



更多关于 ADI 公司的 DSP、处理器以及开发工具的技术资料，

请访问网站：<http://www.analog.com/ee-note> 和 <http://www.analog.com/processor>

如需技术支持，请发邮件至 processor.support@analog.com 或 processor.tools.support@analog.com

利用 Blackfin® 处理器休眠状态实现待机低功耗

作者: Jeff Sondermeyer, Joe Beauchemin, Hiren Desai

Rev 1 – May 5, 2008

引言

Blackfin® 处理器的一个重要优势就是每个 MIPS 的有效功耗很低 (~0.16mW/MIPS)。然而，由于高速存储器的物理特性 (~600 MHz)，处理器上的待机电流 (静态漏电流) 大概在 1-10mA 之间，具体值取决于 Blackfin 处理器的不同型号。便携式媒体应用对待机电流的要求通常不大于几百 μA 。对 Blackfin 处理器，1-10 毫安的待机电流大概要比所要求的要高。便携式媒体播放器 (PMP) 也特别要求较高的 MIPS ($\geq 400\text{MHz}$)。考虑到目前市场上所出现的各种处理器，可以得出一个大概结论：低 MIPS 的处理器待机功耗通常较低，而高 MIPS 的处理器待机功耗通常较高。在处理技术和待机功耗之间有一种间接关系——随着处理技术规模的不断减小，待机功耗变得更大了。所以，如何得到高 MIPS 的同时还能保证低的有效功率和待机功耗呢？

本文就是通过利用 Blackfin 处理器的休眠状态的特性，对该问题进行讨论。根据定义，休眠状态可在保持输入输出能源的情况下，移除内核的电源。但是，当内核的电源被移除之后，处理器的内部 SRAM 的信息将会丢失。大多数 Blackfin 处理器有一个实时时钟 (RTC)，用于周期性地从休眠状态中唤醒内核，还有几种 Blackfin 处理器允许利用其他信号把内核从休眠状态唤醒。本文介绍在休眠周期中用 RTC 唤醒事件保留信息的方法。

对低待机功耗的需求

随着 PMP 的出现，应用对电池的工作时间要求越来越高，要求带负载能够长时间地稳定应用。在这一挑战之外，这些设备的外形尺寸在减小，迫使电池变得更小 (也就是说，电池的额定 mA-h 在降低)。例如，便携式 MP3 播放器 (见图 1) 希望能够利用一个 400 mA-h 的电池在单个负载上连续播放 24 小时。表 1 为一个典型的 MP3 播放器的电力消耗。

对这些应用重要的不只是有效功耗，待机功耗对电池的寿命也非常关键。讨论它们的理由是，*待机功耗* 是为了保持处理器处于一个待命状态所需的功耗。处理器的大部分时间都将处于这种状态，而且必须能够对用户的交互作用 (如按下按键) 给以回应并能恢复正常运作。

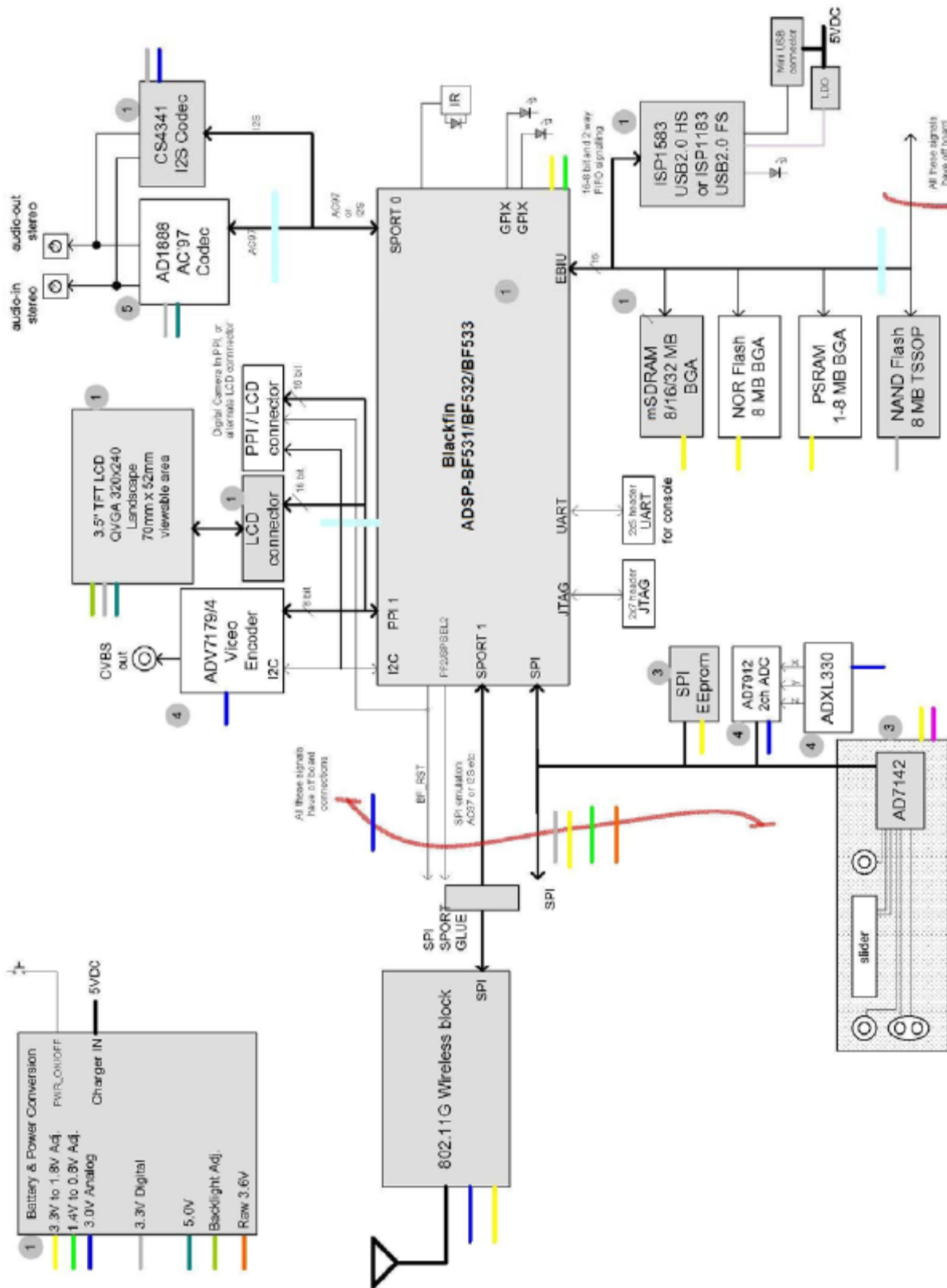


图1 ADSP-BF531 便携式媒体参考设计

休眠状态的使用

Blackfin 处理器的休眠状态是通过关闭内核电源而实现的，它使内核时钟(CCLK)和系统时钟(SCLK)都失去作用。当内核电源被关闭，内部供电电压(VDDINT)就被置为 0V，导致处理器的内部状态丢失。因此，任何内部存储的重要信息(例如，存储器目录，寄存器目录，等等)必须在关闭电源之前将其写入非易失性存储设备中。在文中，这一非易失性存储器就是 mSDRAM。

V_{DDINT} 的移除避免了处理器上的很大一部分泄漏电流，从而显著降低了系统的功耗。该 mSDRAM 存储器在休眠状态下置于自动刷新模式，这样就可以保留其信息。在 ADSP-BF531/BF532/BF533 Blackfin 处理器中，休眠状态可以通过使/RESET 引脚有效或者通过实时钟(RTC)唤醒事件进行唤醒，从而产生一次硬件复位。

Blackfin 处理器的最优解决方案中，对 PMP 市场尤其有益有以下几点：

- Blackfin 处理器在休眠状态下只需要~ 50 μ A 的电流。
- 外部 mSDRAM(1.8 V)在自动刷新模式下的待机电流很低。例如，当 Micron MT48H4M16LF - 1M x 16 x 4-存储区 SDRAM 设备处于自动刷新模式下，四个存储区都打开时，它的电流只有 ~ 100 μ A。
- 它在处理器内核电源开启的瞬间迅速将信息从 mSDRAM 恢复(热重启)至处理器的内存存储器(不到 2 毫秒即可恢复~96 KB)。而较慢的冷重启则是不必要的。

在实现这一系统时硬件和软件的需求是必须考虑的，以下分别进行描述。

软件考虑

我们利用两个 VisualDSP++[®]的例子来说明文所讨论的概念。一个例子运行于 ADSP-BF533 EZ-KIT Lite[®]评估板上。另一例子则说明对便携式媒体应用中通过利用休眠状态完整信息的保存和恢复作用于所需的附加硬件。一个利用 1.8V 的 mSDRAM 的 PMP 参考设计和高效外部处理器内核稳压器用作硬件平台。这个便携式多媒体参考设计(图 1)现在还没有公开实现，但是这里对相关的硬件细节进行了深入讨论。

这两个例子都是利用 VisualDSP++ 4.5 开发。假定便携式应用更青睐于利用一个多线程的 RTOS/OS 环境，所以选择利用了 VisualDSP++Kernel(VDK)。全部工程，BF533_EZKIT_Hibernation_Example.dpj 和 PMP_Hibernation_Example.dpj 都包含在相关的.ZIP 文件里。



BF533_EZKIT_Hibernation_Example.ldr 图像文件必须编程到 ADSP-BF533 EZ-KIT 板的闪存中。需要一个初始化模块来完成休眠状态中信息的恢复。

代码概述

以下是对代码操作的概述。代码操作有四条线程，其中有一个中断服务线程，两个信号灯线程：

线程:

1. 主程序(优先级 5)——配置外设并引发休眠和闪烁 LED 线程。
2. 闪烁 LED(优先级 6)——在 ADSP-BF533 EZ-KIT 板上简单实现 LED4 的 15 次闪烁。
3. 休眠(优先级 7)——关闭外设, 将系统状态存入 SDRAM, 并使处理器处于休眠状态。在线程进入休眠状态之前, 先对 RTC 进行设置并将 SDRAM 置于自动刷新状态。在唤醒前, 处理器返回这一线程, 而外设也重新配置。
4. 空闲(优先级最低的线程)——默认 VDK 线程。没有任务时处理器执行这一线程。

中断:

1. 休眠——利用一个按键使处理器进入休眠状态。按键 SW6 (PF10)设置为输入。在 SW6 按下时, 处理器将激活 EVT_IVG9_Entry 中断处理 (源于 Hibernate_IVG9.c)。这个中断服务线程简单地使能按键信号灯。

信号灯:

1. 按键——当用户按下 SW6(PF10)按键时, 信号灯使能。休眠线程决定于信号灯入口。如果信号灯有效(用户按下 SW6), 线程则继续运行且处理器置于休眠状态。如果信号灯无效(用户没有按下 SW6), 休眠线程模块和调节器将处理器置于空闲线程中。
2. 闪烁——当处理器从休眠状态恢复时, 这个信号灯在休眠线程末端使能。闪烁线程在进入闪烁线路之前决定于信号灯。如果信号灯有效, LED4 闪烁 15 次。如果信号灯无效, 闪烁线程模块和调节器将处理器置于休眠线程。

图 2 对代码操作进行了说明。

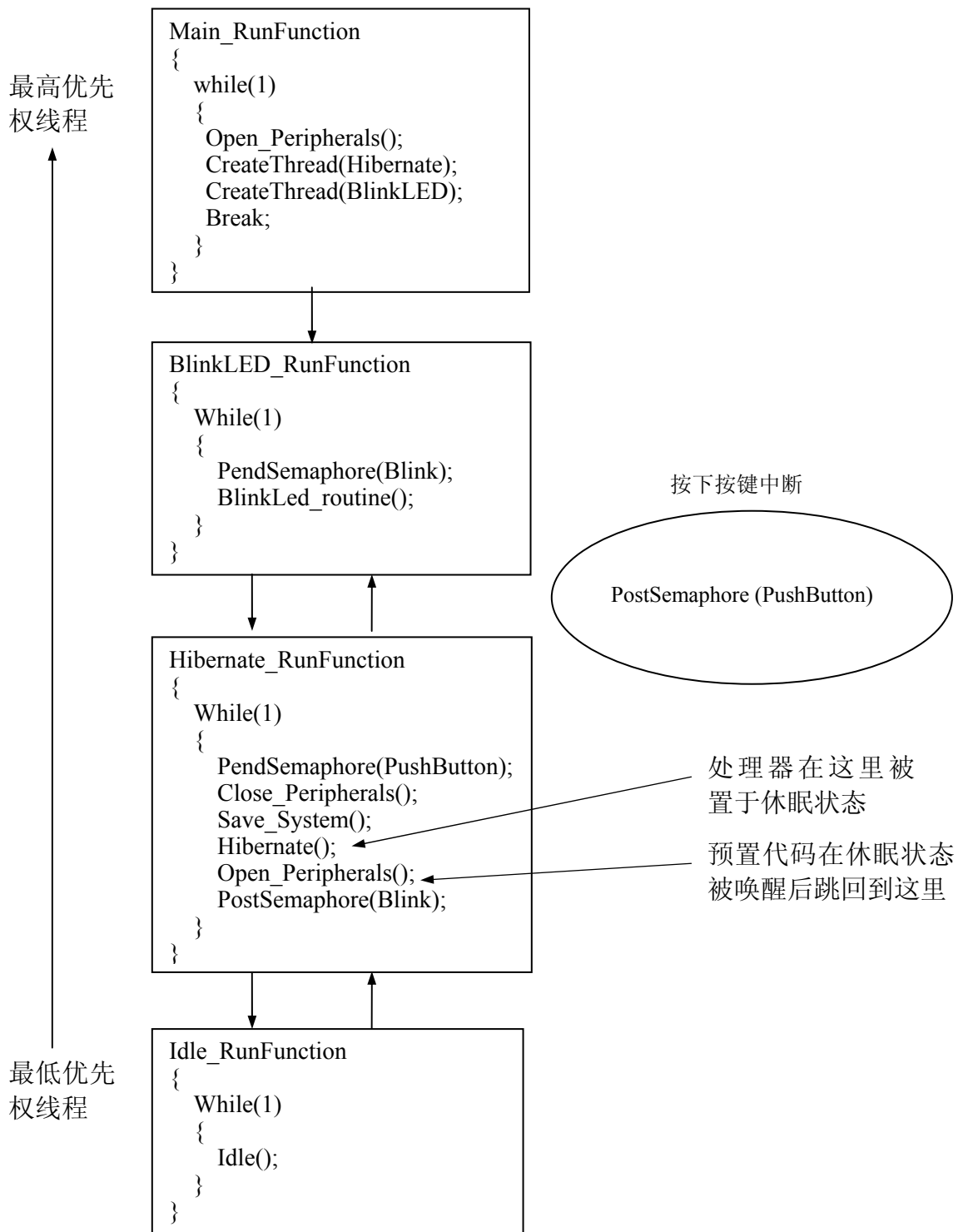


图2 代码操作

SDRAM 用法

在处理器置于休眠状态前，片上存储器的信息存入 SDRAM 存储器。表 2 说明存储什么信息，存到 SDRAM 的什么地方，以及用于存储数据的方法。

片上存储器资源	SDRAM地址	方法
关键字(值为0x12345678)—用于预置代码	0x03FEC	内核读/写
堆栈指针(SP)	0x03FF0	内核读/写
帧指针(FP)	0x03FF4	内核读/写
L1到L3一览表—用于预置代码	0x03FF8	内核读/写
返回地址(RETS)—用于预置代码	0x03FFC	内核读/写
L1指令存储器(0xFFA00000)	0x04000	存储器DMA
L1数据存储区A(0xFF800000)	0x0C000	存储器DMA
L1数据存储区B(0xFF900000)	0x14000	存储器DMA
中间结果暂存器(0xFFB00000)	0x1C000	内核读/写
事件向量表	0x1D000	内核读/写
IMASK	0x1D010	内核读/写

表2 休眠状态之前保存的存储器资源

RTC 休眠唤醒

在处理器置为休眠状态之前设置 RTC，利用其秒表功能生成一个中断。用一个 32.768kHz 的晶体对 RTC 计时，秒表被设置为 80/32768(~2.44 ms)。2.44ms 定时器终止后，RTC 调用一个内部复位，使后台部分跳出休眠状态并迫使调节器重新将电压供给处理器内核和 L1 存储器。



RTC 唤醒只在 BF533_EZKIT_Hibernation_Example.dpj 工程中进行了测试。

在休眠状态唤醒时，处理器启动取决于 BMODE 引脚的状态。在这个例子中，一个初始化模块 (init_code.dxe) 提前确定启动数据流，这在 *ADSP-BF533 Blackfin Booting Process (EE-240)*^[2] 里有详细说明。初始化代码对 SDRAM 存储器进行设置并决定处理器是否从休眠状态中跳出。它通过对存储关键字的 SDRAM 存储器中的 0x3FEC 单元的检测来实现这一功能。如果 0x3FEC 单元里没有 0x12345678 值，处理器将从非休眠硬件复位(冷启动)跳出并要求一次全面启动。对于全面启动，初始化代码简单存储信息并返回启动 ROM 来实现应用软件的全面启动。如果关键字设为 0x12345678，处理器就跳出休眠状态(热启动)，且不需要全面启动。这种情况下，初始代码利用指向 DMA 描述符的 0x3FF8 单元的指针来建立一个存储器 DMA 传输，起始于 L3 存储器 mSDRAM，送回到 L1 存储器。它还恢复表 2 所列出的其余的片上存储器资源。



用于为初始化代码生成内存映像图的连接器描述文件(.ldf)将该代码放在 L1 代码间隔的最后。这样做是为了确保执行代码从 mSDRAM 被 DMA 传输回 L1 中的 0xFFA00000 单元时，该代码不会覆盖掉正在执行任务的代码。

最后，初始化代码跳回存储于 0x3FFC 单元的 RETS 地址，将处理器送回休眠线程(在处理器设为休眠状态前的最后一个正在运行的线程)。这一初始化代码列于附录 A 中。

唤醒时间会根据需要存储的存储器信息的多少而有所不同。表 3 列出了不同的 L1 存储器恢复及它们的相应唤醒时间。

L1 存储器保存/恢复的数量	唤醒时间
8 kBytes of L1 Instruction 8 kBytes of L1 Data Bank B	318 μ s
56 kBytes of L1 Instruction 16 kBytes of L1 Data Bank A 24 kBytes of L1 Data Bank B	1.01 ms
32 kBytes of L1 Instruction 32 kBytes of L1 Data Bank A 32 kBytes of L1 Data Bank B	1.81 ms

表3 休眠状态后的唤醒时间

唤醒时间的测量是从电压供给 VROUT 引脚(ADSP-BF533 EZ-KIT Lite 板上的 U32 MOSFET 的引脚 4)的时刻算起，一直到所有的片上存储器资源存储好(就是重新进入休眠线程之前)为止。这一周期包括供给电压(包括被调用的 RTC 的复位)的时间，启动 ROM 来加载/执行初始化代码的时间，以及初始化代码对片上存储器进行恢复所需的时间。

硬件考虑

对于便携式应用，根据休眠状态需要一些常用的硬件模块执行一次全面保存/恢复。注意，并不是所有的 Blackfin 处理器会用到讨论的所有外部电路。随着 Blackfin 处理器家族的日趋成熟，一些先进的技术已合并了更多尖端的休眠功能。对这些先进特性的进一步了解请参阅 Blackfin 设备硬件参考手册。考虑用于与休眠状态进行有关的主要的子系统。为了明确起见，将从对三个外部补偿调节器设计选项的考察开始。

选项 #1- 利用非 RTC 唤醒的外部变量补偿设计

Blackfin 处理器有一个内置补偿调节器来提供 V_{DDINT} 。在功率敏感度较低的应用中，这个补偿调节器和相关的外部组件将材料降到最少。然而，如在 PMP 应用中所看到的，效率是很关键的。Blackfin 调节器的效率可达 75%左右，但不支持 $V_{DDEXT}=1.8V$ 的情况。因此，对于要求 $V_{DDEXT}=1.8V$ 并且调节器设计效率高达 90%的低功耗系统，必须用到外部调节器。Blackfin 处理器内部稳压器的结构是基于一个简单的电压控制器。它不具有限流功能，同步整流，及软启动性能。一个外部补偿设计可

以用于结合 Blackfin 处理器的内部电压控制器以及：

- 提高动态性能
- 提高效率
- 提高稳定性
- 减小所需PCB板面积
- 减少所需组件的额定
- 允许通过软件对电压进行可变控制



这一设计允许软件对 VR_CTL 寄存器进行写操作，从而利用休眠状态并在 100mV 范围内对 V_{DDINT} 进行调整。然而，内部电压调节控制器在 $V_{DDEXT} < 2.5V$ 时不起作用，且 RTC 也不能用于将处理器内核从休眠状态唤醒。

该话题已经在先前的报纸上出现过，在文相关的.zip 文件中有原文：*Using an External Switching Regulators to Supply an Adjustable Internal Voltage to Blackfin Processors*^[3]。这个设计利用了 Blackfin 处理器的片上调节控制器的 VROUT 输出，因此处理器核电压是软件可调的，调节幅度为 100mV。所需外部电路见图 3。

选项 #2—利用 RTC 唤醒的外部固定补偿设计

如果 Blackfin 的 RTC 需要将内核从休眠状态唤醒，外部补偿设计可能是较好的选择。通常情况下，便携式系统必须在预定的时间间隔被快速(≤ 10 毫秒)唤醒，来执行特定的任务，然后必须返回到低功耗的待机状态。如果 RTC 用来定时唤醒处理器内核，必须有一个信号提示外部补偿调节器恢复处理器内核的供电。一个可以用来完成这项任务的信号就是所有 Blackfin 处理器普遍具有的 VROUT。通常情况下，这一信号由内部电压控制器生成且用于与其他无源元件共同来转换外部 FET 并驱动 V_{DDINT} 。然而，在这种情况下，VROUT 的转换用来提示外部调节器重新向处理器内核供电。

如图 4 所示的廉价的电路可以完成这项任务。这一课题在 *External Circuitry for Hibernate Wakeup Function on Blackfin Processor with a Fixed Voltage External Voltage Regulator*^[4]中有详细介绍，包含在相关的.zip 文件中。



这一设计允许 RTC 唤醒操作使处理器从休眠状态中跳出，甚至在 $V_{DDEXT} < 2.5V$ 时继续工作。不过，内部稳压控制器是不可编程的，意味着 V_{DDEXT} 是固定的。

图 4 的电路中，Blackfin 的 GPIO 引脚(pf6)用来启动休眠过程。因此，软件必须在进入休眠状态时对 PF6 引脚写入一个 1，并在启动休眠状态的同时写一个 0 值。

选项#3—基于 RTC 唤醒的外部变量补偿设计

可以将选项#1 和选项#2 的特性结合起来得到一个理想的解决方案。图 5 说明是根据图 4 的逻辑允许 RTC 唤醒的 ADP2105 补偿调节器。这一方案允许用 2 个 GPIO 对 4 个核电压进行设置。否则，图 5 中的 Core_GPIO_1 就会连接到一个 PWM 模式下的 Blackfin 定时输出提供无限的电压调节级数。



这一设计允许 RTC 唤醒在休眠状态后唤醒外部补偿调节器，并在 V_{DDEXT} 为 3.3 V, 2.5 V, 或 1.8 V 时正常工作。它还提供软件对可变核电压在 100-mV 范围内的调整。然而，内部电压调节控制器只能通过两个 GPIO 引脚对四个离散数值进行编程设计。

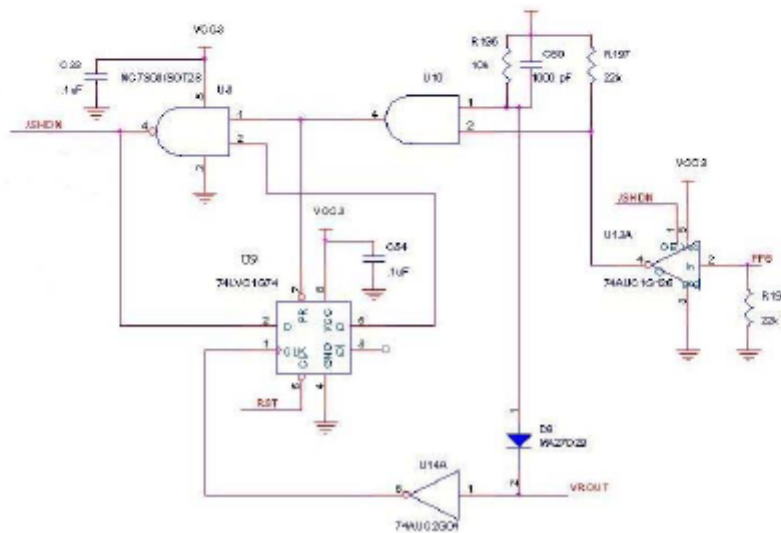
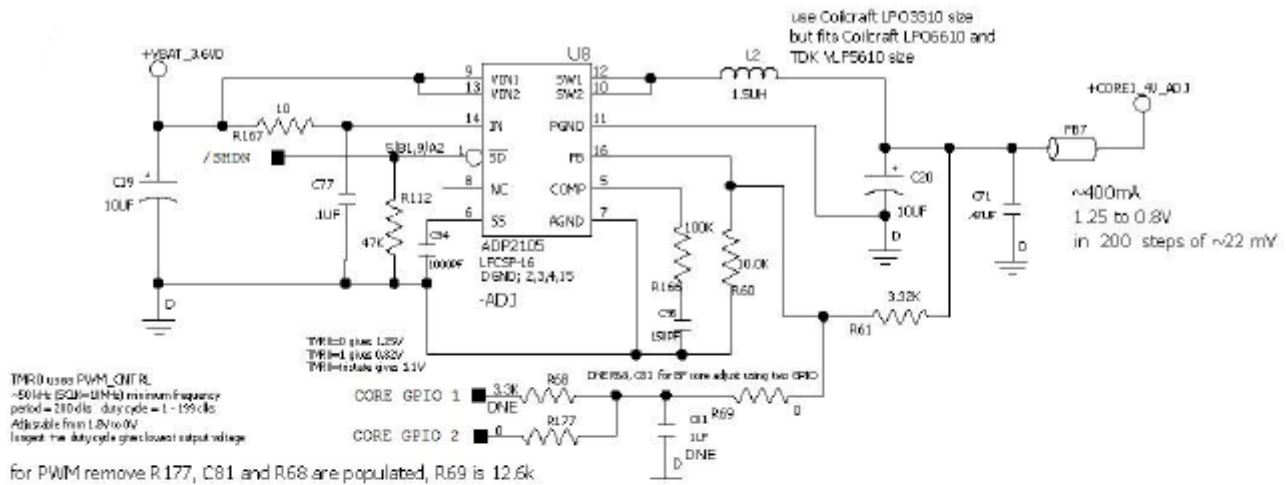


图 5 利用 RTC 唤醒的外部可变补偿设计

处理器内核复位加电

一旦处理器内核功率稳定，就必须提供一个处理器内核复位，它可以使处理器启动。这个需求可以用于上述三个处理器内核补偿设计。



仅对该处理器内核，工作复位(POR)是休眠状态必要的，也是系统 POR 的补充。

绝大多数的工作复位芯片要求 $\geq 1.8V$ 的电源供给。在这种情况下，一旦 Blackfin 处理器的 V_{DDINT} 达到稳定的 1.1V 左右，就必须应用一个清空处理器内核的复位操作(/RESET 必须确认为低，最小值为 275 纳秒，根据 *ADSP-BF531/ADSP-BF532/ADSP-BF533 Blackfin 嵌入式处理器数据手册* ^[4] 而定)。因此，需要一个特别的 POR，它要允许一个外部可调参考电压和一个可调复位定时器(通常精确到 1 毫秒)。Maxim MAX6896 POR 芯片(见图 6)可以提供这个功能。AD 公司的 ADP2105 补偿调节器有一个软启动特性，它允许产品产生一个最低为 750 微秒的稳定的输出电压。这样，从处理器内核加电到实际启动代码的最短时间为：

$$750 \text{ 微秒(降压稳定时间)} + 1 \text{ 毫秒(最小复位脉冲宽度)} = 1.75 \text{ 毫秒}$$

显然，如前所述，根据要保存和恢复信息的多少，还将额外耗费 2 毫秒。因此，从最初的电源供给到完整的信息恢复和代码执行之间存在(总共)4 毫秒的启动延迟。

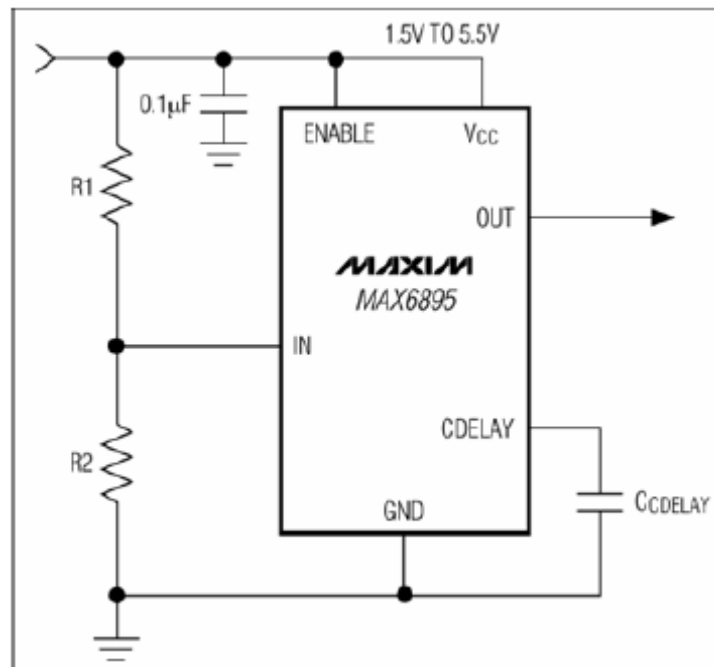


图6 外部电压判定的POR 和可调复位脉冲长度

另外一个要考虑的是 PMP 通常希望从一个用户界面(UI)命令或按钮的按下操作来唤醒内核(如恢复供电)。这常常以一个 UI 中断(例如一个来自 Wi-Fi 模块的中断)的形式出现。如果这个 UI 中断要执行，它就必须嵌入到图 5 中的 U8 或 U10。

在软件控制下进入休眠

怎样进入休眠状态呢？所有 PMP 系统必须让这一特性受内核控制(例如软件控制)。由于调节器是外

部的，就需要有一个机构来关闭内核电源。在利用休眠状态的便携式应用中，设计者往往需要在内核电源被切断时保持外部组件(mSDRAM，闪存，等等)仍然有电。



内核对于外部补偿调节器关闭的要求要能够越过任何由图 4 或图 5 中的 RTC 电路产生的其他信号。

根据 Blackfin 处理器的启动方式，通用的标志输出(GPIO)状态在内核电源被移除后可能是不确定的。内核电源的切换(开和关)过程中，这些 GPIO 的状态不能改变。因此，在图 7 所示的 PMP 参考设计中，要附加外部供电的锁存器(U35 – 74LVC374A)和 D 触发器(U16 – 74AUC1G74)。这两个低成本组件由 1.8-V 的外部电压源供电并在全部休眠过程中保持通电。在热启动或冷启动后，配置/设置阶段必须简单的写这些锁存器。然后，要进入休眠状态，我们要在适当的锁存位(SD5)做一次 0-1 的转换，这样就可以关闭外部调节器。这样做还可以得到提高处理器上可用 GPIO 数量的附带好处。注意在 PMP 第一次打开时，系统的供电开关连接到 D 触发器上的/SET 引脚上使处理器内核补偿调节器被迫处于“开”状态。

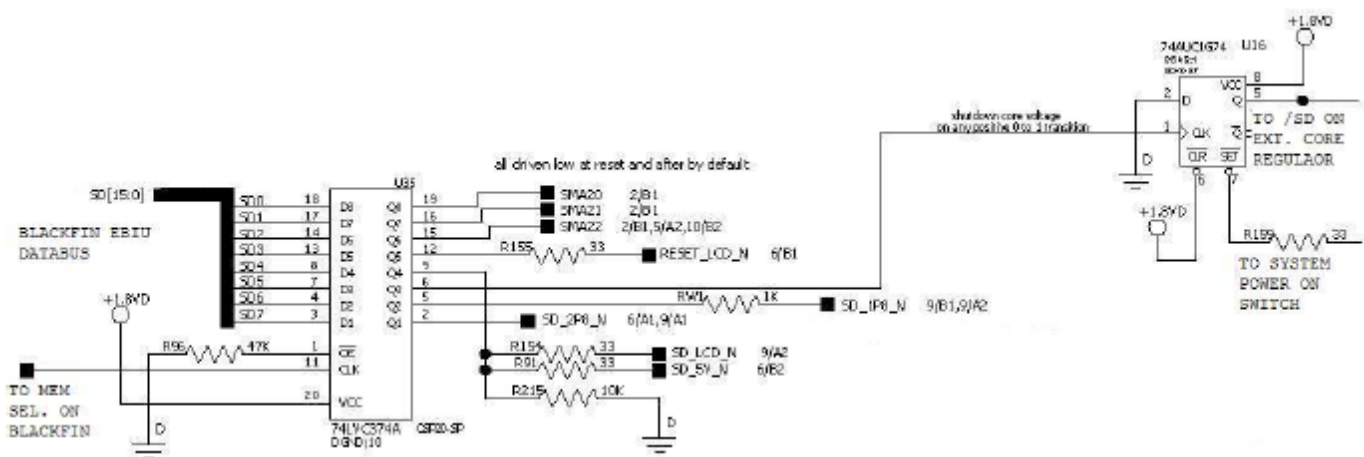


图 7 带有开/关外部补偿调节器的 D 触发器的外部锁存器

三态 SDRAM 时钟使能(SDCKE)

当处理器内核进入休眠状态时，在紧随休眠状态的复位序列执行的同时，mSDRAM 的控制信号将被关闭。如果观察 SDRAM 是怎样工作的，会发现在 SDRAM 处于自动刷新模式时，时钟使能信号被主机驱动至高电平，就会使该部分跳出自动刷新模式。早期的 Blackfin 处理器，像 ADSP-BF531/BF532/BF533，是不提供启动过程(该过程连接到 mSDRAM 设备的 SDCKE 引脚)中保持 SCKE 信号状态灵活性的。其不利的副作用就是当 Blackfin 处理器执行它的启动过程时 mSDRAM 会过早跳出自动刷新模式从而失去其中的信息。因此，PMP 参考设计增加了一个三态缓冲(图 8 中的 U36-74LVC1G125)，这里输出使能(/OE)接入到外部锁存器(74LVC374A-U35)中。在使 mSDRAM 先于休眠状态进入自动刷新后，使能三态 SDCKE。接着，当处理器内核功率恢复时，SDCKE 保持原状，直到处理器内核进行一次热重启 (就是说这部分保持自动刷新状态)。作为安装和配置阶段的一部分，要重新使能 SDCKE。

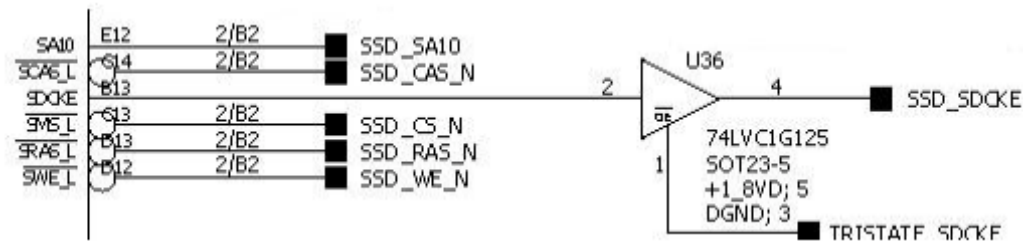


图8 SDCKE 三态缓冲

SDRAM 掉电后长时间保存信息

PMP 参考设计中提到，根据印刷电路板上电容的数量，mSDRAM 可以在系统电源被切断后很长一段时间(大约几分钟)保存它里面的信息。可能会想起在 mSDRAM 中存储了一个关键字(值为 0x12345678)。如果这个关键字有效，就会产生一次热启动(就是从 mSDRAM 中恢复信息)。但实际上存在着产生启动被破坏的风险，这决定于系统内电容上电荷的消耗。因此，为了在休眠状态下完整保存信息从而正常工作，系统必须了解何时发生了相对于热启动的冷启动。也许在硬件上完成这一任务的最简单方法就是将系统复位引脚(/SET)接到外部 D 触发器(74AUC1G74)的预设引脚上。Blackfin 处理器可以在启动时询问这个锁存器以决定处理器是以冷启动(闪存启动)还是热启动(休眠启动)方式启动。很明显，在热启动时不会进行系统的全面复位，仅对处理器内核进行复位。在启动过程的初始化阶段，Blackfin 处理器能够监控这个锁存器。如果它为 1，处理器将从闪存中完成一次全面冷启动。在信息保存阶段，Blackfin 处理器可以向这个锁存器写一个 0。然后，在热启动过程中，处理器对该位进行检验，如果结果为 0，就完成一次休眠信息恢复。



D 触发器必须由 I/O 端口(1.8 V)供电，从而不会与核电压复位电路发生冲突。

结论

文中所讨论的所有休眠操作对于要求完整信息保存及恢复性能的系统都是有必要的。对能够接受完成一次冷启动的时间系统来说，信息的保存与恢复操作就没有必要了(与所讨论的许多硬件一样)。然而，在需要这一特性的地方(比如一个必须定时并迅速唤醒的 PMP)，这个应用就至关重要。这并不是说仅关闭内核电源就没有用处。它在一些应用中确实是有用的。但是，这里的重点是要提供一个具有以下 3 个主要优点的 Blackfin 解决方案：

- 低有效功率(~0.16 mW/MIP)
- 低待机功耗(≤ 300 μW)
- 处理器待机状态的快速唤醒(4毫秒或更短时间内)

附录 A-初始化模块源代码

```

#include "defbf533.h"
#define OFFSET_( x ) ((x) & 0x0000FFFF)
#define UPPER_( x ) (((x) >> 16) & 0x0000FFFF)
#define LOWER_( x ) ((x) & 0x0000FFFF)
#define SYSMMR_BASE 0xFFC00000
#define COREMMR_BASE 0xFFE00000

.section program;
Start_of_Code:
    [--SP] = ASTAT; [--SP] = RETS; [--SP] = (r7:0); [--SP] = (p5:0);

    P1.H = UPPER_(SYSMMR_BASE);    // P1 Points to top of SYSTEM MMR Space
    P1.L = LOWER_(SYSMMR_BASE);
    R0 = 0x10(z);
    W[P1+OFFSET_(SPI_BAUD)] = R0;  // Speed up SPI rate.

CONFIG_SDRAM:
    R0 = 0x0817(Z);
    W[P1+OFFSET_(EBIU_SDRRC)] = R0;    // SDRAM Refresh Rate Control Register
    R0 = 0x0013(Z);
    W[P1+OFFSET_(EBIU_SDBCTL)] = R0;   // SDRAM Memory Bank Control Register
    R0.H = 0x0091; R0.L = 0x998D;
    [P1+OFFSET_(EBIU_SDGCTL)] = R0;   // SDRAM Memory Global Control Register
    SSYNC;

PLL_change:
    P0.L = lo(SIC_IWR); P0.H = hi(SIC_IWR);
    R1 = [P0];           // Save current SIC_IWR
    R0 = 0x1;
    [P0] = R0;          // Enable only PLL relock to break idle

    P1.H = hi(PLL_CTL); P1.L = lo(PLL_CTL);
    R0.L = 0x2C00;      // CLKIN = 27MHz for the ADSP-BF533 EZ-KIT Lite board
    w[P1] = R0;        // MSEL = 010110 = 22X = 22 X 27 = 594 MHz
    idle;              // Required for PLL relock
    [P0] = R1;        // Restore previous SIC_IWR

    P1.L = lo(PLL_DIV); P1.H = hi(PLL_DIV);
    R0.L = 0x5;        // Default for ADSP-BF533 is 0x0005 -> CCLK/5 = 594/5 = 118.8
    w[P1] = R0;

CHECK_KEY:
    P2.H = 0x0000; P2.L = 0x3FEC;    // Key is located at address 0x3FEC
    R0.H = 0x1234; R0.L = 0x5678;    // Expected key value is 0x12345678
    R1 = [P2++];
    CC = R0 == R1;
    IF CC JUMP RESTORE;             // If key matches, don't do a full boot

FULL_BOOT:
    (p5:0) = [SP++]; (r7:0) = [SP++]; RETS = [SP++]; ASTAT = [SP++];
    RTS;                            //If key doesn't match, return back to bootrom for full boot

RESTORE:
    P1.H = UPPER_(SYSMMR_BASE);    // P1 Points to top of SYSTEM MMR Space
    P1.L = LOWER_(SYSMMR_BASE);

```

```

    SP = [P2++]; FP = [P2++];          // Restore Stack and Frame Pointers

// Restore L1 Memory via DMA
R0 = [P2++];          // P2 will point to the Return Address after this update
[P1+OFFSET_(MDMA_S0_CURR_DESC_PTR)] = R0;
R0 += 0x20;
[P1+OFFSET_(MDMA_D0_CURR_DESC_PTR)] = R0;
R0 = (FLOW-0x3000) | (NDSIZE-0x400) | RESTART | WDSIZE_32 | DMAEN(z);
W[P1+OFFSET_(MDMA_S0_CONFIG)] = R0;
R0 = (FLOW-0x3000) | (NDSIZE-0x400) | RESTART | WDSIZE_32 | WNR | DMAEN(z);
W[P1+OFFSET_(MDMA_D0_CONFIG)] = R0;

// Restore Scratchpad Memory via Core reads/writes
P3.H = 0x0001; P3.L = 0xC000;          //P3 = SDRAM Memory
P4.H = 0xFFB0; P4.L = 0x0000;          //P4 = Scratchpad Memory
P5.H = 0x0000; P5.L = 0x0400;          //Restore 4096 bytes
LSETUP(L3_TO_SCRATCH_begin, L3_TO_SCRATCH_end) LC0 = P5;
    L3_TO_SCRATCH_begin:      R0 = [P3++];
    L3_TO_SCRATCH_end:        [P4++] = R0;

// Restore EVT Table via Core reads/writes
P4.H = 0xFFE0; P4.L = 0x2000;          //P4 = Event Vector Table
P5.H = 0x0000; P5.L = 0x0010;          //Restore 16 entries
LSETUP(L3_TO_EVT_begin, L3_TO_EVT_end) LC0 = P5;
    L3_TO_EVT_begin:         R0 = [P3++];
    L3_TO_EVT_end:           [P4++] = R0;

P4.H = 0xFFE0; P4.L = 0x2104;          //Restore IMASK
R0 = [P3++];
[P4] = R0;

TST_RUN_BIT:                          // Wait until L1 Memory DMA is complete
R0 = W[P1+OFFSET_(MDMA_D0_IRQ_STATUS)](z);
CC = BITTST(R0,3);
IF CC JUMP TST_RUN_BIT;                //Wait until DMA is complete

/**** For profiling purposes ****/
R0 = 0x0080;
W[P1+OFFSET_(FIO_DIR)] = R0;           //Enable PF7 as output
W[P1+OFFSET_(FIO_FLAG_S)] = R0;       //Set PF7 high
/*****/
P5 = [P2];
JUMP(P5);

```

参考文献:

- [1] *MT48H4M16LF – 1 Meg x 16 x 4 banks Mobile SDRAM Data Sheet*. Rev C, October 2007. Micron Technology, Inc.
- [2] *ADSP-BF533 Blackfin Booting Process (EE- 240)*. Rev 3, January 11, 2005. Analog Devices, Inc.
- [3] *Using External Switching Regulators to Supply an Adjustable Internal Voltage to Blackfin Processors*. Rev 3, January 30, 2007. Analog Devices, Inc. (included in .ZIP file)
- [4] *External Circuitry for Hibernate Wakeup Function on Blackfin Processor with a Fixed Voltage External Voltage Regulator*. March 2007, Analog Devices, Inc. (included in .ZIP file)
- [5] *ADSP-BF531/ADSP-BF532/ADSP-BF533 Blackfin Embedded Processor Data Sheet*. Rev E, July 2007. Analog Devices, Inc..
- [6] *ADSP-BF533 Blackfin Processor Hardware Reference*. Rev 3.2, July 2006, Analog Devices, Inc.

文档记录

Revision	Description
<i>Rev 1 – May 5, 2008 by J. Sondermeyer, Joe Beauchemin, and Hiren Desai</i>	Initial Release