

低功耗设计指南

作者: *Brant Ivey*
Microchip Technology Inc.

简介

在未来的嵌入式系统市场中，低功耗应用将占据很大的份额。每年都有越来越多的设计人员被要求提供便携、无线和节能的产品设计。本文全面阐述了低功耗嵌入式系统的设计基础，以期降低向低功耗应用过渡的难度。本文讨论的示例侧重于从单片机（MCU）的角度分析功耗。作为应用的“大脑”，MCU 通常功耗最大，对系统功耗的影响也最大。

与所有设计一样，低功耗嵌入式系统设计人员必须在功耗和其他因素（如成本、尺寸和复杂性）之间进行综合考虑。尽管有些低功耗技术不会增加系统成本，但有些却可能需要设计人员做出取舍。本指南将给出若干需要进行取舍的示例。但是，我们无法讨论所有需要取舍的情况，因此嵌入式设计人员始终应考虑节能技术对整个系统的影响。

本设计指南将提到 PIC[®] MCU 上可用的低功耗模式，但不会对这些特性进行详细说明。有关 PIC MCU 器件上可用低功耗模式的信息，请参见 AN1267《纳瓦技术和 nanoWatt XLP™ 技术：Microchip 低功耗器件简介》（DS01267A_CN）。

低功耗基础知识

低功耗的定义因应用的不同而存在很大差异。在一些系统中，工作能源已经足够，但低功耗设计人员想要努力降低运行开销或提高运行效率。在另一些应用中，只能提供有限的电源（如纽扣电池），这决定了系统的功耗要求。要降低这些系统的功耗，侧重点也各不相同，因此非常有必要弄清楚功耗的分布情况以及设计高效的低功耗系统应当从何处着手。

主要功耗源

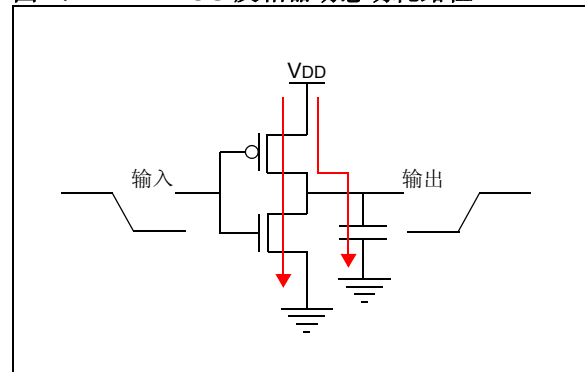
在 CMOS 器件（如单片机）中，总功耗可以分为两大类：动态功耗和静态功耗。动态功耗是单片机在运行和执行编程的任务时消耗的功率。静态功耗是只对器件供电，但不运行代码时消耗的功率。

动态功耗

动态功耗是 MCU 正常工作时消耗的电流。它包括开关 CMOS 电路时造成的功率损失以及器件的有源模拟电路（如 A/D 或振荡器）中偏置电流造成的功率损失。

要了解开关损耗的来源，请参见图 1 所示的 CMOS 反相器。

图 1: CMOS 反相器动态功耗路径



该反相器在输入为 VDD 或 VSS 时没有或只有很少的功耗。但是，如果信号从 VDD 切换为 VSS，则过渡期间 PMOS 和 NMOS 会在线性区域内产生偏置，从而产生 VDD 到地的电流。还需注意的是，在实际系统中，输出总线上会有一些量的负载电容。逻辑电平发生变化时，该总线电容的充电和放电会产生额外的电流消耗。

因单个门电路的动态开关损耗产生的平均功耗可通过如下公式计算：

公式 1： 动态功耗

$$P = V^2 \times f \times C$$

其中， V 代表系统电压， f 代表开关频率， C 代表负载电容。

注意，此公式仅适用于单个 CMOS 器件，不适用于整个 MCU。考虑整个 MCU 时，需要在此公式上乘一个比例因子 (α)，该因子根据器件中全部门电路的开关频率的不同而有所不同。

公式 1 揭示了控制动态功耗的几个重点。首先需要考虑电压，电压项在公式中是以平方形式出现，因此它是对动态功耗影响最大的因素。降低系统工作电压可以显著降低功耗。另一个需要考虑的重点是，可以对系统中哪个部分进行修改。每一个嵌入式系统的要求都不同，这也限制了设计人员调整电压、频率或负载电容的能力。

例如，嵌入式系统的设计人员无法自由控制内部负载电容 C 。该电容取决于 MCU 的内部走线与设计。MCU 制造商可决定是否利用适当的低功耗 IC 设计技术（如门控时钟信号）来限制负载电容的开关频率。系统设计人员则只能通过使能或禁止各 MCU 功能对内部负载电容加以控制。精明的低功耗设计人员应确保程序运行过程中，始终仅使能当前需要的 MCU 功能并关闭其他所有功能。

设计人员可以控制传送到 I/O 引脚的信号的外部负载电容。这些电容可能比器件的内部电容大得多，从而产生显著的损耗。因此，设计人员检查数字开关的杂散电容设计就非常重要。有关 I/O 低功耗设计技术的详细信息，请参见“**硬件设计**”部分。

工作电压主要取决于 MCU 的制造加工工艺。随着制造工艺的几何尺寸的缩小，工作电压以及器件的动态功耗都将继续降低。嵌入式系统设计人员可据此选择工作电压较低的 MCU。但是，如果最低系统电压由系统的其他部分（如传感器或通信接口）决定，则需在成本和功耗之间进行取舍。此时需要加配 MCU 电源稳压器，但这会提高系统成本。有趣的是，在某些情况下，若允许移除稳压器，以更高的电压运行会更节电。此时，稳压器中的功率损耗高于在较高电压下工作而增加的动态电流损耗。

频率通常是影响动态功耗的因素中变化最多的一个。因此，嵌入式设计人员通常通过调整频率来主动控制功耗。系统的最佳工作频率由以下几个因素共同决定：

- 通信或采样速度要求
- 处理性能
- 允许的最大峰值电流

如功耗公式所示，低频率会导致低动态电流。但需要注意的是，执行速度也是影响功耗的因素。在某些情况下，以较高的频率运行会更节电，因为这会加快相应操作的完成速度，从而使系统更快地进入休眠模式减少功耗。另外，在低频率下运行时，动态开关电流有可能不再是系统功耗的主要影响因素。MCU 上模拟电路中的偏置电流所产生的静态功耗将占据主导地位，降低频率因此不再是有效的节能技术。此时，设计人员应该转而专注于降低静态功耗的技术。

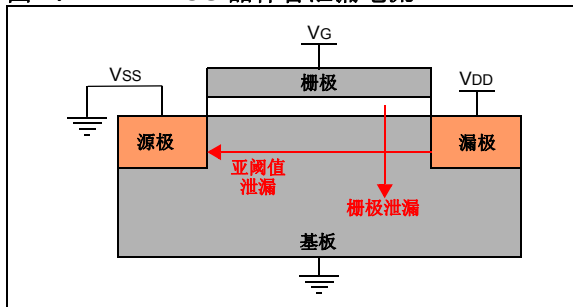
静态功耗

静态功耗包括代码不处于活动状态时为了维持正常的系统操作所需的所有功耗。这通常包括模拟电路的偏置电流、低功耗计时振荡器和泄漏电流。静态功耗是电池供电系统的主要考量因素，因为在应用使用期的大部分时间内，电池供电系统都处于休眠模式。

稳压器和欠压复位 (Brown-out Reset, BOR) 等模拟电路需要一定量的偏置电流，用以在温度和电压发生变化时保持可接受的精度。补偿这类模块的电流消耗时可采用的最佳技术通常有：利用 MCU 固有的灵活性根据需要使能和禁止模拟模块；或使用低功耗低精度模式。

计时功能，如看门狗计时器（Watchdog Timer, WDT）和实时时钟/日历（Real-Time Clock/Calendar, RTCC）也经常被视为系统静态功耗消耗源。虽然这些模块经常开关，可被认为产生动态功耗，但由于它们始终以固定频率运行，并且功耗非常小，因此更适合包含在静态功耗计算中。尽管如此，作为动态电路，它们仍然遵循“动态功耗”部分中提到的功耗优化规则，并且主要受电压、频率和电容的影响。这些振荡器本质上可以看作低频低功耗振荡器。例如，通常会将 32 kHz 晶振驱动器设计成允许晶振在尽可能小的峰-峰电压下工作，同时保持稳定的振荡。有关降低晶振功耗的详细信息，请参见“低功耗晶振设计”部分。

图 2: NMOS 晶体管泄漏电流



泄漏电流是由于 CMOS 器件中使用的 MOSFET 工作不理想所致。随着工艺线宽的减小和晶体管的小型化，FET 会在诸多方面不再与其在理想系统中的表现相同。即使栅极电压低于导通阈值，仍会产生从 FET 漏极到源极的泄漏电流。此电流称为亚阈值泄漏电流。在较狭窄的晶体管中，漏极和源极距离较近的情况下会产生亚阈值泄漏电流。晶体管越窄，泄漏电流越大。此外，泄漏还受温度和电压的影响。对于采用了小型化工艺的 MCU，在高温下以最大电压工作时会产生许多个 μA 的泄漏电流。

与动态功耗类似，嵌入式系统的设计人员也不能完全控制泄漏功耗。选择线宽较大的 MCU 会减少泄漏，但动态电流消耗会更高。温度也不受设计人员控制，而是取决于系统要求。一般来说，减少泄漏最好的办法是降低电压并关闭不需要的电路。一些 PIC[®] MCU 还具有深度休眠之类的特性，它可以关闭器件中的额外电路，使功耗低于一般的休眠状态。

加工工艺权衡因素

在尝试同时优化动态功耗和静态功耗时，将凸显一项非常重要的权衡因素。采用线宽较小的加工工艺可以显著降低动态功耗，但泄漏电流会更高。表 1 比较了常用的几种加工工艺及其相应的静态和动态功耗。在某些情况下，这会使设计人员很难决定应该侧重哪一方面来减少系统功耗。为了选择正确的 MCU 来最大程度降低特定系统的功耗，设计人员必须了解是动态功耗还是静态功耗对系统的影响更大。此过程称作“功耗预算”，我们将在“测量功耗”部分详细进行讨论。正是这种在动态功耗和静态功耗之间的取舍，使得 MCU 设计的优化过程显得非常重要。MCU 制造商在工艺线宽不断减小的同时也始终致力于维持较低的泄漏电流。

表 1: 加工工艺功耗比较

加工工艺	泄漏功耗 (归一化值)	动态功耗 (归一化值)	内核电压
0.35 μm	0.5	2.8	3.0V
0.25 μm	.75	2	2.5V
0.18 μm	1	1	1.8V
130 nm	1.5	.75	1.5V
90 nm	2	0.44	1.2V

AN1416

什么是低功耗？

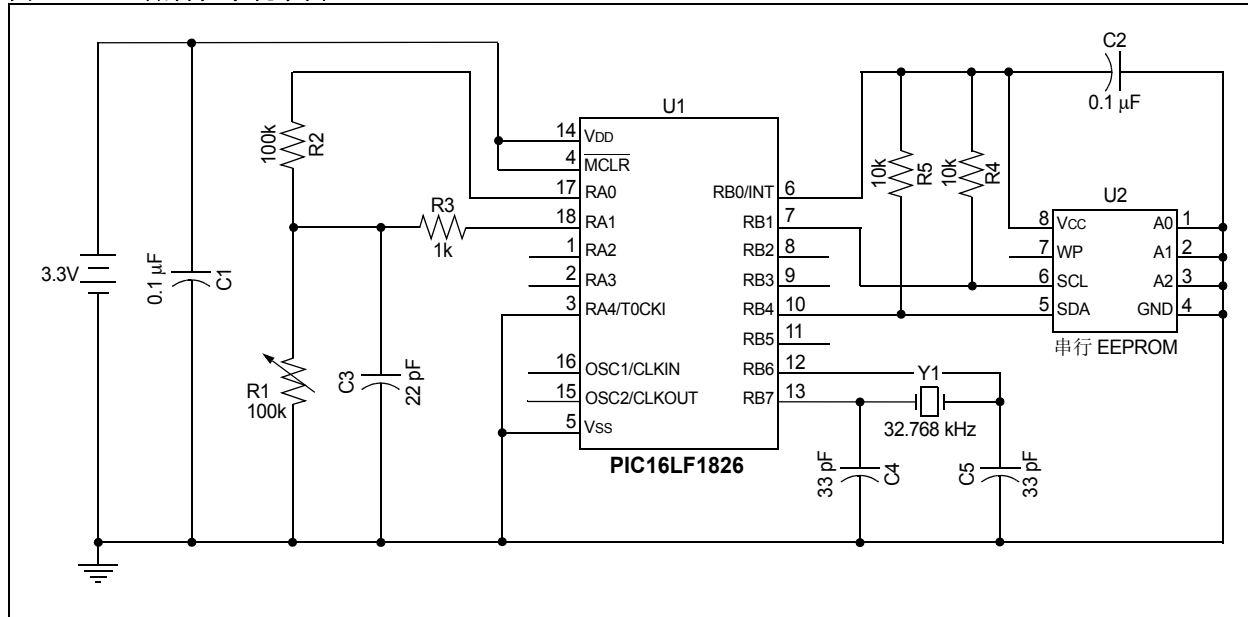
低功耗对不同的系统有不同的含义。一些应用（如电源）注重动态功耗，因为它们必须保持不间断运行。对于这些应用，只需考虑如何尽量降低动态功耗来提高效率。另一方面，电池供电的应用则通常比较关注休眠模式下的功耗，因为它们在大部分时间都处于休眠模式。

但是，运行和休眠模式下的功耗也不是系统功耗中仅有的两个重要方面。例如，有效占空比较低时，唤醒时间对系统有重大影响。假设系统仅唤醒 $10\ \mu\text{s}$ 来读取和存储输入，然后便返回休眠模式。在这种情况下，

将无法起振晶振，因为它需要若干毫秒才能趋于稳定。MCU 必须能够在几微秒内唤醒并工作，才能有效地管理应用。

为了解系统功耗各方面的重要性，我们将使用图 3 中的应用示例来说明各种低功耗设计技术及其权衡因素。数据记录传感器每 $100\ \text{ms}$ 进行一次采样，然后在记满一页（32 个样本）后将数据存储到 EEPROM。系统需要 $50\ \mu\text{s}$ 进行一次采样，将数据存储到 EEPROM 则需要 $5\ \text{ms}$ 。传感器使用 PIC16LF1826，即 3V XLP PIC 单片机。

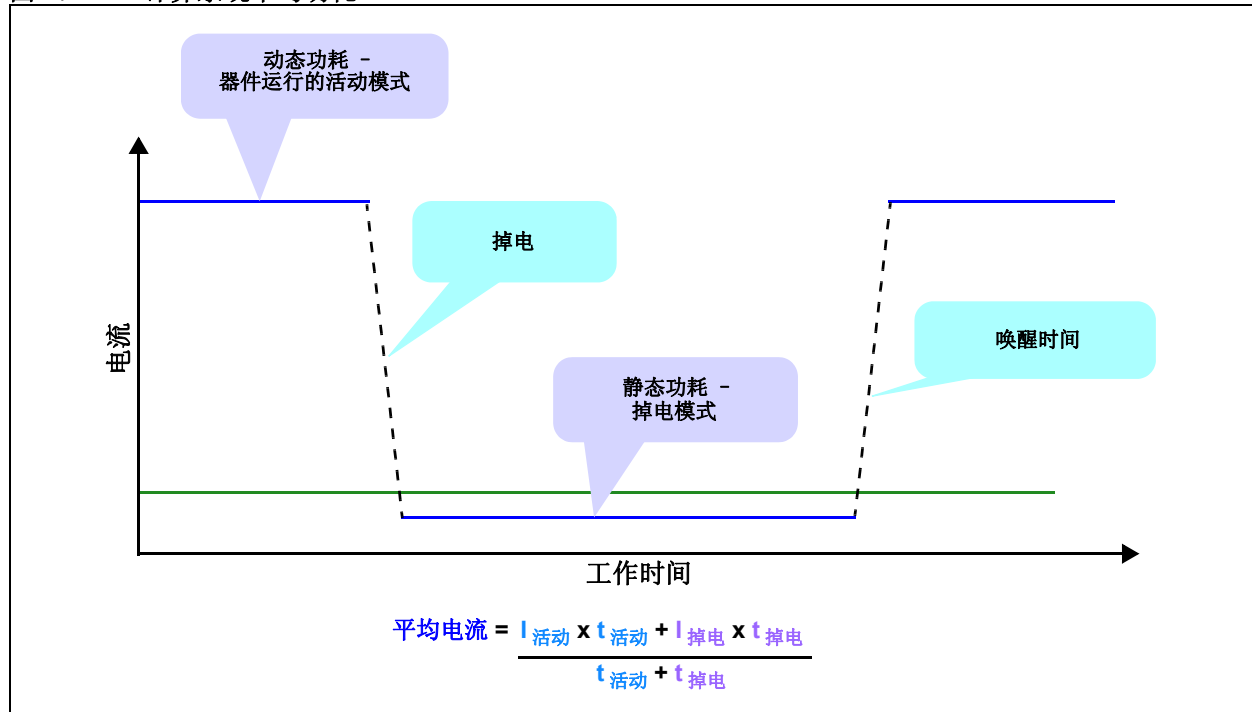
图 3： 低功耗系统示例



测量功耗

测量系统总功耗时，主要考虑平均功耗和最大功耗这两个值。平均功耗等于系统在动态和静态功耗模式下消耗的总能量除以平均系统循环时间，如图 4 所示。平均功耗提供了一个值，可用于精确确定电池的使用寿命或系统的总能耗，因此非常重要。

图 4： 计算系统平均功耗



最大功耗是系统在最坏的情况下需要消耗的功率。为设计适当的系统电源，非常有必要确定系统的最大功耗。例如，许多电池的性能都取决于电池的电流汲取速率，因此了解系统电池的电流输出能力是非常有必要的。

测量功耗看似比较简单，但要准确获取系统的平均功耗，对许多系统来说都很复杂。大多数嵌入式应用都不会提供足够的时间让电流表准确地测量并显示电流值，除非修改代码使其停留在特定状态。虽然可以使用示波器显示电流曲线，但可能无法准确获取低功耗模式下的功耗。因此，要对系统在所有模式下的功耗进行准确建模，通常需要组合使用多种测量方法。

测量静态功耗

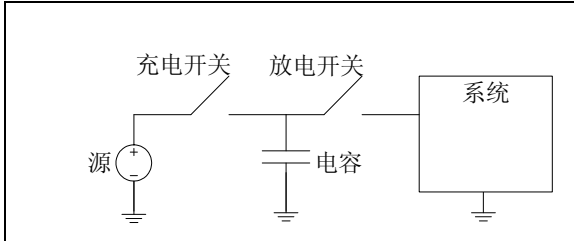
测量静态功耗时需要考虑以下几点：首先，用来测量的电流表需要有足够高的分辨率以准确测量 MCU 的功耗。对于许多 nanoWatt XLP MCU，静态功耗可能低于 100 nA，这个数值已经超出了大多数手持式万用表和一些台式 DMM 的最大分辨率。

另外要注意电流表的压降。将电流表设置到低电流范围来测量静态功耗时，通常在电流表两端会产生明显的压降。这会干扰压敏电路，并且如果出现短暂的高功率脉冲，MCU 将发生欠压复位。此压降值通常并未在产品文档中给出，必须使用额外的万用表或示波器进行测量。

一般来说，静态功耗无法通过示波器进行测量。对于低于 $10\ \mu\text{A}$ 的电流，必须使用至少 $5\ \text{k}\ \Omega$ 的电流检测电阻才能形成可以测量的压降。系统若不是处于低功耗状态，使用这么大的电阻可能会影响其正常工作。

如果没有可以准确测量超低功耗应用的电流表，则可使用电容作为替代方案。MCU 在低功耗模式下工作时，将作为灌电流恒流源。通过电容的恒流放电公式可以相当方便地测量低功耗系统的功耗。如图 5 所示，连接电容（如低泄漏型 $10\ \mu\text{F}$ 陶瓷电容）和应用，使用开关将电容与电源和应用断开。

图 5： 低功耗测量——电容法



要使用电容测量低功耗：

1. 同时导通两个开关，以运行系统，并由电源向电容充电，直到达到 V_{DD} 为止。
2. 断开电源（及任何电压表），使应用通过电容供电运行一段时间。此时间应足够长，使电容放电大约 $100\ \text{mV}$ 。确保在电压下降到超出应用的正常工作范围前断开电容的连接。
3. 从应用断开电容连接，然后连接电压表测量电容的剩余电压。
4. 使用公式 2，根据电压压降差值和放电时间计算电容恒流放电时的电流值。

注： 使用这种方法时，某些电容型号会产生较大的泄漏电流，使结果出现误差。要确定泄漏电流，可在电容不带负载的情况下重复试验等长时间，确定泄漏造成的电压变化量。

例如，如果系统通过 $10\ \mu\text{F}$ 电容供电运行了 10 秒，产生的压降为 $100\ \text{mV}$ ，则通过公式 2 可计算出系统 10 秒内的平均电流损耗为 $100\ \text{nA}$ 。

公式 2： 电容恒流放电

$$I = C \frac{\Delta V}{t}$$

测量动态功耗

使用电流表测量系统在正常工作条件下的动态功耗通常不会很有效。原因是大部分电流表的采样速度都不够快，以致无法在系统执行代码和改变状态时准确地测量其实时功耗。要使用电流表准确测量动态功耗，需要修改系统代码使其保持在特定状态下进行功耗测量。通过这种方法可以提供每种状态下电流消耗的准确数据，但不会提供计算系统平均功耗所需的执行时间信息。

测量动态功耗的有效方法是，使用示波器测量电流检测电阻两端的电压。设计人员可以使用示波器了解系统在工作期间跨越各种状态时的功耗变化，以及在每种状态中花费的时间。这有助于生成完整的应用功耗曲线。选择合适阻值的电流检测电阻很重要。所选的电流检测电阻既应足够大，以提供示波器上可测量的分辨率；但也要足够小，以免系统在高功耗状态下发生电压不足现象。一般来说， 10 到 $100\ \Omega$ 的电阻适合于动态功耗的测量。

创建功耗数据表

收集了静态和动态两种功耗数据后，即可使用这些数据创建系统的功耗数据表。功耗数据表使设计人员能够清楚知道系统的主要功耗来自哪里，以便进行进一步的优化。要创建功耗数据表：

1. 根据功耗的变化情况将应用细分为若干状态。
2. 测量每种状态的功耗和执行时间。
3. 通过功耗乘以时间计算每种状态下的总能耗。

表 2: 功耗数据表计算示例

模式	执行 32 次采样的 模式时间 (ms)	电流 (μA)		电荷 ($\mu\text{A} \times \text{ms}$)
		按器件	模式总计	
休眠 MCU 休眠 传感器关闭 EEPROM 关闭	3200	0.8 0 0	0.8	2560.0
初始化 MCU 唤醒 传感器关闭 EEPROM 关闭	0.32	0.8 0 0	0.8000	0.26
传感器采样 MCU 运行 传感器打开 EEPROM 关闭	1.28	150 16.5 0	166.5	213.12
换算 MCU 运行 传感器关闭 EEPROM 关闭	0.32	1300 0 0	1300	416.00
存储 MCU 运行 传感器关闭 EEPROM 打开	5	150 0 1000	1150	5750.00
总计	3206.92	—	—	8939.38

平均电流 (μA)，总电荷/ 总时间	2.788
峰值电流 (μA)	1300

数据表得出后，优化应用功耗将变得更加简单。功耗数据表清楚地显示系统中功耗最大的状态，设计人员因此可重点降低这些状态的功耗。功耗数据表可以简化系统平均功耗和最大功耗的计算。现在分析一下表 2 中的功耗分布数据，这些数据以图 3 中的应用示例为基础。表中计算了每一种单片机模式的执行时间，还按器件细分了电流消耗。请注意，在此数据表中，存储和休眠是功耗最大的两种模式。由于这两种模式对系统的影响最大，设计人员在使用该系统时应首先着重降低这两种模式下的功耗。

性能与功耗

嵌入式系统始终需要在性能和功耗之间寻找平衡。低功耗设计的关键是创建这样一个系统：能充分利用控制器的各种优点和特性，在功耗预算范围内发挥最佳性能。设计时需考虑以下几种重要的性能影响因素：

- 唤醒时间
- 振荡器速度
- 指令集架构 (Instruction Set Architecture, ISA)
- 外设特性
- 从闪存执行与从 RAM 执行

唤醒时间

唤醒时间对工作时间较短的系统非常重要，因为唤醒过程所消耗的电流与正常工作的消耗相当。

唤醒时间主要是振荡器起振时间（Oscillator Start-up Time, OST）。表 3 中显示了 PIC MCU 上各种常见时钟源的典型起振时间。请注意，晶体振荡器和锁相环（Phase-Locked Loop, PLL）振荡器等大多数高精度

高速时钟源的起振时间都较长。许多 PIC MCU 都具备“双速启动”功能，以此实现在系统上使用长起振时间的时钟源实现快速唤醒。“双速启动”功能将首先使用内部 RC 振荡器进行唤醒，这种振荡器的起振时间一般只有几 μs 。在主时钟源稳定之前，系统会一直使用该振荡器工作，通过运行非时序关键的代码来减少总操作时间，如图 6 所示。

图 6: 双速示波器

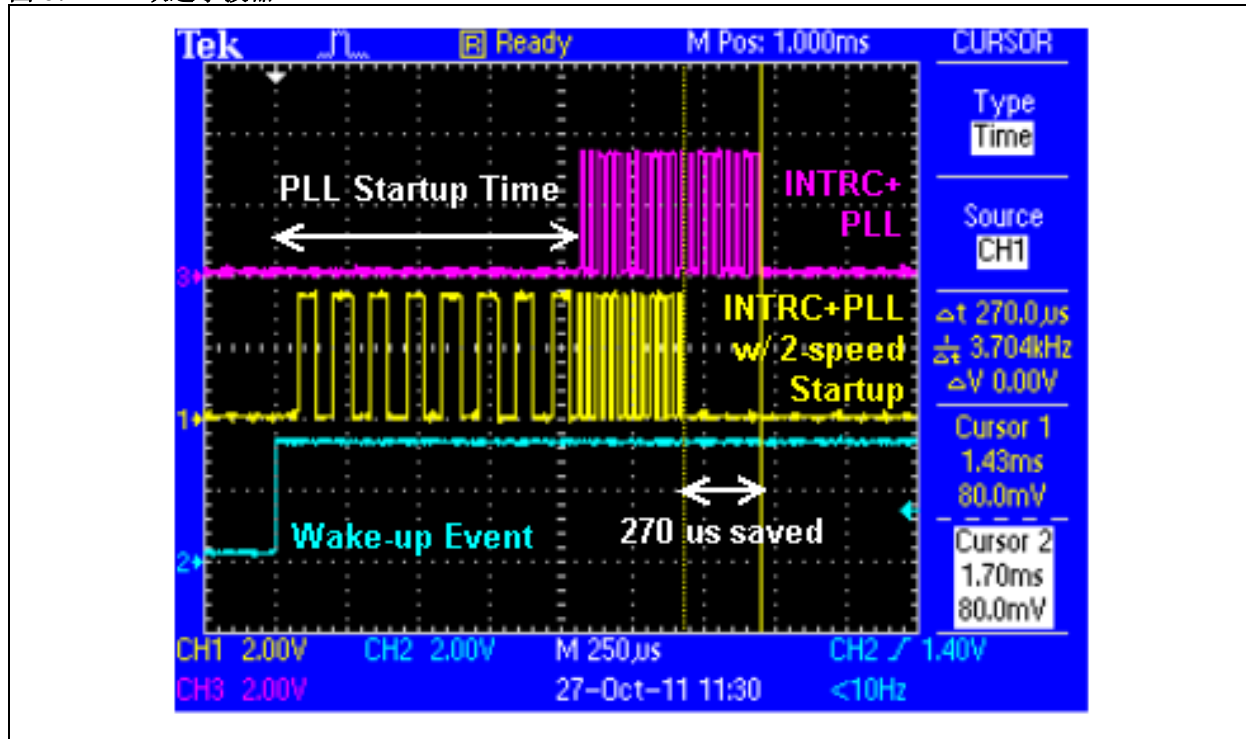


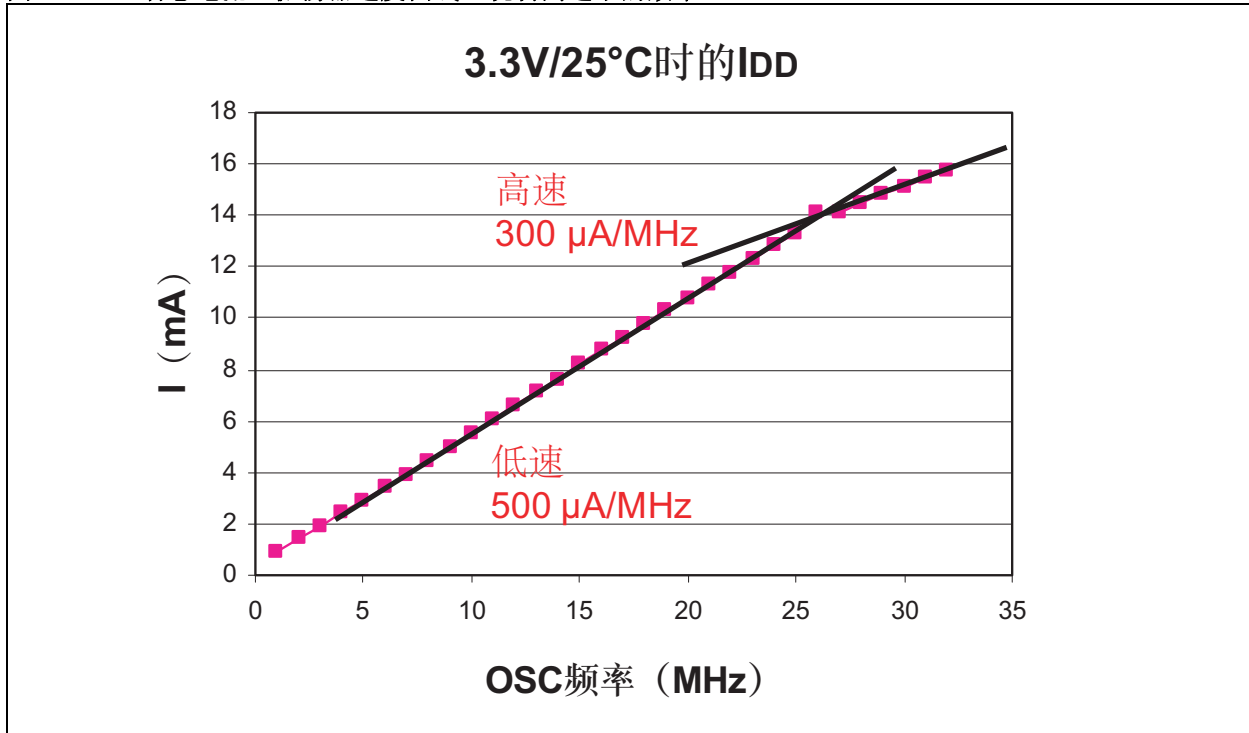
表 3: 各种振荡器的典型唤醒时间

振荡器	频率	起振时间 (ms)
内部快速 RC 振荡器	8 MHz	.001-.010
低功耗 RC 振荡器	31 kHz	0.3
主晶振	8 MHz	0.5-1.0
内部 RC + PLL	32 MHz (8 MHz x 4)	1.0
主晶振 + PLL	32 MHz (8 MHz x 4)	1.5-2.0
辅助晶振	32.768 kHz	100-1000

时钟频率与功率效率

低功耗设计最常见的一个问题是：为最大程度降低功耗，最佳运行频率是多少？以最低速运行更好还是先高速运行后转入休眠模式更好？虽然这取决于所用的 MCU，但通常高速比低速更有效率。这主要受最初给振荡器和 MCU 供电时所需固定电流的影响。频率低时，这种固定偏置电流消耗的功率会占据总功耗的很大一部分。而频率较高时，可以忽略此项。因此频率较高时通常会有较低的 $\mu\text{A}/\text{MHz}$ ，从而工作效率更高。请参见图 7，了解高速如何更高效。

图 7: 动态电流—振荡器速度曲线, 说明高速下的效率



关于这一规则有几点注意事项。如果电源的阻抗较低，可以承受较高的电流，则可适用这一规则。但如果电源阻抗较高（如纽扣电池），则该规则不再成立。如果使用电流较低的电源，则电源的内部阻抗会在电流消耗较高时减少对 MCU 的有效功率输出。这会缩短电池的使用寿命。还会由于电流负载高，导致电池产生输出压降而引发意外的欠压复位，影响系统的工作可靠性。

另外，高速运行时系统电压通常成为重要的因素。很多 MCU 都无法在器件的整个电压范围内全速运行。因此，电池供电的应用高速运行时，必须监视 VDD，以确定电池电压是否已降到全速运行所需的最小电压值。如果 VDD 低于此值，则会导致代码误执行。大多数 MCU 都提供低电压检测（Low-Voltage Detect, LVD）功能，在 VDD 降到接近该范围时中断器件，器件因此可减少系统在低电压下的工作频率，从而延长系统使用寿命。

指令集架构 (ISA)

MCU 制造商倡导的主要动态功耗技术参数是 $\mu\text{A}/\text{MHz}$ 或 $\mu\text{A}/\text{MIPS}$ 。尽管如此，很多设计人员都发现这些技术参数不是很准确，因为它们没有考虑 MCU 的 CPU 架构。必须了解每单位能耗实际执行的工作量，才能从真正意义上对功耗进行比较。不同的架构在单个时钟周期执行的工作量会有所不同。这使得 MCU 的指令集架构也成为一项重要的功耗考虑因素。时钟方案、每指令周期数和可用指令都会对器件性能产生很大的影响，并直接影响一个周期中能完成的工作量，从而影响功耗。

首先需要考虑的是 MCU 时钟比率和每指令周期数。时钟比率指输入系统时钟频率 (Fosc) 与内部指令时钟频率 (Fcy 或 SYSCLK) 之比。8 位 PIC MCU 具有 4 分频架构，而 16 位 PIC MCU 具有 2 分频架构。PIC32 器件具有 1 比 1 的时钟比率。查看器件的技术参数时需着重考虑时钟比率。有时器件功耗会根据 Fosc 来指定，有时则根据 Fcy 指定。指定 Fosc 的器件以 MHz（如 8 MHz）为速度单位，而指定 Fcy 的器件则使用 MIPS（如 8 MIPS）为速度单位。

但是，比较功耗时不能只考虑时钟比率。每指令周期数是另外一个需要注意的方面。所有 PIC MCU 都采用 RISC 架构，大部分指令的执行只需要一个周期。尽管如此，还有很多其他 MCU（甚至许多标榜为 RISC 的器件）需要多个周期来执行一条指令。由于存在这种差别，因此很难直接比较不同架构的功耗。在有些情况下，如果 MCU 上大部分的指令需要一个给定周期数来执行，则可计算出平均周期/指令值用于功耗比较；其他情况过于复杂而无法得到该值。如果确实无法得到该值，通常只能通过基准测试来比较功耗。

基准测试是考虑了“指令集架构”的唯一一种比较完善的器件比较方法。它也是耗时最多的比较方法。对于简单的基准测试，可以参考一些开源的基准测试标准来获得初步的比较结果。通过将基准测试完成时间与数据手册上的功耗技术参数结合，通常能够合理地比较器件功耗。尽管如此，这些基准测试通常也只是使用器件的 CPU，不会使用器件的外设或高级功能。要得到理想的比较结果，可以筛选出系统中运行时间较长和功耗较大的部分对应用进行简化，然后将简化的系统作为更准确的基准来执行功耗预算分析。

正确使用外设

单片机上的外设功能有助于大幅降低功耗。但是，在判断哪些外设功耗高哪些外设功耗低时经常会弄错。例如，比较 UART 和 SPI 时，许多设计人员会选择 UART 作为低功耗模块。因为它需要更少的 I/O 翻转，并且以低速运行，但这种低速实际上会使 UART 的功耗更大，因为 SPI 完成一项事务的速度通常比 UART 快得多。尽管 SPI 可能在执行事务期间功耗较大，但之后器件可进入休眠状态，从而消除最大的功耗源——CPU。表 4 列出了 MCU 的一些常用功能，并给出了这些功能的估算功耗的大致基准。当然，这些值会随着供应商、外设设置和应用的不同而发生变化，但应该能在有多种模块可选时提供决定依据。

以下几节将提供有关如何使用不同外设降低系统功耗的更多技巧。

表 4：常用外设的功耗

串行通信	电流 (μA)	发送 10 个字节所需时间 (ms)	总电荷 (μA × ms)
UART (57.6k)	200	1.74	347.22
I ² C™ (400 kHz)	1000	0.25	250.00
SPI (4 MHz)	700	0.02	14.00
模拟模块			
模拟模块	电流 (μA)	转换 10 个采样所需时间 (ms)	总电荷 (μA × ms)
低速 A/D (最大 100 ksps)	250	0.1	25
高速 A/D (最大 500 ksps)	1000	0.02	20
低速比较器	10		
高速比较器	100		
DAC/CVREF	5		
低功耗模块			
低功耗模块	电流 (μA)		
欠压复位 (BOR)	5		
看门狗定时器 (WDT)	0.5-1.0		
实时时钟和日历 (RTCC)	0.5-1.0		
定时器 (31 kHz)	0.5-1.0		

模数转换器 (A/D)

模数转换器 (A/D) 本质上是模拟模块，其中许多用于执行转换的电路（如参考电压电路、信号放大器、信号缓冲器等）不会随转换速度发生明显变化。由于这些模拟电路的功耗通常占了 A/D 功耗的大多数，因此以高于应用所需的速度运行 A/D 并在采样之间禁止 A/D 通常很有好处。A/D 应设置为使用尽可能快的转换时钟，并将采样时间缩短至保持测量精度所必需的最小值。

DMA 与 FIFO 缓冲区

直接存储器访问 (Direct Memory Access, DMA) 控制器是功能强大的可降低功耗的工具。DMA 通过分担 CPU 的数据传输任务来提高性能。任何可减少 CPU 运行时间的功能都有助于大幅降低功耗，因为为 CPU 提供时钟是 MCU 中最耗能的任务。

很多能够在休眠模式下工作的外设具有内置 FIFO 缓冲区，可通过延长休眠时间来降低功耗。FIFO 将存储已接收或已采样的数据，并在缓冲区充满后中断。这样器件仅需唤醒一次来处理数据。这要比 CPU 在完全唤醒状态下处理每个传输任务节能得多。

欠压复位 (BOR)

欠压复位 (BOR) 保护电路对于低功耗应用来说是一把双刃剑。一方面，该电路可以防止应用在电池电量耗尽或高瞬变电流导致电源电压下降时发生误执行。另一方面，即使器件处于低功耗模式下，该电路仍保持高功耗。通常，可保持精度的平均 BOR 具有大约 5 μA 的偏置电流。

为实现最佳功耗性能，利用 BOR 的灵活特性非常重要。以下是一些可用于降低功耗的 BOR 特性：

- **休眠模式下自动禁止 BOR。** 很多 PIC MCU 能够在器件进入低功耗模式时自动禁止 BOR。由于 BOR 的主要用途是防止器件在低电压时误执行代码，在 CPU 不运行时通常不再需要 BOR。因此，在休眠模式下自动禁止模块，在 CPU 唤醒时自动重新使能模块是很有价值的。
- **精度和功耗设置控制。** 一些 PIC MCU 的 BOR 可对电流消耗进行编程。这允许设计人员根据所需精度和电压来选择最适合系统的电流范围。
- **低功耗 BOR 或深度休眠 BOR 模式。** 有时，应用不需要持续工作的 BOR 来防止误执行。但是，完全禁止 BOR 电路可能会使 MCU 在电源电压降至接近于晶体管阈值电压（而没有一直降至地电压）时锁定。因此，许多新型 PIC MCU 采用低功耗 BOR (Low-Power BOR, LPBOR) 来提供下行保护，以确保 MCU 始终能上电复位 (Power-on Reset, POR)。这种模式的使能电流通常约 50 nA，明显小于完全运行的 BOR。

唤醒源

正确选择唤醒源是经常被忽略的 MCU 特性，该特性有助于降低功耗。现代 MCU 具有各种唤醒源，可通过器件上的几乎任何外设模块唤醒。尽管许多应用按照传统基于定时器唤醒来检查事件，但使能够专门捕捉特定事件的外设可降低功耗。例如，I²C™ 和 SPI 模块可以在休眠时接收数据，之后唤醒器件。这样就不必每隔几毫秒唤醒一次器件来检查传入传输，从而节省能耗。

另一个示例是使用实时时钟（Real-Time Clock, RTC）唤醒相对于使用定时器模块唤醒的优点。使用定时器模块计时通常需要每秒唤醒一次器件来更新时钟和检查报警条件。但是，实时时钟和日历（RTCC）允许器件在休眠状态下计时，直到发生报警，从而降低功耗。

从闪存执行与从 RAM 执行

为 MCU 的闪存供电和读取该闪存是功耗最大的操作之一。从 RAM 读取所消耗的电能比从闪存读取所消耗的电能少。为了利用此差异，许多高端 MCU（如 PIC32）提供了从 RAM 执行代码的选项，以帮助降低动态功耗。但是，仍需要考虑一些权衡因素和开销才能确定这种方式是否适合特定应用。

要从 RAM 运行，必须将代码从闪存复制到 RAM。此任务会消耗大量时间和电能。为了使该任务有执行价值，从 RAM 运行的函数通常应该有大小限制，或者应该是计算密集型函数。

对于某些器件架构来说，一个需要考虑的潜在权衡因素是，由于存在将单条存储器总线用于执行和操作的限制，从 RAM 运行可能会降低性能。例如，PIC32 器件中使用的哈佛架构通常使用一条存储器总线从闪存读取，用另一条总线从 RAM 读取。这样器件可以高速运行，而不会使 CPU 停顿。但是，从 RAM 操作时只使用一条存储器总线，这在执行某些写入 RAM 的指令时可能使处理器停顿。在估量从 RAM 执行的价值时，请注意这些器件上的潜在差异。

软件设计

软件设计对低功耗系统设计有重大影响。如果在编写软件时未使用合适的技术，则为了实现低功耗而进行的最佳低功耗器件选择和硬件优化操作很可能就白费了。以下几节提供了用软件降低功耗的一般技巧。

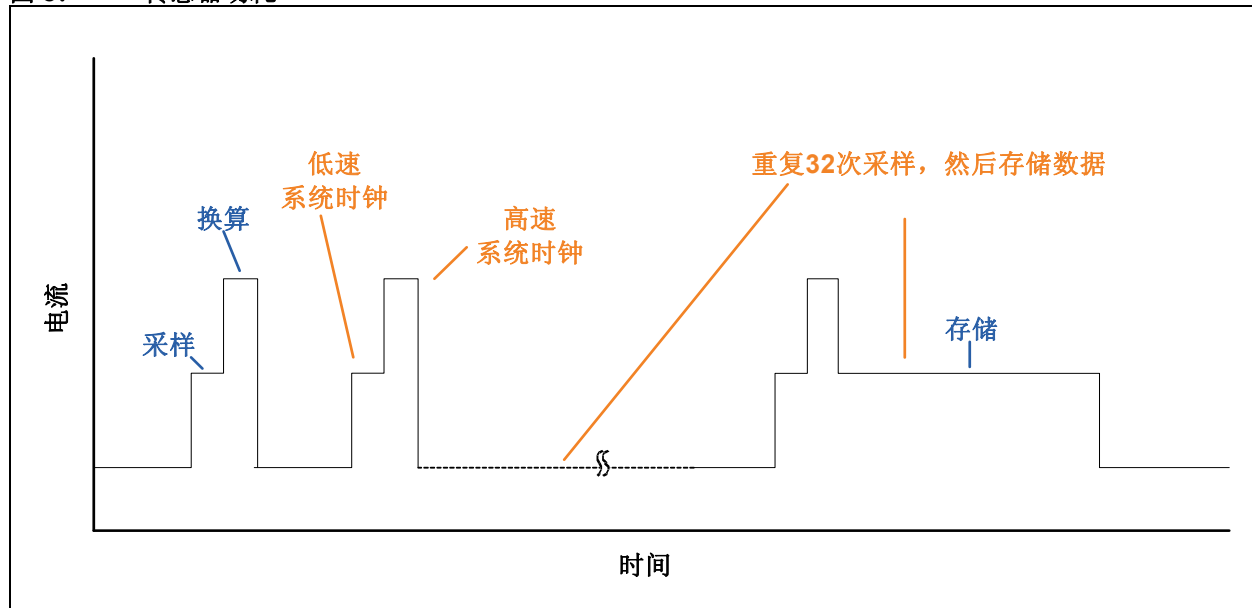
条件代码执行

一种可在许多应用中有效使用的软件技术是利用条件唤醒和代码执行结构。例如，图 8 给出了图 3 所示的传感器的功耗图。该传感器系统在写入 EEPROM 之前执行 32 次采样。这样系统在大多数唤醒事件中运行非常短

的时间，并确保尽可能少地执行高电流的写闪存操作，从而将功耗降至最低。如果每个采样都单独写入，系统将消耗大量的电流。

此外，该系统还使用变化的系统时钟来进一步提高能效。该系统在采样和存储过程期间使用低频时钟源，只有在换算过程中才会使用高频时钟。这是因为系统中使用的 A/D 和 EEPROM 的采样和存储时间是固定的。在此期间降低频率可减少电流消耗，而不会增加运行时间。换算时间仅取决于 MCU 的性能，并随着工作频率的变化而变化。因此，在此期间最好快速运行，随后让系统恢复到休眠模式。

图 8: 传感器功耗



空闲和打盹

通常，可以很容易判断何时在应用中使用低功耗模式（如休眠），因为在系统可进入休眠模式时有明显的停机时间。动态节能模式（空闲和打盹）则没有那么明显。例如，系统应使用空闲或打盹模式的情况有：

- **替换等待循环。**低功耗应用应避免出现让 CPU 执行费时操作的循环，如轮询某个位。系统设计人员应将这些循环替换为对空闲模式的调用，如下所示：

将：

```
while(!ADCInterruptFlag);
```

替换为：

```
while(!ADCInterruptFlag)
{
    Idle(); // 在 ADC 中断时唤醒
}
```

这在代码中具有相同的效果，但功耗要低得多，因为直到 A/D 中断唤醒器件，才会为 CPU 提供时钟。请注意，这不会影响器件的其他功能，因为其他中断可以唤醒器件并被正确处理，随后器件会恢复为空闲模式。如果 CPU 在等待标志时需要执行一些处理，则应考虑打盹模式而不是空闲模式，让 CPU 在等待的同时以低速运行。

- **运行高速外设。**假设系统的主要工作是保持高分辨率 PWM 输出。高分辨率要求使系统保持高频时钟。在这种情况下，由于需要保持 PWM，不能使用休眠模式来降低功耗。同样，空闲模式可能也不可行，因为 CPU 需要在必要时监视系统状态以更新 PWM。即使这样，CPU 也不需要采用与 PWM 相同的速度运行才能执行此作业。在这种情况下，打盹模式是理想选项，允许 CPU 低速执行必要任务。

中断与轮询

在从系统中清除耗能的等待循环时，低功耗系统设计人员应考虑从基于轮询的系统切换到基于中断的系统。利用中断可以使系统更频繁地使用节能模式（如休眠和空闲），而不是等待轮询事件。考虑图 3 所示并在表 2 中详细说明了的传感器。如果此应用每次采样都需要大量处理时间，则在空闲模式下等待、在获取采样时轮询 A/D

标志是没有意义的。相反，可采用基于中断的方法，这样系统可在获取当前采样的同时处理前一个采样。如果系统很简单，不需要很多处理，或者有需要频繁运行的后台任务，轮询可能是更好的选项，因为它允许更简单紧凑的代码而无需额外电流消耗。

功耗优化和 C 语言

C 语言是编写嵌入式程序的强大工具，但与汇编语言相比，C 语言剥夺了设计人员对代码的许多控制权。因此，一些工程师对在需要极低功耗的应用中使用 C 语言犹豫不决，而更加喜欢使用汇编语言来手动优化代码。

虽然经验丰富的工程师使用汇编语言编写的代码确实几乎总是比 C 语言代码更有效率，但目前的编译器已经过优化来尽量缩减这种差距。因此，使用 C 语言来获得汇编语言所不具备的代码可移植性和维护改进通常是值得的。实际上，随着项目复杂度的提高，C 语言编译器比手写汇编语言更高效的可能性也在增加。汇编语言在用于非常小的程序或大型程序中的小程序段时最有价值，因为这些小程序需要密集处理，可以很容易进行手动优化。

在 C 语言或汇编语言中存在可由编程器完成而无法由编译器优化器执行的优化。一个可大幅改进代码的优化示例是针对通用情况简化程序流程。考虑这样一种情况，A/D 采样已完成，并且与前一个采样相同。优化过的代码在执行耗时的处理步骤之前，会将当前采样与前一个采样以及其他常用值进行比较。如果新采样与这些值中的一个相匹配，则代码可使用预先计算的结果并避免大量处理时间。在某些情况下，这种对程序架构的修改会显著减少总执行时间。对于大多数应用来说，许多应用功能中均有一些情况比其他情况出现得更频繁，在这些情况下可利用上述方法。

硬件设计

电路板级注意事项

设计低功耗硬件时需要遵循几条明确的要求：

- 最大化电流路径中的阻抗
- 最小化高速开关路径中的阻抗
- 最小化泄漏电流
- 最小化工作占空比

以下建议在系统的各种电路中使用上述规则，以将这些元件的功耗影响降至最低。

按钮

按钮功耗的主要问题是，按钮通常以非常慢的人类速度时域进行操作，而不使用高速嵌入式系统时域。即使非常快的按钮操作也需要数百毫秒。如果按钮上有 $10\text{ k}\Omega$ 上拉电阻，则意味着几百毫秒的 $300\text{ }\mu\text{A}$ 电流消耗，这在低功耗系统中是相当可观的。增加上拉电阻值可有助于降低功耗，但高阻值可能会减小抗噪能力。此外，增加上拉电阻还会增加信号的时间常量，这会延长检测到高电压前的上升时间。

更好的方法是使用 MCU 的内部上拉电阻作为按钮的电压源。由于可在代码中动态使能和禁止内部上拉，只要检测到按钮按下，MCU 即可禁止上拉，去除接地电流路径。这也在其他方面改进了按钮电路，因为提供了简单的去抖动功能并减少了外部元件数。对于需要能够检测按住状态的按钮，可以定期重新使能上拉来确定按钮是否处于按住状态。

LED

LED 也存在功耗问题，原因与按钮类似。LED 在运行时的电流消耗很大（ 2 mA 至 50 mA ），而且必须运行很长时间才对用户有意义。以下是用于降低 LED 功耗的两种主要技术：

- 使用 PWM 以较低占空比驱动 LED。这种方法能够在必要时动态降低亮度，从而使代码可在一定程度上控制 LED 功耗。
- 通过增加限流电阻的大小来以较低的电流驱动 LED。该技术对于新型高亮度和高效率 LED 尤其有用。这些 LED 在额定电流下可发出眩目的灯光。但是，对于不需要高亮 LED 的系统，可以用低得多的电流驱动 LED 并使其仍保持高度可见性。尽管额定电流为 2 mA 的 LED 在低于额定值工作时可能不是很亮，但额定电流为 20 mA 的高亮度 LED 在低于 1 mA 的电流下驱动时通常仍可保持高可见性。

连接和控制 I/O 引脚

在许多应用中，I/O 引脚的不正确使用是意外高功耗的常见原因。在 PIC 单片机上使用双向 I/O 时需要遵守几条规则，以将功耗降至最低。

未使用的端口引脚

默认情况下，PIC 单片机的 I/O 引脚在上电时用作输入。数字输入引脚在输入电压接近 V_{DD} 或 V_{SS} 时功耗最低。如果输入电压在 V_{DD} 和 V_{SS} 之间，则数字输入缓冲器内的晶体管会在线性区域内偏置，并消耗大量电流。在悬空的未使用 I/O 引脚上，引脚电压可能漂移到 $V_{DD}/2$ 或处于振荡状态，导致未使用的 I/O 引脚消耗大量电能。如果将开关信号耦合到 I/O 引脚上，则引脚可消耗多达 $100\text{ }\mu\text{A}$ 的电流。

未使用的 I/O 引脚应保持未连接状态，但应配置为输出引脚并驱动到任一状态（高电平或低电平），或配置为输入引脚并使用外部电阻（约 $10\text{ k}\Omega$ ）拉至 V_{DD} 或 V_{SS} 。如果此类引脚可以配置为模拟输入，则关闭数字输入缓冲器，以防止悬空信号导致过度电流消耗。这些方法中的任意一种都可防止悬空节点情况，从而将功耗降至最低。

模拟输入

模拟输入的阻抗非常高，因此消耗的电流非常小。如果施加的电压通常在 VDD 和 VSS 中间，则模拟输入的电流消耗将低于数字输入。有时，当数字输入必须切换到低功耗状态时，可以并适合将数字输入配置为模拟输入。

数字输入和输出

只要数字输入被拉至 VDD 或 VSS，就仅消耗引脚输入泄漏电流。

除了流经引脚为外部电路供电的电流，数字输出引脚不消耗任何其他电流。应密切注意外部电路，以确保输出引脚驱动的状态功耗最低。如果外部电路掉电，需确保与其连接的任何 I/O 引脚均驱动到低电平，以防止灌电流通过已禁止的电路。

对于具有高开关频率的数字输入和输出，应尽量减少走线长度并去除不必要的元件来确保总线上没有杂散电容。

调整上拉电阻大小

具有上拉或下拉电阻的 I/O 线路可能是应用中的高能耗源，经常是功耗优化的目标。如前文所述，按钮和 LED 的功耗取决于所用电阻的大小。对于低功耗应用，一般规则是使用尽可能大的上拉电阻。但是，设计人员应注意这样做对电路的影响。例如，在 I²C 模块上，上拉电阻的大小决定了总线的最大速度。使用过大的上拉电阻将增加总线的时间常量并使通信速度下降。这会导致电流消耗的净增长。

电容泄漏电流

所有电容在电流通过电介质时都会产生少量电荷损失，即使完全充电后也是如此。这种损失通常称为“电容泄漏电流”。泄漏电流量取决于电容的大小和类型。通常，钽电容和电解电容的泄漏电流较高，而陶瓷电容和薄膜电容的泄漏电流较低。

表 5: 常见电容的泄漏电流

电容泄漏电流 (10 μF)	
电解电容	5 μA
钽电容	1 μA
陶瓷电容	20 nA
薄膜电容	5 nA

针对不同的电容，泄漏参数有所不同。在一些情况下，泄漏电流是直接指定的。在其他情况下，泄漏电流由电容的绝缘电阻 (Insulation Resistance, IR) 决定。绝缘电阻通常以兆欧或兆欧 × 兆法拉为单位指定。要通过后一种方式确定泄漏电流，请使用以下公式：

公式 3: 计算电容泄漏电流

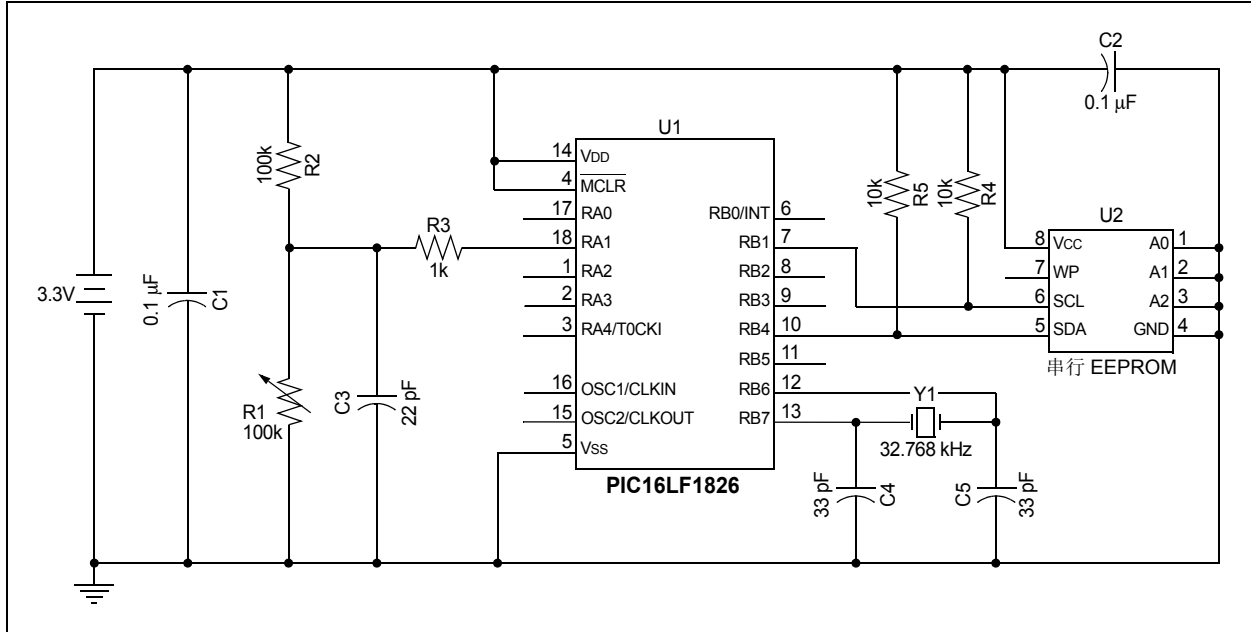
$$I (nA) = \frac{V \times C}{IR (M\Omega \times MF)}$$

请注意，某些电容的初始泄漏电流可能较高，这是因为电介质中的化学物质带有极性。电流开始时较高，当电容处于完全充电状态后将降至额定泄漏电流值。为低功耗系统选择电容时，如果可能，最好选择泄漏电流低的电容。

外部电路供电

MCU 只是整个系统中的一个组件，系统中所有器件的功耗构成了系统的总功耗。尽管许多现代 IC 设计有低功耗模式，但其他组件对低功耗的含义有不同的标准。虽然 MCU 可能具有功耗接近 100 nA 的休眠模式，但外部收发器等器件通常在待机模式下消耗几微安的电流。因此，经常需要让 MCU 控制对系统中外部电路的供电，以尽量降低总功耗。一些可由 MCU 轻易控制且功耗较高的外部电路包括模拟分压器、LCD 背光、传感器和收发器（如 RF 和 RS-232）。考虑图 9 所示的系统。通过修改系统使分压器 (R1) 和串行 EEPROM (U2) 由 MCU 的 I/O（分别为 RA0 和 RB0）控制，从而大幅减少系统泄漏电流。对于不能由 MCU 的 I/O 直接控制的高功耗器件，使用 PIC 器件驱动 MOSFET 并为电路供电。进行这些修改会使设计类似于图 3 中经过优化的低功耗系统。

图 9: 未优化的低功耗传感器系统



使用这种技术时需要注意几个问题。首先，在元件上电或掉电前，需确保连接到已禁止的外部电路的所有 I/O 引脚都处于正常状态。一般来说，操作 I/O 的最佳步骤如下：

- 给器件上电：
 1. 使所有已连接的 I/O 处于三态。
 2. 给外部器件上电，等待器件起振。
 3. 将 I/O 配置为工作状态。
- 使器件掉电：
 1. 使所有已连接的 I/O 处于三态。
 2. 使外部器件掉电。
 3. 将 I/O 作为输出驱动为低电平。

在更改器件状态前，I/O 引脚应处于三态，以确保在系统上电或掉电时没有总线冲突。禁止外部电路时，最好将已连接的 I/O 引脚驱动为低电平并配置为输出，以确保禁止这些线路浮动。

使用此技术时，请考虑器件重新上电与静态功耗相比较的成本。对于浪涌电流高、有大电容需要充电或上电时间长的外部器件，存在与上电相关的高电流消耗。对这些元件重新上电所产生的电流消耗造成的电荷损失可能多于通过断电节省的电量。确保对整个系统执行功耗预算，以确定哪种状态功耗更低。

电源设计

低功耗系统电源的选型和设计可能对系统的功耗有很大影响。线性稳压器、开关稳压器和其他 PMIC 都需要一定量的电能才能工作，许多器件专为具有高功耗要求的系统进行了调整。对于电池供电的系统，电源中可能无法或完全不需要使用稳压器。

开关稳压器通常为应用提供较高效率，以较高功率工作。这使得开关稳压器适用于线路供电的低功耗系统，专注于提高效率和降低动态功耗。尽管如此，在低电源电流时，其效率也非常低。升压稳压器在 50 mA 时的效率可达 95%，而在负载电流为 10 μA 时可能只有大约 20% 或更低的效率。在设计低功耗系统时应记住，系统大部分时间处于休眠状态，因此从总体上看线性稳压器是更有效的选择。

在某些情况下，运行不带稳压器的电池供电系统可能很理想。特别是：LiFeS₂（锂AA电池）、LiMnO₂（纽扣电池）和碱性电池都在大多数低功耗单片机的 1.8-3.6V 工作电压范围内工作。有关电池化学性能的详细信息，请参见本文档的电池部分。通过设计能够使用电池直接供电的系统，可以显著延长系统使用寿命。

考虑设计为由两节碱性AA电池供电的低功耗系统。这些电池具有线性放电曲线，在电池寿命的前 50% 期间从 3.6V 放电至 2.7V。如果使用低压降 2.5V 稳压器，在稳压器低于最小工作电压前将仅使用大约一半的电池电量。电池中剩余的电量（从 2.7V 降至 1.8V 时）无法得到使用，因而会浪费一半电池电量。此外，稳压器在工作期间将有一定量的静态电流消耗，进一步减少电池的有效使用寿命。不使用稳压器工作时有两个主要的权衡因素。一个权衡因素是应用损失了一个参考电压。但是，大多数现代MCU都有一个内部参考电压，可用于替代稳压器来确定当前 VDD电压并执行模拟转换。另一个权衡因素是，应用的所有元件都需要支持较宽的电压范围，这会提高成本。

低功耗晶振设计

尝试降低 MCU 功耗的主要影响之一已体现在低功耗 32.768 kHz晶振的驱动器中。为了最大程度地降低电流消耗，典型晶振驱动器的驱动强度已变得非常低，以致于晶振现在可以在低于 1 μ A 的电流下有效运行。结果是设计晶振电路时必须非常小心，并且要进行良好的测试，以确保晶振正确起振并精确运行。

低功耗晶振设计的基本要点是将晶振尽可能接近 MCU 放置，并且在可能的情况下，使接地环包围晶振驱动器电路，以防止耦合噪声。

此外，应确保为晶振和驱动器选择了大小适当的大容量电容，这一点也非常重要。请注意，这与许多早期晶振设计有很大不同。32 kHz 晶振的平均负载电容通常为 12.5 pF，这使得大容量电容约为 22 pF。但是，由于晶振驱动器电流已下降，可能无法再有效起振具有如此高负载电容的晶振。在这些系统中应使用具有理想负载电容（3.7 pF 至 6 pF）的低功耗晶振，以获得更可靠的低功耗运行。

有关如何设计低功耗晶振驱动器电路的更完整的分析，请参见 AN1288, “*Design Practices for Low-Power External Oscillators*”; AN849, “*Basic PICmicro[®] Oscillator Design*” 和 AN949, “*Making Your Oscillator Work*”。

电池注意事项

许多低功耗系统是电池供电的移动或远程设备。以下几节将介绍如何设计基于电池的应用，重点在于使用电池时的一些关键设计决策。

电池化学成分

电池的化学成分对电池的参数和性能有很大影响。每种化学成分都有特定的电压放电曲线、内部阻抗和自放电电流。一些最常见的一次性电池（不可充电）的化学成分有碱性（标准 AA/AAA）、二氧化锰（纽扣电池）和锂离子二硫化物（锂 AA/AAA）。最常见的可充电电池的化学成分为氢化镍（可充电 AA/AAA）和锂离子。

表 6 汇总了各种电池化学成分的特性。

电压曲线描述了电池的输出电压在电池放电时的变化。大多数电池会在数据手册中提供图表，以显示在特定放电电流下的电压曲线。电压曲线会影响系统的稳压和工作电压要求。例如，碱性电池具有倾斜的放电曲线，它随着电池的使用线性下降。要求电压稳定的应用需要使用稳压器来确保一致的工作电压。相反，大多数锂化学物质具有非常平坦的放电曲线。它们会输出恒定电压，直到电池使用寿命结束时，电压输出才会迅速下降。具有平坦放电曲线的锂化学物质较难通过电量计测量，因为无法通过测量电池电压的方式确定剩余电量。但是，相对于碱性电池，系统能够在电池电压下降过低之前使用更高百分比的电池电量。

电池的内部阻抗是电池化学性质的结果。由于电池需要通过化学反应来释放能量，电池无法对系统的电力需求做出即时反应。因此，电池上的高电流短脉冲经常导致输出电压骤降，设计人员在创建电池供电的系统时必须考虑到这一点。纽扣锂电池尤其如此，由于其有很高的内部电阻，通常有 10 mA 左右的电流限制。工作电流高于此限制时，将由于内部电阻上的压降而浪费大量电池电量。

自放电电流由电池内持续的化学反应产生。即使不从电池中汲取电流时，内部化学反应也在进行。这意味着一些电池电量会随时间丢失。碱性电池和大多数蓄电池的自放电电流很高，因此不适合长期应用。一次性锂电池的自放电电流非常低，每年的损失通常不到电池电量的 1%。电池制造商通常将自放电电流指定为保存期的一个组成部分，表示电池在不使用状态下保留其大部分电量（通常为 80%）的年数。

表 6: 常见电池化学成分的电气特性

化学成分	类型	典型外形	标称电压	电压曲线	自放电 (%/月)	标称内部电阻
碱性	一次性	AA/AAA	1.5V	倾斜	0.08%	150-300 mΩ
Li/MnO ₂	一次性	纽扣	3.0V	平坦	0.05%	10k-40k mΩ
Li/FeS ₂	一次性	AA/AAA	1.5V	平坦	0.30%	90-150 mΩ
锂离子	蓄电池	多种	3.6V	平坦	20%	30-40 mΩ
Ni/MH	蓄电池	AA/AAA	1.2V	倾斜	30%	30-40 mΩ

估算电池寿命

使用计算出的功耗预算中的平均电流可以确定电池能够为应用供电的时长。表 7 给出了在使用表 4 中的功耗预算示例所描述的系统平均功耗时，典型电池类型的使用寿命。计算电池寿命的公式如公式 4 所示：

公式 4: 计算电池工作寿命

$$\text{寿命 (小时)} = \frac{\text{电量 (mAh)}}{\text{系统电流} + \text{电池自放电电流 (mA)}}$$

表 7: 低功耗传感器的电池寿命估算

电池	电量 (mAh)	自放电 (%/年)	电池寿命 (年)
CR2032	220	1%	8.3
锂 AAA	1150	1%	36.7 ⁽¹⁾
碱性 AAA	1150	4%	17.5 ⁽¹⁾
锂离子 ⁽²⁾	850	20%	4.4

注 1: 该时间超过电池的典型保存期，表示电池可能在完全放电前已物理失效。

2: 电量随电池规格的不同而不同；这里采用的是典型值。

请注意，公式 4 是用于生成初始估算值的简单工具。针对 AAA 型电池查看表 7 中的公式结果可以看出，它预测的寿命超出了这些电池类型的典型保存期。电池可能在完全放电前由于其他原因（如包装泄漏）而物理失效。设计人员在计算电池的设计寿命时应该了解电池的所有限制条件。

为了简化功耗预算和电池寿命估算的过程，Microchip 提供了一个名为 XLP Battery Life Estimator 的工具，使用此工具可以根据使用的电池和系统功耗曲线对电池寿命进行基本计算。此工具支持大多数 nanoWatt XLP PIC MCU 器件，并可配置为适合各种电池类型。可以从 www.microchip.com/ble 下载此工具。

结论

低功耗设计是一个复杂的主题，其内容远远超过单个文档的覆盖面。本应用笔记介绍了低功耗系统设计人员在设计过程中会涉及到的多个主题。设计人员必须了解系统中的功耗源、元件的关键功耗相关参数、硬件和软件功耗优化技术以及功耗和性能之间的关系。在了解对于应用至关重要的低功耗设计的各个方面后，设计人员可专注于深入这些主题，以设计出真正优化功耗的系统。

参考资料

Microchip 低功耗设计中心，
www.microchip.com/xlp

Energizer 产品数据手册，Energizer Inc.，
<http://data.energizer.com/>

B. Ivey, AN1267, 《纳瓦技术和 nanoWatt XLP 技术：
Microchip 低功耗器件简介》(DS01267A_CN)，
Microchip Technology Inc.，2009。

Microchip 应用部门，《PIC[®] 单片机低功耗技巧与诀窍》
(DS01146B_CN)，Microchip Technology Inc.，2009。

请注意以下有关 **Microchip** 器件代码保护功能的要点：

- **Microchip** 的产品均达到 **Microchip** 数据手册中所述的技术指标。
- **Microchip** 确信：在正常使用的情况下，**Microchip** 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 **Microchip** 数据手册中规定的操作规范来使用 **Microchip** 产品的。这样做的人极可能侵犯了知识产权。
- **Microchip** 愿与那些注重代码完整性的客户合作。
- **Microchip** 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。**Microchip** 承诺将不断改进产品的代码保护功能。任何试图破坏 **Microchip** 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 **Microchip** 产品性能和使用情况的有用信息。**Microchip Technology Inc.** 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 **Microchip Technology Inc.** 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip** 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。**Microchip** 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 **Microchip** 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 **Microchip** 免于承担法律责任，并加以赔偿。在 **Microchip** 知识产权保护下，不得暗中以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、**Microchip** 徽标、dsPIC、FlashFlex、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC、SST、SST 徽标、SuperFlash 和 UNI/O 均为 **Microchip Technology Inc.** 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL 和 The Embedded Control Solutions Company 均为 **Microchip Technology Inc.** 在美国的注册商标。

Silicon Storage Technology 为 **Microchip Technology Inc.** 在除美国外的国家或地区的注册商标。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、SQI、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA 和 Z-Scale 均为 **Microchip Technology Inc.** 在美国和其他国家或地区的商标。

SQTP 是 **Microchip Technology Inc.** 在美国的服务标记。

GestIC 和 ULPP 为 **Microchip Technology Inc.** 的子公司 **Microchip Technology Germany II GmbH & Co. & KG** 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2013, **Microchip Technology Inc.** 版权所有。

ISBN: 978-1-62077-078-8

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



MICROCHIP

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:

<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 **Atlanta**

Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

波士顿 **Boston**

Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 **Chicago**

Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 **Cleveland**

Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 **Dallas**

Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 **Detroit**

Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯

Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 **Los Angeles**

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 **Santa Clara**

Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 **Toronto**

Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州

Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港特别行政区

Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门

Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海

Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄

Tel: 886-7-213-7828
Fax: 886-7-330-9305

台湾地区 - 台北

Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹

Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 **Australia - Sydney**

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 **India - Bangalore**

Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 **India - New Delhi**

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 **India - Pune**

Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 **Japan - Osaka**

Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 **Japan - Tokyo**

Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 **Korea - Daegu**

Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 **Korea - Seoul**

Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 **Malaysia - Kuala Lumpur**

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 **Malaysia - Penang**

Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 **Philippines - Manila**

Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 **Singapore**

Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 **Thailand - Bangkok**

Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 **Austria - Wels**

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 **Denmark - Copenhagen**

Tel: 45-4450-2828
Fax: 45-4485-2829

法国 **France - Paris**

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Munich**

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 **Italy - Milan**

Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 **Netherlands - Drunen**

Tel: 31-416-690399
Fax: 31-416-690340

西班牙 **Spain - Madrid**

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 **UK - Wokingham**

Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12