

# 初始化 Hercules™ ARM® Cortex™-R4F 微控制器

Sunil Oak

## 摘要

这份应用报告提供了 Hercules 系列中 TMS570LS31x 和 RM4x 系列微控制器产品的简要概述和初始化过程。本文中用到的“Hercules 微控制器 (MCU)”是指这些系列微控制器中的任一部件。

此文档还显示了源文件的代码段，这些源文件由 HALCoGen 工具生成。这个文档中用到的所有代码结构在也由同一工具生成的头文件中定义。

HALCode生成器工具可从以下URL 下载：<http://www.ti.com/tool/halcogen>。

## 内容

1	方框图 .....	2
2	Hercules 微控制器的标准初始化序列 .....	3
3	参考书目 .....	14

## 图片列表

1	器件方框图 .....	2
2	针对方框图的颜色图例 .....	2
3	FMPLL 方框图 .....	6
4	VIM 中断地址内存映射 .....	13

## 图表列表

1	Hercules 微控制器上的时钟源 .....	7
2	Hercules 微控制器上的时钟域 .....	10

Hercules is a trademark of Texas Instruments.  
 Cortex is a trademark of ARM Limited.  
 ARM is a registered trademark of ARM Limited.  
 All other trademarks are the property of their respective owners.

# 1 方框图

1 节显示了一个扩展集 TMS570LS31x 微控制器的高级方框图。要获得与 TMS570LS 系列或 RM4x 系列微控制器相关的任一派生器件的实际方框图，请参见器件专用数据表。

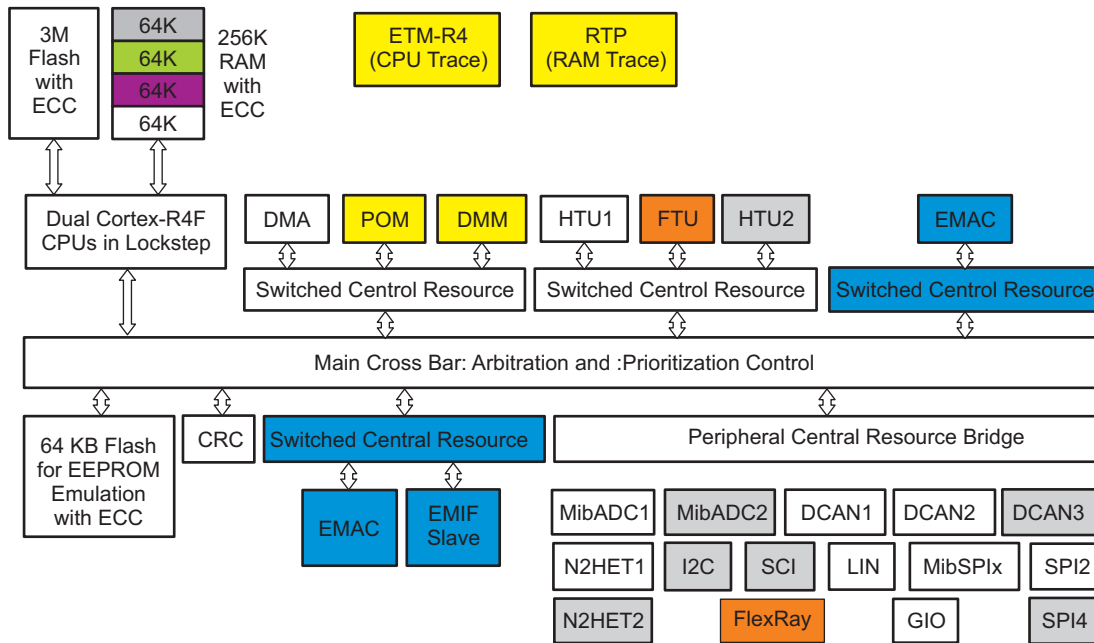


图 1. 器件方框图

此方框图包括一个在微控制器上执行的单独内核功率域的颜色编码表示（请见图 2）。在初始化期间，按照应用的要求，这些功率域可被单独打开 (ON) 或关闭 (OFF)。

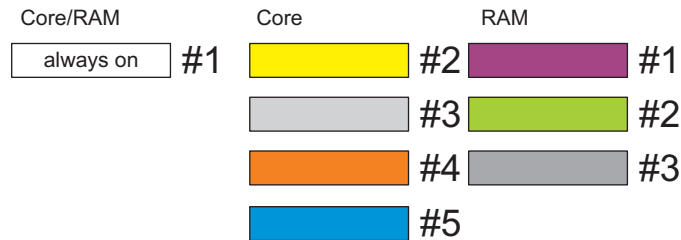


图 2. 针对方框图的颜色图例

## 2 Hercules 微控制器的标准初始化序列

在下面总结了 Hercules MCU 上的关键特性初始化和配置的基本顺序，在下面的部分中对很多步骤进行了详细说明。与这份应用报告一同提供的源代码示例演示了很多建议的步骤。初始化序列的某些部分不是强制的。例如，非安全应用可以选择不使用针对闪存和 RAM 存取的纠错码 (ECC) 特性。每个应用还必须有其特定的意外处理机制：复位处理程序，中止处理程序等。使用 HALCoGen 生成的代码包括针对每个例外的标准处理例程。这些例程需要按照应用的要求进行修改。

1. 启用 Cortex-R4F CPU 内的浮点单元 (FPU) (请见 2.1 节)。
2. 初始化 CPU 寄存器和 FPU 寄存器，其中包括堆栈指针 (2.2 节)。
3. 启用闪存接口模块响应一个由 CPU 访问闪存时标示的 ECC 错误 (2.3 节)。
4. 启用 CPU 的事件总线报告机制 (2.4 节)。
5. 针对闪存存储器存取 (CPU 的 ATCM 接口)，启用 CPU 的单纠错双纠错 (SECEDED) 逻辑 2.5 节
6. 处理复位的原因以确定是否继续启动序列 (2.6 节)
7. 检查加电期间是否出现任何错误信令模块 (ESM) 组 3 错误。如果在加电期间出现任何 ESM 组 3 错误，那么继续代码执行是不安全的，并且微控制器初始化过程可在这一点上被停止。序列中随后的步骤假定在加电期间没有 ESM 组 3 错误。
8. 用针对分压器 (电阻分压器) 最后级的最大值配置 PLL 控制寄存器 (2.7 节)。
9. 启用锁相环 (PLL)(2.8 节)。
10. 运行 eFuse 控制器启动检查，并在 eFuse 控制器 SECEDED 逻辑上启动自检 (2.9 节)。
11. 将外设从复位中释放，并启用到所有外设的时钟 (2.10 节)。
12. 设置器件级复用选项，以及输入/输出 (I/O) 复用。
13. 等待 eFuse 控制器 ECC 逻辑自检完成并校验其结果。
14. 针对所需的等待状态和管道模式来设置闪存模块 (2.11 节)。
15. 设置闪存组和供电模式 (2.12 节)。
16. 调谐 LPO (2.13 节)。
17. 运行嵌入在闪存模块内部的 SECEDED 逻辑电路的自检 (2.14 节)。
18. 等待主 PLL 输出变成有效。
19. 将器件时钟域映射到所需的时钟源 (2.15 节)。
20. 逐步减少电阻分压器的值以获得针对 PLL1 和 PLL2 的目标 PLL 输出频率。
21. 在 CPU 自检控制器上运行诊断校验 (2.16 节)。一个 CPU 复位在 CPU 自检完成时被置为有效。因此，导致复位处理程序的初始化步骤将被重复。
22. 运行 CPU 内置自检 (LBIST)(2.17 节)。在 CPU 自检完成时，一个 CPU 复位被置为有效。因此，导致复位处理程序的初始化步骤将被重复。
23. 运行 CPU 比较模块 (CCM-R4F) 诊断校验 (2.18 节)。
24. 运行存储器自检控制器诊断校验 (2.19 节)。
25. 使用可编程内置自检 (PBIST) 控制器来启动 CPU RAM 自检，并且等待此自检完成并通过 (2.20 节)。
26. 使用系统模块硬件初始化机制来初始化 CPU RAM，这样的话，用于 CPU RAM 的 ECC 区域也被初始化 (2.21 节)。
27. 启用针对 CPU RAM 存储器存取的 CPU 单纠错双纠错 (SECEDED) 逻辑 (CPU 的 B0TCM 和 B1TCM 接口) (2.22 节)。
28. 使用 PBIST 控制器在所有片载双端口 SRAM 上启动自检 (2.23 节)。
29. 针对主数据 RAM 存取运行 CPU SECEDED 逻辑电路自检 (B0TCM 和 B1TCM) (2.24 节)。
30. 针对主闪存存储器存取运行 CPU SECEDED 逻辑自检 (ATCM) (2.25 节)。
31. 等待所有片载双端口 SRAM 上的自检完成并通过。
32. 使用 PBIST 控制器启动除 CPU RAM 之外的所有片载单端口 SRAM 自检 (2.26 节)。
33. 等待所有片载单端口 SRAM 上的自检完成并通过。
34. 启动针对所有其它片载 SRAM 的自动初始化 (2.27 节)。
35. 检查是否所有 RAM 的初始化过程已完成；如果未完成的话，请等待。

36. 检查针对所有外设存储器的奇偶校验错误检测机制 (2.28 节)。
37. 启用 CPU 的专用矢量化中断控制器 (VIC) 端口 (2.29 节)。
38. 在矢量中断管理器 (VIM) 存储器中编辑所有中断处理例程地址 (2.30 节)。
39. 配置所有中断通道的 IRQ/FIQ 中断优先级 (节 2.30.1)。
40. 在 CPU 内部启用所需的中断 (IRQ 和/或 FIQ) (2.31 节)。
41. 在 VIM 控制寄存器内部启用所需中断 (节 2.30.2)。
42. 设置到错误信令模块 (ESM) 输入的应用响应 (2.32 节)。
43. 初始化复制表格、全局变量和构造函数 (2.33 节)。
44. 验证双时钟比较器 (DCC) 模块可真正地检测并标示一个频率错误。
45. 将 DCC 模块配置成持续监控 PLL 输出。
46. 验证对于所有总线主控的内存保护单元 (MPU) 保护违反被标记为一个错误, 发送给 ESM。
47. 使用循环冗余码校验 (CRC) 和直接存储器存取 (DMA) 运行整个闪存背景检查。
48. 运行针对模数转换器 (ADC) 的偏移误差校准例程。
49. 运行模数转换器 (ADC) 模拟输入通道自检。
50. 检查所有外设的 I/O 回路。
51. 为总线主控设置 MPU。
52. 设置数字窗口安全装置 (DWWD) 模块处理窗口尺寸以及保护违反发生时的模块响应 (复位或不可屏蔽中断 (NMI))。
53. 配置 N2HET1 至 N2HET2 监控功能性。
54. 使用外设中央资源 (PCR) 控制器寄存器为外设配置所需的访问许可。
55. 配置外部安全伴随器件, 例如 TI TPS6538x, 用于在线诊断操作。
56. 设置实时中断 (RTI) 模块用来生成应用所需的周期中断。
57. 调用主应用程序 (2.35 节)。

## 2.1 启用浮点协处理器 (FRU)

在 CPU 复位时浮点协处理器被禁用，并且必须在应用需要浮点运算时被启用。如果在 FPU 禁用时执行一个浮点指令，将会生成一个未定义的指令异常情况。

## 2.2 初始化 Cortex-R4F 寄存器

Hercules 系列微控制器包括运行在锁步运转模式下的双 Cortex-R4F CPU。一个内核比较模块 (CCM-R4) 比较每个 R4F CPU 的输出信号。两个 CPU 输出间的任何差异都会被标记为一个高严重等级故障。对于两个 CPU，并不保证 CPU 内部寄存器在同样状态下加电。CPU 使用一个函数调用将内部寄存器推入堆栈，这样会检测到一个内核比较错误。因此，CPU 内部内核寄存器需要在进行任何函数调用前被初始化至一个预先定义的状态。

CPU 的调用-返回堆栈包含一个 4 条目循环缓冲器。当 CPU 预取单元 (PFU) 检测到一个采用的过程调用指令时，PFU 将返回地址压入调用-返回堆栈。在 ARM 和 Thumb 指令集中，被 PFU 识别为过程调用的指令为：

- BL immediate
- BLX immediate
- BLX Rm

当返回堆栈检测到一个取返回指令时，PFU 从返回堆栈的顶部位置发出一个取指令，并且弹出返回堆栈。在 ARM 和 Thumb 指令集中，被 PFU 识别为过程返回的指令为：

- LDMIA Rn{!}, {...,pc}
- POP {...,pc}
- LDMIB Rn{!}, {...,pc}
- LDMDA Rn{!}, {...,pc}
- LDMDB Rn{!}, {...,pc}
- LDR pc, [sp], #4
- BX Rm

## 2.3 在闪存接口模块中启用对 ECC 错误的响应

闪存模块具有一个闪存错误检测和地址 0xFFFF87008 上的纠错控制寄存器 1 (FEDACCTRL1)。这个寄存器控制在闪存模块内执行的 ECC 功能，其中包括对 Cortex-R4F CPU 内 SECDDED 逻辑的支持。这个寄存器的位 3-0 组成了 EDACEN 字段。EDACEN 缺省配置为 0x5。此应用必须将 EDACEN 配置为 0xA，以便启用闪存模块对 CPU 的 SECDDED 逻辑的支持。

## 2.4 启用 Cortex-R4F CPU 的事件信号发送机制

Cortex-R4F CPU 有一个专用的事件总线，此总线被用来表示已经发生的事件。这个事件信号发送在复位时被禁用，并且必须被启用。闪存模块和 RAM 模块接口捕捉 CPU 发出的 ECC 错误事件。这使得应用能够进一步调试导致 ECC 错误的准确地址。

通过将性能监控单元的“性能监控控制寄存器，c9”的“X”位置位可启用 CPU 事件信号发送。

## 2.5 启用针对 ATCM 接口的 Cortex-R4F CPU ECC 校验

CPU 具有内部 ECC 逻辑，此逻辑保护所有 CPU 到 ATCM（闪存）接口的访问。缺省情况下不使用这个逻辑，而必须通过将系统控制协处理器的辅助控制寄存器，c1 的 ATCMPCEN 位置位来启用。

## 2.6 处理复位原因

针对不同的复位情况，每个应用具有不一样的耐受水平。

## 2.7 配置 PLL

Hercules 包含一个调频锁相环 (FMPLL) 宏，这个宏可使输入振荡器频率被倍乘至一个更高的频率，这个频率高于外部谐振器或晶振可方便实现的频率。此外，FMPLL 可实现从固定晶振或谐振器中生成很多不同频率选项的灵活性。

FMPLL 使得应用能够在 FMPLL 的已选基频信号输出上附加一个“调制频率”信号。这通过将其展开在基频附近的受控频率范围内来减少输出信号的电磁能量。缺省情况下这个模式被禁用，可在噪声敏感应用中启用这个模式。

Hercules 微控制器还包含一个第二非调制 PLL 宏。这个 PLL#2 可被单独配置成生成一个用特定用途的第二高频时钟源，诸如 80MHz 的 FlexRay 通信时钟源等。

### 2.7.1 FMPLL 方框图

图 3 显示了一个 FMPLL 宏的高级方框图。

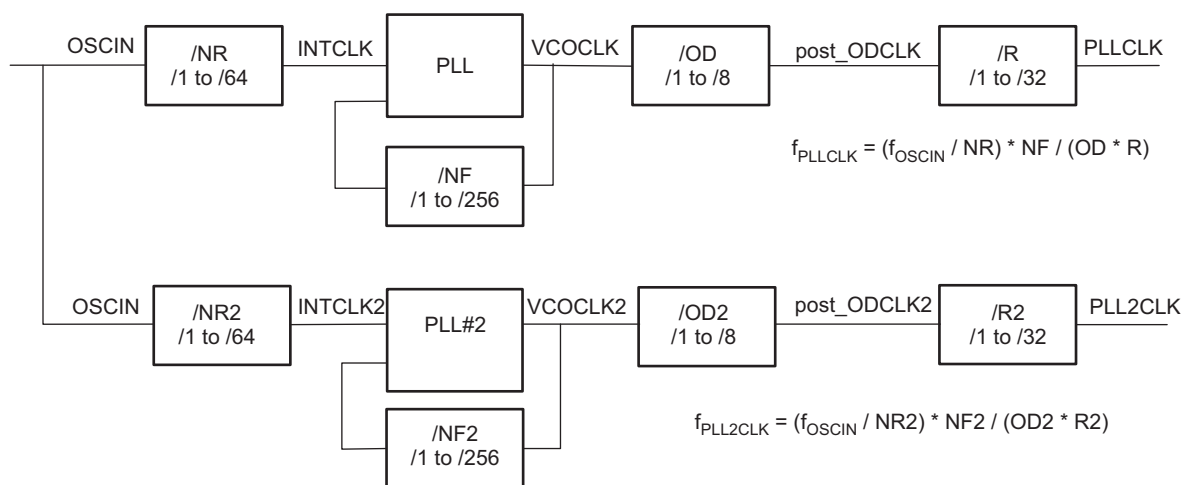


图 3. FMPLL 方框图

参数  $f_{\text{OSCIN}}$ 、 $f_{\text{post\_ODCLK}}$  和  $f_{\text{HCLK}}$  是数据表技术规格。要确定这些频率的最小/最大限值，请参见器件专用数据表。

注：FMPLL 花费  $(127 + 1024 \cdot \text{NR})$  个振荡器周期来锁定至目标频率，因此建议在器件初始化时尽可能的配置 FMPLL 并将其启用。

### 2.7.2 FMPLL 配置

使用两个位于 Hercules 微控制器上系统模块内的控制寄存器，PLL 控制 1 寄存器 (PLLCTL1) 和 PLL 控制 2 寄存器 (PLLCTL2)，来配置 PLL1。

使用系统模块内的一个单个 PLL 控制 3 寄存器 (PLLCTL3) 来配置 PLL2。

## 2.8 启用时钟源

### 2.8.1 Hercules 微控制器上的可用时钟源

表 1 中列出了 Hercules 微控制器支持 7 个不同的时钟源。

表 1. Hercules 微控制器上的时钟源

时钟源编号	时钟源名称	说明
0	OSCIN	这个是主振荡器，通常由一个外部谐振器或晶振驱动。这个是到 FMPLL 和 FMPLL2 宏的唯一可用输入。OSCIN 频率必须介于 5MHz 和 20MHz 之间。
1	FMPLL#1 输出	这个是 FMPLL 的输出，此输出由 OSCIN 生成为输入时钟。FMPLL 输出时钟频率一定不能超过器件专用数据表中规定的最大器件频率。FMPLL 特有一个调制模式，在这种模式中，调制频率被附加到 FMPLL 输出信号上。
2	未执行	源 #2 上未连接时钟信号。这个时钟源一定不能被启用或从任一时钟域中选择。
3	EXTCLKIN1	外部时钟输入 #1。这个时钟源只有在真正的外部时钟源被连接至已确定的 EXTCLKIN1 器件端子上时才被启用。更多信息请参见器件专用数据表。
4	低频 (LF) LPO	这是内部基准振荡器的低频输出。LF LPO 通常是一个 80KHz 的信号，并且通常用于低功耗模式使用情况。
5	高频 (HF) LPO	这是内部基准振荡器的高频输出。HF LPO 通常是一个 10MHz 的信号，并且用作监控主振荡器的基准时钟。
6	FMPLL#2 输出	这是第二 FMPLL 的输出，此输出由 OSCIN 生成为输入时钟。FMPLL 输出时钟频率一定不能超过器件专用数据表中规定的最大器件频率。
7	EXTCLKIN2	外部时钟输入 #2。这个时钟源只有在真正的外部时钟源被连接至已确定的 EXTCLKIN2 器件端子上时才被启用。更多信息请参见器件专用数据表。

### 2.8.2 用于启用和禁用时钟源的控制寄存器

Hercules 微控制器上有 7 个可用的时钟源：

- 时钟源 0, 4 和 5 被启用，而时钟源 1, 3, 6 和 7 在任何系统复位时被禁用。
- 时钟源 2 未被执行，并且一定不能在应用中启用它。
- 系统模块时钟源启用寄存器 (CSDIS) 的每个位控制相同编号的时钟源：位 0 控制时钟源 0，位 1 控制时钟源 1，以此类推。
- 还有专用时钟源禁用设定 (CSDISSET) 和时钟源禁用清零 (CSDISCLR) 寄存器来使应用避免使用读取-修改-写入操作。
- 在将任一位置位时，相应的时钟源被禁用。
  - 一旦时钟域消失或者使用此时钟源的第二时钟源 (FMPLL, FMPLL#2) 将被禁用，此时钟源只能被禁用。

### 2.8.3 示例时钟源配置

```

systemREG1->CSDISCLR = 0x0000000U
                        | 0x00000001U // Enable clock source 0
                        | 0x00000002U // Enable clock source 1
                        | 0x00000010U // Enable clock source 4
                        | 0x00000020U // Enable clock source 5
                        | 0x00000040U; // Enable clock source 6
    
```

上面的配置启用时钟源 0, 1, 4, 5 和 6。

在被启用的时钟源里，编号 0, 4 和 5 在缺省情况下被启用，并且在处理器从加电复位中释放时变成有效。这些是主振荡器以及内部基准振荡器的两个输出。

时钟源 1 和 6 是两个 PLL 输出。FMPLL 以及 FMPLL#2 具有已定义的启动时间，而他们的输出在这个时间之前不可用。在将 PLL 输出用于任一时钟域前，此应用必须等到针对这些时钟源的有效状态标志被设定。示例初始化序列使用这个 PLL 锁定时间来执行所有初始化操作，这些操作不必在为应用选中的最大运行频率上完成。

## 2.9 运行 eFuse 控制器 SECEDED 逻辑自检

电子可编程熔丝 (eFuse) 被用来在加电复位 (nPORRST) 置为无效后配置此部件。eFuse 值被读取并作为加电复位序列的一部分被载入到内部寄存器内。这称为 eFuse 自动载入。eFuse 受到单位纠错、双位错误检测 (SECEDED) 编码的保护。这些熔丝在器件的最初工厂测试被设定。eFuse 控制器被设计成一旦器件被包装，就不能更改 eFuse 的状态。

对于安全系统，对于应用来说，在器件复位后检查 eFuse 控制器的状态十分重要。与 eFuse 控制器错误以及校验这些错误的应用序列相关的更多细节，请参见器件专用技术参考手册的《eFuse 控制器》一章。

## 2.10 释放到外设的复位和时钟

外设被保持在复位状态，需要应用明确地使外设脱离复位状态。这可通过将时钟控制寄存器 (CLKCNTL) 的外设启用 (PENEA) 位置位来完成。

到外设模块的时钟也在系统复位时被禁用，需要由应用明确启用。这可通过将与外设中央资源 (PCR) 控制寄存器占用的外设选择象限相对应的位置位以清除外设模块断电状态（外设断电清零寄存器 [0:3] (PSPWRDWNCLR<sub>x</sub>)）来完成。要获得与每个外设的外设选择象限相关的更多信息，请参见器件专用数据表。

## 2.11 配置闪存存取

Hercules 系列微控制器上的闪存存储器是非易失性电可擦除和可编程存储器。

Hercules 微控制器包含一个数字模块，此模块管理全部到闪存存储器的存取。对于总线主控时钟速度最高为 45MHz 的情况，不需任何等待状态即可完成闪存存取。如果总线时钟频率高于 45MHz，那么根据总线时钟速度的不同，任一闪存存取需要适当数量的等待状态。Hercules 系列微控制器支持最高 180MHz 的时钟速度。对于实际允许的最大速度和相应地址以及数据等待状态的数量，请参见器件专用数据表。

假定应用需要一个 180MHz 的 CPU 时钟速度。对于任何闪存存储器存取，这需要 1 个地址等待状态和 3 个数据等待状态。这些等待状态需要在闪存模块寄存器内配置。

此闪存模块还特有一个管道运行模式。当这个模式被启用时，此模块从闪存存储器中读取 128 位数据，并且将它们保持在缓冲器中，CPU 可在无需任何等待状态的情况下读取这些缓冲器。CPU 能够从管道缓冲器中读取 32 或 64 位指令或数据。

闪存模块内的闪存读取控制寄存器 (FRDCNTL) 控制等待状态和管道模式。

Hercules MCU 还有一个独立的闪存组（组 #7），此闪存组专门用于数据存储。这个组可被用来仿真一个 EEPROM。可通过闪存模块内的单独 EEPROM 仿真配置寄存器 (EEPROM\_CONFIG) 来配置到这个闪存组的存取。到 EEPROM\_CONFIG 寄存器的写入操作必须首先通过配置闪存状态机写入使能控制寄存器 (FSM\_WR\_ENA) 来启用。

一旦到 FSM 控制寄存器的存取被启用，可配置闪存组 7 的读取访问。

## 2.12 配置闪存组和供电模式

Hercules 系列微控制器上的闪存组和电荷泵支持三个不同的运转模式以优化功耗。

- 激活模式



- 闪存组感测放大器和感测基准被启用
- 闪存电荷泵的所有电路被启用
- 待机模式（只适用于闪存组）
  - 闪存组感测基准被启用，但是感测放大器被禁用
- 休眠模式
  - 闪存组感测放大器和感测基准被禁用
  - 闪存电荷泵的全部电路被禁用

缺省情况下，在任何系统复位后，闪存组和电荷泵处于激活状态。闪存模块允许应用为闪存组和电荷泵配置“工作在降”电源状态。当在用户可配置的时间内没有检测到对闪存组的访问时，闪存组和电荷泵自动将电源模式切换至所选的低效运行状态。

此闪存模块还包含专门的定时器，此定时器在激活和所选的低效运行状态之间对闪存组和电荷泵进行自动排序。在非激活电源状态中对任一闪存组的读取访问将把所选的组和电荷泵“唤醒”至激活电源状态。只允许对处于激活状态的组进行编辑和擦除操作。

闪存组控制寄存器 (FBAC) 控制闪存组的电源状态。

闪存电荷泵存取控制寄存器 (FPAC1, FPAC2) 控制闪存电荷泵的电源状态。

### 2.13 配置振荡器监视器

HF LPO 时钟源被用作一个监控主振荡器的基准时钟。如果振荡器频率落到范围之外，将会检测到一个故障： $\{f_{\text{HFLPO}}/4, f_{\text{HFLPO}} * 4\}$ 。

HF LPO 频率在并且随着工艺角以及内核电源 (VCC) 和温度的变化而发生很大变化。*Hercules* 微控制器使得应用能够调谐 HF LPO，这样应用能够选择 HF LPO 的运行频率点。这反过来确定了振荡器频率的有效范围。

器件测试期间，一个调谐值被写入地址为 0xF008\_01B4 的闪存存储器的一次性可编程部分 (OTP)。这个 OTP 字的位 31:16 包含一个 16 位值，为了初始化对 HF LPO 的调谐，可将这个值编辑进低功率振荡器监控器控制寄存器 (LPOMONCTL)。

或者，此应用可使用双时钟比较 (DCC) 模块来确定 HF LPO 的调谐设置。DCC 模块可实现两个时钟频率的比较：一旦 HF LPO 被确定处于 OTP 内最初 HFTRIM 设置的范围内，晶体振荡器可被用作一个基准，HF LPO 和 LF LPO 可针对此基准进行进一步的调谐。更多信息请参见器件专用技术参考手册。

### 2.14 运行闪存模块 **SECEDED** 逻辑自检

闪存模块从闪存组 0 的 TI OTP 区域内地址 0xF0080140 上读取“复位配置矢量”。这是一个 64 位值，这个值被用来配置器件电源域等。闪存模块具有内置 **SECEDED** 逻辑来检测这个矢量中的任何单位错误或检测并标出这个矢量中的双位错误。如果在从 OTP 中读取数据时检测到双位错误，那么 ESM 组 3 错误情况被标出，并且 nERROR 信号被置为低电平。如果在从 OTP 中读取数据时检测到一个单位错误，那么这个错误由 **SECEDED** 逻辑纠正 - 无标志被置位，也不向 ESM 发送错误信号。

在闪存组 0 的 TI OTP 扇区内有专用位置，这些位置被设定成具有单位和双位错误。具体来讲，对地址 0xF00803F0 的 32 位或 64 位读取会导致一个单位错误指示，而对地址 0xF00803F8 的 32 位或 64 位读取会导致一个双位错误指示。这些位置应用准备就绪，这样可确保闪存接口模块在从 OTP 读取数据时能够检测单位和双位错误。

## 2.15 时钟域

现在所有进一步的初始化步骤需要以应用最大的运行频率执行。现在此应用必须等待 PLL 锁定它们的目标频率，然后将器件时钟域映射至所需的时钟源。Hercules 微控制器上提供多个时钟域以简化使用这些时钟域的不同模块的配置和可控性（请见表 2）。

表 2. Hercules 微控制器上的时钟域

域名称	时钟名称	注释
CPU 时钟域	GCLK	GCLK 控制全部 CPU 子系统，其中包括浮点单元 (FPU) 和内存保护单元 (MPU)。
系统总线时钟域	HCLK	HCLK 与 GCLK 共用同一时钟源，并且时钟频率始终与 HCLK 一样。
系统外设时钟域	VCLK_sys	VCLK_sys 被用于诸如 VIM, ESM, SYS 的系统模块。使用一个 1 至 16 的可编程分频器，VCLK_sys 分频自 HCLK。
外设时钟域	VCLK, VCLK2, VCLK3	VCLK 是主外设时钟，并且与 VCLK_sys 同步。VCLK2 是第二外设时钟，并且被保留用于增强型定时器模块 (NHET) 和相关的传输单元 (HTU)。使用一个 1 至 16 的可编程分频器，VCLK2 也分频自 HCLK。 $f_{HCLK}$ 必须是 $f_{VCLK2}$ 的整数倍， $f_{VCLK2}$ 必须是 $f_{VCLK}$ 的整数倍。使用一个 1 至 16 的可编程分频器，VCLK3 也分频自 HCLK，并且被用于 TMS570LS3x 微控制器上的以太网和 EMIF 模块。
异步时钟域	VCLKA1, VCLKA2 和 VCLKA4	这些时钟域被保留用于具有严格抖动限制的专门通信模块。针对这些通信模块的协议（例如，控制器局域网 (CAN), FlexRay, 以太网）不允许已调制时钟用于波特率生成。异步时钟允许此时钟源用于从 GCLK, HCLK 和 VCLKx 时钟域中分离的波特时钟。
实时中断时钟域	RTI1CLK	这个时钟用于生成 RTI 模块的周期中断。

### 2.15.1 将时钟域映射至时钟源

Hercules 微控制器上的系统模块包含可将时钟域映射到任一可用时钟源的寄存器。

GCLK, HCLK 和 VCLKx 域的时钟源由 GCLK, HCLK, VCLK 和 VCLK2 源寄存器 (GHVSR) 选择。

VCLKA1 和 VCLKA2 域的时钟源通过外设异步时钟源寄存器 (VCLKASRC) 选择。

用于 VCLKA3 和 VCLKA4 域的时钟域由外设异步时钟配置 1 寄存器 (VCLKACON1) 选择。

RTI1CLK 域的时钟源通过 RTI 时钟源寄存器 (RCLKSRC) 选择。

### 2.15.2 示例时钟域映射

```

systemREG1->GHVSR    = (0U << 24U) // Use main oscillator as wake up source for GHV CLK
                    | (0U << 16U) // Use main oscillator for HV CLK when GCLK is off
                    | (1U);      // Use FMPLL as current source for GHV CLK
systemREG1->VCLKASRC = (6U << 8U) // Use second PLL output for FlexRay bit timing
                    | (0U);      // Use main oscillator for DCANx bit timings
systemREG1->RCLKSRC  = (1U << 8U) // Set the RTI1CLK divider to divide-by-2
                    | (0U);      // Use FMPLL as source for RTI1CLK
    
```

### 2.15.3 配置 VCLK, VCLK2 和 VCLK3 频率

VCLK 和 VCLK2 时钟信号分频自 HCLK 时钟信号。这些是可由系统模块时钟控制寄存器 (CLKCNTL) 配置的独立分频器。

注:

- VCLK2 频率也必须为 VCLK 频率的整数倍数。
- 在配置 VCLK2 和 VCLK 的分频比之间必须有一些延迟。

VCLK3 时钟信号也分频自 HCLK 时钟信号。这个分频器位于时钟控制寄存器 2 (CLK2CNTL) 内。

### 2.16 运行一个 CPU 自检控制器 (STC) 诊断校验

这涉及到在 STC 校验模式下运行一个 CPU 自检间隔。STC 自检模式会导致一个卡 0 错误，由于会出现一个 STC 故障，所以这个错误将被引入两个 CPU 中的一个。如果未指示 STC 故障，这就意味着 STC 不能够检测 CPU 内的错误，而器件运行是不可靠的。STC 自检配置和执行的相关信息请参见器件专用技术参考手册。CPU 将在 STC 自检完成时复位。复位处理程序例程可认为器件初始化从序列的下一步开始。

### 2.17 运行 CPU 自检 (LBIST)

CPU 自检的配置和执行的相关信息请参见器件专用技术参考手册。CPU 将在 STC 自检完成时复位。复位处理程序例程可认为器件初始化从序列的下一步开始。

### 2.18 运行一个 CPU 比较模块 (CCM-R4F) 诊断校验

CCM-R4F 比较每个 CPU 时钟周期上的双 Cortex-R4F CPU 输出。任何不匹配被标示为一个 ESM 组 2 错误。这样可确保两个 CPU 确实运行在锁步模式中。CCM-R4F 模块也使得应用能够使用内置自检例程来测试不同的错误条件。要了解如何在自检模式中配置 CCM-R4F，请参见器件专用技术参考手册。

### 2.19 运行可编程内置自检 (PBIST) 控制器诊断校验

PBIST 引擎被用来在所有片载存储器上运行存储器测试例程。对于应用来说，依靠这个引擎来检测和报告一个存储器故障情况十分关键。因此，对于应用来说，在使用它来进行实际的片载存储器测试前，有必要测试这个错误检测和报告机制。这可通过选择在一个 ROM 存储器上运行一个 RAM 测试来完成。这个测试必须生成一个存储器测试故障。这个应用可寻找错误标志来确保 PBIST 控制器确实能够检测和报告一个存储器测试故障。要了解如何在已选择的片载存储器上配置 PBIST 控制器开执行特定的存储器测试算法，请参见器件专用技术参考手册。

### 2.20 使用 PBIST 控制器来启动一个 CPU RAM 自检

首先测试 CPU RAM，这样此应用可以继续执行，而在以后测试其它存储器。要了解如何配置 PBIST 控制器，请参见器件专用技术参考手册。

### 2.21 初始化 CPU RAM

用于片载存储器自动初始化的系统模块硬件也会初始化相关的 ECC 或奇偶校验位置。这个机制现在也被用来初始化 CPU RAM。这个过程将 CPU RAM 全部清零，并且也编辑相应的 ECC 位置。

### 2.22 启用针对 BxTCM 接口的 Cortex-R4F CPU 的 ECC 校验

CPU 具有内部 ECC 逻辑，此逻辑保护 CPU 对于 BxTCM (RAM) 接口的所有访问。缺省情况下不适用这个逻辑，必须通过将系统控制协处理器的辅助控制寄存器，c1 的 B1TCMPEN 和 B0TCMPEN 位置位来启用。

### 2.23 使用 PBIST 控制器启动对所有双端口存储器的自检

测试单端口与双端口片载 SRAM 使用不同的算法。使用可编程 BIST (PBIST) 引擎执行片载存储器自检的相关信息，请参见器件专用技术参考手册。

## 2.24 运行用于 TCRAM 存取的 CPU ECC 逻辑的自检

CPU TCRAM 先初始化，这样所有 TCRAM 被清零，而相应的正确 ECC 位置被设定。针对 TCRAM 存取的 CPU ECC 逻辑测试包含破坏 ECC 位置来产生单位和两位 ECC 错误。要获得针对 TCRAM 存取的 CPU ECC 逻辑测试序列，请见器件专用技术参考手册或初始化示例项目。请注意，产生一个双位 ECC 错误的 TCRAM 位置读取会使 CPU 采取数据中止异常操作。初始化示例项目还包括一个示例数据中止处理程序。

## 2.25 运行用于程序闪存存取的 CPU ECC 逻辑的自检

闪存接口模块支持一个诊断模式（模式 7），此模式使得应用能够测试用于程序闪存存取的 CPU ECC 逻辑。要了解用于程序闪存存取的 CPU ECC 逻辑的测试序列，请参见器件专用技术参考手册或初始化示例项目。请注意，产生一个双位 ECC 错误的闪存位置读取会使 CPU 采取数据中止异常操作。此初始化示例项目还包括一个示例数据中止处理程序。

## 2.26 使用 PBIST 控制器启动所有单端口存储器的自检

由于在之前已经验证过 CPU RAM，它可以排除在这个测试之外。使用可编程 BIST (PBIST) 执行片载存储器自检的相关信息，请参见器件专用技术参考手册。

## 2.27 片载 SRAM 自动初始化

Hercules 微控制器上的系统模块使得所有片载 SRAM 在硬件内初始化。由于全部片载存储器都支持某种形式的错误检测，所以这一功能特别重要。CPU 数据 RAM 支持 ECC，而外设存储器支持奇偶校验错误检测。自动初始化机制还在需要时初始化 ECC 或奇偶校验存储器。

## 2.28 运行所有外设 RAM 奇偶校验保护机制的自检

对这个微控制器上大多数外设 RAM 的存取收到奇偶校验错误检测的保护。每个具有针对其相关存储器的奇偶校验错误检测的外设也包括一个自检模式，以确保它确实能够在一个外设 RAM 存取发生时检测和报告一个奇偶校验错误。在由相关外设启用之前，这些自检机制可由应用使用。

## 2.29 启用 Cortex-R4F CPU 的矢量中断控制器 (VIC) 端口

CPU 具有一个专用端口，此端口使得 VIM 模块能够提供一个中断处理例程的地址，连同中断 (IRQ) 信号。相对于 CPU 必须将等待的中断解码并且需要首先确定将被处理的最高优先级中断，这提供了更快的中断处理例程访问。

CPU 复位时，VIC 端口被禁用，并且必须由应用启用。通过将 CPU 系统控制寄存器 (SYS) 内的 VE 位置位来启用 VIC。

## 2.30 矢量中断管理器 (VIM) 配置

Hercules 微控制器上的 VIM 模块支持中断请求通道和中断生成源的灵活映射。通道编号和中断模块间的缺省映射在器件专用数据表中定义。中断通道编号还定义了通道间的固有优先级，编号低的通道具有更高的优先级。也就是说，优先级以下的顺序减少：通道 0 → 通道 1 → 通道 2 ... 通道 95。

对于这份应用报告，假定应用倾向于保持通道间的缺省优先级顺序。要了解用于改变中断通道和中断源间映射的控制寄存器的细节，请参见器件专用技术参考手册。

VIM 模块包含一个存储器，这个存储器保存中断处理例程（用于应用中启用的每个中断）的起始地址。这个存储器起始地址为 Hercules 微控制器上的基地址 0xFFFF82000。它由 32 位的 97 个字组成。图 4 中显示了 VIM 地址内存映射。

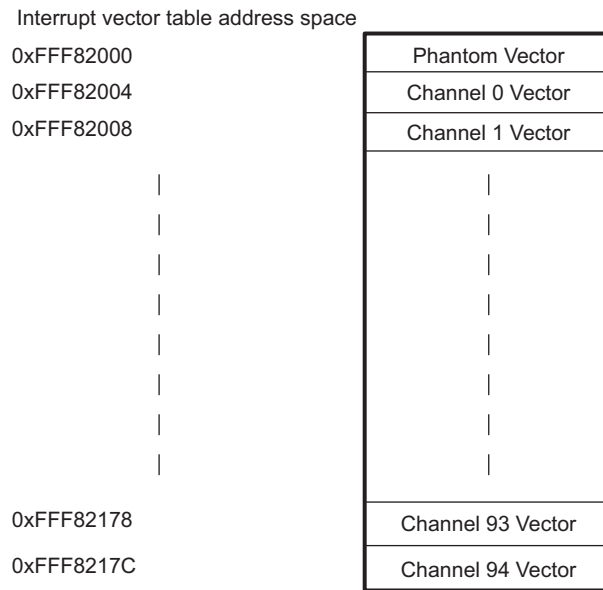


图 4. VIM 中断地址内存映射

### 2.30.1 将中断配置为快速中断或正常中断

对 VIM 的每个中断请求可被配置成发送给 CPU 的快速中断请求 (FIQ) 或正常中断请求 (IRQ)。FIQ/IRQ 程序控制寄存器 (FIRQPRx) 可实现这个选择。

中断请求 0 和 1 始终为 FIQ。所有其它中断缺省为 IRQ 中断。

注：一个被映射到 FIQ 的中断请求不能使用 CPU 的 VIC 端口。

### 2.30.2 启用和禁用中断

可使用中断使能设置来启用或禁用每个中断。中断请求 0 和 1 始终启用，并且不能被禁用。当一个中断被禁用时，如果此时中断条件被生成，但是未生成针对 Cortex-R4F CPU 的 IRQ 或 FIR 异常情况，那么这个中断并不防止中断标志被置位。

### 2.31 在 Cortex-R4F CPU 内启用中断

缺省情况下和 CPU 复位后，中断 (IRQ 和 FIQ) 在 Cortex-R4F CPU 内被禁用。将 Cortex-R4F CPU 内当前程序状态寄存器 (CPSR) 的“I”位清零可启用支持中断，而快速中断 (FIQ) 可通过将 CPSR 的“F”位清零来启用。

### 2.32 设置到组 1 错误的错误信令模块 (ESM) 响应

ESM 使得应用能够选择对组 1 分类中错误的模块响应。这些是最低严重性的错误，可由应用通过生成一个到 CPU 的中断来处理。ESM 还提供在外部 nERROR 引脚上标示任何组 1 错误的功能。

### 2.33 编译程序需要的额外初始化

如果源程序用 C 语言或 C++ 编写，那么 TI 编译程序要求创建 C/C++ 运行环境。其中包括：

- 复制表初始化，如果需要的话
- C/C++ 中定义的全局和静态变量的初始化

- 全局构造函数初始化
- 进行函数调用以分支进入主应用程序

这些针对每个编译程序的要求也许会有所不同。必须参考编译程序参考手册来确定所使用的编译程序的特定要求。

### 2.34 其它初始化步骤未在本文档中说明

下面是器件初始化期间应用能够执行额外操作列表

- 验证 DCC 模块能够检测并报告一个频率不匹配错误。
- 配置 DCC 模块以持续监视 PLL 输出频率。
- 这个微控制器上的几个总线主控包含他们自己的存储器保护单元以防止对存储器映射特定部分的存取。建议用户确保可检测出对这些 MPU 限制的保护违反，并将这些违反标记为 **ESM** 错误。
- 为每个总线主控配置 MPU。
- 使用 CRC 和 DMA 运行程序闪存存储器的背景校验
- 对嵌入式 ADC 模块的任一偏移错误进行校准。
- 在全部 ADC 输入上运行自检以确保他们没有开路或短接至电源或接地。
- 运行全部外设信号的 I/O 回路检验
- 配置窗口安全装置模块处理窗口尺寸以及对窗口违反的模块响应。
- 配置 N2HET1/N2HET2 监控功能。
- 设置 RTI 模块以按需生产周期中断。
- 使用 PCR 寄存器为外设配置所需的访问权限。
- 配置任一外部安全伴随芯片，例如 TI TPS6538x，用来在线诊断操作。

### 2.35 调用主应用

当使用 C/C++ 时，这是一个正常函数调用。它可以是一个分支指令或到执行此应用程序的例程的名称的分支指令连接。

例如：

```
main();
exit();
```

## 3 参考书目

- 《TMS570LSxxx7 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS162](#))
- 《TMS570LSxxx5 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS164](#))
- 《TMS570LSxxx4 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS165](#))
- 《RM48Lx50 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS174](#))
- 《RM48Lx40 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS175](#))
- 《RM48Lx30 16/32 位精简指令集计算机 (Risc) 闪存微控制器数据表》([SPNS176](#))
- 《TMS570LS31/21 16/32 位精简指令集计算机 (RISC) 闪存微控制器技术参考手册》([SPNU499](#))
- 《RM48 16/32 位精简指令集计算机 (RISC) 闪存微控制器技术参考手册》([SPNU503](#))

## 重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或隐含权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	<a href="http://www.ti.com.cn/audio">www.ti.com.cn/audio</a>	通信与电信	<a href="http://www.ti.com.cn/telecom">www.ti.com.cn/telecom</a>
放大器和线性器件	<a href="http://www.ti.com.cn/amplifiers">www.ti.com.cn/amplifiers</a>	计算机及周边	<a href="http://www.ti.com.cn/computer">www.ti.com.cn/computer</a>
数据转换器	<a href="http://www.ti.com.cn/dataconverters">www.ti.com.cn/dataconverters</a>	消费电子	<a href="http://www.ti.com.cn/consumer-apps">www.ti.com.cn/consumer-apps</a>
DLP® 产品	<a href="http://www.dlp.com">www.dlp.com</a>	能源	<a href="http://www.ti.com.cn/energy">www.ti.com.cn/energy</a>
DSP - 数字信号处理器	<a href="http://www.ti.com.cn/dsp">www.ti.com.cn/dsp</a>	工业应用	<a href="http://www.ti.com.cn/industrial">www.ti.com.cn/industrial</a>
时钟和计时器	<a href="http://www.ti.com.cn/clockandtimers">www.ti.com.cn/clockandtimers</a>	医疗电子	<a href="http://www.ti.com.cn/medical">www.ti.com.cn/medical</a>
接口	<a href="http://www.ti.com.cn/interface">www.ti.com.cn/interface</a>	安防应用	<a href="http://www.ti.com.cn/security">www.ti.com.cn/security</a>
逻辑	<a href="http://www.ti.com.cn/logic">www.ti.com.cn/logic</a>	汽车电子	<a href="http://www.ti.com.cn/automotive">www.ti.com.cn/automotive</a>
电源管理	<a href="http://www.ti.com.cn/power">www.ti.com.cn/power</a>	视频和影像	<a href="http://www.ti.com.cn/video">www.ti.com.cn/video</a>
微控制器 (MCU)	<a href="http://www.ti.com.cn/microcontrollers">www.ti.com.cn/microcontrollers</a>		
RFID 系统	<a href="http://www.ti.com.cn/rfidsys">www.ti.com.cn/rfidsys</a>		
OMAP应用处理器	<a href="http://www.ti.com.cn/omap">www.ti.com.cn/omap</a>		
无线连通性	<a href="http://www.ti.com.cn/wirelessconnectivity">www.ti.com.cn/wirelessconnectivity</a>	德州仪器在线技术支持社区	<a href="http://www.deyisupport.com">www.deyisupport.com</a>

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122  
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司