EB398/D
版本 0，08/2002

防止由于代码跑飞而导致 MCU 应用故障的技术

作者：Peter Topping
East Kilbride

引言

MC68HC(9)08 系列 MCU 具备防止代码跑飞的功能，而且即使发生了代码跑飞，它也能防止应用出现故障。导致代码跑飞的原因可能是错误的代码、超出规范允许范围运行 MCU、或者是严重的 EMI 或电气噪声事件。从定义来看，并没有明确指出在代码跑飞期间会出现什么情况，但是由于它是超出规范运行环境而导致的，很容易破坏程序计数器，从而导致 MCU 出现不可预测的行为。

在可能发生这种现象的 MCU 应用中，我们建议采取下文中介绍的各种预防措施。但是即使采取了这些推荐预防措施，在异常情况下，仍然存在着较小的代码跑飞的可能性。正因为如此，我们还介绍了能够防止 MCU 或应用硬件在这种情形下遭到损坏的技术。在下列应用中，这些预防措施尤为重要：使用了任何类型的片上或外部非易失性存储器（闪存、EEPROM 或备份 RAM）的应用、以及外部硬件有可能进入一种异常状态，甚至是破坏状态的应用。

根据定义，在代码跑飞后，MCU 的运行是不可预测的，甚至不能相信其 I/O 端口会继续输出可接受的状态。这就有可能使其端口进入一种不可预测的状态，进而导致外部硬件也进入不可预测的状态。

在使用非易失性存储器的应用中，储存器中的内容有可能被 MCU 的失控行为破坏。如果闪存或 EEPROM 存储器中包含应用代码，问题会更严重。如果代码遭到破坏，整个应用可能都无法运行，而且不可能通过局部的重新编程来恢复。这就要求我们通过在线重新编程，甚至可能必须更换包含 MCU 的整个 PCB，来修复这些设备。

freescale™
飞思卡尔半导体

在基于 ROM 的简单应用中，可能不使用非易失性存储器，也无需关心 I/O 出现无法预测的状态。在这种情况下，采取预防措施来防止代码跑飞就显得不那么紧迫，尽管它仍然具备一些作用，例如在电压下降时防止 LED 指示灯或其它显示器的混乱闪烁。

即使事先没有迹象表明一个应用可能出现某种有害的故障，我们还是强烈建议您采取所有必要的预防措施，以防止代码跑飞，这是一个非常好的做法。这些建议都是一些老生常谈，但人们有时会置若罔闻，因此也并非总是被采用的。虽然我们在这里讨论的是 MC68HC(9)08 这款特定的器件，但这些技术适用于所有 MCU 的应用。

1. 防止代码跑飞

在上电和掉电过程中，代码跑飞都是潜在的风险。这两种情况要以不同方式进行处理。

1a. 掉电保护

防止代码跑飞的最重要的方法就是低电压禁止功能（LVI）。当 Vdd 电源降低到规定的最小值以下时，LVI 能让 MCU 保持复位状态。尽管 LVI 可以在电源故障时保护应用系统，但它最重要的作用还是在有意关断电源时出现的瞬时状态下提供保护。如果没有采用某种形式的 LVI，则在每次关闭电源时，代码跑飞的可能性会非常大。实施 LVI 有两种可选方式：内部和外部。

显而易见，片上 LVI 是我们的首选，因为它不会产生任何额外成本。在使用带内部 LVI 的 MCU（所有 MC68HC(9)08）、而未采用外部 LVI 的应用中，应该始终启用内部 LVI。两个 CONFIG 寄存器位 LVRSTD 和 LVIPWRD 都应为零。这是它的默认状态，但是在带 3 伏选项的芯片中，要注意使用正确的电压。默认的电压是 3 伏，因此，在 5 伏应用中，应该将相应的位(LVI5OR3)置 1。即使 CONFIG 寄存器中的默认值是正确的，也应在应用代码的开头，写入这个寄存器和所有其它的一次性写入寄存器。这样可以提供更高的安全性，因为即使在代码跑飞后，试图写入这些寄存器的操作也会被禁止。

7	6	5	4	3	2	1	0
	LVISTOP	LVIRSTD	LVIPWRD	LVI5OP3			

CONFIG 寄存器中的 LVI 控制位

在规定可以低至 2.7 伏电压运行的芯片中，例如 MC68HC(9)08KX8 或 MC68HC(9)08GP32，LVI 可以在 5 伏应用中提供出色的保护。它可以保证在 3.9 至 4.5 伏之间的某个电压下启动，从而确保电源电压在远没有降到规定的最低 Vdd 电压之前，就将复位信号保持为低。为了得到最好的结果，应该使用法定的 3 伏总线速度（不超过 4MHz）。这可以确保到低至 2.7 伏的整个电压范围下设备都正常运行。

如表 1 所示，MC68HC(9)08 的 LVI 的 5 伏规范能保证当电压低于启动点时（在 3.9 到 4.5 伏之间），将复位口维持低电平。在只有 5 伏的 MCU 中，如 MC68HC(9)08AZ60，在 4.5 伏以上可以保证正常运行。尽管在 LVI 启动点之上，设备应该能正常运行，但实际上是无法保证的。在对安全性要求很高的应用中，应该使用外部 LVI。这样可以选择在电源电压降到 4.5 伏以下之前启动，就百分之百地确保了不会要求芯片在额定电压范围之外执行代码。图 1 中给出了适用的外部 LVI 器件，包括 MC33064 和 MC34064，以及相应的电路。

表 1. LVI 启动电压

V_{DD}		LVI 启动点（电压下降）	
	最小	典型	最大
3 伏	2.45	2.60	2.70
5 伏	3.90	4.25	4.50

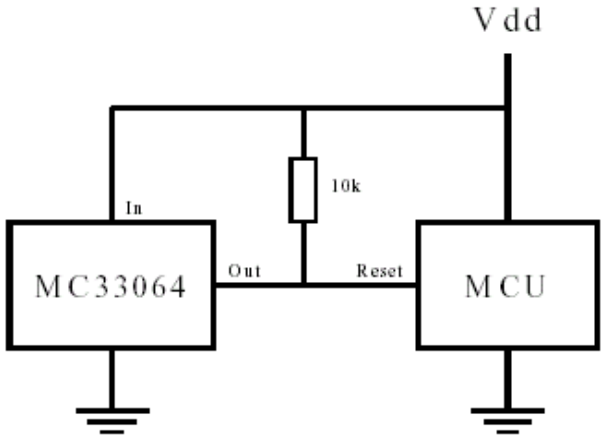

图 1：外部 LVI

当电压下降时，有些应用可能需要将状态信息保存到内部或外部的非易失存储器中。LVI 就可以用于这一用途，但应注意避免超出规范以外的运行。如果启用了 LVI，但没有启用复位功能，则可以通过轮询 LVI 状态寄存器中的 LVIOUT 位，来根据需要启动信息转移。然而，只有当电源电压仍然有效时，才可能进行操作。因此只有在采用 5 伏 LVI 启动点，而且按规定可以低至 3 伏运行的应用中，这种方法才是适用的。只要采用 4MHz 或更低的总线速度，就可以实现这一功能，例如在 908GP32 或 908KX8 中。

如果采用了这种方法，设计者就必须确保存储电容足够大，能在电压达到 3 伏 [1] 之前，有充裕的时间来执行所需的代码。很显然，这个方法不能提供全面的 LVI 掉电保护，我们一般不推荐使用这一方法。

1. 如果想要代码在低于规范的 Vdd 上运行，则应在 RAM 上运行，而不是在闪存上运行。这是因为与闪存相比，RAM 能在更低的电压下正确读取数据。但是，如果 Vdd 低于规定的最小值，这两种方法都不能保证正确的运行。

还有一些启动这种操作而不使用 LVI 的方法，从而能打开 LVI 的全部功能（包括复位功能）以备使用。然而，它们需要一个 MCU 引脚和一些外部硬件。当然，只需要一个 2 脚的稳压器或 3 脚的低电压检测（LVI）芯片就可以了。可能的方法包括：将 LVI 芯片（如 MC34064）的信号送入轮询口，或者用一个 A/D 口监测带隙稳压器（如 LM385-1.2）。在后一种方法中（见图 2），随着电源电压下降，A/D 参考电压也随着 Vdd 一起下降，而外部参考信号还保持有效电平。结果就是随着电源电压的降低，参考信号的 A/D 测量值反而上升，而测量值就可用于在希望的电压下启动相应的程序。

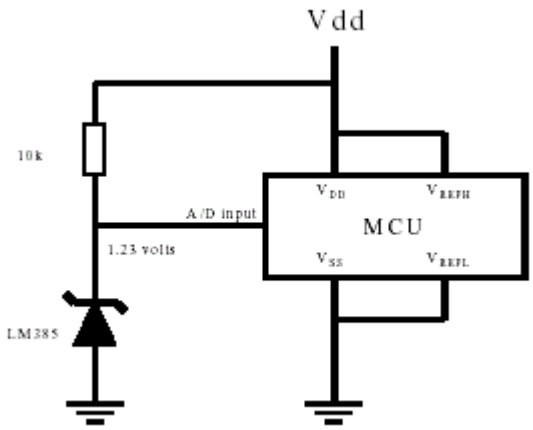

图 2：外部电压参考信号

在有些电池应用中，可能有两个甚至三个阈值，在不同的阈值处要采取不同的措施（保存信息、关断大功耗硬件等）。采用 A/D 的方法通过一个 MCU 引脚就能满足这个要求。

1b. 上电保护

掉电时，内部 LVI 可以控制复位脚，但在上电过程中也要当心。内部的上电复位（POR）电路在振荡器启动前（一般是在 Vdd 达到 2 伏的几毫秒后），会保持复位脚为低电平，然后再持续 4096 个时钟周期的低电平。如果这段时间还不够长，无法确保电源电压在复位脚变为高平前达到规定的范围，那么就需要采取措施，使复位脚维持更长时间的低电平。诸如 MC34064 之类的外部 LVI（见图 1）可以在电源电压低于启动点前保持复位脚为低电平。如果不采用外部 LVI，最常见的方法就是在复位脚上加一个电容，以及上拉电阻器，如图 3 所示。这样能充分延迟其上升时间，以使电源达到规定的最小值。设计人员必须利用他们对系统电源的了解，确定合适的时间常数。典型参数是 10kΩ - 100kΩ和 0.1µF - 1.0µF。如果采用这种复位电路来提供上电保护，则应该始终启用内部 LVI 以提供掉电保护。

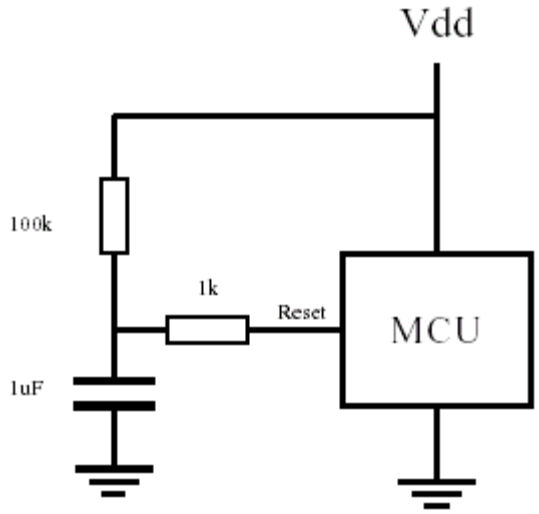

图 3. 简单的外部上拉和电容

图 3 中与复位脚串连的电阻是可选的。由于内部生成的低电平复位信号在输出至引脚前有缓冲，因此这个电阻并非绝对必需。然而，我们还是建议您用示波器检查一下复位脚在上电和掉电阶段的动作。加上串连电阻后，外部电容的存在不会影响到内部产生的低电压环境，因此可以更清晰地观察复位信号。

1c. 复位脚的整体控制

图 4 中给出了掉电和上电期间，电源电压以及复位脚的典型动作。掉电时，当电源电压超出规定的范围，内部 LVI 会拉低复位脚并且一直保持低电平，直到电源重新建立。如果采用这项功能而且运行正常，那么对电源电压的下降时间就没有特别的要求。这样就允许采用电源衰减时间很长的设计，例如使用较大的存储电容，或者掉电时系统的电流消耗很低。

只要加在 LVI 上的电源电压还能让其电路正常运行，LVI 就可以在复位脚上保持低电平。如果深入观察复位脚上的信号就会发现，一旦电源降得非常低，复位脚会回漂到十分之几伏的电压。这是正常的。这种情况只有在电源电压已经降得非常低（不足 1 伏），不可能执行任何代码，也不会破坏闪存时，才会发生。

MC68HC(9)08 规范中有电源电压须从 100mV（有些器件是 200mV，详见数据手册）以下开始的要求，以保证内部上电复位（POR）。当电压接近为 0 时，由于半导体器件停止导通，使得有些应用的掉电衰减时间常数可能会太长。在系统设计过程中，应该考虑到这一点，以保证调电后剩余的 Vdd 降到起始电平所需要的时间不至于长得不合理。例如，可以采取在主电源电容上跨接一个附加阻性负载的形式。

电源的初始上升斜率还应该足够快，以保证上电复位。所需的最小斜率取决于特定 MCU 使用的 Vdd。举例来说，MC68HC908GP32 的 5 伏电压规范给出的数字是 35 伏/秒。大多数 MC68HC08 设备都是这个数字（3 伏为 20 伏/秒），但 908AZ60A 和 908AB32 规定 5 伏电压下为 20 伏/秒。

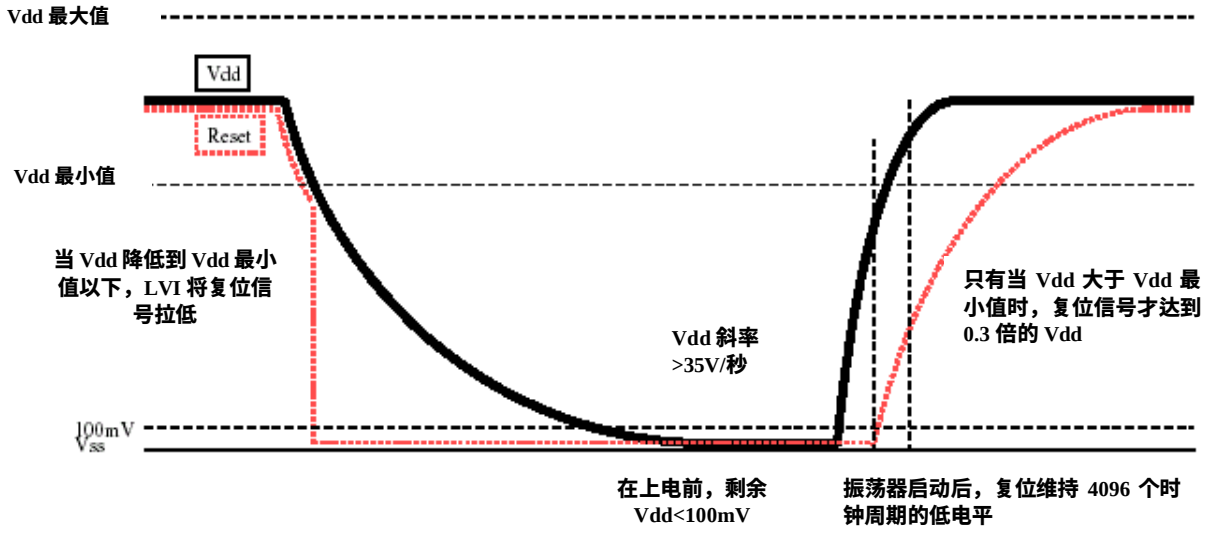

图 4. Vdd 和复位脚的典型波形

图 4 显示的是采用图 3 所示的外部 RC 电路，而且 POR 电路的运行正常时，复位脚的动作。POR 保持复位脚为低电平，直到振荡器启动并运行了 4096 个周期。这时引脚被释放，在外部 RC 电路的控制下变为高电平。设计者应该根据对电源的了解来选择 RC 时间常数，使复位脚只有在 Vdd 超过其规定最低值之后，才能升至规定的 Vil 值（0.3xVdd）。

2. 最大程度减小代码跑飞的影响的技术

即使采取了推荐的各种预防措施，在异常情况下仍然存在很小的代码跑飞的可能性。正因为如此，还应该采用下面的预防措施，防止在意外情况下对 MCU 或应用硬件造成损坏。

2a. 未使用的闪存或 ROM 地址

根据定义，代码跑飞就是对 MCU 程序计数器（PC）的破坏。此时 MCU 可能不受控制地从存储器程序代码区的未用地址中执行程序，为避免可能有害的影响，应该保持在其中填充已知的安全代码。无论代码保存在 ROM、闪存，还是在 EEPROM 中，这条原则都是适用的。在 ROM 应用中，唯一需要关心的是 I/O 动作，以及保存在 EEPROM 中的数据遭到破坏的可能性。在基于闪存的应用中，还存在代码破坏的可能性。

所有未使用的地址都应该包含一条合理而且适宜的指令。该指令可能是一个以 SWI 指令结尾的 NOP 序列，或者更简单，都是 SWI 指令。这么做以后，不受控地在这些地址中执行代码就会发生软件中断。然后程序员就能确定到底发生了什么意外。

SWI 中断矢量应该指向一个相应的故障处理程序。这可能是一个不给 COP 置数的无限循环（见后面）、执行 STOP 的指令，或是跳转到应用软件的起始点。

对于不同的应用，最适宜的策略不尽相同。顾名思义，STOP 指令的作用是：停止继续执行指令，I/O 配置停留在当前的状态。如果采用无限循环，COP 会强制复位，但产生复位所需的最长时间取决于 COP 更新周期，以及在一个周期的哪个确切位置失去控制。如果采用最后一种方法（跳转到应用的起始点），程序员应该记住还没有复位过，要对堆栈指针进行适当的处理。代码在打开中断前，应该进行必要初始化。如果没有复位，一次性写入寄存器还是锁住的。

另外一种可能的方法是用非法的操作码（如$32）填充未被采用的地址。如果它被执行到，就将强制发生非法操作码复位。但是，在剩余的空间上保留默认的$00或$FF 是不适宜的，因为它们是合法的指令。BRSET0 ($00)的动作取决于 portA 的数据寄存器，而索引指令 STX（$FF）会将索引寄存器保存到不想要的地址中。所以不推荐这两种方法。

所有未使用的中断矢量都应该填充错误处理程序的地址，以防止该中断意外发生。

我们还建议采用所提供的其它功能来监测不希望出现的现象。例如，当使用 PLL 时，就应该激活失锁中断功能，如果时钟速度可疑，就可用它来确定如何动作。

2b. 内部和外部 COP

如果未采用外部看门狗芯片，就应该启用片上 COP，并在正常的代码执行期间定时写入。这应该在主代码中完成，而不是在中断程序里做。之所以这样推荐是因为即使主后台任务出现故障，中断也可能继续正确进行。通过正确使用 COP，能够抓住大多数代码跑飞的情形，而通过其它方法是无法检测的。结合对未使用地址的适当填充,在程序计数器发生破坏的情况下,COP 能够提供充分的复位保护。

应该选择尽可能小的而且符合系统要求的看门狗超时周期。很显然，更短的超时周期可以提供更好的保护，但对闪存破坏的保护度还是有限的。现在的趋势是更快地写入大块的闪存，这意味着可以非常快地进行修改（以微秒计），因此 COP 只能提供有限的保护。所以 COP 应该与本文中介绍的其它保护方法结合起来使用。

2c. 908 MCU 中的闪存块保护

代码跑飞还有一个特定的风险：应用的非易失性存储器有被破坏的可能性，包括内部或外部的闪存或 EEPROM。在这类应用中采用上文中介绍的所有预防代码跑飞的措施是非常重要的。即使采取了所有这些预防措施，启用片上闪存保护功能仍不失为一种有益的做法。

在 MC68HC908 芯片中,采用闪存块保护寄存器(FLBPR)的形式,通过对 FLBPR 的配置，可以保护用户在闪存中选择的地址范围。如果与应用不冲突，建议对整个地址范围进行保护。执行代码占用的所有地址范围应该一直被保护。如表 2 所示，在所有 0.5µm MC68HC908 芯片中，FLBPR 默认的擦除值(\$FF)会取消所有保护。\$FF 以外的其它值可以保护 FLBPR 自身以及一个由其实际数值确定的地址范围。一旦得到保护，FLBPR 就只能通过将 V_{TST} 高电加到 IRQ 脚的方法进行修改。唯一的例外是 MC68HC908AZ60A，它不用在 IRQ 上加高压就能实现 FLBPR 的修改。

在 MC68HC908 系列的不同型号中，根据不同的内存地址分配，FLBPR 与保护范围起始地址的映射各不相同，如表 2 所示。LFBPR 对应于保护区起始地址的指定位。高端的固定位为 1（在 908AZ60A 的闪存块 2 中，第 15 位为零），低端位是 0。

该表只是块保护机制的简要说明，各个器件的数据手册才是最权威的信息来源。终止地址都是$FFFF（908AZ60A 的闪存 2 为$7FFF）。这种方法导致大多数 908 设备的块保护起始地址的解析度都是 64 字节或 128 字节。这与每种器件的块（或页）的大小相对应 [1]。对表中的器件来说，FLBPR 本身就是闪存寄存器，但是在有些 908 器件中（如 908JL/JK），它却作为读写寄存器来实施，因此在每次复位或上电后，必须由应用软件写入。

表 2. 闪存块保护寄存器(FLBPR)

器件	FLBPR 地址	起始地址				解析度	全部保护
		Bit 15	Bit 14	Bit 13	FLBPR 映射		
908AZ60A - 1 块 2	$FF80 $FF81	1 0	- -	- -	位 14-7 位 14-7	128 字节 128 字节	$00 $00
908AB32 908GP32 908MR16/32	$FF7E	1	-	-	位 14-7	128 字节	$00
908EY16 908GT16	$FF7E	1	1	-	位 13-6	64 字节	$00
908MR8 908GR8 908KX8	$FF7E	1	1	-	位 13-6	64 字节	$00

如果没有采取适当的预防措施，发生代码跑飞后，小 RAM 器件（如 MC68HC908KX8）闪存破坏的风险会更高，因为这些器件包含基于 ROM 的闪存擦除和写入程序（这些程序用于闪存和 EEPROM 的烧写/擦除循环及测试）。风险在于程序计数器的破坏可能导致非正常地执行这些 ROM 中的代码，进而破坏闪存。

在 MC68HC908GP32 或 MC68HC908AZ60A 等较大的器件中，ROM 内没有闪存子程序。如果符合系统的要求，用户应该避免将他们的闪存修改子程序包含在应用软件中。这并不妨碍代码现场升级的能力，因为应用程序可以包含较小的载入引导程序，用于必要时下载相应的闪存子程序，然后在 RAM 中执行。此外，08 系列的监控器模式还可与 P&E 的 PROG08SZ 软件结合使用。然而，即使是对于这些低风险器件，闪存保护功能也应该一直启用。

1. 在以前的 0.65µm 器件中（如非“A”的 908AZ60）上，闪存块保护寄存器的工作原理是不同的。

3. 总结

1. 启用 LVI。在 5 伏应用中，目前的 LVI 规范能防止 3.9 伏电压以下的代码跑飞，并且可以用于确保电源下降时的复位。在 908KX8 或 908GP32 等电压可以低至 2.7 伏的器件中，LVI 能在 5 伏应用提供更好的保护。
2. 用合理的指令（如 SWI）来填充应用代码的未使用字节（ROM 或闪存）。SWI 矢量（及所有其它未使用矢量）应该编程指向相应的错误处理子程序。
3. 启用 COP，并以合适的频率清空计数器，较短的超时周期可以提供更好的保护。
4. 在基于闪存的应用中，充分启用闪存块保护功能。

How To Reach Us:

USA/Europe/Locations Not Listed:
Freescale Literature Distribution
P.O.Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:
Freescale Semiconductor Japan Ltd.
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
825-26668334

Learn More: For more information about Freescale products, please visit **www.freescale.com**

freescale™
飞思卡尔半导体

Rev. 0
08/2002