用，当 CALL = “0”时，则为跳入。模块执行完公共部份后，判断 CALL 标志，转入相应的分支程序。

3. DECKY程序模块方框图

键扫描显示程序的粗略方框图如图六所示。

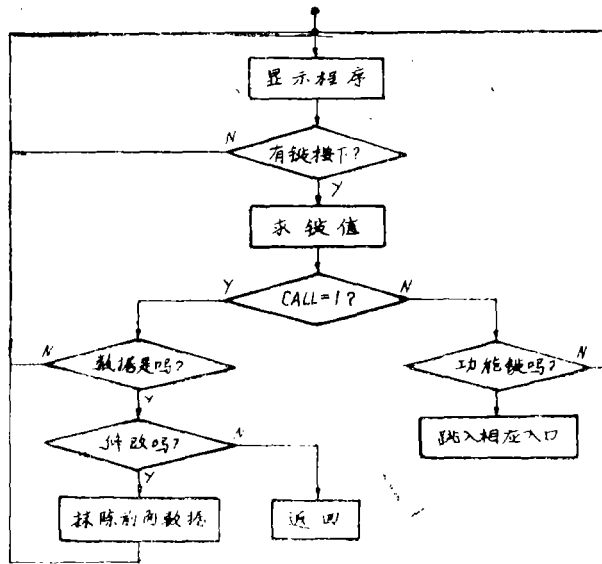


图 6 DECKY程序框图

该程序前半部份是公共的，判断CALL标志后，是调用，程序执行左半部份，是跳入，程序执行右半部份。

4. DECKY程序模块的具体使用

使用该模块时，如要调用，首先将CALL标志置“1”，在程序返回时，先复位CALL标志，再执行返回指令。跳入该程序时不必考虑CALL标志。在光栅单色仪中，人一机信息交换时，用CALL指令调用，扫描测量完毕程序跳入DECKY显示终止状态，等待新的命令。

值得指出的是，该模块在两种方式下执行时，求得键值后，对键值的处理是相互隔离的，这给操作带来很大方便。这种程序模块的设计，往往是要根据仪器的具体技术要求和作过程进行，具体问题在处理方法上并不是唯一的。

## 七、数据的迭加测量

数据的测量，是智能仪器的重要功能，所获得数据的质量直接影响到仪器的主要技术指标。

### 1. A/D变换器抖动的消除

影响A/D变换器的因素是很多的，这些因素的变化，都会导致A/D变换器的抖动，引起数字量的变化。表五是一个12位A/D变换器在不同模拟量输入情况下的输出情况。

表5 12位A/D变换器输出数据

输入模拟量 (mV)	102	805	994	1103	1398	2499	2604	4001
第一次输出数据	79	632	788	875	1111	1985	2064	3169
第二次输出数据	78	640	789	873	1107	1979	2062	3168
第三次输出数据	80	640	791	873	1106	1979	2064	3169
第四次输出数据	82	639	790	874	1106	1980	2063	3171
第五次输出数据	81	638	791	875	1107	1979	2062	3171
理论数码值	81	638	790	874	1108	1980	2063	3170

注：A、工作电压、参考电压5.168V B、A/D变换为逐次比较式 C、输入模拟量用干电池

由表五看出，第一次输出数据相对理论值，有的偏差较大，后面的相对偏差小一些。对偏差大的几个数据进行分析，不难看出，其输入模拟量相对上次A/D变换时的输入模拟量的变化幅度都较大，而变化幅度小的，偏差也小。这种A/D变换的抖动，基本上属于A/D变换器自身的问题。用程序的方法将第一次的数据抹去不用，消除这种抖动。在模拟量变化比较大，采样间隔较大的情况下，采用这种消抖动的方法，数据的质量相对要好一些。

### 2. 迭加测量的实现

表6 12位A/D变换输出数据的迭加和平均

输入模拟量(mV)	102	805	994	1103	1398	2499	2604	4001
第二次输出数据	78	640	789	873	1107	1979	2062	3168
第三次输出数据	80	640	791	873	1106	1979	2064	3169
第四次输出数据	82	639	790	874	1106	1980	2063	3171
第五次输出数据	81	638	791	875	1107	1979	2062	3171
迭 加	321	2557	3161	3495	4426	7917	8251	12691
平 均 值	80	639	790	874	1107	1979	2063	3170
理 论 值	81	638	790	874	1108	1980	2063	3170

使用迭加法采集数据，有两个目的。其一是将迭加采到的数据进行算术平均，以便消除一些 A/D 变换器以外的因素所引起的数据抖动。其二是为了提高信噪比、特别在一些弱信号测量中，将迭加采到的数据，进行数字滤波处理，获得高质量的数据。

表 6 所列数据是在表五基础上，去掉第一次的输出数据进行迭加和算术平均获得的数据。不难看出，平均值与理论值比较，最终结果，其精度已经接近 12 位 A/D 变换器的理想精度。

实现迭加测量的程序如下：

```

LDA: LD A, 4 ; 四次取数计数
      LD BC, 0
      DI
      OUT (A/D), A ; 起动 A/D 变换
      EI
      HALT ; 等待中断
LDA4: DI
      OUT (A/D), A
      EI
      HALT
      ADD HL, BC ; 求迭加和
      LD B, H
      LD C, L ; 和送 BC
      DEC A
      JR NZ, LDA4 ; 四次未完返回
      LD L, (NDD)
      LD (HL), B
      INC HL
      LD (HL), C
      INC HL
      LD (NDD), HL
      HALT

```

注：NDD 中的内容是存数地址指针

实践中应用迭加测量方法获得了较好效果，这种方法在一般仪器中都是可行的。

## 八、结 束 语

本文所讨论的问题，都是在科研题目实施中的一些体会和所遇到的一些具体实际问题，以及对这些问题的具体处理方法。智能仪器中的计算机系统设计问题涉及面是很广泛的。针对不同种类的仪器，遇到的实际问题和处理解决的办法也必然各有所不同。在上述涉及到的一些科研题目中，主要参加者有乔燕燕、李湜、王盾、胡君、宣殿武等人。

### 参 考 文 献

- [1] 北京工业大学, 《TP801Z80单板机使用手册》1981.4.
- [2] 周明德, 《微型计算机硬件、软件及其应用》1983.
- [3] A. Lesea R. Zaks, 《Microprocessor interfacing techniques》1978.
- [4] Gerge D. Kraft, 《Mini/Microcomputer Hardware Design》1979.
- [5] ALAN. R. Miller, 《The 8080/Z80 Assembly Language》1982.
- [6] Robert C. Bruce, 《Software Debugging for microcomputer》1981.
- [7] 杨润生, 王敬觉, 《微型计算机及其应用》1983.
- [8] G. B. Clayton, 《Data Converters》1982.

## The Design of a Computer System in Intelligence Instrument

Xing Zhongbao

### Abstract

This paper discusses design problems of a computer system in intelligence instrument and processing method to meet requirements in our works, Considering basic design method of the computer software, hardware and interface.