

一般以微伏、毫伏计，因此地线更要小心设置。放大电路本身采用一点入地，不能一个电路多点接地，否则地线中的电位差将对放大电路产生干扰；A/D前置放大电路一般浮空；内存放大电路的印刷电路板上一点入地。这类放大器的地线一定要远离功率地和噪声地(即继电器、电动机等的地)。

6.3.3 隔离与耦合

在抗干扰措施中，还采用各种隔离与耦合的方式来提高系统的抗干扰能力。使用这种方法可以让两个电路相互独立而不形成一个回路，例如在系统中既有数字电路，又有模拟电路，当输入的模拟信号很小时，数字电路会对模拟电路产生较大的干扰，所以在实际的电路设计中应该避免数字电路和模拟电路之间有共同的回路，即将二者加以隔离。此外，检测系统中单片机与数字电路、脉冲电路、开关电路的接口，一般也用光电耦合器进行隔离，以切断公共阻抗环路，避免长线感应和共模干扰。高增益的放大器(>60 dB)，需要在输入级设级间耦合。在需要采用较长信号传输线的场合，可以采用屏蔽与光电耦合相结合的办法。

常用的隔离方法有光电耦合器件隔离、继电器隔离、隔离放大器隔离和隔离变压器隔离等。光电耦合器件响应速度比变压器、继电器要快得多，对周围电路无影响，并且体积小、重量轻、价格便宜、便于安装，线性光电耦合器用在模拟电路中的信号线性变换场合，也用在放大器的隔离中，如图 6.6 所示为采用光电耦合器隔离前级放大和后级放大电路的方法。其中， I_b 为偏置电流，两个光电耦合器组成互补的形式，以改变放大电路的线性度，减少温度影响。虽然线性光电耦合器的线性度好，但其转换精度较低，信号的动态范围也较小。所以现在大量使用的是用于数字量、开关量变换的光电耦合器。如图 6.7 所示为几种使用光电耦合器进行隔离的方式。

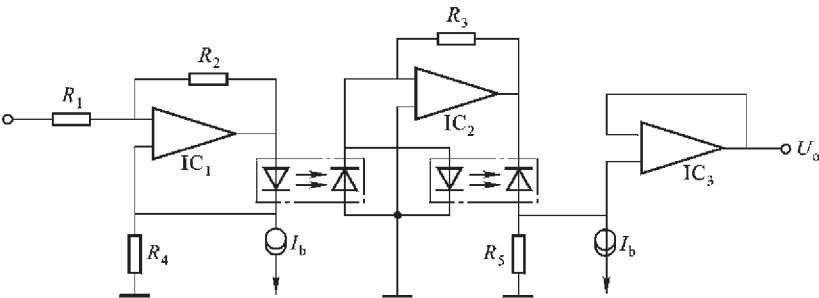


图 6.6 放大器间的光电隔离

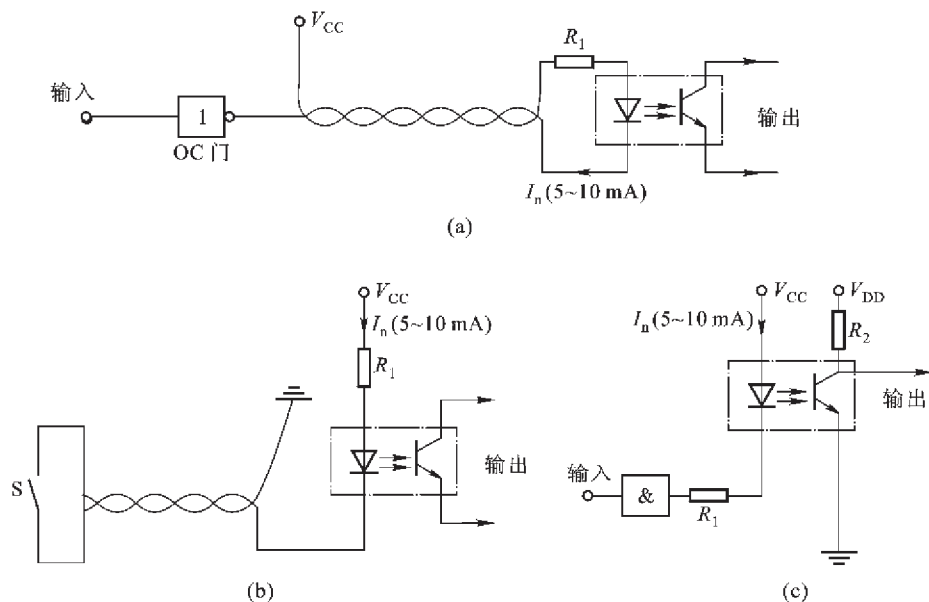


图 6.7 采用光电耦合器隔离的方法
(a) OC 门和光电耦合器的连接 (b) 接点和光电耦合器的连接
(c) 和门和光电耦合器的连接

6.3.4 布线抗干扰措施

在检测系统中，印制板上电力线、信号线等线路的布局、板上器件空余管脚的安排、测试设备与仪器仪表的信号传输线的连接等，都是实际应用中要考虑的问题。

(1) 走线原则

在长线传输中，为了防止窜扰，行之有效的办法是采用交叉走线法。长线传送时，应遵循功率线、载流线和信号线分开，电位线和脉冲线分开的原则。在传送 $0 \sim 50 \text{ mV}$ 的小信号时，更应该如此。

电力电缆最好用屏蔽电缆，并且单独走线，与信号线不能平行，更不能将电力线与信号线装在同一电缆中。

(2) 元器件空余输入端的处理

电路设计中常常会出现器件管脚空余的现象，一般不能将这些管脚随意处置，特别是元器件空余输入端，处理不好往往可能造成较大的干扰输入，所以应采取一定的处理方法，以降低干扰。实践中常采取如下方法：

把空余的输入端与使用输入端并联，这种方法简单易行，但增加了前级电路的输出负担；

把空余的输入端通过一个电阻接高电平,这种方法适用于慢速、多干扰的场合;
把空余的输入端悬空,用一反相器接地,这种方法适用于要求严格的场合,但多用了—个组件。

(3) 数字电路的抗干扰措施

每一块数字电路组件上,都有高频去耦电容,一般为 $0.01 \sim 0.02 \mu\text{F}$ 。在布局上这些电容应充分靠近集成块,并且不应集中在印刷板上某—端。每块印刷板上的电源输入端也应加 $10 \sim 100 \mu\text{F}$ 的去耦电容。直流配电线的引出端也应尽可能地做成低阻抗传输线的形式。

前面介绍过快速逻辑电路会产生高频干扰,所以这些电路也应按高频电路来处理。所有装有大量逻辑电路的印制板,都必须有良好的接地。实际设计中,这个地可以是低阻泄流排,或者用印制板上大面积的铜箔作接地,其接地面积应占印制板面积的 60% 以上。这个接地面对供电回路提供一个低感回路,并可为信号交连电路提供一个阻抗固定的线路。此外,在检查印制板的接地时要注意是否接触良好、可靠。

各种逻辑电路产生噪声的程度也是不同的,其中 TTL 产生的噪声最大,而 HIL 产生的噪声最小。一般来说,开关速度越快,噪声越大(ECL 除外,其电路的供电电流,在导通和截止时都一样,在门的开关中,电流变化率为零,因此它产生的噪声低)。另外,门电路的速度与传播延时成比例,ECL 的速度最高,HIL 最慢,通常 TTL 的速度较 ECL 略低,但噪声却要大 10 倍。

数字电路对噪声敏感。RTL 对噪声极为敏感,而 HIL 和 CMOS 最不敏感。

6.3.5 软件抗干扰措施

干扰不仅影响检测系统的硬件,而且对其软件系统也会形成破坏。如造成系统的程序弹飞、进入死循环或死机状态,使系统无法正常工作。因此,软件的抗干扰设计对计算机检测系统是至关重要的。此外,前面介绍的干扰抑制技术,都是采用一定的硬件措施。随着抗干扰技术理论与实践的不断深入,除了研究常规的抗干扰措施外,掌握信号与噪声的规律、区分噪声与信号的性质的研究等也是非常重要的。实际的工程应用中,仅采用硬件措施往往满足不了需要,所以应该寻求软件方法。在软件方法中已有不少有效的措施,如数字滤波、选频和相关处理等,这些软件处理程序可以方便地提取淹没在噪声中的有用信号。而实践中将硬件方法和软件方法结合起来,可以达到良好的干扰抑制效果。设计实践证明:软件抗干扰不仅效果好,而且降低了产品成本。在系统运行速度和内存容量许可的条件下、应尽量采用软件抗干扰设计。

(1) 干扰对系统软件的影响

外界的干扰对智能仪器仪表中单片机系统产生干扰后,可能引起 RAM、

程序计数器获得的总线上的数字信号发生错乱，CPU 根据错误数据进行运算，得到错误的操作数和结果，在没有任何纠错措施的情况下，这个错误会一直传递下去。CPU 读取到错误的地址码后，会造成程序运行偏离正常轨道，引起程序失控。程序失控后，有时可能又会回到正常运行状态，但是已经造成一系列不良后果，或者对后续程序的正常运行埋下隐患，这样的情形甚至可能使得程序进入死循环，造成系统的完全瘫痪。

计算机系统内的片内 RAM、扩展 RAM 以及片内的各种特殊功能寄存器等，都可能因外界干扰引起状态的变化。这些数据状态的正常与否，关系着整个装置或系统的显示、控制、运行等，是非常重要的。如果某个中断专用寄存器的数据发生错误，中断设置方式的改变，可能造成有用中断的关闭和未用中断的开启，使意外的非法中断得以进入。此外，造成数据误差、控制失灵、某些部件的工作状态改变等，都是干扰可能对系统软件造成的影响。这些问题使得在系统软件的设计与维护中必须采取措施，以维持其正常、稳定的运行状态。

(2) 软件抗干扰的主要措施

1) 数字滤波

检测系统的输入信号和外界的干扰有时是随机的，故其特性往往只能从统计的意义上描述。此时，经典滤波方法就不可能把有用的信号从测量结果中分离出来，而数字滤波具有较强的自适应性。所谓数字滤波，就是通过一定的计算程序对采样信号进行平滑处理，提高其有用信号，消除或减少各种干扰和噪声的影响，以保证系统的可靠性。

例如，对于 N 次等精度数据采集，存在着系统误差和因干扰引起的粗大误差，使采集的数据偏离真实值。此时，可以采用算术平均值的方法，求出平均值作为测量示值。还可以在此基础上，将剔除了粗大误差的测量数据的平均值作为测量结果示值。这样既剔除了粗大误差，又可以消除一定的系统误差。在综合考虑适当的 N 值后，可以在满足测量精度要求的前提下，拥有足够的测量速度。式(6.10)为该数字滤波方法的表达式。

$$\bar{X} = \frac{1}{N-m} \sum_{i=1}^{N-m} X_i \quad (6.10)$$

式中， X_i 为第 i 次的测量值， m 为粗大误差数。

对于主要去掉脉冲性质的干扰为目的场合，可以采用中值滤波法。即对某一个被测量连续采样 N 次(一般取奇数)，然后将这 N 次的采样值从小到大或从大到小排队，再取中间值作为本次采样示值。另外，这个数字滤波方法只要改变循环次数，则可以推广到对任意采样值进行中值滤波，而且 N 值越大，滤波效果就越好。但是在实际应用中不可能把 N 值取得太大，一般取 $5 \sim$

9 即可，以控制总的采样时间。

上面简要介绍的数字滤波方法主要适用于变化过程比较快的场合，基本上属于静态滤波，如压力、流量等参量。对于慢速随机变量，这些滤波方法效果并不太好。所以，要采用动态滤波的方法，即一阶滞后滤波方法。式(6.11)为该法的表达式。

$$\bar{Y}(k) = (1 - \alpha) X(k) + \alpha \bar{Y}(k-1) \quad (6.11)$$

式中， $X(k)$ ——第 k 次采样值；

$\bar{Y}(k)$ ——第 k 次采样后的滤波结果输出值；

$\bar{Y}(k-1)$ ——上次滤波结果输出值；

α ——滤波平滑系数， $\alpha = \tau / (\tau + T)$ ；

τ ——滤波环节的时间常数；

T ——采样周期。

通常采样周期要远小于滤波环节的时间常数，也就是输入信号的频率快，而滤波环节时间常数相对大，这是一般滤波器的概念，所以这种数字滤波器相当于硬件电路中的 RC 滤波器。对于 τ 、 T 的选取，可以根据具体情况而定。一般 τ 越大，滤波的截止频率就越低，相当于 RC 滤波电路中电容增大，而硬件电路中的电容增加是有限的，数字滤波器中的 τ 值则是可以任意取值的，这也是数字滤波器可以作为低通滤波器的原因。滤波系数 α 的确定可以根据选用的 τ 和 T 计算得到，其值一般为小数，所以 $(1 - \alpha)$ 也是小数。

每一种数字滤波方法都有其各自的特点，在实际应用中要根据具体的测量参数来选用合适的方法。除了可以根据所测参量变化的快慢情况来选取外，要注意滤波效果与所选择的采样次数 N 有关， N 越大，效果越好，只是花费时间越长。所以，在考虑实际滤波效果达到要求的前提下，应采用运行时间较短的程序。此外，在热工和化工过程 DDC 系统中，要结合实际情况决定是否一定用数字滤波方法。因为不适当的数字滤波方法有可能将要控制的波动滤掉，从而降低控制效果甚至不能控制。

2) 软件陷阱

软件陷阱是通过指令强行将捕获的程序引向指定地址，并在此用专门的出错处理程序加以处理的软件抗干扰技术。前面提到干扰可能会使程序脱离正常的运行轨道，软件陷阱技术可以让弹飞了的程序安定下来。从汇编语言的角度，软件陷阱可用一条长跳转指令来实现，如 MC-51 单片机指令可用 `LJMP ERTREAT` (错误处理程序入口)，为加强捕捉效果，一般还结合指令冗余，即为抑制 CPU 受干扰后把操作数当成指令码来执行，而加入的不可能把操作数当成指令码来执行的两个字节的空操作指令 `NOP`，所以软件陷阱一般有三条指令构成

```
NOP
NOP
LJMP ERTREAT
```

程序中安置软件陷阱的位置是很考究的，通常可以在未使用的中断向量区、未使用的程序存储区、表格末端、程序区的断裂点等位置设置相应的陷阱。

应该注意的是，当程序因干扰而弹飞，进入到非正常区域，若被这一位置事先设置好的软件陷阱捕捉，这时应引导程序先期进入出错处理程序，而不要简单地将程序引导到某段程序的起始处。

在未使用的中断向量区设置陷阱，利用的是各种 CPU 所约定的中断入口地址。例如 MCS-51CPU 有 5 个中断源，对应每个中断源的中断子程序的入口地址都有明确的规定，不能任意选择。

中断源	中断入口地址
外部中断 INT0	0003H
定时器中断 T0	000BH
外部中断 INT1	0013H
定时器中断 T1	001BH
串行通道中断	0023H

所以中断服务子程序可以放在程序存储区的任何位置，但在上述中断入口地址单元必须放一条跳转指令，指向相应的中断服务子程序(对于软件设计中开放了的中断而言)。这时陷阱的设置就是在未使用的中断所对应的中断入口地址单元中放一条跳转指令，让因干扰而激活的中断服务进入出错处理程序。

在程序寄存器 EPROM 的使用中，常常会遇到剩余大量未编程的空间的情况。这些未被使用的单元一般呈现 0FFH 状态，对于 8051 指令系统则是 MOV R7, A 这一指令。如果程序弹飞到此，将进入不再跳转的假停机状态。如果在这些未编程的空间，以每隔一定距离设置一陷阱，加上一条跳转指令，引导程序到出错处理程序，就可以解决该问题。

在程序区中不能任意设置陷阱，否则将影响程序的正常执行，故一般是在程序的断裂点处设置陷阱。单片机指令系统中 SJMP、LJMP、AJMP、RET、RETI 等指令执行后，程序计数器 PC 的值应正常跳变，而不应顺次向下运行，否则将出现错误。这时设置软件陷阱就是在正常程序执行不到的地方放置冗余指令和跳转指令，以捕捉弹飞的程序并引导到出错处理程序。

实际上，在编制汇编语言时，结合程序流程框图为各个主、子程序插入软件陷阱，并在系统运行中加以调试、调整，可以收到良好的软件抗干扰效果。

3) “Watchdog” 技术

“Watchdog”俗称看门狗，即监控定时器，是计算机检测系统及智能仪器仪表系统中普遍采用的抗干扰和可靠性措施之一。

“Watchdog”有多种用法，其主要的应用则是用于因干扰引起的系统程序弹飞的出错检测和自动恢复。它实质上是一个可由 CPU 复位的定时器，原则上由定时器以及与 CPU 之间的适当的输入/输出接口电路组成，如振荡器加上可复位的计数器构成的定时；各种可编程的定时器/计数器(如 Intel 8253 \ 8254 等)；单片机内部的定时/计数器等。

如图 6.8 所示为“Watchdog”的工作原理，两个计时周期不同的定时器 T_1 、 T_2 ，如两定时器的时钟源相同，设定 $T_1=1.0\text{ s}$ ， $T_2=1.1\text{ s}$ ，用 T_1 定时器的溢出脉冲 P_1 同时对 T_1 、 T_2 定时器清零，那么只要 T_1 定时器工作正常，则 T_2 定时器永远不会有计时溢出。一旦定时器 T_1 不再计时， T_2 定时器就会计时溢出，产生溢出脉冲 P_2 ，表明 T_1 出了故障。这里 T_1 是应用程序的周期， T_2 即是“Watchdog”，利用输出脉冲 P_2 并进行巧妙的程序设计，可以检测出系统的出错，使弹飞的程序重新回到正常轨道中。

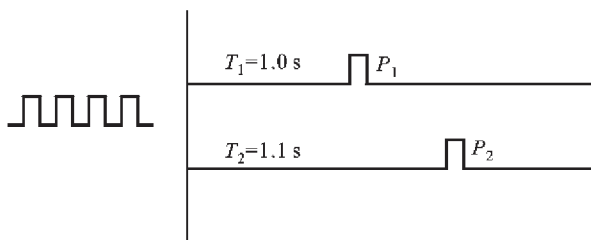


图 6.8 “Watchdog”工作原理

如图 6.9 所示为“Watchdog”的构成，实际上它是一个和 CPU 构成闭合回路的定时器。当程序飞到一个临时形成的死循环里，或者 PC 指针落到 EPROM 芯片范围外，即全地址区之外时，系统将完全瘫痪。“Watchdog”可以代替人工对系统自动复位，使 CPU 从死循环或弹飞状态回到正常的程序流程中。

“Watchdog”的输出连接到 CPU 的复位端或中断输入端。“Watchdog”的每一次溢出输出都将引起系统的复位，使系统重新初始化；或产生中断，使系统进入故障处理程序，进行必要的处理，自动恢复正常的运行程序。只要在定时器的定时到脉冲产生之前，即“Watchdog”动作之前，CPU 访问定时器一次，则定时器均会重新开始计时，故也不会有定时到脉冲的输出；否则，如有定时到脉冲输出，说明定时器所监视的程序进程发生了问题，需要系统复位或中断。

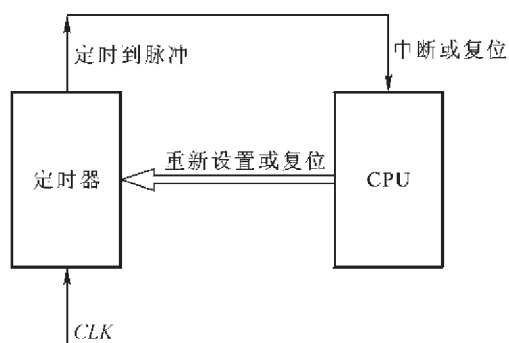


图 6.9 “Watchdog” 的构成

在一般的应用中，可以把一个程序分成若干个进程或者任务，如图 6.10 所示，然后以不同的进程或任务分别设置“Watchdog”的定时值，从而达到监控每一个进程或任务的目的。

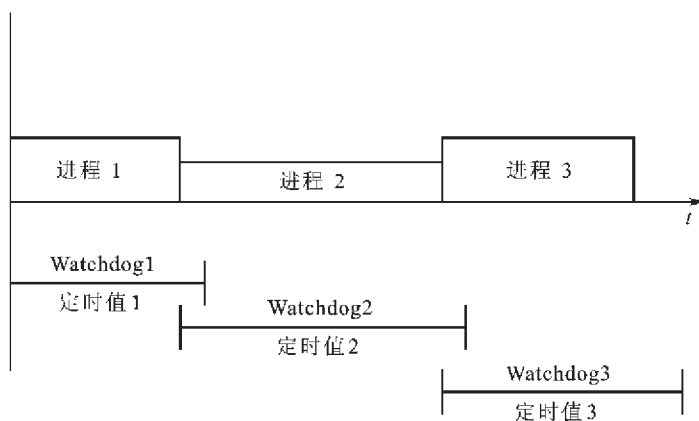


图 6.10 以进程设置“Watchdog”的定时值