

i.MX 6SoloX功耗测量

本应用笔记将帮助用户设计功耗管理系统。本报告通过几个用例说明在Freescale SABRE SD平台上进行的i.MX 6SoloX应用处理器电流消耗测量。读者可以为i.MX 6SoloX芯片选择合适的电源域，并了解不同情况下的预期芯片功耗。

附注

本应用笔记中的数据基于小样本尺寸进行的经验测量，因此，无法确保所呈现的结果。

内容

1	i.MX 6SoloX 电压源概述	2
2	i.MX 6SoloX 处理器的内部功耗测量	4
3	用例和测量结果.....	9
4	降低功耗.....	18
5	用例配置和用法指南	21
6	修订历史记录	39



1 i.MX 6SoloX 电压源概述

i.MX 6SoloX 处理器具有若干个电源域（电压电源轨）和内部电源域。图 1 显示了这些电源轨的连接以及内部电源域的分布情况。

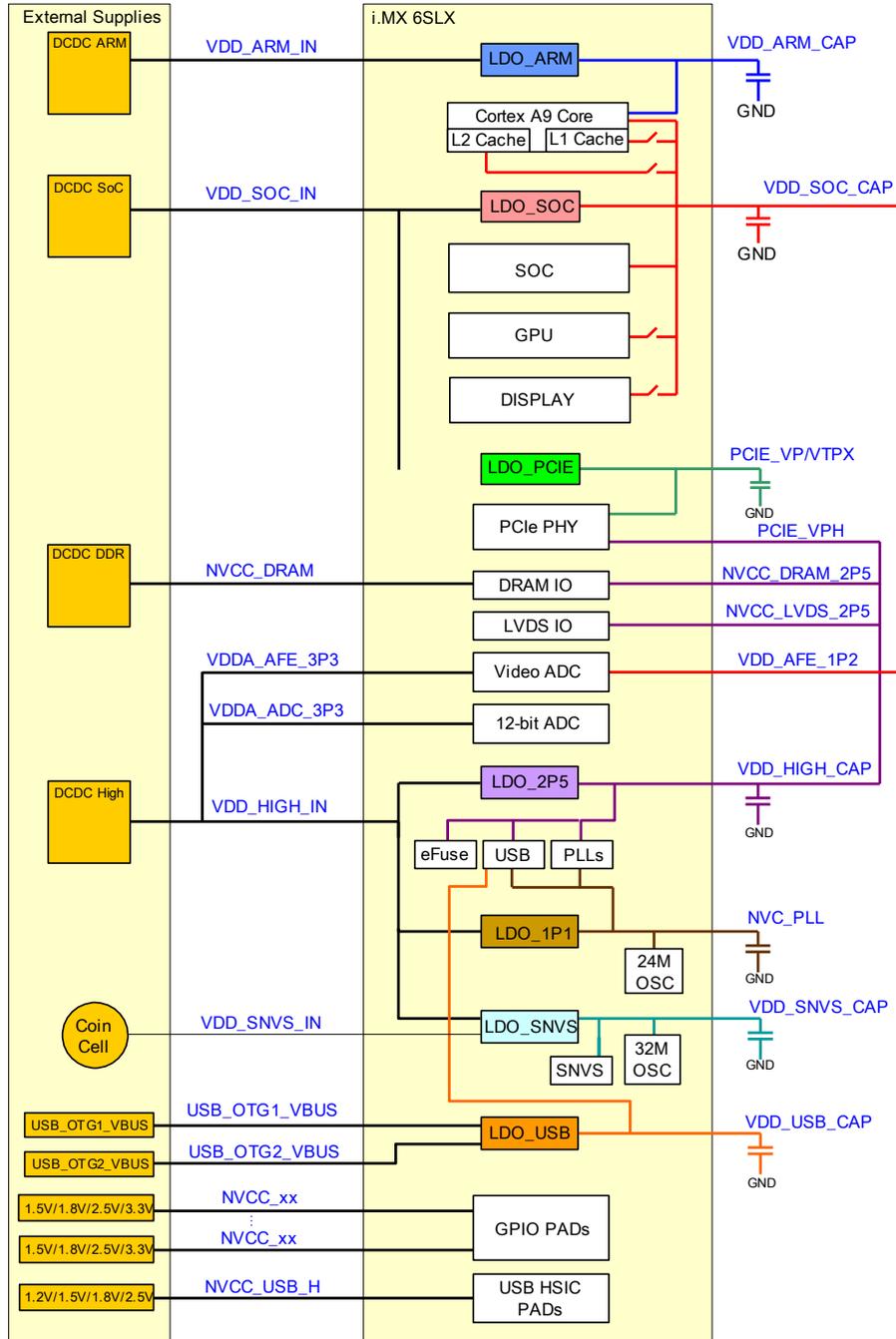


图1 i.MX 6SoloX电源轨

附注

每个电源轨的建议工作条件，以及每个 I/O 电压电源引脚组的详细说明，请参见 i.MX 6SoloX 消费电子产品的数据手册 (IMX6SXCEC)。

有关 i.MX 6SoloX 电源轨的更多详细信息，请参见 *i.MX 6SoloX 应用处理器参考手册 (IMX6SXRМ)* 的电源管理单元 (PMU) 章节。

2 i.MX 6SoloX 处理器的内部功耗测量

几个用例（[章节 3“用例和测量结果”](#)中描述）均运行于 SABRE SD 平台 (Rev B)。这些测量主要针对以下电源域进行：

- VDDARM_SOC_IN 平台的电源
- VDDHIGH_IN — PLL、DDR 预驱动器、PHY 和部分其他电路的电源

这些电源域消耗处理器的大部分内部功率。相关用例还有其他电源域的功耗。但是，这些电源域的功耗并不取决于特定用例，而取决于是否使用了这些模块。除了在深度睡眠模式以外，SNVS 的功耗相对可以忽略不计。

NVCC_*功耗主要取决于板级配置和组件。因此，它未包括在 i.MX6SoloX 内部功耗分析中。添加 NVCC_DRAM 该功耗供参考。

这些电源在不同用例中的功耗如[表 2](#)至[表 8](#)所示。

附注

除非另有说明，所有测量均针对典型工艺硅在室温（大约 26 °C）下进行。

2.1 VDDHIGH 功耗

电压 VDD_HIGH 域通过 2.5V LDO (LDO_2P5) 生成。

此电压域为以下电路供电：

- 片上 LDO
- 带隙
- MLB
- eFUSE
- PLL 模拟部分
- DDR I/O 的预驱动器 (NVCC_LVDS_2P5)

它也可以为以下域供电（取决于电路板连接）：

- PCIe
- LVDS 电桥
- DDR I/O 的差分输入缓冲器

2.2 DDR I/O 电源

DDR I/O 通过为 DDR I/O 焊盘供电的 NVCC_DRAM 提供。此电源的目标电压取决于所用的 DDR 接口。不同 DDR 接口的目标电压如下：

- 1.5 V (DDR3)
- 1.35 V (DDR3L)
- 1.2 V (LPDDR2)

NVCC_DRAM 电源的功耗受以下各种因素的影响：

- DDR 接口的活动量
- 片上终端 (ODT) — 使能/禁用、用于 DDR 控制器和 DDR 存储器的终端值
- 用于 DDR 控制的板终端和地址总线
- DDR 焊盘配置（如驱动强度）
- 板布局
- DDR 存储器器件的负载

附注

- 由于上述原因，下表中提供的测量值可能因系统而异。所提供的数据仅供参考，不应视为规范。
- 在 Freescale SABRE SD 平台上测得的电流还包括板载 DDR3L 存储器器件的电流。此板（在其上进行测量）包括两个 DDR3L 器件，总容量为 1 GB。SABRE SD 平台采用“T”拓扑结构，实现的板存储器路由无需板级电阻终端。这进一步减少了 DDR I/O 电源使用。

2.2.1 片上终端 (ODT) 设置

片上终端 (ODT) 是 DDR3/DDR3L SDRAM 的一种特性，允许 DRAM 打开/关闭每个 DQ、DQS、DQS#和 DM 信号的终端电阻。ODT 特性旨在提高存储器通道的完整性，方法是允许 DRAM 控制器独立地打开/关闭任意或所有 DRAM 器件的终端电阻。

使用较弱的 ODT 设置可以显著降低 DDR I/O 的功耗。所需的 ODT 设置取决于系统，并且会随板设计的不同而不同。应谨慎选择这些设置，以确保在满足 DDR 参数的 JEDEC 要求的同时实现功耗优化。

因此，系统设计人员需要调整 Linux BSP 版本中使用的默认设置，以适合于不同的系统。

2.3 测量过程中的电压水平和 DVFS 使用

除了 VDDARM_SOC_IN 外，所有电源的电压水平应设为 i.MX 6SoloX 消费电子产品数据手册 (IMX6SXCEC) 中定义的典型电压水平。

VDDARM_SOC_IN 电源需要特殊说明。为了省电，可以在用例的运行时间过程中使用 DVFS（动态电压和频率调整）更改 VDDARM_SOC_IN 电压。在低功耗模式下，可以将这些电源的电压更改为待机电压水平。

2.3.1 VDDARM 电压水平

VDDARM 的目标电压水平因所使用的 DVFS 设定点不同而不同，后者由 DVFS（也称为 CPUFREQ）驱动器选择。影响所选设置点的因素很多，其中 CPU 负载为最重要的因素。其他因素包括 CPU 延迟要求、热限制和外设 I/O 性能要求。测量所使用的电压和频率设定点如表 1 所示。

附注

正式的操作点请参见 i.MX 6SoloX 消费电子产品数据手册 (IMX6SXCEC) 的工作范围表。

大多数测量可以使用这些电压水平执行，并且本文档中出现的功耗数据基于这些值。如果在不同的电压水平下进行测量，则功耗将会随电压的变化而变化。在使用 DVFS 的实际应用中，软件配合硬件将会根据用例要求自动调整电压和频率值。

计算功耗所使用的电压为这些设定点之间的平均电压。它取决于在每个设定点上所花的时间量。

2.3.2 VDDSOC 电压水平

旁路 LDO_SOC 时，VDD_SOC_IN 的近似标称目标电压水平因 VDD_SOC_CAP 设定点的不同而不同。测量中使用的 VDD_SOC_CAP 设置参见表 1。正式的操作点请参见 i.MX 6SoloX 消费电子产品数据手册 (IMX6SXCEC) 的工作范围表。

表1. VDDARM、VDDSOC和VDDPU电压水平（仅供参考）

ARM频率	LDO状态	VDDARM_SOC_IN	VDD_ARM_CAP	VDD_SOC_CAP
996 MHz	已旁路	1.250V	1.250V	1.250V
792 MHz	已旁路	1.175V	1.175V	1.175V
396 MHz	已旁路	1.175V	1.175V	1.175V

2.4 温度测量

在某些用例中，需测量芯片温度。可以在热校准器件上使用片上热传感器来进行温度测量。测量温度时，建议等待直到温度稳定。

附注

测得的温度仅供参考，并且由于电路板、外壳、热扩散技术等不同，不同系统上测得的温度也不同。即便使用相同类型的电路板，测得的温度也可能因各种因素而不同，例如环境、硅变化和测量误差。

2.5 所用的硬件和软件

用于测量的软件版本如下：

- Yocto rootfs（版本：Oct.31 Yocto daily build）、Linux Kernel 版本：L3.10.53-1.1.0_ga + g80bec27。
- 用于测量的板为 Freescale SABRE SD 平台。
- 使用 Agilent 34401A 6 ½ 数字万用表执行测量。

2.6 用于功耗测量的板设置

使用默认电源电压进行功耗测量。默认的输入电压如下：

- VDDARM_SOC_IN 1.375 V 时
- VDD_HIGH_IN 3.0 V 时
- NVCC_DRAM 1.35 V 时

因此，使用不同的设置（如可配置的独立 ARM 直流交换机）后，可以通过降低 VDDARM_IN 输入电压水平进一步优化系统功耗，从而实现所需的操作点。此设置可能会导致较高的系统成本，因此，需要在成本和系统功耗之间实现平衡。

2.7 Freescale SABRE SD 平台上的测量点

可以通过测量各测量点处的平均压降，并将其除以电阻值以获得平均电流来获得功耗值。SD 板上 0.02 Ω 电阻的容差为 1%。不同电源域的测量点如下：

- VDDARM_SOC_IN — 芯片域电流于 SH10 上测得，此测量建议的电阻值为 0.02 Ω 。
- VDDHIGH_IN — VDDHIGH 域电流于 R375 上测得，此测量建议的电阻值为 0.02 Ω 。
- DDR3L I/O 和存储器 — 此域中的电流包括 NVCC_DRAM 电流和板载 DDR3L 存储器器件的总电流。此域中的电流于 R373 上测得，此测量建议的电阻值为 0.02 Ω 。

3 用例和测量结果

以下章节介绍的主要用例和子类型，可构成在 SABRE SD 平台上进行 i.MX 6SoloX 内部功耗测量基准。

3.1 低功耗模式用例

3.1.1 用例 1 — 深度-睡眠模式 (DSM)

此模式在 Linux BSP 中被称为“睡眠模式”或“暂存于 RAM”。这是可能的最低功耗状态，此时外部电源仍开启。

用例如下：

- ARM 平台采用门控电源。
- L1 缓存外设采用门控电源。
- SoC 调节器已旁路。
- 所有 PLL（锁相环）和 CCM（时钟控制器模块）生成的时钟关闭。
- CKIL (32 kHz) 输入开启。
- 所有模块禁用。
- 已应用偏置。
- 所有模拟 PHY 掉电。
- 外部高频晶振和片上振荡器掉电（通过断言 CCM 中的 SBYOS 位）。
- 通过断言 PMIC_STBY_REQ 将 VDDARM_SOC_IN 降至 0.975 V。在此模式下，外部电阻负载不会产生任何电流。

表 2 显示了此用例用于 i.MX 6SoloX 处理器时的测量结果。

表2. 深度睡眠模式 (DSM) 测量结果

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	0.972	1.021	1.050
VDD_HIGH_IN	2.973	0.743	0.25
总功耗（无DDR3L I/O + 存储器）	-	1.764	-
DDR3L I/O + 存储器3	1.365	22.318	16.350
总功耗	-	24.082	-

附注

有关此用例和设置的其他详细信息，请参见[章节 5 “用例配置和用法指南”](#)。

3.1.2 用例 2 — 系统闲置模式

用例如下：

- 如果 Kernel 处于最低闲置水平，则 ARM 采用电源门控。
- 如果 ARM 采用电源门控，则所有 PLL 关闭。
- 操作系统开启
- LCD 关闭。
- 屏幕未刷新。

此用例模拟器件在某段时间内处于闲置状态且显示器在定时器到期之后关闭的情况。

表 3 显示了此用例用于 i.MX 6SoloX 处理器时的测量结果。

表3. 系统闲置模式测量结果

电源域	L3.10.53-1.1.0_ga+g80bec27		
	电压 (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.177	10.684	9.077
VDD_HIGH_IN	2.990	1.653	0.553
总功耗（无DDR3L I/O + 存储器）	-	12.337	-
DDR3L I/O + 存储器	1.354	22.327	16.490
总功耗	-	34.664	-

附注

有关此用例和设置的其他详细信息，请参见[章节 5 “用例配置和用法指南”](#)。

3.2 音频播放用例 — MP3 音频播放

用例步骤如下：

1. MP3（MPEG-1 音频层 3）解码由 ARM 完成。
2. 通过 SAI（同步音频接口）运行音频播放。
3. 流（比特率为 128 kbps、采样频率为 44100 Hz 的 mp3 文件）来自 SD（安全数字）卡。

定时器到期之后，LCD 关闭。在 LCD 关闭状态下测量数据。

表 4 显示了此用例用于 i.MX 6SoloX 处理器时的测量结果。

表4. MP3音频播放测量结果

电源域	L3.10.53-1.1.0_ga+g80bec27		
	电压 (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.176	108.899	92.6
VDD_HIGH_IN	2.983	46.156	15.426
总功耗（无DDR3L I/O + 存储器）	-	155.055	-
DDR3L I/O + 存储器	1.352	48.3	35.752
总功耗	-	203.355	-

附注

有关此用例和设置的其他详细信息，请参见[章节 5“用例配置和用法指南”](#)。

3.3 Dhrystone 基准

Dhrystone 是一种复合基准测试，用于测量处理器和编译器的整数计算性能。Dhrystone 基准测试的小尺寸使其适合于 L1 缓存，因此可以最大程度减少对 L2 缓存和 DDR 的访问。

3.3.1 用例 1 — Cortex-A9 (996 MHz) 上的 Dhrystone 基准测试, Cortex-M4 (227 MHz) 上的 CoreMark 基准测试

在此用例中，Dhrystone 测试由两个内核执行。Cortex-A9 以 996 MHz 频率在环路中运行测试，Cortex-M4 以 227MHz 频率在环路中运行测试。

表 5 显示了此用例用于 i.MX 6SoloX 处理器时的测量结果。

表5. Cortex-A9 (996 MHz) 上的Dhrystone基准测试, Cortex-M4 (227 MHz) 上的CoreMark基准测试

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.262	697.222	552.474
VDD_HIGH_IN	2.985	78.207	26.2
总功耗（无DDR3L I/O + 存储器）	-	775.429	-
DDR3L I/O + 存储器	1.348	28.575	21.198
总功耗	-	804.004	-

3.3.2 用例 2 — Cortex-A9 (996 MHz) 上的 Dhrystone 基准测试, Cortex-M4 闲置

在此用例中, Dhrystone 测试由单个内核执行。ARM 处理器以 996 MHz 频率在环路中运行测试。M4 内核闲置。测量 ARM 的运行功率。

表 6 显示了此用例用于 i.MX 6SoloX 处理器时的测量结果。

表6. Cortex-A9 (996MHz) 上的Dhrystone基准测试, Cortex-M4闲置

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.261	657.990	521.8
VDD_HIGH_IN	2.985	53.775	18.015
总功耗 (无DDR3L I/O + 存储器)	-	711.765	-
DDR3L I/O + 存储器	1.348	28.165	20.894
总功耗	-	739.93	-

3.4 图形用例 — 3D 游戏基准测试, MM07

此用例具有以下特性:

- XGA 分辨率, 使用 MM07 (Taiji) 基准测试。
- 帧速率为 3.109 fps。
- 显示器使用 LVDS, 具有 XGA 分辨率。

图形从 SD 卡加载到 DDR (双倍数据传输速率) 存储器, 由 GPU3D 处理, 然后复制到 DDR 中的显示器缓冲器中。它随后被 PXP 采用并显示在 LCD 显示器 (通过 LVDS) 上, 刷新速率为 60 Hz。

在此用例中, 测量是在 DVFS 禁用且 CPU 速率设为 396 MHz 的情况下进行。

表7. 3D游戏MM07基准测量结果 — DVFS禁用

电源域	L3.10.53-1.1.0_ga+g80bec27		
	电压 (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.181	383.824	324.999
VDD_HIGH_IN	2.983	121.286	40.659
总功耗 (无DDR3L I/O + 存储器)	-	505.110	-
DDR3L I/O +1 存储器	1.350	225.646	167.145
总功耗	-	730.756	-

附注

有关此用例和设置的其他详细信息, 请参见[章节 5 “用例配置和用法指南”](#)。

3.5 典型最大功耗 — Cortex-A9 (996 MHz) 上的 Dhrystone 基准测试和 3D 游戏基准测试 (MM07, OpenGL ES2.0), Cortex-M4 (227 MHz) 上的 CoreMark 基准测试

此用例旨在提供常见用例的功耗，这是高度非典型的，但当计划在极端条件下工作时，可能与某些系统相关。

此用例在 LVDS 显示器上同时运行：

- 具有 XGA 分辨率、通过 LVDS 端口的 3D 图形。

此用例具有以下特性：

- 图形为 3D 游戏基准 — MM07。
- 两个 ARM 内核均为重负荷。
- 最大频率用于 ARM、GPU 和 DDR 时钟。

表8. SABRE SD平台上的典型最大功耗测量结果

电源域	L3.10.53-1.1.0_ga+g80bec27		
	电压 (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.265	939.389	742.6
VDD_HIGH_IN	2.982	131.769	44.188
总功耗 (无DDR3L I/O + 存储器)	-	1071.158	-
DDR3L I/O + 存储器	1.351	266.371	197.166
总功耗	-	1337.529	-

3.6 非多媒体用例

3.6.1 用例 1 — USB-至-eMMC 文件传输

在此用例中，总共有 1 GB 数据从 USB 设备传输至 eMMC 器件。每次复制的数据大小为 1 MB，重复 1000 次。SDMA 用于执行至 eMMC 主机控制器的数据传输。表 9 显示了传输测量结果。

表9. USB至eMMC文件传输测量结果

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.178	215.132	182.625
VDD_HIGH_IN	2.984	67.728	22.697
总功耗（无DDR3L I/O + 存储器）	-	282.86	-
DDR3L I/O + 存储器	1.350	141.904	105.114
总功耗	-	424.764	-

3.6.2 PCIe 数据传输用例

在此用例中，通过 PCIE 1G 以太网传输数据 100s。表 10 显示了传输测量结果。

表10. PCIe数据传输测量结果

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.260	645.498	512.3
VDD_HIGH_IN	2.980	135.369	45.426
总功耗（无DDR3L I/O + 存储器）	-	780.867	-
DDR3L I/O + 存储器	1.354	223.545	165.1
总功耗	-	1004.412	-

3.7 SNVS

在此用例中，除了 VDD_SNVS 域外，板上的大部分电源轨关闭，VDD_SNVS 用于保持 RTC 以及 SNVS 域中的其他逻辑处于上电状态。表 11 显示了 SNVS 测量结果。

表11. SNVS测量结果

电源域	电压 (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (uW)	I (uA)
VDD_ARM_SOC_IN	0	0	0
VDD_HIGH_IN	0	0	0
SNVS_IN	3.020	87.731	29.050
总功耗（无DDR3L I/O + 存储器）	-	87.731	-
DDR3L I/O + 存储器	0	0	0
总功耗	-	87.731	-

4 降低功耗

总系统功耗取决于软件优化程度以及系统硬件的设置。以下是有助于降低系统功耗的建议列表。部分已在 Linux BSP 中使用。可以在单独的用户系统上进一步优化。

附注

计划在未来的 BSP 版本上进一步实施功耗优化。请参考 Freescale 网站，以获取最新版本的 BSP。

- 无论是否使用时钟或模块，请通过配置时钟控制器模块 (CCM) 中的 CCGR 寄存器来应用时钟选通。
- 减少运行的 PLL 数目 — 主要应用于音频播放模式或闲置模式。
- 内核 DVFS 和系统总线调整 — 为 ARM 应用 DVFS 并调整 AXI、AHB 和 IPG 总线时钟的频率可以大大降低 VDDARM 和 VDDSOC 域的功耗。但是，由于工作频率降低，因此访问 DDR 需要的时间更长，这会增加 DDR I/O 和存储器的功耗。需要针对每种模式权衡考量，以量化对系统功耗的整体影响。
- 尽可能将 i.MX 6SoloX 置于低功耗模式 (WAIT、STOP)。详情参见 i.MX 6SoloX 应用处理器参考手册 (IMX6SXR) 的章节“时钟控制器模块 (CCM)”。
- DDR 接口优化：
 - 请小心使用 DDR 存储器板路由，使 PCB 走线尽可能短。
 - 可能的话，请使用较低的 ODT（片上终端）设置。使用的终端将会严重影响 DDR 接口引脚的功耗。
 - 为 DDR 接口引脚使用合适的输出驱动器阻抗，以实现良好的阻抗匹配。选择能够提供所需性能的尽可能最低驱动强度，以节省通过 DDR I/O 引脚的电流。
 - 谨慎选择板载电阻，使浪费的电流最低 — 例如，选择 CLK 和 CLK_B（使用 DDR3L 存储器）之间的阻抗匹配电阻。
 - 如可能，在较低性能用例下，切换至 DLL 关闭模式可大大降低 DDR 频率。这将禁用或减少终端，并且它将降低驱动强度。因此，可以大大降低 DDR 接口引脚的功耗。
 - 当 DDR 存储器处于自动刷新模式时，浮动 i.MX 6SoloX DDR 接口引脚（设为高 Z），并将 DDR_SDCKE0 和 DDR_SDCKE1 保持为低值。如果通过使用外部下拉电阻将 DDR_SDCKE0 和 DDR_SDCKE1 保持为低值，请确保在此模式下这些引脚上不存在任何板载终端。
 - 如可能（取决于系统稳定性），请将 DDR 输入引脚配置为 CMOS 模式，而不是差分模式。方法是将 IOMUXC 中对应寄存器中的 DDR_INPUT 位清零。建议在以低频运行（如处于 DLL 关闭模式时）时使用此设置。

- 使用以低 I/O 电压运行的 LV DDR3L 存储器设备可以将 I/O 功耗进一步降低 20%。
- 使用最新工艺技术的 DDR 存储器设备可以大大降低 DDR 器件和 DDR I/O 的功耗。

浮动 i.MX 6SoloX DDR 接口引脚涉及各步骤如下所示。

附注

通过内部 RAM 而不是 DDR 存储器运行代码时，将执行以下所有编程步骤。此代码不可缓存。

进入挂起（深度睡眠模式）之前要执行的步骤：

1. 读取 MAPSR 寄存器 MMDC 中的省电状态（已使能自动省电），以确保 DDR 处于自动刷新状态。
2. 进行以下操作：
 - a) 如果没有用于 DDR 控制和地址总线的板载终端，则将所有 DDR IF I/O 的 DSE（驱动强度选择，在 IOMUXC 中）设为 0（高 Z），除了 CKE0 和 CKE1。
 - b) 如果 DDR 控制和地址总线具有与 VTT 连接的板载终端电阻，如使用 SODIMM 的情况：
 - 方案 1：

对于 (2a)，保持 SDCKE0/1 有效，这将导致引脚中的部分其他电流共用 IOMUXC_SW_PAD_CTL_GRP_CTLDS 寄存器中相同的 DSE 控制。引脚为 DRAM_CS0、DRAM_CS1、DRAM_SDBA2、DRAM_SDCKE0、DRAM_SDCKE1 和 DRAM_SDWE。
 - 方案 2（需要在 DRARM_SDCKE0/1 引脚上使用板载下拉电阻）
 - 将终端电阻的电源设为浮动（可通过具有 GPIO 功能的部分引脚实现）。
 - 将所有 DDR IF I/O 的 DSE（驱动强度选择，在 IOMUXC 中）设为 0（高 Z）。
3. 进入挂起模式。

退出挂起模式之后要执行的步骤：

1. 将 DDR I/O 的所有设置恢复为所需值。
2. 系统进入运行模式。

附注

如果系统可确保无主机访问 DDR，则可将以下应用到包括深度睡眠模式在内的其他情形：可以相同方式浮动 DDR 引脚，即便未进入挂起模式，并且可以手动将 DDR 置于自动刷新模式，以节省耗电。当 CPU 未运行或其通过内部 RAM 运行时会发生此情况。

5 用例配置和用法指南

5.1 深度睡眠模式

在此用例中，除了用于系统唤醒的 32 kHz 时钟外，所有时钟和 PLL 均关闭。

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4。
2. 启动 A9 镜像文件。
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)。
4. 运行以下命令，使系统进入 DSM 模式：

```
echo mem > /sys/power/state
```

5. 测量功耗并记录结果。

5.2 系统 Idle 模式

5.2.1 系统 Idle 模式 — 时钟配置

表 12 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表12. 系统闲置模式时钟配置

时钟名称	频率 (MHz)
OCRAM	24
AHB	3
CPU	0
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	1

5.2.2 系统闲置模式 — PLL 配置

表 13 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表13. 系统闲置模式PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	0
PLL2 — 系统总线PLL	0
pll2 396m pfd	0
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	0
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.2.3 系统闲置模式 — 系统设置

全部断开，除了 SD 和 LVDS 之外。

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后再启动 M4
2. 通过“x11=false”启动 A9 镜像文件
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)。
4. 运行以下脚本，让系统进入 governor 模式：

```
#!/bin/bash
echo 8 > /proc/sys/kernel/printk
ifconfig eth0 down
ifconfig eth1 down
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 1 > /sys/class/graphics/fb0/blank
```

5. 测量功耗并记录结果

5.3 音频播放

5.3.1 音频播放 — 时钟配置

表 14 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表14. 音频播放时钟配置

时钟名称	频率 (MHz)
OCRAM	24
AHB	24
CPU	396
GPU2D	0
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	49.5

5.3.2 音频播放 — PLL 配置

表 15 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表15. 音频播放PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	0
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.3.3 音频播放 — 系统设置

- SD 启动
- 连接 XGA LVDS 面板

5.3.4 音频播放 — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4，并启动 A9 镜像文件。
2. 通过 SD 卡中的文件系统启动板
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)。
4. 运行以下脚本在 400M 下进行测量：

```
#!/bin/sh
echo 8 > /proc/sys/kernel/printk
ifconfig eth0 down
ifconfig eth1 down
```

```
echo conservative > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 0 > /sys/class/graphics/fb0/blank
```

6. 播放音频:

```
pacmd set-default-sink 1;while [ 1 ];
do gplay-1.0 ./128kbps_44khz_s_mp3.mp3;done;
```

7. 测量功耗并记录结果

5.4 3D 游戏

5.4.1 3D 游戏 — 时钟配置

表 16 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表16. 3D游戏时钟配置 — CPU频率为396 MHz

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D内核	720
GPU3D AXI	720
MMDC CH0	396

5.4.2 3D 游戏 — PLL 配置

表 17 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表17. 3D游戏PLL配置 — CPU频率为396 MHz

PLL名称	频率 (MHz)
PLL1 — 系统PLL	0
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	720
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.4.3 3D 游戏 — 系统设置

- SD 启动
- 连接 XGA LVDS 面板

5.4.4 3D 游戏 — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4。
2. 启动 A9 镜像文件并启动至 SD rootfs 的板。
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)。
4. 运行以下脚本在 400M 下进行测量：

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor echo 396000 >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed echo 0 >
/sys/class/graphics/fb0/blank
```

5. 复制 3Dmark_es20 应用程序 MM07 文件夹：basemark_es2.0 至本地 SD。
6. 运行 MM07 (Tajji) 应用程序并记录 fps。
7. 测量功耗并记录结果。

5.5 Cortex-A9 (996 MHz) 上的 Dhrystone，Cortex-M4 (227 MHz) 上的 CoreMark

5.5.1 Cortex-A9 (996 MHz) 上的 Dhrystone，Cortex-M4 (227 MHz) 上的 CoreMark — 时钟配置

表 18 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表18. Dhrystone时钟配置

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	396

5.5.2 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 (227 MHz) 上的 CoreMark — PLL 配置

表 19 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表19. Dhrystone PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	996
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	454
pll3 720m pfd	0
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.5.3 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 (227 MHz) 上的 CoreMark — 系统设置

- SD 启动
- 连接 XGA LVDS 面板

5.5.4 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 (227 MHz) 上的 CoreMark — 步骤

1. 将 M4 CoreMark 镜像文件载入 TCM, 然后再启动 M4。
2. 启动 A9 镜像文件并启动至 SD rootfs 的板。
3. 运行以下脚本在 1G 下进行测量:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

4. 运行 dry2 并测量

```
while true; do dry2 ; done
```

5. 测量功耗并记录结果。

5.5 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 idle

5.6.1 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 idle — 时钟配置

表 20 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表20. Dhrystone时钟配置

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	396

5.6.2 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 idle — PLL 配置

表 21 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表21. Dhrystone PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	996
PLL2 — 系统总线PLL	528
p1l2 396m pfd	396
p1l2 352m pfd	0
p1l2 594m pfd	0
PLL3 — OTG USB PLL	0
p1l3 508m pfd	0
p1l3 454m pfd	0
p1l3 720m pfd	0
p1l3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.6.3 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 闲置 — 系统设置

- SD 启动
- 连接 XGA LVDS 面板

5.6.4 Cortex-A9 (996 MHz) 上的 Dhrystone, Cortex-M4 闲置 — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI, 然后启动 M4。
2. 启动 A9 镜像文件并启动至 SD rootfs 的板。
3. 在 M4 控制台上输入“S”, 将 M4 更改为低频模式 (12 MHz)。
4. 运行以下脚本在 1G 下进行测量:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

5. 运行 dry2 并测量

```
while true; do dry2 ; done
```

6. 测量功耗并记录结果。

5.7 最大功耗

Cortex-A9 (996 MHz) 上的 Dhrystone 基准测试和 3D 游戏基准测试 (MM07, OpenGL ES2.0), Cortex-M4 (200 MHz) 上的 CoreMark 基准测试。

5.7.1 最大功耗: 时钟配置

表 22 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表22. 最大功耗时钟配置

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D内核	720
GPU3D AXI	720
MMDC CH0	396

5.7.2 最大功耗 — PLL 配置

表 23 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表23. 最大功耗PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	996
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	454
pll3 720m pfd	720
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.7.3 最大功耗 — 系统设置

- SD 启动
- 连接 XGA LVDS 面板

5.7.4 最大功耗 — 步骤

1. 将 M4 CoreMark 镜像文件载入 TCM，然后启动 M4
2. 启动 A9 镜像文件并启动至 SD rootfs 的板，连接至 XGA LVDS 显示器
3. 运行以下脚本在 1G 下进行测量：

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

4. 复制 3Dmark_es20 应用程序 MM07 文件夹：basemark_es2.0 至本地 SD
5. 运行 dry2 并测量

```
while true; do dry2 ;
done&
```

6. 运行 MM07(Taiji) 应用程序并记录 fps
7. 测量功耗并记录结果

5.8 USB 至 eMMC 文件传输

5.8.1 USB 至 eMMC 文件传输 — 时钟配置

表 24 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表24. USB至eMMC文件传输时钟配置

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	396

5.8.2 USB 至 eMMC 文件传输 — PLL 配置

表 25 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表25. USB至eMMC文件传输 PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	0
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	0
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.8.3 USB 至 eMMC 文件传输 — 系统设置

- SD 启动插入到槽 3
- 将 eMMC 卡插入槽 4 或焊接 eMMC 芯片的改造板

5.8.4 USB 至 eMMC 文件传输 — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4。
2. 启动 A9 镜像文件并启动至 SD rootfs 的板。
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)。
4. 运行以下脚本：

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
```

```
echo 1 > /sys/class/graphics/fb0/blank
```

5. 运行以下命令，将 USB 中的 1G 数据传输至 eMMC：

```
dd if=/dev/sda1 of=/dev/mmcblk0p1 bs=1M count=1000
```
6. 测量功耗并记录结果。

5.9 PCIE 数据传输

5.9.1 PCIE 数据传输 — 时钟配置

表 26 中的时钟配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表26. PCIE数据传输时钟配置

时钟名称	频率 (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D内核	0
GPU3D AXI	0
MMDC CH0	396

5.9.2 PCIE 数据传输 — PLL 配置

表 27 中的 PLL 配置与版本 L3.10.53-1.1.0_ga+g80bec27 一致。

表27. PCIE数据传输 PLL配置

PLL名称	频率 (MHz)
PLL1 — 系统PLL	0
PLL2 — 系统总线PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3 — OTG USB PLL	0
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4 — 音频PLL	0
PLL5 — 视频PLL	0
PLL6 — ENET PLL	100
PLL7 — 主机USB PLL	0
PLL8 — MLB PLL	0

5.9.3 PCIE 数据传输 — 系统设置

- SD 启动插入到槽 3
- 将 PCIE 以太网器件连接至板，然后将 PCIE 以太网端口直接连接至 PC（可支持 1 Gb/s 服务器）

5.9.4 PCIE 数据传输 — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4
2. 启动 A9 镜像文件并启动至 SD rootfs 的板
3. 在 M4 控制台上输入“S”，将 M4 更改为低频模式 (12 MHz)
4. 运行以下脚本：

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down
```

```
echo 1 > /sys/class/graphics/fb0/blank
```

5. 手动将 eth2 ip 设为 \$LOCALIP，然后将服务器 ip 设为 \$SERVERIP
6. 在服务器上运行“iperf -s”；
7. 运行“iperf -d -c \$SERVERIP -n 500M -B \$LOCALIP D3-t 100”
8. 测量功耗并记录结果

5.10 SNVS

5.10.1 SNVS — 步骤

1. 将 M4 Idle 应用镜像文件载入 QSPI，然后启动 M4。
2. 启动 A9 镜像文件并启动至 SD rootfs 的板。
3. 按住 SW2（开/关）键约 5 秒，让板掉电。
4. 测量功耗并记录结果。

5.11 重要命令

在 Uboot 控制台中

- `printenv` — 显示环境变量。
- `setenv` — 更新环境变量。
 - `setenv <name> <value> ...`
 - 将环境变量“name”设为“value ...”
 - `setenv <name>`
 - 删除环境变量“name”
- `saveenv` — 保存环境变量更新。
- `bootargs` — 传送至 `kernel`，这被称为 `kernel` 命令行。在 Linux 控制台中
- `cat /proc/cmdline` — 显示命令行
- `cat /sys/devices/virtual/thermal/thermal_zone0/temp` — 将温度印至屏幕（应校准芯片）
- `cat /sys/kernel/debug/clk/clk_summary` — 将所有 `clks` 印至屏幕。

6 修订历史记录

表 28 提供了此应用笔记的修订历史记录。

表28. 修订历史记录

版本编号	日期	重大变更
修订版0	05/2015	初次公开发行版。

How to Reach Us:

Home Page:

freescale.com

Web Support:

freescale.com/support

本文档中的信息仅供系统和软件实施方使用Freescale产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale销售产品时遵循以下网址中包含的标准销售条款和条件：

freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited. ARM Cortex™-A9 is the trademark of ARM Limited.

© 2015 Freescale Semiconductor, Inc.

© 2015 飞思卡尔半导体有限公司

