

MC9S08GB/GT 的低功耗模式

作者: Donnie Garcia 和 Scott Pape
德克萨斯州奥斯汀
8/16 位应用和系统工程部

介绍

本应用笔记介绍如何在使用 MC9S08GB/GT 单片机时达到低功耗。MC9S08GB/GT 属于低成本、高性能的 HCS08 8 位新型单片机系列。本产品系列新增的部分功能包括: 40MHz HCS08 CPU、增强型指令集和背景调试控制器 (BDC), 它为系统级实时调试提供了一个简便的接口。如需此产品更详细的介绍, 请参考其数据手册, 即飞思卡尔文献资料“MC9S08GB60/D”。网上的数据手册会提供最新的技术参数 (<http://www.freescale.com.cn>)。

MC9S08GB/GT 单片机增加了一些特别适合实现低功耗的功能。这些功能为用户提供了极大的灵活性, 能为各种不同应用提供理想的运行状态。

系统时钟生成

系统时钟可以由外部 (晶振、谐振器及方波) 或内部时钟源生成。而且, 可以用锁频环 (FLL) 将外部或内部时钟源提升到一个更高的频率。MC9S08GB/GT 可以使用低频 (32 kHz–100 kHz) 或高频 (1 MHz–16 MHz) 晶振以及谐振器。当系统启动时 (从停止状态启动或复位), 单片机会采用内部时钟源, 从而缩短系统启动时间。根据应用的要求选择最佳的系统时钟生成方式, 可以降低功耗。表 1 列出了不同时钟模式的配置注意事项。要了解 MC9S08GB/GT 时钟选项的更多资料, 请参考 AN2494/D:《在 MC9S08GB/GT 中配置系统外围时钟》(*Configuring the System Peripheral Clocks in MC9S08GB/GT*)。

表 1. ICG 配置注意事项

	时钟参考源 = 内部	时钟参考源 = 外部
启用 FLL	<p>(FEI) FLL 启用-内部时钟源</p> <p>$4\text{ MHz} < f_{\text{Bus}} < 20\text{ MHz}$</p> <p>中等功耗（若振荡频率范围 = 高的情况，则低于 FEE）</p> <p>中等时钟精度（IRG 调整后）</p> <p>系统成本最低（无需外部器件）</p> <p>IRG 打开。DCO 打开。⁽¹⁾</p>	<p>(FEE) FLL 启用-外部时钟源</p> <p>$4\text{ MHz} < f_{\text{Bus}} < 20\text{ MHz}$</p> <p>中等功耗（若振荡频率范围 = 低，则低于 FEI）</p> <p>高时钟精度</p> <p>中/高系统成本（需要晶振、谐振器或外部时钟源）</p> <p>IRG 关闭。DCO 打开。</p>
旁路 FLL	<p>(SCM) 自时钟模式</p> <p>该模式主要提供快速可靠的系统启动。</p> <p>$3\text{ MHz} < f_{\text{Bus}} < 5\text{ MHz}$（默认）</p> <p>$3\text{ MHz} < f_{\text{Bus}} < 20\text{ MHz}$（由过滤位）</p> <p>中等功耗</p> <p>低精度。</p> <p>IRG 关闭。DCO 打开并开环。</p>	<p>(FBE) FLL 旁路-外部时钟</p> <p>f_{Bus} 范围 $\leq 8\text{ MHz}$，当使用晶振或谐振器。</p> <p>功耗最低</p> <p>时钟精度最高</p> <p>系统成本中/高（需要晶振、谐振器或外部时钟源）</p> <p>IRG 关闭。DCO 关闭。</p>

1. IRG 通常耗电 $100\mu\text{A}$ 。根据不同的输出频率，FLL 和 DCO 耗电典型值在 0.5 至 2.5 mA 之间。为了尽量降低耗电和抖动现象，应尽可能选择最小的 N 和 R。

工作模式

复位后，正常工作模式为“运行”模式，此时 CPU 处于工作状态，外围设备可以启用。执行 WAIT（等待）指令后，单片机进入等待模式。在等待模式中，因为 CPU 没有时钟信号，功耗会降低。若要进一步降低功耗，可使用停止模式。执行 STOP（停止）指令后，可进入 3 种停止模式中的一种。它们分别是：停止 1、停止 2 和停止 3，每种模式能提供不同的低功耗运行等级。下表介绍了这些停止模式的动作。

表 2. 停止模式动作

模式	CPU、数字外围设备、闪存	RAM	ICG（内部时钟发生器）	ATD（模/数转换器）	KBI（键盘中断）	稳压器	I/O 管脚	RTI（实时中断）
停止 1	关	关	关	关闭 ⁽¹⁾	关	关	复位	关
停止 2	关	备用	关	关闭	关	备用	状态保持	选择性打开
停止 3	备用	备用	备用 ⁽²⁾	关闭	选择性打开	备用	状态保持	选择性打开

1. 根据 ATDPU 的状态，进入 ATD 停止模式或断电模式。

2. 停止 3 中可以配置运行晶体振荡器。请参见 ICG 寄存器。

实时中断（RTI）

实时中断可以用来退出停止 2 或停止 3 模式。在停止 3 中，它可以配置为使用外部或内部时钟源。在停止 3 中，使用内部时钟源比使用外部时钟源更省电。在停止 2 中，只能使用内部时钟源。可以配置实时中断模块来实现 8ms 至 1.024s 的实时中断功能。 1KHz 参考时钟的误差在 $\pm 30\%$ 左右，因此当实时中断采用内部时钟源时，唤醒时间会是近似值。

低电压检测（LVD）

MC9S08GB/GT 单片机能够在停止 3 模式中启用或关闭低电压检测。必须注意，如果要在“停止”状态中使用低电压检测，那么唯一可以采用的是停止 3 模式。如果在 SPMSC1 中的 LVDSE 位置 1，那么不管 SPMSC2 中的 PDC 和 PPDC 位是什么值，一旦执行 STOP（停止）指令，就会进入停止 3 模式。

工作电压范围

MC9S08GB/GT 单片机的指定工作电压范围从 3.6 V 到低至 1.8V（参见表 3）。低于 2.08V 运行时，最大总线速度应降至 8 MHz 或更低。电压较低时，所有模式的最终功耗也会降低。

表 3. 直流特性（环境温度 = -40 至 85 °C）

参数	符号	最低	典型	最高	单位
电源电压（运行、等待和停止模式）					
$0 < f_{\text{Bus}} < 8 \text{ MHz}$	V_{DD}	1.8	—	3.6	V
$0 < f_{\text{Bus}} < 20 \text{ MHz}$		2.08	—	3.6	

内部稳压器

MC9S08GB/GT 采用内部稳压器提供一个大约 2.4V 的内部电源，给 CPU 和大多数外围设备供电。当 V_{DD} 下降到低于 2.4 V 时，稳压器被旁路；当 V_{DD} 上升到高于 2.4 V 时，此稳压器可以防止运行电流的大幅上升。当单片机处于运行和等待模式时，稳压器总是开启的。在停止 2 和停止 3 模式中，稳压器处于一种宽松调节的状态，从而降低功耗。在停止 1 中，稳压器关闭。

低功耗模式介绍

停止模式

随着 HCS08 单片机系列的引入，两种新的停止模式也随之诞生，从而形成 3 种完整的停止形式。这 3 种停止模式分别称为停止 1、停止 2 和停止 3。停止 3 与 HC08 单片机的停止模式相同，而停止 1 和停止 2 则是两种新形式。

停止 1

停止 1 是完全断电的模式，此时内部稳压器处于关闭状态。这样就可以为单片机提供最低的待机电流。在停止 1 中，单片机的所有内部线路，包括 RAM 和所有寄存器都处于断电状态。这意味着进入该模式后，寄存器和 RAM 中的所有内容将丢失。同时，在停止 1 模式中所有外围设备都被关闭，包括低电压检测、实时中断、键盘中断和模/数转换器等。设备内部的时钟也都停止。I/O 脚驱动电路的内部电源关闭，输出脚下拉至地。I/O 脚恢复为高阻输入。

要退出停止 1 模式，可将复位或 $\overline{\text{IRQ}}$ 脚置低。在停止 1 模式中，无论进入停止 1 前 IRQ 如何配置，此时总是低电平有效。在停止 1 模式中 IRQ 和复位都自动内部上拉。

退出停止 1 模式后，单片机的寄存器配置与 POR（上电复位）发生后相同。不管由复位还是 $\overline{\text{IRQ}}$ 脚唤醒单片机，CPU 都读取复位矢量，然后开始执行代码。

停止 2

停止 2 提供的待机电流低于停止 3，但高于停止 1。停止 2 是部分断电模式，这时内部稳压器将进入“宽松的”调节模式，从而通过降低稳压器的输出功率来降低电流消耗。

在停止 2 中，RAM 保持供电，所有 I/O 管脚都锁定在它们进入该模式前的状态。配置为输入的管脚仍旧是输入，输出管脚则输出最后已知的状态。但是，所有由稳压器供电的外围设备（如低电压检测和键盘中断）都进入断电状态，不能继续使用。模/数转换器也关闭，不能继续使用。

尽管 I/O 管脚在停止 2 中保持原有状态，但所有寄存器的电源都被关闭。因此，在进入停止 2 前，需要保存的寄存器值，如 SCI、定时器或端口数据，都应复制到 RAM 中。

要退出停止 2 模式，可将复位或 $\overline{\text{IRQ}}$ 管脚置低。无论 $\overline{\text{IRQ}}$ 在进入停止 2 前如何配置，现在总是低电平有效。在进入停止 2 模式前，必须开启 $\overline{\text{IRQ}}$ 功能。

在停止 2 中，除了复位或 $\overline{\text{IRQ}}$ 管脚，还可以不依赖外部输入，开启并使用实时中断进行唤醒。但是，只有内部 1-kHz 实时中断振荡器才能用作停止 2 中的实时中断时钟源。一旦发生实时中断，将退出停止 2，类似发生过上电复位。

与停止 1 一样，退出停止 2 会导致寄存器重置为上电复位值，但有下述的例外。在写入 1 至 SPMSC2 的 PPDACK 位前，SPMSC2 的 PPDF 位将保持置 1，而且 I/O 管脚维持锁定当前状态。PPDF 位可以用作的跳转标志，从而执行停止 2 的恢复程序。为了继续保持 I/O 管脚的当前状态，应先将 RAM 中保存的寄存器值复制到它们各自的位置，然后再写 PPDACK 位。此时未被恢复的所有寄存器都会回复为上电复位值，其对应的 I/O 管脚也将回复为上电复位状态。停止 2 模式恢复后，在写入 PPDACK 之前，由于 I/O 被锁定，外围设备不能开始正常运行。

停止 3

HCS08 系列单片机的停止 3 与 HC08 系列的停止模式功能相同。所有 I/O 管脚的状态都锁定为执行停止命令前的状态。停止 3 中有一些其它两种停止模式没有的选项。停止 3 是唯一能够在停止期间提供 LVD 保护的停止模式。实际上，如果 SPMSC2 寄存器的 LVDSE 位设置为 1，单片机只能进入停止 3 的停止模式。

此外，该模式还可以将 OSCSTEN 位置 1，这样时钟发生器可以个别运行但单片机其余部分的时钟则处于关闭状态。如有必要，可以使用 OSCSTEN 选项来避免振荡器启动时间过长。这时，RTI 也能用外部时钟源作参考时钟产生实时中断。在定时精度要求严格的应用中，使用外部参考时钟可以提供准确的实时中断间隔。

从停止 3 退出比从停止 1 和停止 2 退出更平滑。如果使用中断源（如 $\overline{\text{IRQ}}$ 、键盘中断或实时中断）退出停止 3 模式，单片机先提供中断服务，然后继续执行停止指令后面的指令。退出停止 3 后不需要初始化外围设备。当然也可以利用复位管脚退出停止 3 模式。在这种情况下，单片机将读取复位矢量，寄存器和外围设备将回到复位状态。

等待

等待模式的功耗低于运行模式。在这种模式中，供给 CPU 的时钟处于关闭状态，从而减少功耗。其它外围设备在等待模式中都可开启。在该模式中，任何中断都可用于退出等待。常见的应用是先执行 WAIT（等待）命令，然后等待 SCI 或 SPI 中断，这样操作能持续运行。若由中断退出等待，单片机先执行中断服务程序，然后继续执行 WAIT（等待）命令后面的指令。

低功耗模式的使用

要进入 3 种停止模式的任何一种，都必须正确配置两个寄存器的三个位。在系统选项寄存器（SOPT）中，停止模式允许位（STOPE）必须设置为 1。该寄存器在复位后只能写入一次，因此必须注意在同

一次写入中配置其它选项。如果 STOPE 位清零，又开始执行 STOP（停止）指令，这时 STOP 指令将被当作非法操作码，单片机会强制复位。

在系统电源管理状态和控制寄存器 2（SPMSC2）中的两个位，断电控制位（PDC）和部分断电控制位（PPDC），决定执行 STOP（停止）指令时，进入 3 种停止模式中的哪种模式。

此外，为了能够进入停止 2 或停止 1 模式，SPMSC1 的 LVDSE 位必须清零。如果这一位没有清零，那么唯一能够进入的停止模式只能是停止 3。

表 4 简要概括了各种停止模式的退出源和退出后的状态。

表 4. 停止模式选择和退出源

模式	SPMSC2		退出源	退出后的状态
	PDC	PPDC		
停止 1	1	0	IRQ 或复位	上电复位
停止 2	1	1	IRQ、复位或实时中断	上电复位（SPMSCR 的 PPDF 位置 1）
停止 3	0	无关	IRQ、复位、实时中断或键盘中断	如果由复位退出，则上电复位；否则从中断向量开始继续正常运行。

停止 1

当 PDC 位设置为逻辑 1，PPDC 位设置为逻辑 0 时，执行 STOP（停止）指令后进入停止 1。停止 1 模式通过关闭内部稳压器及由其供电的所有片上外围设备，得到最小的电流消耗。同时，I/O 管脚和所有内存也都关闭。

如果功耗最为重要，而且又不需要单片机自唤醒，停止 1 是最合适的情形。只有 $\overline{\text{RESET}}$ 或 $\overline{\text{IRQ}}$ 的下降沿才能从该模式中唤醒单片机。

由于单片机的绝大部分在停止 1 期间都会断电，进入停止 1 只需要最少的软件处理。主要的考虑是在 SOPT 寄存器中允许 STOP 指令，以及通过 SPMSC2 寄存器的 PDC 位选择停止 1。此时不需要配置各个外围设备，因为进入停止 1 后它们都会自动断电。

既然 I/Q 管脚处于复位状态，那么所有 I/O 管脚都将回复为默认的输入状态并断开内部上拉。如果在停止模式中开启外部振荡器（ICG 控制寄存器 1 的 OSCSTEN 位），那么进入停止 1 后，该位的值将被忽略，时钟电源将被关闭。

如果在停止模式中启用 LVD 模块，停止 1 就不能采用。LVD 开启后，进入停止 1，会导致单片机进入停止 3 模式。

$\overline{\text{RESET}}$ 和 $\overline{\text{IRQ}}$ 管脚将自动配置为停止 1 的唤醒管脚。不需要软件或外部上拉。

从停止 1 唤醒后，单片机的启动与上电复位类似。由于所有寄存器回复为上电复位状态，并且 RAM 的电源也曾关断，所以没有能指示单片机是从停止 1 模式唤醒的机制。

由于上电复位会令内部 4-MHz 时钟驱动系统总线时钟，因此“停止恢复”过程相当快，从而在进入正常程序流之前，能快速执行代码以完成寄存器恢复。最长延迟是内部稳压器从关闭到开启，然后稳定所需的时间。从唤醒信号的下降沿到执行第一条指令的延迟通常为 50 μ s（当 $V_{DD} = 3\text{ V}$ ）和 80 μ s（当 $V_{DD} = 2\text{ V}$ ）。

停止 2

当 PDC 位和 PPDC 位都设置为逻辑 1 时，执行 STOP（停止）指令会进入停止 2。停止 2 的电流消

耗高于停止 1，但低于停止 3。RAM 继续保持供电，保持它的值和 I/O 管脚都锁存当前的状态。

采用停止 2 模式时，需注意以下几点以确保正常操作：

- 必须打开 $\overline{\text{IRQ}}$ 脚或将其外部上拉。
- 停止时 LVD 必须关闭 (LVDSE = 0)。
- 如果在停止 2 中采用实时中断，只有内部时钟源工作。
- OSCSTEN 位在停止 2 中没有作用。此时钟源总是电源关闭。
- 只有 RAM 保持供电，其它所有 I/O 寄存器唤醒后将复位。
- 从停止 2 进入后，在改变 I/O 管脚的状态前，必须保持 PPDF 标志清零。

必须通过写入 $\overline{\text{IRQ}}$ 状态和控制寄存器中 (IRQSC) 的中断管脚允许位 (IRQPE) 来开启 $\overline{\text{IRQ}}$ 引脚。否则进入停止状态后，单片机会马上从停止 2 唤醒，除非 $\overline{\text{IRQ}}$ 管脚上有外部上拉。IRQ 中断不需要启动 (IRQSC 的 IRQIE 位)。

$\overline{\text{IRQ}}$ 脚将自动配置为停止 2 的唤醒脚。不需要软件或外部上拉。

如果在停止模式中启动了 LVD 模块，就不能采用停止 2。在停止模式中启用了 LVD 后再进入停止 2，单片机会进入停止 3 模式。

在停止 2 中采用实时中断模块作为一个唤醒源时，必须使用内部时钟源，因为在停止 2 中外部时钟源不能保持供电。

如果在停止模式中开启外部振荡器 (ICG 控制寄存器 1 的 OSCSTEN 位)，然后进入停止 2，那么该位将被忽略，时钟电源会被关闭。

一般来说，应在 SOPT 寄存器中允许停止指令，并且 SPMSC2 寄存器的 PDC 和 PPDC 位必须设置为逻辑 1。

未提及的外围设备不需要任何特殊处理，因为进入停止 2 后它们的电源会自动被关闭。

如果要求有最低的功耗，同时又能保持 RAM 内容和 I/O 状态，则停止 2 是最适合的。由于实时中断模块能在停止 2 中运行，所以单片机不需要外部输入也可以唤醒。

从停止 2 唤醒后，单片机的启动过程类似 POR (上电复位) 发生后的情形。但是，与停止 1 不同的是，SPMSC2 寄存器的 PPDF 可以指示单片机是从停止 2 而不是从标准 POR 唤醒的。

通过使用 PPDF 和 PPDACK 位，用户代码能够在进入停止 2 前，将所有需要的寄存器值保存到 RAM 中，然后在唤醒后恢复这些值。如果在 PPDACK 写为逻辑 1 前，端口寄存器已经保存和恢复，那么其 I/O 状态将被保留。那些没有重新配置成停止 2 锁存状态的端口管脚都将恢复到复位状态。同样，没有重新配置成进入停止 2 之前的状态的所有外围设备也将恢复到复位状态。

进入和退出停止 2 的典型代码执行顺序如下：

```

; Constant declarations
IRQSCinit:    equ    $10                ; enable the IRQ pin
SOPTinit:    equ    $A0                ; enable COP and STOP
SRTISCinit:  equ    $17                ; enable int and select 1.024 sec timeout
SPMSC2init:  equ    $03                ; PDC & PPDC both set
SPMSC2st2:   equ    $07                ; PPDACK, PDC & PPDC both set
PPDFmask:    equ    $08                ; mask for PPDF bit in SPMSC2 reg
...
1) System initialization after reset
Start:        lda    SPMSC2            ; Check if coming from stop2
              and    #PPDFmask
              bne    Stop2rec         ; If so, branch to recovery code
              lda    #SOPTinit        ; Else, treat as normal POR
              sta    SOPT              ; init the System Options
              lda    #SPMSC2init
              sta    SPMSC2           ; init the SPMSC2 reg
              mov    #IRQSCinit,IRQSC ; init the IRQ pin
...
2) Entering stop with RTI enabled
              jsr    SaveRegs
              lda    #SRTISCinit      ; Enable RTI module
              sta    SRTISC
              stop
...
3) After RTI times out, a POR will execute and code restarts at reset vector but this time PPDF
will be set

Start:        lda    SPMSC2
              and    #PPDFmask
              bne    Stop2rec
...
Stop2rec:     jsr    LoadRegs
              lda    # SPMSC2st2
              sta    SPMSC2
              bra    Main
...
; Begin Main code execution
Main:

```

注意常量 SPMSC2st2 不仅设置 PPDACK 位以清除 PPDF 标记, 还将 PDC 和 PPDC 位设置为逻辑 1。这是因为它们都是一次性写入位。如果在一次写操作中未能将这些位设置为逻辑 1, 那么下一个 STOP (停止) 指令会进入停止 3 模式, 而不是停止 2。当然, 如果需要, 用户可以在这条指令中选择开启停止 1 或停止 3, 而不是停止 2。

与停止 1 相同, POR 会令系统总线时钟由内部 4-MHz 时钟驱动, 停止恢复过程相当快, 从而允许代码迅速执行以恢复寄存器设置。稳压器恢复完全调节状态的延迟与停止 1 相同, 大概是 $V_{DD} = 3 \text{ V}$ 时为 $50\mu\text{s}$, $V_{DD} = 2 \text{ V}$ 时为 $80\mu\text{s}$ 。

停止 3

停止 3 模式不能提供最低的 I_{DD} 电流, 但它功能多, 而且在所有停止模式中影响最小。只要 SPMSC2 的 PDC 位设置为 0, 即可进入停止 3。此外还应注意, 如果在停止模式下启动了 LVD, 或进入背景调试模式时也启动了 LVD (BDCSCR 中的 ENBDM 位已经设定为 1), 则此时唯一能够进入的停止模式是停止 3。当 ENBDM 位设置为 1 后执行停止指令, 背景调试逻辑的系统时钟仍然有效, 因此仍能继续背景调试通信。

如果用户需要能够轻松退出的停止模式, 可以采用停止 3。使用内部时钟或 FLL 时, 停止恢复时间通

常为 100 μs 左右。在使用 FLL 对参考频率进行倍频的应用中，停止 3 有一个优点，在通过中断从停止 3 中恢复时，能保留 DCO 的原有设置。就是说在停止恢复时，DCO 将用先前定义好的系统时钟配置进行设置。

与其它停止模式不同的是，如果停止 3 通过中断退出，是不需要进行任何初始化或重新配置的。中断发生后，CPU 开始处理堆栈操作，引导其进入中断服务程序。执行中断返回 (RTI) 指令后，CPU 会回复到紧跟在 STOP (停止) 指令后的那一条指令处。

另一个可能需要采用停止 3 模式的情况是必须使用键盘中断 (KBI) 模块的时候。MC9S08GB/GT 有 8 个键盘中断脚 - 每一个管脚都能从停止 3 中唤醒芯片。键盘中断模块不能在停止 1 或停止 2 中使用，但是在某些应用中，必须有多种退出停止模式的信号。

停止恢复时间

停止 1 和停止 2 的停止恢复时间非常接近。从停止 1 或停止 2 恢复时，内部稳压器需要一段时间来重新建立 2.4 V 电压。因此，停止 1 和停止 2 的恢复时间与 V_{DD} 有关。如果 V_{DD} 为 3V，那么停止恢复时间大约为 50 μs；如果 V_{DD} 为 2V，则停止恢复时间大约为 80 μs。表 5 列出了使用 M68DEMO908GB60 演示板和低功耗模式代码时的测量值。

表 5. 停止 1 和停止 2 的大致停止恢复时间

模式	V _{DD} = 3 V	V _{DD} = 2 V
停止 1	50 μs	80 μs
停止 2	50 μs	80 μs

停止 3 的启动时间在很大程度上取决于所采用的时钟配置。停止 3 在不同的 V_{DD} 下的最快恢复时间应为 100 μs 左右。除非 ICG 配置为 FLL 旁路外部时钟模式 (FBE)，这样的停止恢复时间都是能得到的。100 μs 是室温下的典型测量值。在 FBE 模式中，停止 3 的恢复时间因外部时钟频率而有所差异。例如，使用 32-kHz 晶体时，在晶体未达到稳定状态之前，是不会开始停止恢复操作的。在试验室测试中，采用 32-kHz 晶体时停止 3 的恢复时间约为 180 ms–300 ms。如果 OSCTEN 位置 1 (在停止模式中开启振荡器)，则恢复时间为 2.42 ms。这个延迟是由于 16 个周期的计数和中断读取开销造成的。如果外部时钟频率提高，停止恢复时间将减少。使用 32-kHz 晶体对停止恢复时间进行测量是最坏的情形，因为低频晶体需要很长的启动时间。

如果使用复位退出停止 3，停止恢复时间与停止 1 和停止 2 大致相等。停止 3 恢复时间数据参见表 6。

表 6. 停止 3 的大致停止恢复时间测量值

时钟源	恢复时间
使用内部参考时钟	100 μs
外部 32kHz 晶体 (OSCTEN = 0)	180 ms–300 ms
外部 32kHz 晶体 (OSCTEN = 1)	2.4 ms

特殊低功耗模式下的 I_{DD} 典型值

表 7 和表 8 的数据测自 M68DEMO908GB60 演示板。表后所提供的代码可以用作进入不同停止模式的样板程序。

表 7. 室温下 V_{DD} = 3.12 V 时的 I_{DD} 典型值

模式	电流
停止 1	57 nA
停止 2	590 nA
启动实时中断的停止 2	890 nA
停止 3	750 nA
启动实时中断的停止 3	1.1 μA
启动实时中断的停止 3 (带外部 32-kHz 时钟源)	14.5 μA

表 8. 室温下 V_{DD} = 2.0 V 时的 I_{DD} 典型值

模式	电流
停止 1	17 nA
停止 2	400 nA
启动实时中断的停止 2	700 nA
停止 3	525 nA
启动实时中断的停止 3	850 nA
启动实时中断的停止 3 (带外部 32-kHz 时钟源)	10.5 μA

创建实现低功耗功能的代码

这里列出了一些有关创建实现低功耗功能的代码的一些重要注意事项。所有 I/O 管脚都要初始化为低驱动能力的输出或带上拉的输入。这样就可以保证不会因浮动输入而产生额外的电流消耗。

为了达成尽可能低的 I_{DD} 电流，在停止模式中应关闭 LVD。在停止模式中，代码是不执行的，因此低电压导致代码跑飞的风险本身就是最小的。不过，若在停止 2 或停止 3 中使用实时中断，风险就会增加。如果在停止状态中启动了 LVD，则不能进入停止 1 和停止 2；但在停止 3 中，程序代码应确定在停止状态中关闭 LVD。对 OSCSTEN 位也是如此。如果在停止模式中仍旧允许振荡器运行，则无法实现最低的功耗。

在处理停止 2 的恢复时，一定要注意在将 PPDACK 位置 1 前，所有寄存器和 I/O 都处于锁定状态。例如，在置 PPDACK 位前，端口管脚的状态不能改变。

以下是在这篇应用笔记中用于进行测量的汇编软件的列表。

```

Metrowerks HC08-Assembler
(c) COPYRIGHT METROWERKS 1987-2003
Rel. Loc  Obj. code  Source line
-----
1          ;*****
2          ;*      Copyright (c) Freescale 2003
3          ;*****
4          ;*File name:      Low_Power_Modes.asm Current Release Level:  1.0
5          ;*Last Edit Date: 06-May-03      Classification:      ES
6          ;*
7          ;*Include Files: 9S08GB60v1r3.inc MC68HC9S08GB60 MCU definitions
8          ;*Assembler:   CW Assembler V5.0.13      Version:      3.16
9          ;*Target Device: MC68HC9S08GB60
10         ;*Documentation: GB60 Low Power Modes AN2493
11         ;*****
12         ;* Author:      Donnie Garcia
13         ;* First Release: 06-May-03
14         ;*
15         ;* Update History:
16         ;*
17         ;* Rev      Date      Author  Description of Change
18         ;* -----
19         ;* 1.0      5-06-03      DG      Initial Release
20         ;*
21         ;*****
22         ;*****
23         ;* This code is used along with M68DEMO0908GB60 board to demonstrate
24         ;* Stop Modes
25         ;* The measurements in AN2493 where taken using this code and the
26         ;* Demo Board
27         ;* For Measurement purposes all headers/jumpers (Except the Power_Sel
28         ;* jumper) were removed from the demo board
29         ;* When using Stop2 or Stop1 in order to re-establish BDM connection
30         ;* PTG0/BKGD should be held low on power up, then released.
31         ;*****
32         ;*      StopSelect and WakeSelect are used to configure the code
33         ;* To test Stop1
34         ;* StopSelect = %00000001 WakeSelect = Dont Care
35         ;*
36         ;* To test Stop2
37         ;* StopSelect = %00000010 WakeSelect = %00000000
38         ;*
39         ;* To test Stop2 RTI
40         ;* StopSelect = %00000010 WakeSelect = %00000010
41         ;*
42         ;* To test Stop3
43         ;* StopSelect = %00000100 WakeSelect = %00000001
44         ;*
45         ;* To test Stop3 RTI Internal

```

```

46      ;* StopSelect = %00000100 WakeSelect = %00000010
47      ;*
48      ;* To test Stop3 RTI External
49      ;* StopSelect = %00000100 WakeSelect = %00000100
50      ;*****
51      include "9S08GB6Ov1r3.inc"
52      ;*****
53      ;SELECT STOP MODE AND WAKE UP SOURCE HERE
54      ;
55      0000 0004 StopSelect: equ      %00000100 ;Select Stop Mode Here
56      ;                |||
57      ;                | |--Stop1 Mode selected
58      ;                | +---Stop2 Mode selected
59      ;                +----Stop3 Mode Selected
60      ;If more than 1 mode is selected the lowest stop mode will be set
61      ;If no selection is made stop3 is chosen
62
63      0000 0002 WakeSelect: equ      %00000010 ;Select Method of wake up (Stop2,3)
64      ;                |||
65      ;                | |--KeyBoard wake selected (For Stop3)
66      ;                | +---RTI Internal wake selected
67      ;                +----RTI external wake Selected (For Stop3)
68      ;If No selection is made KBI is selected
69      ;*****
70
71
72      ;IMPORTANT REGISTER INITIS
73      ;
74      0000 0014 initSPMSC1: equ      %00010100 ;Disable LVD in stop
75      ;                |||||
76      ;                |||||
77      ;                |||||
78      ;                ||||| +----LVDE Enable LVD
79      ;                ||| +----LVDS Disable LVD in stop
80      ;                ||| +----LVDR Enable LVD reset protection
81      ;                | +-----LVDIE
82      ;                | +-----LVDACK
83      ;                +-----LVDF
84
85      0000 0000 initSPMSC2: equ      %00000000 ;This register sets stop mode
86      ;                |||||
87      ;                ||||| +--PPDC
88      ;                ||||| +--PDC
89      ;                ||| +----PPDACK
90      ;                ||| +----PPDF
91      ;                || +-----LVWV
92      ;                | +-----LVDV
93      ;                | +-----LVWACK
94      ;                +-----LVWF
95      ;
96
97
98      0000 0063 initSOPT:   equ      %01100011 ;COP and STOP enable controls
99      ;                |||  |||
100     ;                |||  | +--RSTPE --- Reset pin enabled
101     ;                |||  +--BKGDPPE -- BKGD pin enabled
102     ;                | +-----STOPE --- STOP allowed
103     ;                | +-----COPT --- long timeout 2^18
104     ;                +-----COPE --- COP off

```

```

105
106
107      0000 003C  initICGC1:  equ    %00111100 ;Clock Generator Control 1
108      ;                                     0| | | | |xx      this setting for 32 kHz xtal
109      ;                                     | | | | |+---OSCSTEN - keep osc on in stop mode
110      ;                                     | | | | |+----CLKS0 - select PLL engaged external
111      ;                                     | | | | |+----CLKS1 /   (FEE) mode
112      ;                                     | | | | |+-----REFS ---- enable oscillator amplifier
113      ;                                     +-----RANGE --- 32.768 kHz crystal
114
115
116      0000 0021  initICGC2:  equ    %00100001 ;Clock Generator Control 2
117      ;                                     | | | | | | | | should write MFDx before ICGC1
118      ;                                     | | | | | | | | +--RFDO \
119      ;                                     | | | | | | | | +--RFD1 --- post-PLL divider
120      ;                                     | | | | | | | | +--RFD2 *
121      ;                                     | | | | | | | | +----LOCKRE --- loss of clock doesn't reset
122      ;                                     | | | | | | | | +----MFDO \
123      ;                                     | | | | | | | | +-----MFD1 --- PLL loop multiplier N
124      ;                                     | | | | | | | | +-----MFD2 *
125      ;                                     +-----LOLRE --- loss of lock doesn't reset
126
127
128      0000 0000  LED          equ    0
129      ;*****
130      ;
131      org      RamStart
132
133      0080      StopSet      RMB     1          ;Used to select stop mode
134      0081      WakeSet     RMB     1          ;Used to select Wait Mode
135      0082      PTFD_STORE  RMB     1          ;Used to store PTF
136
137      ;
138      org      RomStart
139
140      START:
141
142      1080 AE 63      ldx      #initSOPT
143      1082 CF 1802   stx      SOPT          ;Disable COP and enable STOP
144
145      1085 AE 14      ldx      #initSPMSC1
146      1087 CF 1809   stx      SPMSC1       ;Disable LVD in stop
147
148      108A C6 180A   lda      SPMSC2       ;how did we get here?
149      108D A4 08     and      #mPPDF       ;was it a wake-up from STOP2?
150      108F 26 1C     bne     Stop2Recovery ;If = 0 was normal reset
151      ;If = 1 Stop2 Recovery is needed
152
153      ;This begins the path of a normal reset (Not stop2 recovery)
154
155      INIT:
156      ;*****
157      ;FIRST setup SPMSC2 to to the proper stop mode
158      1091 A6 04     lda      #StopSelect
159      1093 B7 80     sta      StopSet
160
161      1095 00 80 0E   brset   0,StopSet,Set_Stop1
162      1098 02 80 02   brset   1,StopSet,Set_Stop2
163

```

```

164
165 109B 20 10      Set_Stop3:      bra      StopSelectDone ;Reset state of SPMSC2 selects stop3
166
167
168 109D B6 00      Set_Stop2:      lda      initSPMSC2 ;enable stop2
169 109F AA 03      ora      #(mPDC|mPPDC)
170 10A1 C7 180A    sta      SPMSC2
171
172
173 10A4 20 07      bra      StopSelectDone
174
175
176 10A6 B6 00      Set_Stop1:      lda      initSPMSC2 ;enable stop1
177 10A8 AA 02      ora      #(mPDC)
178 10AA C7 180A    sta      SPMSC2
179
180
181
182
183
184
185
186
187 10AD A6 02      StopSelectDone:
188 10AF B7 81      ;*****
189 10B1 00 81 08   Stop2Recovery:      ;Initialize before PDACK
190 10B4 02 81 0F   ;*****
191 10B7 04 81 13   ;Now set up the selected wakeup source
192 10BA 20 1F      lda      #WakeSelect
193
194
195 10BC 18 17      sta      WakeSet
196 10BE 12 16      brset   0,WakeSet,InitKBI
197 10C0 14 16      brset   1,WakeSet,InitRTIint
198 10C2 18 01      brset   2,WakeSet,InitRTIext
199 10C4 20 15      bra      WakeSelectDone
200
201 10C6 AE 17      InitKBI:
202 10C8 CF 1808    bset   KBIPE4,KBIPE ;Enable Keyboard Pin
203 10CB 20 0E      bset   KBIE,KBISC ;Enable Keyboard Interrupts
204
205 10CD AE 37      bset   KBACK,KBISC ;Clear Pending Keyboard Interrupts
206
207 10CF CF 1808    bset   PTAPE4,PTAPE ;Enable Pullup for Keyboard pin
208 10D2 6E 21 49   bra      WakeSelectDone
209 10D5 6E 3C 48   InitRTIint
210 10D8 07 4A FD   ldx   #$17 ;Enable RTI Interrupts 1s timeout
211
212
213
214
215
216
217 10DB 6E 10 14   stx   SRTISC
218
219 10DE 6E EF 03   bra      WakeSelectDone
220 10E1 6E FF 07   InitRTIext
221 10E4 6E FF 0B   ldx   #$37 ;External clock bit set
222 10E7 6E FF 0F   ;Enable RTI Interrupts long timeout

```

```

223 10EA 6E FF 13      mov     #$ff,PTEDD ;EDDR
224 10ED 6E FF 43      mov     #$ff,PTFDD ;FDDR
225 10F0 6E FF 47      mov     #$ff,PTGDD ;GDDR
226
227 10F3 6E 00 00      mov     #$00,PTAD ;ADR
228 10F6 6E 00 04      mov     #$00,PTBD ;BDR
229 10F9 6E 00 08      mov     #$00,PTCD ;CDR
230 10FC 6E 00 0C      mov     #$00,PTDD ;DDR
231 10FF 6E 00 10      mov     #$00,PTED ;EDR
232 1102 6E 01 40      mov     #$01,PTFD ;FDR
233 1105 6E 00 44      mov     #$00,PTGD ;GDR
234
235 1108 C6 180A      lda     SPMSC2 ;how did we get here?
236 110B A4 08        and     #mPPDF ;was it a wake-up from STOP2?
237 110D 27 13        beq     MainLoop
238
239 110F 4E 82 40      mov     PTFD_STORE,PTFD ;Replace PTF with stored info
240 1112 C6 180A      lda     SPMSC2 ;acknowledge Stop2 recovery
241 1115 AA 07        ora     #(mPPDACK|mPDC|mPPDC)
242 1117 C7 180A      sta     SPMSC2
243
244 111A B6 40        lda     PTFD ;Toggle LED here for Stop2
245 111C A8 01        eor     #mPTFD0
246 111E B7 40        sta     PTFD
247 1120 2E FE        bil     * ;Wait while IRQ is low (Debounce)
248
249
250 1122 B6 40        MainLoop lda     PTFD
251 1124 B7 82        sta     PTFD_STORE ;Store PTF state into RAM
252 1126 8E          stop
253 1127 20 F9        bra     MainLoop
254
255 ;*****Interrupt Service Routines*****
256 kbi_isr
257 1129 14 16      bset    KBACK,KBISC ;Acknowledge KB Interrupt
258 112B B6 40      lda     PTFD ;Toggle LED Here
259 112D A8 01      eor     #mPTFD0
260 112F B7 40      sta     PTFD
261 1131 80          rti
262
263 1132 C6 1808      rti_isr lda     SRTISC
264 1135 AA 40      ora     #mRTIACK
265 1137 C7 1808      sta     SRTISC ;Acknowledge RTI Interrupt
266 113A B6 40      lda     PTFD ;Toggle LED Here
267 113C A8 01      eor     #mPTFD0
268 113E B7 40      sta     PTFD
269 1140 80          rti
270
271 ;*****Vectors*****
272 org     Vrti
273 FFCC 1132      fdb     rti_isr
274 org     Vkeyboard
275 FFD2 1129      fdb     kbi_isr
276 org     Vreset
277 FFFE 1080      fdb     START

```

结论

可以采用多种方式配置 MC9S08GB/GT 以实现低功耗。本文介绍的 3 种停止模式为用户应用提供了不同的解决方案。I_{DD} 电流数据显示，在所有停止模式中，工作电流都可以保持很低。多种停止模式的灵活性，再加上其它功能（如实时中断、运行电压范围和时钟配置选择）使 MC9S08GB/GT 成为低功耗应用中的理想单片机。

下面是各种低功耗模式的功能总结：

- 停止 1 是完全断电模式，2V 电压下的典型电流为 20 nA
 - 所有 RAM 和寄存器内容会丢失
 - 通过 IRQ 或复位可退出停止 1；退出需要外部事件
- 停止 2 是部分断电模式，2V 电压下的典型电流为 400 nA
 - 保持 RAM 内容、I/O 状态锁定
 - 通过 IRQ、复位或内部自动唤醒定时器退出
 - 需要对要使用的各外围设备进行初始化
- 停止 3 与 M68HC08 单片机的停止模式相当，2V 电压下的典型电流为 500 nA
 - 通过 IRQ、键盘中断、低电压检测、内部自动唤醒定时器或复位退出
 - 允许时钟发生器运行，但不提供给外围设备，因此可以使用外部参考时钟
 - 不需要初始化外围设备
- 等待模式关闭 CPU 的时钟，但可以提供给外围设备，2 V 电压、1 MHz 频率下的典型电流为 420μA（FBE）
 - 通常用于等候中断时，如 SPI 接收中断
 - 立即处理中断服务程序
- 运行模式时，2 V 电压、1 MHz 频率下的典型电流为 640μA

本应用笔记包括 2 个压缩文件 AN2493SW1.zip 和 AN2493SW2.zip。AN2493SW1.zip 只包含实现低功耗模式的软件的汇编和头文件。

AN2493SW2.zip 则包括完整的 CodeWarrior 项目文件夹。项目文件夹第一级中有扩展名为“.mcp”的 CodeWarrior 项目文件。安装 CodeWarrior 后，双击此文件可打开项目。项目已经被汇编，列表文件（扩展名为“.lst”）在“bin”子文件夹中提供。此外，S 记录文件（扩展名为“.s19”）也在该文件夹中。

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™, 飞思卡尔™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2005

