TMS570LS31x/21x 和 RM48x 器件安全手册 赫丘利斯™ ARM® 安全微控制器

用户指南

Literature Number: ZHCU035A
February 2012

内容

图片列表

图表列表

TMS570LS31x/21x 和 RM48x 器件安全手册 赫丘利斯™ ARM® 安全微控制器的安全手册

1 简介

作为一个系统和设备制造商或者设计人员，您有责任确保您的系统（和任一 TI 硬件或者包含在您系统内的软件组件）符合全部应用安全、规定、和系统级性能要求。 本文档中的所有应用和安全相关信息（包括应用说明、建议安全措施、推荐 TI 产品、和其它材料）只用作参考。 您了解并同意对在安全应用中使用 TI 组件负责，并且您（作为买家）同意对在此类应用中的造成的所有损失、索赔、诉讼、或者费用为 TI 辩护、保护 TI 不受伤害。

本文档是针对德州仪器 (TI) 的 赫丘利斯™ 安全微控制器系列产品的安全手册。 此产品系列使用一个常见的安全架构，此架构在专注多应用产品中被执行 。 这本安全手册涉及的产品实现包括：

- TMS570LS 车用安全微控制器
  - TMS570LS31x
  - TMS570LS21x
- RM4xx 工业用安全微控制器
  - RM48x

这本安全手册提供系统开发人员所需的信息以帮助他们使用一个受支持的赫丘利斯微控制器来创建一个安全系统。 这个文档包含：

- 扩展集产品架构概述
- 用于减少系统故障的开发过程的概述
- 针对随机故障管理的安全架构的概述
- 架构分区、实施的安全机制、和推荐用法的详细资料

下面的信息记录在《*TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器安全分析报告摘要*》(SPNU521) 和《*RM48x 赫丘利斯 ARM 安全微控制器安全分析报告摘要*》(SPNU522)，在本文档中就不再赘述：

- 芯片级上 MCU 故障率估计摘要
- 在安全公制计算中所使用的假设
- 芯片级 ISO 26262 标准安全公制摘要
- 芯片级 IEC 61508 标准安全公制摘要

下面的信息记录在《*TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器详细安全分析报告摘要*》(SPNU523) 和《*RM48x 赫丘利斯 ARM 安全微控制器详细安全分析报告摘要*》(SPNU527)，在本文档中就不再赘述：

- 定性故障模式和影响分析 (FMEA)
- 针对一个示例实现的定性故障树分析 (FTA)

- 用于估计器件故障率的错误模式适合于启用定制故障率计算
- 带有至器件子模块级细节的定量 FMEA（也被成为 FMEDA、故障模式、影响和诊断分析）适合于启用基于诊断的定制应用的计算

下面的信息记录在*《安全案例报告》*中，在此文档中就不再赘述：

- 显示遵守目标安全标准的证明汇总
- 目标标准符合性评估的结果

下面的信息记录在*《安全案例数据库》*中，在此文档中就不再赘述：

- 根据需要，可提供符合性目标安全标准的细节

使用本文档的用户应该大体上熟悉赫丘利斯产品系列。 可从以下网站获得更多信息http://www.ti.com/hercules。 本文档的目的是与相关数据表、技术参考手册、和其它处于开发阶段产品的文档一起使用。 这个技术内容部分是为了简化开发、减少内容重复、并避免混淆。

更多关于*《安全案例报告》*和*《安全案例数据库》*的信息，请通过http://www.ti.com与您的 TI 销售代表联系。

2 赫丘利斯 TMS57031x/21x 和 RM48x 产品概述

65nm 赫丘利斯产品系列是经验证的 TMS570LS 130nm 安全 MCU 系列产品到 65nm 制造工艺的革命。图 1中显示了一张产品扩展集架构的简化图。 这张图只是此架构的基本表示而非包含全部内容。 例如，此系列中的产品可以按照外设的数量、总线主控外设的数量、或者内存数量进行升级-但是程序设计者的模型仍然保持一致。

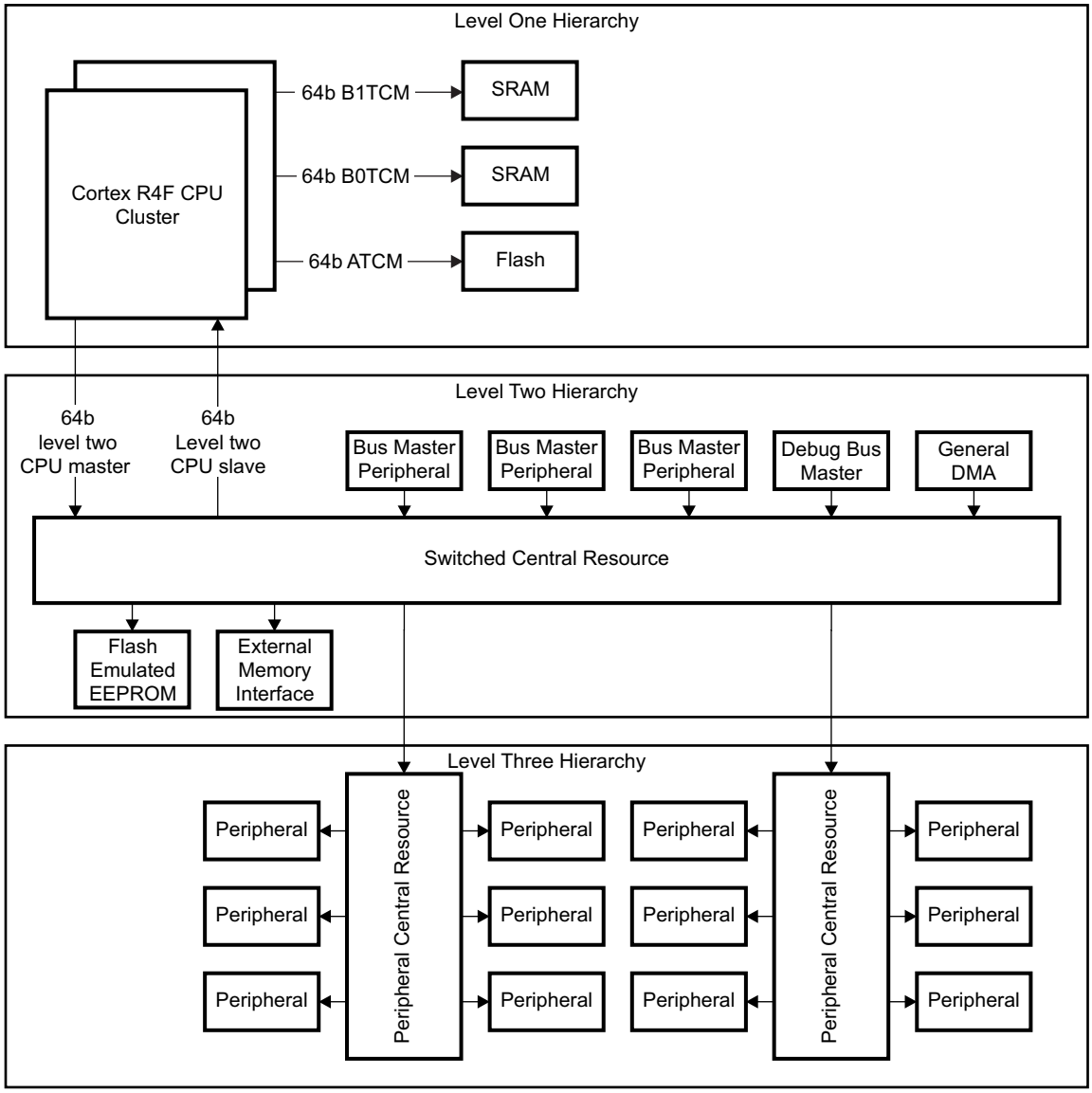


图 **1.** 赫丘利斯产品架构概述

赫丘利斯产品架构采用已经验证的 ARM Cortex™-R4F CPU，此 CPU 使用紧密耦合存储器配置。 此 Cortex-R4f CPU 由一个锁步配置内的检测器 Cortex-R4F CPU 实现。 这一配置在提供正确 CPU 运行逐周期检查的同时保持一个简单、易用的方法来使用单核程序设计者模型。 三级 64 位紧密耦合存储器 (TCM) 接口实现对主 CPU 存储器的访问。 这个 TCM 接口在每一个时钟周期内可实现到 SRAM 和闪存的多达三个并行访问。 此架构是一款经改良的哈佛 (Harvard) 程序并且数据访问并不只限于特定内存通道。 一个独立的 64 位二级总线主控接口提供到二级存储器等级的访问，而一个 64 位二级受控接口可实现非 CPU 总线主控到一级存储器的访问。

二级器件等级由一个开关中心资源（也被称为一个总线矩阵或者交叉开关矩阵 (crossbar)）控制。 这是一个器件级互连，此互连允许多个总线主控访问多个总线受控，还提供优先级、路径选择、解码、和仲裁功能。 到二级器件层次的总线主控包括 CPU、总线主控外设、调试总线主控、和通用直接存储器访问 (DMA) 控制器。 二级器件层次上的总线受控包括闪存 EEPROM 仿真存储器、外部存储器接口 (EMIF)、到一个或者更多外设总线段的访问、和一个 Cortex-R4F 受控端口可实现二级总线主控到一级紧密耦合存储器的访问。

三级层次主要由外设组成。 外设被分成一个或者多个外设总线段，由一个外设中心资源统一管理。 这个外设中心资源为总线事务处理目标外设提供地址解码功能。

2.1 目标应用

赫丘利斯 MCU 系列针对通用安全应用。 依据 ISO 26262-10:2011，为了支持环境安全因素 (SEooC) 开发，在初步设计阶段，就已经对多种安全应用进行了分析。 目标应用示例包括：

- 车辆刹车系统，包括轮胎防锁死系统 (ABS)、带有牵引控制的轮胎防锁死系统 (ABS+TC)、和电子稳定性控制系统 (ESC)
- 电机控制系统，特别是电子助力转向 (EPS) 系统和电动汽车 (EV) 动力传动系统
- 通用安全计算，例如主动安全系统中的集成传感器集群处理和车辆策略生成
- 工业自动化，例如用于安全流程控制的可编程逻辑控制器 (PLC) 和可编程自动化控制器 (PAC)

在目标系统的要求有所重叠的情况下，TI 已经尝试按照最严格的要求来设计器件。 例如，ESC 应用中针对定时器逻辑电路的故障耐受时间间隔通常为 100ms。 在一个 ESC 应用中，故障耐受时间间隔通常为 10ms。 在这种情况下，TI 已经按照 < 10ms 的故障耐受时间间隔来执行定时器子系统分析。

虽然 TI 在开发这些器件时考虑了特定的应用情况，但是这不应限制客户执行其它系统。 借助于所有安全组件，组件安全概念到系统安全概念的合理化转化必须由系统集成人员来完成。

2.2 产品安全约束

对于按照很多安全标准开发的安全组件，组件安全手册将提供产品安全约束列表。 对于一个简单组件，或者被开发用于一个单一应用的更加复杂的组件，这是一个合理的答复。 然而，赫丘利斯产品系列既是复杂设计，又非针对一个单一、特定应用而开发的器件。 因此，一个单一的产品安全约束集不能管理该产品所有可行的使用。 *《TMS570LS31x/21x 赫丘利斯 ARM详细安全分析报告》*(SPNU523) 和 *《RM48x 赫丘利斯 ARM 详细安全分析报告》*(SPNU527) 提供了带有相关产品安全约束的常见系统中的一个赫丘利斯产品的实现示例。

3 针对系统故障管理的赫丘利斯开发过程

对于一个安全开发，有必要管理系统和随机故障。 德州仪器 (TI) 已经针对安全半导体创建了一个唯一的开发过程，此过程大大减少了系统错误的可能性。 这一工艺建立在一个作为安全开发基础的标准质量管理 (QM) 开发工艺之上。 然后这一过程由一个第二层开发活动所补充，此开发活动为针对 IEC 61508 和 ISO 26262 的特定安全开发。

在 2007 年，为了依据 IEC 61508 标准进行产品开发，TI 首次认识到对这个标准开发过程进行补充的必要。 TI 与安全行业领头羊 exida 咨询公司密切合作以确保此开发符合标准的要求。 2008 年，按照 IEC 61508 第一版针对安全开发的过程被执行。 这一过程已经在多个微控制器新技术开发中被执行，这些新技术目前正应用于安全系统中。 本文档中所描述的赫丘利斯系列产品和安全架构在 IEC 61508 开发流程下开始开发。

到 2009年中，很明显，新出台的 IEC 61508 第二版和 ISO 26262 功能安全标准将要求增强型工艺流程能力。 由于这些草拟的标准缺还不成熟，在最终草案出台前，不太可能执行一个能够确保符合标准的开发工艺。 在 2009 年中，TI 与 ISO 26262 工作组一起，以期更好的理解并影响与微控制器硬件组件开发有关的标准。 作为 US 技术顾问组 (TAG) 和 ISO 26262 国际工作组的一成员，TI 在下列领域有突出贡献：

- ISO 26262:5-2011，附录 D - 描述故障描述的信息部分和推荐的硬件组件针对方法，TI 详尽的芯片故障模式描述知识和诊断方法的有效性加强了这一部分内容
- ISO 26262:10-2011，第 9 款-描述环境安全因素开发的信息部分，此技术使商用现货 (COTS) 安全组件的使用成为可能并将其合法化。
- ISO 26262:10-2011，附录 A - 描述如何将 ISO 26262 应用于微控制器的信息部分，这一部分受到了 TI 在将 IEC 61508 应用于微控制器开发中所学习到的经验和教训的影响

在 2010 年中，TI 开始开发一个符合 IEC 61508 第二版本和 ISO 26262 草案基准 18 的工艺流程。 在 ISO 26262 国际工作组中，TI 与 Yogitech 一起工作并发现此团队有互补功能。 已经建立起一个针对工程服务和安全咨询服务的合作伙伴关系以加速全新安全相关产品的开发。 Yogitech 现有的 fRMethodology 开发工艺和 TI 的 IEC 61508 开发工艺被整合在一起并被加强以创建一个全新的满足 IEC 61508 和 ISO 26262 标准的工艺。 随着 ISO 26262 标准的不断发展，这一工艺已经经历了一个持续改进的过程。 本文档所涉及的应用于第一代赫丘利斯芯片的工艺包含了针对 ISO 26262 草稿基准 21（2011 年 7 月）和 ISO 26262:2011 国际标准发布

3.1 TI 标准 MCU 汽车用开发流程

德州仪器 (TI) 从事针对安全和非安全汽车应用的汽车用微控制器开发已经超过二十年。 汽车市场对于产品的质量管理和高可靠性有着很强的需求。 虽然不是明确针对符合功能安全标准进行开发，TI 标准 MCU 汽车开发过程已经特有了很多管理系统故障所必须的要素。 这个开发过程可以被看成是质量管理 (QM)，但是并未达到 IEC 61508 安全完整性级别 (SIL) 或者 ISO 26262 汽车安全完整性级别 (ASIL)。 TI 标准 MCU 汽车开发过程经认证符合 ISO TS 16949，此认证由 Det Norske Veritas 认证公司 （Katy，德克萨斯州）在证书 CERT-07319CC10-2004-AQ-HOU-IATF 下评定（IATF 证书编号 0113679）。 此开发经认证也符合 ISO 9001:2008，此认证评估由 DNV 认证 B.V.（荷兰） 在证书 CERT-06185-2003-AQ-HOU-RvA 修订版本 2 下评定。

此标准过程将开发分成以下三个阶段：

- 商业机会预先筛分
- 程序计划编制
- 创建
- 验证、采样、和辨别
- 证明
- 产量增加和持续生产

标准过程显示在图 2中。

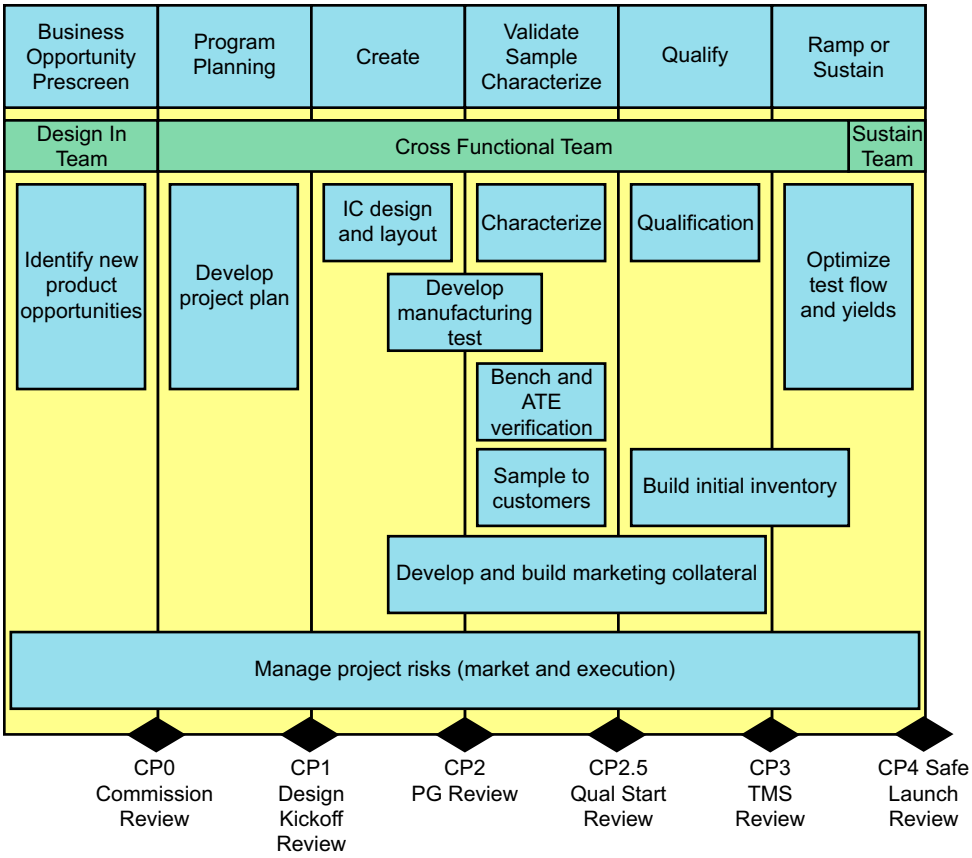


图 2. TI 标准 MCU 汽车 QM 开发过程

3.2 TI MCU 汽车 IEC 61508 开发过程

德州仪器 (TI) 在 2008 年开发了一个用于开发安全汽车应用的最初工艺。 这一工艺是针对 IEC 61508 第一版标准开发的，由可用的委员会第二版本草稿进行补充。 此工艺被开发为一个活动的附加层，除了标准 MCU 汽车 QM 开发工艺，还应执行这些活动。 这个被应用于 TMS570LS20216S 产品开发的工艺已经由 exida 认证 S.A.（证书号 TI 071227 C001）评定适合于在 IEC 61508 SIL 3 中使用。

这个工艺中的全新关键活动包括：

- 任命一个负责所有安全相关活动的安全经理
- 开发一个安全计划以跟踪安全相关活动
- 安全要求的生成、应用、和验证
- 定性 (FMEA) 和定量 (FMEDA) 安全分析的执行
- 对安全手册和安全分析报告进行修改以支持客户开发

3.3 Yogitech fRMethodology 开发工艺

fRMethodology 是 YOGITECH 私有的针对安全设计探索的“白盒子”方法，此方法包括：

- fRFMEA，按照 IEC 61508 和 ISO 26262 来执行一个集成电路 FMEA 的方法
- fRFI，一个在基于来自 fRFMEA 的输入的集成电路执行故障注入的工具

由于评估和验证一个给定集成电路的安全故障失效比率的流程符合 IEC 61508（证书号 Z10 06 11 61674 001），fRMethodology 已经 TÜV SÜD 批准。 作为 ISO-TC22-SC3-WG16 (ISO 26262) 意大利和国际工作组的成员，YOGITECH 发挥了积极的作用从而将此方法扩展至 ISO 26262。 在 ISO 26262 国际工作组中，YOGITECH 负责部分 10 的附录 A，例如，如何在一个 ISO 26262 应用的环境中处理微控制器。

而且，由于 YOGITECH 在模拟设计和模拟验证方面的坚实经验，YOGITECH 将 IEC 61508 和 ISO 26262 的要求扩展至模拟电路领域。 YOGITECH 一个用于模拟电路验证的工具，AMSvkit，此工具已经被世界上的几家公司使用http://www.amsvkit.yogitech.com/。

YOGITECH 的 fRMethodology 符合 ISO 26262-10:2011，附录 A。它主要包括：

- 将组件或者系统分成基本部件（“敏感区域”）
- 计算它们的故障率
- 使用那些故障率来计算安全标准
- 使用故障注入来验证结果
- 通过改变架构或者技术参数来实现对那些标准的敏感度分析
- 传递到客户号来比较不同的架构

3.4 赫丘利斯增强型安全开发工艺

赫丘利斯增强型安全开发工艺是现有 TI 和 Yogitech 的针对功能安全开发流程的融合。 工艺开发的目的在于吸取每一个流程的最佳方面并将它们结合在一起，形成同类产品中最佳的系统故障减少能力。

工艺流程以符合 IEC 61508 和 ISO 26262 基准 21 为目标，并经历持续改进的过程以包含未来 ISO 26262 工作组草案的全新特性。 TI 和 Yogitech 将这些功能安全标准设定为目标是因为他们相信这些标准代表了半导体功能性安全开发的技术发展水平。 虽然未将其它功能性安全标准作为目标，我们认为以业界技术发展水平为目标开发出的产品能够很容易地被应用于其它功能性安全系统。

这个组合的工艺流程中的关键要素包括：

- 在系统级设计、安全概念、和基于 TI 在安全系统开发方面的专业知识的要求方面的假定
- 组合定性和定量或者相似安全分析技术包含 TI 和 Yogitech 所知的芯片故障模式和诊断技术的汇总
- 基于多重工业标准以及 TI 制造数据的故障评估
- Yogitech 的代表技术发展水平的用于已宣称诊断范围验证的故障注入技术的应用
- 两个公司通过针对 IEC 61508 多重安全开发和参与 ISO 26262 国际工作的过程所获得的经验和教训的整合

图 3用图例显示了这些覆盖在标准 QM 开发流程顶部的活动。

Phase 0 Business Opportunity Prescreen	Phase 1 Program Planning	Phase 2 Create	Phase 2.5 Validate, Sample, and Characterize	Phase 3 Qualify	Phase 4 Ramp or Sustain
Determine if safety process execution is necessary	Define SIL/ASIL capability	Execute safety design	Validate safety design in silicon	Qualification of safety design	Implement plans to support operation and production
Execute development interface agreement (DIA) with lead customers and suppliers	Generate safety plan	Qualitative analysis of design (FMEA and FTA)	Release safety manual	Release safety case report	Update safety case report (if needed)
	Initiate safety case	Incorporate findings into safety design	Release safety analysis report	Update safety manual (if needed)	Periodic confirmation measure reviews
	Analyze system to generate system level safety assumptions and requirements	Develop safety product preview	Characterization of safety design	Update safety analysis report (if needed)	
	Develop component level safety requirements	Validation of safety design at RTL level	Confirmation measure review	Confirmation measure review	
	Validate component safety requirements meet system safety requirements	Quantitative analysis of design (FMEDA)			
	Implement safety requirements in design specification	Incorporate findings into safety design			
	Validate design specification meets component safety requirements	Validation of safety design at gate/layout level			
	Confirmation measure review	Confirmation measure review			

图 **3.** 赫丘利斯增强型功能性安全开发工艺

4 针对随机故障管理的赫丘利斯产品架构

对于一个安全开发，有必要管理系统和随机故障。 赫丘利斯产品架构包括很多安全机制，当正确使用时，安全机制能够检测并对随机故障做出响应。 此文档的这一部分对于针对 MCU 的架构安全概念进行了说明。

4.1 针对安全分析的安全岛原理和架构划分

RM4x 和 TMS570 赫丘利斯处理器共享一个被称为“安全岛”原理的共同安全架构概念。 这个基本概念涉及到硬件诊断应用和软件诊断间的一个平衡以在管理功能安全的同时平衡成本。 在“安全岛”方法中，一个元件的内核集被持续分配来运行硬件安全机制。 这个元件内核集，其中包括电源和时钟以及复位、CPU、闪存存储器、SRAM 和相关到闪存和 SRAM 的互连，需要确保所有软件功能的正确执行。 为了在其它器件元件上，例如外设，提供基于软件的诊断，一旦这些元件的正确运行被确认，软件就能够在这些元件上执行。 通过在客车空间中生成多个安全产品，这一概念已经被证实可行。

图 4用图例显示了覆盖在赫丘利斯产品架构扩展集配置上的安全岛方法。

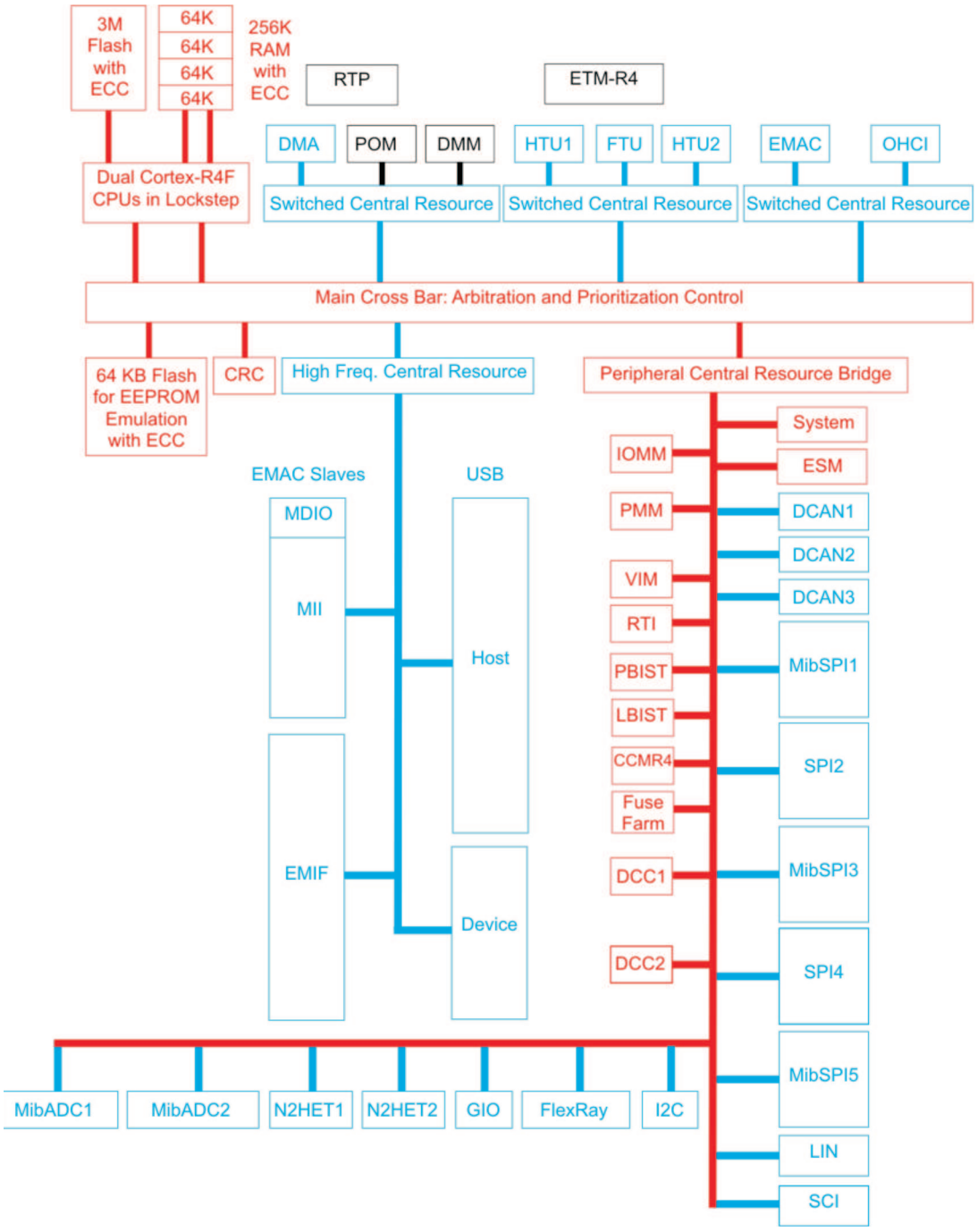


图 **4.** 赫丘利斯 **MCU** 用于安全分析的部分

图 4用图例显示了三个架构划分：

- “安全岛层（红色）”-此区域为所有处理操作所需的逻辑电路。 这个逻辑电路受到主板硬件诊断和特定使用假定的重要保护以确保对于安全运行的高级别信心。 一旦这个区域是安全的，它可被用于在其它设计元件上提供综合性的软件诊断。
- “混合层”（蓝色）-这个区域是包含大多数安全外设的区域。 这个区域相对不太依赖于硬件诊断。 软件诊断和应用协议被置于它们之上以提供必需诊断覆盖的剩余项。
- “离线层”（黑色）-这个区域的逻辑电路具有最少的或者没有集成硬件诊断。 这一层中的很多特性只用于调试、测试、和校准功能；在安全运行期间，闪存未激活。 这个区域的逻辑电路可被用于安全运行，假定系统集成人员已经添加了适当的软件诊断或者系统级措施。

4.2 系列变量管理

赫丘利斯系列架构支持多个产品变量。 这些产品能够作为唯一芯片设计被执行或者它们可以是共享芯片设计，在这些设计中，有些元件，即使出现在芯片中，也被禁用或者不被技术规范所相信。 只有在特定器件数据表和技术参考手册中进行明确说明的扩展集架构的元件才能被确保出现并运行。 当进行赫丘利斯平台开发时，建议安全概念应基于扩展集产品架构以在系列变量范围内启用最大扩展性。 上一个部分中显示的扩展集架构针对安全手册介绍部分中记录的所有器件部件号有效。

4.3 运行状态

赫丘利斯 MCU 产品有一个运行状态的共用架构定义。 这些运行状态应该由系统开发人员在他们的软件和系统级设计概念中进行观测。 运行状态状态机显示在图 5中并说明如下。

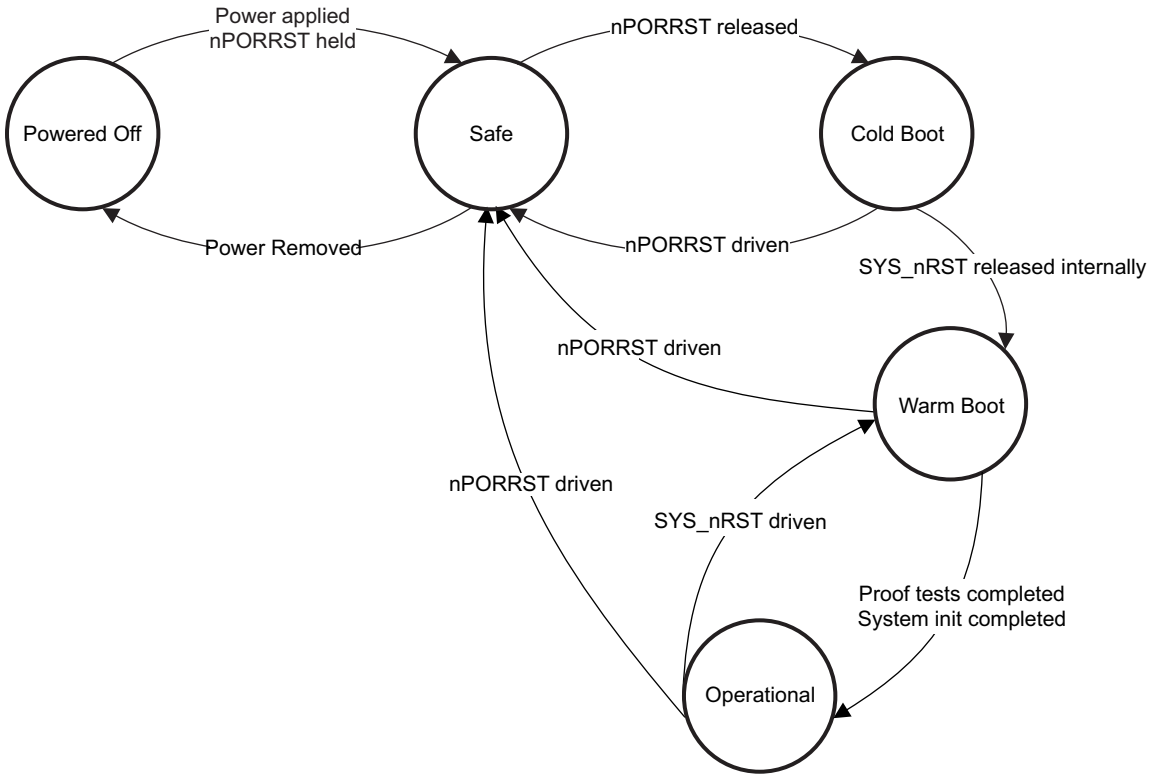


图 5. 赫丘利斯 MCU 运行状态

- “断电”-这是赫丘利斯 MCU 的初始运行状态。 内核或者 I/O 电源均未加电，器件处于非功能状态。 这个状态只能转换到安全状态，并且只能通过安全状态到达此状态。
- “安全”-在安全状态中，赫丘利斯 MCU 被加电但还不可用。 nPORRST（加电复位、也被成为冷启动）由系统置成有效，但是在电源缓慢上升为稳定状态之前不被释放。 如果电源不在一个最小运行范围内，内

部电压监视 (VMON) 安全机制也会继续将 nPORRST 置为器件内部有效。 当产品处于安全状态时，CPU 和外设不可用。 输出驱动器是被保持在一个只输入状态的三态和输入/输出引脚。

- “冷启动”-在冷启动状态中，关键模拟元件、数字控制逻辑电路、和调试逻辑电路被初始化以为未来使用。 CPU 保持供电状态但不可用。 当冷启动过程完成时，SYS_nRST 信号被内部释放，导致热启动级。 SYS_nRST 信号过渡改变能在 SYS_nRST I/O 引脚上被外部监控。
- “热启动”-热启动模式将信号逻辑电路复位并启用 CPU。 CPU 开始从闪存存储器中执行软件并且器件的软件初始化开始。 没有硬件连环显示热启动已经完成；这由软件决定。
- “可用”-在可用模式期间，器件能够支持安全功能性。

4.4 错误管理

当诊断检测到一个故障，这个错误必须被标出 赫丘利斯产品架构使用一个被称为外设错误信令模块 (ESM) 的外设逻辑电路来提供来自内部安全机制的故障指示集合。 ESM 提供了一些机制来将错误按照严重性分类并提供可编程错误响应。 在表 1中，对 ESM 中的错误分类进行了汇总。

表 1. ESM 错误标示概要

错误组	中断响应	错误引脚响应	注释
1	可编程中断和可编程中断优先级	可编程响应	对于一般为非关键严重的错误
2	生成不可屏蔽的中断	错误引脚被激活	对于一般为关键严重的错误
3	无中断响应	错误引脚被激活	对于那些在 CPU 中执行的诊断发现的关键错误

当一个错误被标明后，MCU 或者系统对此错误做出响应。 赫丘利斯产品的错误响应有多种可能。 系统集成人员负责确定应该采取哪种错误响应并确保此响应与系统安全概念一致。

- CPU 异常结束-对于在 CPU 中执行的诊断，这个响应在 CPU 中直接执行。 在异常中断期间，此程序序列将环境转化为一个异常中断句柄并且软件有机会来管理此故障。
- CPU 中断-这个响应可由 CPU 外的诊断来执行。 在软件有机会来管理此故障的地方，一个中断允许 CPU 外部事件来生成一个程序序列环境并将其转化为中断句柄。
- SYS_nRST 生成-这个响应使得器件能够从操作状态变为热启动状态。 SYS_nRST 可以从一个外部监视器生成或者在内部由软件复位或者安全装置生成。 当不可能恢复到操作状态时，重新进入热启动状态使得软件恢复成为可能。
- nPORRST 生成-这个响应允许器件的状态变为冷启动状态、热启动状态、或者操作状态。 当不能从热启动状态中恢复时，有可能从这个状态重新进入冷启动状态来尝试恢复。 如果需要，也有可能进入断电状态来执行一个系统级安全状态。 这个响应可从内部电压监视器生成，但是主要由 MCU 外部监视器驱动。

ESM 提供了可由 CPU 读取的多个寄存器来确定诊断的当前状态和 nERROR 引脚状态。 对于严重分组 2 中的错误，提供一个不是由 SYS_nRST 复位的影子寄存器。 这使得热复位有可能重新初始化以识别一个启动外部复位的分组 2 错误。

对于 CPU，有可能手工触发 nERROR 引脚响应来测试系统运行或者向外部逻辑电路通知一个内部故障，此故障不是被自动显示给 ESM。 CPU 负责清除 ESM 中显示的错误，其中包括清除 nERROR 引脚响应。

可以通过使用 TI 系统基础芯片（专为与赫丘利斯系列一起使用而开发）来简化外部错误响应的系统级管理。

5 赫丘利斯架构安全机制和使用假设

作为一个系统和设备制造商或者设计人员，您有责任确保您的系统（和任一 TI 硬件或者包含在您系统内的软件组件）符合全部应用安全、规定、和系统级性能要求。 本文档中的所有应用和安全相关信息（包括应用说明、建议安全措施、推荐 TI 产品、和其它材料）只用作参考。 您了解并同意对在安全关键应用中使用 TI 组件负责，并且您（作为买家）同意对在此类应用中的造成的所有损失、索赔、诉讼、或者费用为 TI 辩护、保护 TI 不受伤害。

在这一部分中，对赫丘利斯架构每个主要功能块的安全机制进行了总结并给出了使用方面的一般假设。 这些信息应该被用于确定采用安全机制的策略。 每一个安全机制的细节可在用于 MCU 的特定器件技术参考手册中找到。 硬件安全机制的有效性记录在《*TMS570LS31x/21x 赫丘利斯 ARM 详细安全分析报告*》(SPNU523) 和《*RM48x 赫丘利斯 ARM 详细安全分析报告*》(SPNU527) 中。

在这个部分中，TI 将针对安全机制使用的技术建议分成了几个类别。 不应该认为 TI 建议绝对无错。 有很多不同的方法来执行安全系统并且替代的安全机制也有能提供支持来实现所需的安全标准。 建议的类别如下：

- 强制的-一个强制的标志表明在正常功能运行期间安全机制一直可用并且不能由用户禁用。
- 强烈推荐-一个强烈推荐的标志表明，TI 相信这个安全机制能够提供很难由其它方法执行的且具有较高价值的诊断。 由于安全机制的启用或者禁用需要用户的干预，用户保留在他们的设计中使用或者不使用此安全机制的权利。
- 推荐-一个推荐标志表明，TI 相信这个安全机制能够提供可使用其他方法执行的有价值的诊断。 由于安全机制的启用或者禁用需要用户的干预，用户保留在他们的设计中使用或者不使用此安全机制的权利。
- 可选-一个可选标志表明，TI 相信这个安全机制能够提供可使用其他方法执行的价值较低的诊断。 由于安全机制的启用或者禁用需要用户的干预，用户保留在他们的设计中使用或者不使用此安全机制的权利。

根据安全标准和目标端设备的不同，也许需要对单点故障和潜在故障进行管理。 依照 ISO 26262：2011，当一个功能中的故障出现时，被考虑的潜在故障为：违反一个安全目标并在安全机制中导致一个故障的功能。 在故障耐受时间间隔内，并不需要对潜在故障进行测试，但是可以在引导时间、关断时、或者定期执行对此类故障的测试，这可由系统开发人员确定。 本部分中所描述的很多安全机制可被用作初次诊断、针对潜在故障的诊断，或者二者兼备。 当从潜在故障管理角度来考虑系统设计时，针对通过软件执行的任何初次诊断，请注意将 CPU 和内存故障考虑在内。

5.1 电源

赫丘利斯器件系列产品要求一个外部器件为正常运行提供所需的电压和电流。 为内核逻辑电路和 I/O 逻辑电路提供了独立的电源轨（包括模数转换器 (ADC)、闪存泵和振荡器）。

5.1.1 嵌入式电压监控器(VMON)

赫丘利斯平台组装有一个简单嵌入式电压监控器，此监控器能够大体检测超出范围的电源电压。 VMON 持续运行并不要求软件配置或者 CPU 开销。 VMON 监视内核以及 I/O 电源。 当电源远远高于或者低于额定电压（对于产品特定电压值，请参阅特定器件数据表）时，VMON 从内部驱动 nPORRST（加电复位）引脚。 这个响应将器件保持在安全运行状态。 当电源处于范围之内，VMON 将不会干预 nPORRST 信号。 要获得更多与 VMON 操作相关的信息，请参阅特定器件数据表。

VMON 是一个持续处于运行状态的诊断。 不能将 VMON 诊断禁用。 VMON 诊断的系统内测试通常是不可行的，这是因为需要对外部电压进行严密控制来触发 VMON 错误响应。 如果应用不当，这样一个电压会引起对 MCU 的永久损坏。 VMON 的使用是强制的。

SPNU511 — http://www-s.ti.com/sc/techlit/SPNU511

5.1.2 外部电压监视器

赫丘利斯平台强烈建议使用一个外部电压监视器来监控所有的电压轨。 这个电压监视器应该被配置成过压和欠压阀值与目标器件所支持的电压范围相匹配（特定器件数据表对此进行了注释）。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的外部电压监视器来定义。

5.1.3 注释

- 可以通过使用一个 TI 为赫丘利斯系列开发的系统基础芯片来简化对系统级上电压监视的管理。
- 器件可由用于组合在系统印刷电路板 (PCB) 上的多个电源轨来执行。 为了电源诊断的正确运行，建议在每一个成组的电源轨安排一个电压监视器。
- 外部电压监视器的共模故障分析也许对于在电压生成和监视电路中确定从属关系有所帮助。

5.2 电源管理模块 (PMM)

这个电源管理模块 (PMM) 的任务是控制可开关电源域。 根据所使用的系列变量的不同，可以执行一个或者多个电源域。 电源域可由 TI 在制造时被永久配置或者它们也可由用户设计。 为了确定您的 MCU 上所支持的电源域，请阅读特定器件数据表。 对于编程信息，请参阅特定器件技术参考手册。

5.2.1 电源状态控制器 (PSCON) 的锁步

对于每一个被执行的电源轨，一个电源状态控制器(PSCON) 执行控制逻辑电路。 在赫丘利斯平台内，每一个 PSCON 与一个锁步内的诊断 PSCON 一起执行以检验一个逐周期基础上的正确电源状态控制。 锁步比较检测到的所有错误被以信号的形式告知 ESM。

在加电复位状态期间，PSCON 锁步比较默认被启用。 为了测试比较的功能性，PSCON 锁步比较可由软件禁用。 锁步比较特性的测试由一个软件触发的硬件自检支持。 硬件自检测试能够针对匹配和不匹配输入启动比较测试。 也可用错误强制来测试系统级错误响应。 PSCON 锁步的使用是强制的。

5.2.2 用于控制寄存器的特权模式访问和程序序列

PMM 的设计包括支持无意识控制寄存器编程避免的特性。 这些特性包括多重寄存器写入序列来配置电源域以及写入命令限制来优先处理总线主控事务。 一个更高级的总线主控过滤保护阻止除 CPU 之外的任一总线主控的访问。 非法事务会导致一个到违反总线主控的总线错误响应。

这个安全机制的运行是连续的并且不能由软件变更。 这个机制可以由不正确事务的软件初始化测试 这个安全机制的使用是强制的。

5.2.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.2.4 写入配置的软件回读

为了在电源管理模块中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确运行。 为了支持这个软件测试，强烈建议将电源管理模块内存空间配置为一个使用 Cortex-R4F 内存保护单元 (MPU) 的严格订购的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.2.5 注释

- PSCON 在锁步比较自检期间继续正常运行，但是比较功能不会出现。
- 当 CPU 处于一个暂停调试状态，不执行 PSCON 输出比较。

5.3 时钟

赫丘利斯器件系列产品主要为同步逻辑器件并且同样要求用于正确运行的时钟信号。 时钟管理逻辑电路包括时钟源、时钟生成逻辑电路，此逻辑电路包括锁相环路 (PLL) 的时钟倍乘、时钟分配器、和时钟分配逻辑电路。 这些被用于设计时钟管理逻辑电路的寄存器位于系统模块内。

5.3.1 低功耗振荡器时钟检测器 (LPOCLKDET)

低功耗振荡器时钟检测器 (LPOCLKDET) 是一个可被用于检测主时钟振荡器故障的安全诊断。 LPOCLKDET 采用嵌入式高频、低功耗振荡器 (HF LPO)。 时钟检测电路工作方式为检验一个其它时钟上升沿之间的某一个时钟（振荡器或者 HF LPO）上的上升沿。 结果就是除了标记不正确、频率重复，电路也会由于瞬态情况发生故障。 时钟检测窗口的低端会在至少 12 个 HF LPO 周期内忽略一个瞬态低相位。 请注意，这个瞬态响应的过滤不会改变输入频率范围。

加电复位状态期间，LPOCLKDET 电路默认被启用。 此诊断可通过软件禁用。 强烈建议使用 LPOCLKDET。

5.3.2 PLL 差异检测

PLL 逻辑电路包括一个能够检测一个 PLL 输出时钟差异的嵌入式诊断。 差异是由基准时钟和反馈时钟间的相位锁定损失造成。 错误响应和指示取决于系统模块内的 PLL 控制寄存器的设计。 ESM 错误指示可被生成或者被屏蔽。 此外，万一检测到一个错误，则有可能生成一个内部复位或者从振荡器时钟返回运行状态。 要获得更多与设计这个诊断相关的信息，请参阅特定器件技术参考手册。

只要 PLL 被启用，PLL 差异检测诊断被激活并锁定在一个目标频率上。 此诊断不能由软件禁用，但是错误指示和错误响应可由软件修改。 强烈建议使用 PLL 差异检测诊断。

5.3.3 双时钟比较器 (DCC)

一个或者多个双时钟比较器 (DCC) 被使用为多用途安全诊断。 DCC 可被用于检测不正确频率和时钟源之间的漂移。 DCC 由两个计数器块组成：一个计数器块被用作一个基准时基而另外一个被用作测试时钟。 基准时钟和处于测试中的时钟均可由软件进行选择，可作为时钟频率的预计比率。 与预计比率的偏差会生成一个错误指示到 ESM。 与所执行的时钟选择选项相关的更多信息，请参阅特定器件数据表。 对于 DCC 设计细节，请见技术参考手册。

默认情况下，DCC 诊断并不启用而必须由软件启用。 可通过软件将这个诊断禁用并对其进行配置。 强烈建议将 DCC 用作一个对时钟的诊断。 DCC 模块所采用的循环校验提供了一个自我校验的固有电平（自动覆盖），可考虑将此电平应用于延迟故障诊断中。

5.3.4 外部时钟输出监控 (ECLK)

赫丘利斯平台提供将经选择的内部时钟信号输出用作外部监控的功能。 通过编辑系统模块内的寄存器，可由软件对此特性进行配置。 要确定所执行的外部时钟输出的数量和与可被输出的内部时钟相映射的寄存器，请参阅特定器件数据表。

默认情况下，ECLK 输出上内部时钟的输出并不启用而必须由软件启用。 可通过软件将这个诊断禁用并对其进行配置。 可选择将 ECLK 特性用于对内部时钟的外部监控。

5.3.5 内部安全装置

赫丘利斯平台支持一个在实时中断 (RTI) 模块中实现的内部安全装置。 内部安全装置有两个运行模式：数字式安全装置 (DWD) 和数字窗口模式安全装置 (DWWD)。 这两个运行模式互相排斥，设计人员可选择使用其中的任一模式，但不能同时使用。 对于此内部安全装置的编程细节，请见特定器件技术参考手册。

DWD 是一个传统的单阀值安全装置。 用户为安全装置设定一个超时值并且必须在超时计数器终止前提供一个到安全装置的预先确定的“第一”响应。 超时计数器的终止或者一个不正确的“第一”响应会触发一个错误响应。 在检测到一个故障时，DWD 能够发布一个内部（热）系统复位或者一个 CPU 非屏蔽中断。 复位后，DWD 不启用。 一旦由软件启用，除了系统复位或者加电复位，DWD 不能被禁用。 DWD 功能的使用是可选的。

DWWD 是一个传统的窗口式安全装置。 用户设定一个上界和下界来创建一个时间窗口，在此期间软件必须提供一个到安全装置的预先确定的“宠物”响应。 时间窗口期间内正确响应接收故障或者一个不正确的“宠物”响应会触发一个错误响应。 在检测到一个故障时，DWWD 能够发布一个内部（热）系统复位或者一个 CPU 非屏蔽中断。 复位后，DWWD 不启用。 一旦由软件启用，除了系统复位或者加电复位，DWWD 不能被禁用。 与 DWD 工具相比，时间窗口的使用可实现额外时钟故障检测模式。 建议使用 DWWD 功能。

5.3.6 外部安全装置

当使用一个外部安全装置时，由于安全装置能够采用与被监控的系统分离的时钟、复位、和功率，有可能使用 MCU 时钟系统来减少共模故障。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的外部安全装置来定义。 赫丘利斯平台强烈建议使用一个外部安全装置而非内部提供的安全装置。

5.3.7 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.3.8 写入配置的软件回读

为了在系统模块中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确运行。 为了支持这个软件测试，强烈建议将系统模块内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.3.9 注释

- 可以通过使用一个 TI 为赫丘利斯系列开发的系统基础芯片来简化对系统级上外部安全装置功能的管理。
- 用户能够通过编辑 HF LDO 中的调整值来改进 LPOCLKDET 诊断的精度。 这将要求客户在制造测试期间通过与一个经校准的时钟源进行比较来确定 LPO 调整值。
- 有很多安全装置工具可被用来提供时钟和 CPU 诊断。 总的来说，由于能够减少共模故障，TI 建议使用一个外部安全装置而非内部安全装置。 由于额外的故障模式（此故障模式可被一个更高级的安全装置检测出来），TI 建议使用一个与单一阀值安全装置相对的程序序列、窗口式的、或者问答式的安全装置。
- 在 ECLK 引脚上驱动一个高频时钟输出有可能会产生电磁干扰 (EMI) 。

5.4 复位

赫丘利斯器件系列产品需要一个外部冷启动复位和加电复位 (nPORRST) 来将所有异步和同步逻辑电路置于一个已知状态。 作为启动过程的一部分，加电复位会生成一个内部热启动 (nRST) 信号来将大多数数字逻辑电路复位。 nRST 信号在器件级上作为 I/O 引脚提供；当被内部置为有效时，此信号将接通并可由外部驱动来生成一个热复位。 有关复位功能的更多信息，请见特定器件数据表。

5.4.1 热复位的外部监控 (nRST)

nRST 热复位信号被执行为一个 I/O。可使用一个外部监控器来检测对内部热复位控制信号状态的预期的或者意外改变。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的外部监控器来定义。 这一特性的使用是可选的。

SPNU511 — http://www-s.ti.com/sc/techlit/SPNU511

5.4.2 最后一次复位原因的软件检查

此系统模块提供了一个状态寄存器 (SYSESR)，此寄存器锁存大多数近期复位事件的原因。 一个检查这个寄存器的状态以确定最近一次复位事件原因的启动软件通常由软件开发人员执行。 这些信息可被软件用来管理故障恢复。 强烈建议使用 SYSESR 来检查最近一次复位的原因。

5.4.3 软件热复位生成

此系统模块为软件提供生成一个内部热复位 (nRST) 的功能。 这通过在 SYSECR 控制寄存器中写入适当的控制位来完成。 软件可以利用这一特性来尝试故障恢复。 软件热复位的使用是可选的。

5.4.4 nRST 和 nPORRST 上的毛刺脉冲过滤

毛刺脉冲滤波器在器件的冷复位和热复位引脚上生效。 这些构造过滤出了输入复位引脚上的噪声和瞬态信号峰值以减少复位电路的意外激活。 毛刺脉冲滤波器持续运行并且它们的运行方式不能由软件改变。 毛刺脉冲滤波器的使用是强制的。

5.4.5 影子寄存器

在器件上使用一个二级冷复位和热复位机制可允许影子寄存器的执行。 影子寄存器只在加电复位时被复位。 这些寄存器可被用于存储器件状态或者其它关键信息，这些信息在系统状态被热复位改变之后仍然保持。 此系统模块包括影子寄存器，通过软件，此寄存器可被用于支持故障恢复。 强烈建议启动软件使用影子寄存器状态信息。

5.4.6 外部安全装置

一个外部安全装置可提供二级诊断。 要获得此类诊断的更多信息，请见 外部安全装置。

5.4.7 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.4.8 写入配置的软件回读

为了在系统模块中确保内存映射复位控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确运行。 为了支持这个软件测试，强烈建议将系统模块内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.4.9 注释

- 可以通过使用一个 TI 为赫丘利斯系列开发的系统基础芯片来简化对系统级上复位的管理。
- 由于受监控的复位信号与内部安全装置相互作用，内部安全装置不是一个用于复位诊断的可行选项。

5.5 系统模块

系统控制模块包含到接口时钟、复位、和其它系统相关控制和状态逻辑电路的内存映射寄存器。 此系统控制模块也负责生成系统复位的同步并传递系统热复位 nRST。

5.5.1 优先模式访问和多位使能密钥

系统模块设计包括一些特性以支持意外控制寄存器编程避免。 这些特性包括限制写入命令以优先处理总线主控事务处理和执行用于关键控制的多位密钥。 此多位密钥对于意外激活避免特别有效。 要获得更多寄存器安全机制和错误响应方面的信息，请见特定器件技术参考手册。

这个安全机制的运行是连续的并且不用由软件变更。 可通过生成软件事务且检查器件响应来测试这个机制。 这个安全机制的使用是强制的。

5.5.2 写入配置的软件回读

为了在系统模块中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确运行。 为了支持这个软件测试，强烈建议将系统模块内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.5.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.5.4 注释

- 根据目标标准，用户能够选择在系统模块中执行一个针对静态配置寄存器的定期软件测试。 这个测试能够提供一个针对由软件错误引起的中断的附加诊断覆盖。
- 检查时钟和复位部分，这是因为这些特性由系统模块严密控制。

5.6 错误信令模块 (ESM)

ESM 提供板载硬件诊断错误的统一集合和优先级排序。 更多信息请见 错误管理

5.6.1 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.6.2 错误路径报告的软件测试

一个软件测试可被用于注入诊断错误并检验报告的正确性。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用一个错误路径报告的引导时间软件测试。 建议使用一个错误路径报告定期软件测试。

5.6.3 影子寄存器

在器件上使用一个二级冷复位和热复位机制可允许影子寄存器的执行。 影子寄存器只由加电复位来复位。 这些寄存器可被用于存储器件状态或者其它关键信息，这些信息在系统状态被热复位改变之后仍然保持。 错误信令模块包括影子寄存器，此寄存器可被用于通过软件来支持故障恢复。 强烈建议由启动软件使用影子寄存器状态信息。

5.6.4 写入配置的软件回读

为了在 ESM 中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确运行。 为了支持这个软件测试，强烈建议将 ESM 内存空间配置为一个使用 Cortex-R4F 内存保护单元 的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.6.5 注释

- ESM 错误路径的软件测试可与硬件诊断的引导时间延迟测试组合在一起以减少引导时间。
- 对 ESM 错误路径的测试用可能导致 nERROR 诊断输出生效。 系统集成人员应该确保系统能够管理或者从 nERROR 事件中恢复

5.7 CPU 子系统

赫丘利斯产品系列依赖 ARM Cortex-R4F CPU 来提供通用处理。 Cortex-R4F 是带有嵌入式安全诊断的高性能 CPU。 R4F 也被设计用于轻松整合到一个 1oo1D 锁步配置中。 这些方面使 Cortex-R4f 成为一个用于功能安全产品的性能优良的 CPU。

5.7.1 锁步 CPU 诊断

赫丘利斯产品系列包括一个锁步处理器诊断。 这个特性包括一个增加的与应用 CPU 组装到一个 1oo1D（带有诊断通道的单通道）配置中的诊断 Cortex-R4F CPU。 应用 CPU 和诊断 CPU 处理一样的输入信号，这使得两个 CPU 运行一样的软件。 诊断 CPU 和应用 CPU 应生成一样的输出。 内核比较模块 (CCM) 比较所有与 ESM 误比较的 CPU 输出和标记。

从加电复位开始，锁步诊断持续运行。 当 CPU 被置于一个暂停调试状态时锁步检验被禁用并且只能在一个随后的复位后被恢复。 当执行 CCM 的自测检验功能的时候，锁步功能也可以被暂时禁用。 锁步诊断的使用是强制的。

在复位之后 CPU 的第一个周期期间，有必要执行一个短初始化代码来将所有 CPU 寄存器设定成一个已知状态。 这个代码序列可在特定器件数据表中找到。 必须在复位后将这个代码序列作为第一条指令执行。

通过软件触发的硬件，CCM 逻辑电路提供自检和错误强制功能。 这个自检确保 CCM 比较逻辑电路工作正常。 错误强制功能使您能够测试到一个锁存误比较的系统级响应。 强烈建议在复位时使用自检和错误强制。

5.7.1.1 缓解共模故障的措施

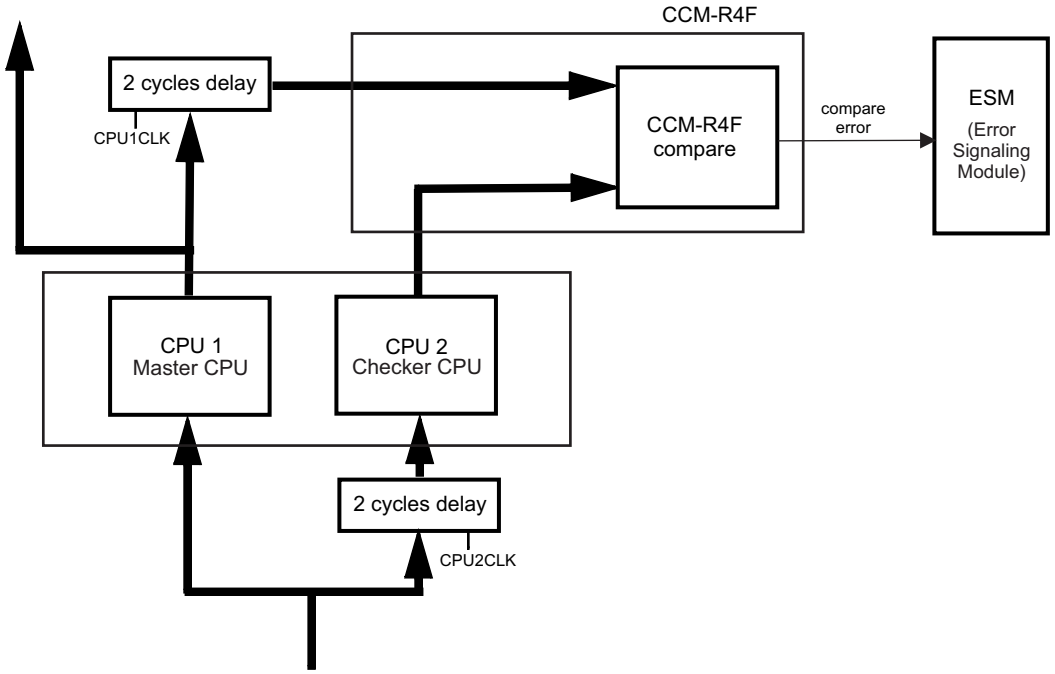
赫丘利斯锁步 CPU 子系统设计包括多个缓解共模故障的最佳做法：

- 物理多样性：
  - 物理核心硬核之间的距离至少 100µm 远。
  - 一个核心翻动和旋转的方向与其它核心相关。 例如，如果一个核心被认为位于“北方”，那么第二个核心将会如图 6中显示的那样“移动到西面”

图 6. 多种 CPU 物理定向

- 时间多样性：
  - 如图 7中所示，定时延迟块被插入以将 CPU 的运行推迟两个周期。

图 7. 锁步时间多样性



  - CPU 时钟域被分成两个时钟树，这样时钟通过两个独立的路径被传递到两个 CPU。
- 电源多样性：
  - 每个核心都有一个专用电源环。

5.7.2 CPU 逻辑内置自检 (LBIST) 自检控制器 (STC)

赫丘利斯系列架构支持硬件逻辑 BIST (LBIST) 引擎自检控制器 (STC) 的使用。 这个逻辑电路用于在晶体管级锁步 CPU 上提供一个非常高的诊断覆盖。 为了快速执行高质量的制造测试，这个逻辑电路采用被插入器件的一样的测试设计 (DFT) 结构，但是使用的是一个内部测试引擎而非外部自动测试设备 (ATE)。 这个技术的效率已经被证明远高于基于软件的逻辑测试，对于一个最新 CPU 中使用的复杂逻辑结构更是如此。

LBIST 测试必须由软件触发。 用户可以选择运行所有测试，或者根据可被分配给 LBIST 诊断的执行时间只运行这些测试的一个子集。 这个时间分片测试特性使得 LBIST 能够被高效用作一个运行时间诊断，此诊断可以根据安全关键环路执行测试时间片，也可以作为一个在 MCU 初始化期间对 CPU 故障的综合性测试。

由于此测试的高效率，LBIST STC 的执行会在每个时钟周期内引起比正常软件执行期间高很多的晶体管开关电平。 STC 内执行的软件控制使得用户能够在测试期间减少 CPU 时钟。 这一特性使用户能够在流耗更高的快速执行或者流耗较小的较慢速执行之间做出折中的选择。

测试时，LBIST 机制要求 CPU 从器件逻辑的剩余部分隔离。 在执行 LBIST 之前，还需执行一个完整环境保存。 当测试执行完成时，CPU 将被复位。 器件逻辑的其余部分将继续正常运行。 CPU 复位后，软件应该读取系统模块 SYSESR 来识别复位的原因并可随后恢复 CPU 环境。

LBIST 逻辑包括对诊断正常运行进行测试的功能。 由于对诊断的测试时间是确定的，一个能够检测故障的超时计数器也被包括在内以使测试能够在预计的时间内完成。 此外，有可能强制生成一个输入错误来在系统级检验错误检测和错误响应的传播。 这个测试的执行步骤如下：

- 启用 STCSTSCR 寄存器中的 self_check_key 和 fault_ins 位。
- 启用 STC 测试间隔零并执行复位

- 一旦测试完成，应该将 STC 全局状态寄存器中的故障位置为 1。
- 禁用 STCSTSCR 寄存器中的 self_check_key 位和 fault_ins 位中的一个或者全部。
- 通过对 STCGCR 中的位 0 进行编程来重新启动自检，这将使自检重新启动。
- 一旦测试完成，应该将 STC 全局状态寄存器中的故障位置为 0。

强烈建议在启动时使用 LBIST 逻辑。 在正常执行期间定期执行 LBIST 逻辑电路是可选的。 LBIST 模块采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.7.3 CPU 内存保护单元 (MPU)

Cortex-R4F CPU 的赫丘利斯工具包括一个 MPU。 MPU 逻辑可被用于提供器件内存中软件任务的空间分离。 根据每一个任务的需求，操作系统控制 MPU 并改变 MPU 设置。 违反一个已设置的内存保护策略会导致一个 CPU 异常中断。 强烈建议使用 MPU。

MPU 也可被用于配置内存系统的内存排序策略。 默认情况下，所有外设访问是严格排序的-这意味着所有在序列中完成的事务被终止并且没有写入事务被缓冲。 如果需要，操作系统可以配置到器件的访问-这意味则写入被缓冲。 这样可以改进一个严格排序模型的性能，此改进是以损失一些确定性为代价的。 强烈建议系统模块和其它被认为具有关键配置的模块被设定为严格排序访问模型。

由于 MPU 处于 CPU 内核内部，正常运行由锁步 CPU 机制进行检验。 此外，当执行 CPU 测试时，LBIST STC 诊断提供了一个 MPU 检验。 附加的基于软件的对 MPU 正常运行的测试和错误响应是可选的。

5.7.4 使用性能监控单元的在线参数描述

Cortex-R4F CPU 包括一个性能监控单元 (PMU)。 这个逻辑电路用于调试和代码参数描述目的，但是它也可被用作安全机制。 PMU 包括一个 CPU 周期计数器以及三个附加计数器，这些计数器可被设定为不同的 CPU 事件计数。 对于一个可被监控的 CPU 事件完整列表，请见《*Cortex-R4 和 Cortex-R4F 技术参考手册*》，此文档位于http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0363e/index.html。 可被监控的 CPU 事件示例包括：

- CPU 中的诊断检测到的 ECC 和奇偶错误的周期数量
- CPU 处于活锁状态的周期数量
- 已经执行的指令数量
- 采取一个例外的周期数量

借助于这些可用的信息，有可能生成一个软件例程来定期检查 PMU 计数器的值并将这些值与正常运行期间内预计的参数相比较。 默认情况下，PMU 不启用且必须通过软件进行配置。 由于 PMU 在 CPU 内部执行，它的正常运行由锁步诊断在逐周期基础上进行检查，也可通过执行 LBIST STC 诊断来进行检查。 将 PMU 用于在线诊断参数描述是可选的。

5.7.5 内部和外部安全装置

一个内部或者外部安全装置能够提供二次诊断。 对于这些诊断的更多信息，请见 内部安全装置或者 外部安全装置。

5.7.6 无效操作和指令陷阱

Cortex-R4F CPU 包括针对无效操作的诊断和可被用作安全机制的指令。 很多此类陷阱在复位后不启用且必须由软件配置。 强烈建议安装软件句柄以支持硬件无效操作和指令陷阱。 CPU 无效操作和指令陷阱的示例包括：

- 无效指令
- 浮点下溢和溢出
- 浮点除零

- 优先级违反

5.7.7 写入配置的软件回读

为了确保 CPU 协处理器控制寄存器的正常配置，强烈建议软件执行一个测试来确定所有控制寄存器写入的正常运行。 CPU 控制寄存器并不是内存映射的且必须通过 CPU 协处理器读写命令进行访问。

5.7.8 注释

- 相对于诸如 TI LBIST STC 硬件机制，很多安全微控制器采用一个 CPU 功能性的基于软件的测试。 TI 不建议在诸如 Cortex-R4F 的中等或者高级复杂程度的 CPU 上执行这些测试。 与等效的 LBIST STC 解决方案相比，基于软件的选项具有较高的内存成本、较低的检测能力、和更长的执行时间。
- 由于锁步诊断，无需执行一个 CPU 控制寄存器配置的定期测试。 一个 CPU 中的控制寄存器干扰不应该影响第二个 CPU。

5.8 初级嵌入式闪存

初级嵌入式闪存存储器是一个非易失性内存，此内存与 Cortex-R4F CPU 内核的 ATCM 端口紧密耦合。 虽然也可进行数据访问，但是 ATCM 闪存存储器主要用于 CPU 指令访问。 对于闪存存储器的访问要经历多个 CPU 周期。 一个闪存逻辑提供多个管道式读缓冲来改进连续地址提取方面的 CPU 访问时间。

5.8.1 闪存 ECC

片载闪存存储器由单纠错、双纠错 (SECDED) 误差校正代码 (ECC) 诊断支持。 它通过一个 64 位宽的数据总线接口 (ATCM) 被连接至 Cortex-R4F CPU。 在这个 SECDED 机制中，一个 8 位代码字被用于当 ECC 数据在 64 位数据总线上进行计算时存储该数据。

用于 ATCM 闪存访问的 ECC 逻辑位于 Cortex-R4F CPU 内。 所有 ATCM 事务处理在数据有效载荷上带有 ECC。 ECC 评估由 CPU 内部的 ECC 控制逻辑电路完成。 这个机制提供 CPU 和闪存存储器间传输的端到端诊断。 检测到的无法校正的错误会导致一个处理器异常中断或者总线错误，这取决于请求的主控。 检测到的能够校正的错误可以选择校正或者不校正此错误，这取决于校正功能是否被启用。 包括 ECC 错误的内存的地址记录在 CPU 内。 更多信息，请见《*Cortex-R4 和 Cortex-R4F 技术参考手册*》此手册位于http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0363e/index.html。

可将错误检测事件从 CPU 输出到闪存包装程序，然后从闪存包装器输出到 ESM。 默认情况下，这一功能性不启用并且必须由软件配置。 Cortex-R4F PMU 必须首先被设定为输出事件到一个外部监控器。 然后闪存包装程序必须被配置为将可校正的和不可校正的事件输出到 ESM。

用于闪存的 ECC 逻辑在复位时被禁用并且必须在 CPU 和闪存包装程序内被配置。 诊断在系统控制协处理器中有用于检验、校正、和读取、修改以及写入功能的独立控制，这些控制必须由软件启用。 由于 ECC 诊断在 CPU 内部执行，所有它的运行状态可由锁步功能性持续检验，也可由 LBIST STC 测试。 强烈建议使用闪存 ECC。 ECC 模块采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.8.2 硬错误高速缓存和活锁

如果 ECC 校正被启用，被校正的数据值被存储在一个内部单入口硬错误高速缓存中。 由于闪存是有特殊编程要求的非易失性内存，所以不能自动将经校正的数据值重写入闪存。

一个单一指令和它的数据不能有多于一个可校正错误。 在检测到多于一个可校正错误的情况下，有可能会使硬错误高速缓存过载并使处理器处于一个不可用的活锁状态。 生成一个活锁的情况包括：

- 一个 64 位非对齐 32 位 Thumb-2 取指令中的两个单一位错误
- 指令数据有效载荷中的一个单一位错误之前的载入指令中的单一位错误 (LDR 或者 LDM)。

活锁由 ESM 标出并且通常要求尝试执行一个针对恢复的复位。 一个闪存接口事务处理上的活锁可以是一个闪存存储器中严重永久性故障的标志。

硬错误高速缓存和活锁功能性的使用是强制的。 这一特性在复位时被启用并且不能由软件禁用。

5.8.3 闪存包装程序地址 ECC

除了 CPU 中的 ECC 功能性，闪存包装程序将 ECC 的功能延伸至包含地址。 采用的标准 SECDED ECC 系统有用于 64 位数据的 8 位编码。 海明等式的扩展可使超过 64 位的数据的附加位被使用同样的海明距离编码成为 8 位代码。 通过将事务处理的 TCM 总线地址组合到闪存存储器 ECC 计算中，闪存包装程序设计利用了这一优势。 所有存储在闪存存储器中的值含有被添加到闪存 ECC 中的地址。 当从闪存读取数据时，闪存包装程序将把地址组件从 ECC 中剥去并将重新生成的 ECC 代码提供给 CPU。 地址中的错误会导致 CPU 内的一个多位 ECC 故障。

复位时地址 ECC 特性被禁用。 没有针对此特性的独立控制。 只要 ECC 在闪存包装程序中被启用，这个特性就被启用。 强烈建议使用这一特性。 闪存包装程序地址 ECC 模块所采用的循环校验提供了一个自我校验的固有电平（自动覆盖），可考虑将此电平应用于延迟故障诊断中。

5.8.4 ATCM 地址总线奇偶校验

到闪存存储器的片载 ATCM 总线连接由一个地址信号上的奇偶诊断支持。 此奇偶诊断校验由 CPU 生成并由闪存包装程序进行评估。 检测到的错误由闪存包装程序信号传输给 ESM 并且在闪存包装程序内对错误地址进行捕捉。

这个诊断在复位时被启用。 这个诊断可通过对闪存包装程序 FPAR_OVR 寄存器内的 BUS_PAR_DIS 码的编程禁用。 强烈建议使用 ATCM 地址总线奇偶诊断。

5.8.5 硬件冗余校验码 (CRC) 闪存内容检查

这个平台包括一个硬件循环冗余校验 (CRC)，此校验执行 ISO CRC-64 标准多项式。 通过计算一个针对所有闪存内容的 CRC 并将得出的值与一个之前生成的“极佳” CRC 相比较，此 CRC 模块能被用于测试闪存内容的完整性。 读取到 CRC 的闪存内容可由 CPU 或者 DMA 来完成。 结果比较、故障指示、和故障响应由管理此测试的软件负责。 强烈建议在启动时执行一个闪存内容的 CRC 完整性检查。 建议在运行时间内定期执行 CRC 完整性检查。 硬件 CRC 模块所采用的循环校验提供了一个自我校验的固有电平（自动覆盖），可考虑将此电平应用于延迟故障诊断中。

5.8.6 闪存存储器阵列中的位复用

赫丘利斯架构中执行的闪存模块执行一个位复用机制，这样被存取用来生成一个逻辑 (CPU) 字的位在物理上不相邻。 这一机制有助于减少会导致逻辑多位故障的物理多位故障的可能性；相反的它们多表现为多个单一位故障。 由于 SECDED 闪存 ECC 不能校正一个逻辑字中的单一位故障，这个机制提高了闪存 ECC 诊断的有效性。 位复用是此架构的强制特性并且不能由软件更改。

5.8.7 闪存扇区保护

通过闪存包装程序的软件配置可以防止扇区上的写入操作。 扇区保护寄存器，组扇区使能寄存器 (BSE)，包含一个针对闪存组中每一个扇区的位，这个位能够启用或者禁用对扇区的写入操作。 BSE 寄存器只能在特权模式下被写入，同时软件 PROTLIDIS 保护位被置为高电平。 这一机制能够减少闪存存储器意外编程的可能性。 强烈建议使用闪存扇区保护特性。

5.8.8 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.8.9 写入配置的软件回读

为了在闪存包装程序中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将闪存包装程序内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.8.10 注释

- 当从 CPU 输出 ECC 错误事件到闪存包装程序并从闪存包装程序输出到 ESM 时，应该小心，错误不是由被丢弃的预取或者其它分支不连续性造成的。
- 通过执行一个闪存内容的 CRC 回读，同时在闪存上启用 ECC，就有可能并行执行闪存的两个诊断。
- 根据产品配置的不同，闪存模块也许会有唯一的电源引脚。 强烈建议如电源部分描述的那样在这些引脚上执行电压监控。
- 根据产品配置的不同，闪存模块也许会有唯一的测试信号引脚。 对于这些信号的正确板级管理，请见特定器件数据表。

5.9 闪存 EEPROM 仿真 (FEE)

赫丘利斯平台架构包括将闪存存储器中的一个独立组用作闪存仿真 EEPROM (FEE) 的功能。 FEE 只被用于数据存储，而不能用作一个 CPU 指令内存。 FEE 是一个二级内存，CPU 可通过 AXI 主控端口（此端口与用于主闪存组的紧密耦合内存接口相对）对此内存进行访问。 FEE 内存中的 EEPROM 仿真由运行在 CPU 上的特定 FEE 驱动器进行管理。

5.9.1 FEE 数据 ECC

片载 FEE 内存由 SECDED ECC 诊断支持。 与主闪存存储器不同，FEE 内存使用一个在闪存包装程序内执行的本地 SECDED ECC 控制器。 这样可以实现 EEPROM 仿真支持所需的额外灵活性，但是也确实使得 CPU 和 FEE 内存间的完全事务路径没有端到端诊断。

FEE SECDED ECC 控制器使用与主闪存存储器内一样的 ECC 算法；数据的每 64 位执行 8 位代码。 所有 ECC 故障的检测在闪存包装程序内部执行。 错误响应通过总线错误提供给 CPU 和一个错误信号提供给 ESM。 错误响应有附加的可编程性以支持由 TI 用户设计的多重 EEPROM 仿真策略。 故障地址记录在闪存包装程序内。

用于 FEE 的 ECC 逻辑电路在复位时被禁用并且可在闪存包装程序内配置。 强烈建议使用 FEE ECC。 FEE ECC 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.9.2 硬件CRC FEE 内容检查

这个平台包括一个硬件 CRC，此校验执行 ISO CRC-64 标准多项式。 通过计算一个针对所有 FEE 内容的 CRC 并将得出的值与一个之前生成的“黄金” CRC 相比较，此 CRC 模块能被用于测试 FEE 内容的完整性。 读取到 CRC 的 FEE 内容可由 CPU 或者 DMA 来完成。 结果比较、故障指示、和故障响应由管理此测试的软件负责。 根据用于支持 FEE 的软件驱动器机制的不同，也许有必要执行这个由 CPU FEE 驱动器驱动的检查。 强烈建议在启动时执行一个 FFE 内容的 CRC 完整性检查。 建议在运行时间内定期执行 FEE CRC 完整性检查。 硬件 CRC 模块所采用的循环校验提供了一个自我校验的固有电平（自动覆盖），可考虑将此电平应用于延迟故障诊断中。

5.9.3 FEE 中的位复用

赫丘利斯架构中执行的 FEE 模块执行一个位复用机制，这样被存取用来生成一个逻辑 (CPU) 字的位物理上不相邻。 这一机制有助于减少会导致逻辑多位故障的物理多位故障的可能性；相反的它们多表现为多个单一位故障。 由于 SECDED FEE ECC 不能校正一个逻辑字中的单一位故障，这个机制提高了闪存 ECC 诊断的有效性。 位复用是此架构的强制特性并且不能由软件更改。

5.9.4 FEE 扇区保护

通过闪存包装程序的软件配置可以防止 FEE 扇区上的写入操作。 扇区保护寄存器，BSE，包含一个针对 FEE 组中每一个扇区的位，这个位能够启用或者禁用对扇区的写入操作。 BSE 寄存器只能在特权模式下被写入，同时软件 PROTLIDIS 保护为被置为高电平。 这一机制能够减少 FEE 存储器无意编程的可能性。 强烈建议使用 FEE 扇区保护特性。

5.9.5 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.9.6 已写入配置的软件回读

为了在 FEE 包装程序中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将 FEE 包装程序内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.9.7 注释

- 由于所有应用 FEE 访问通过一个软件驱动器运行来管理 EEPROM 仿真，任何诊断必须遵守 FEE 驱动器的功能性。
- 根据产品配置的不同，FEE 模块也许会有唯一的电源引脚。 强烈建议如电源部分描述的那样在这些引脚上执行电压监控。
- 根据产品配置的不同，FEE 模块也许会有唯一的测试信号引脚。 对于这些信号的正确板级管理，请见特定器件数据表。

5.10 初级嵌入式 SRAM

初级嵌入式 SRAM 是一个非易失性内存，此内存与 Cortex-R4F CPU 内核的 BTCM 端口紧密耦合。 虽然也可用于指令存取，BTCM SRAM 主要用于 CPU 数据存取。 SRAM 的存取时间比闪存存储器快很多，这样在最大 CPU 频率时无需等待状态。

执行两个 64 位 BTCM 接口：BTCM0 和 BTCM1。 CPU 可在一个周期内生成两个 BTCM 存取：一个在 BTCM0 上，另外一个在 BTCM1 上。 SRAM 寻址被插入到两个以 64 位为基础的 BTCM 接口之间，这样大大减少了多个 BTCM 主控（PFU，LSU，和 AXI-S）间的仲裁时间。

5.10.1 数据 ECC

片载 SRAM 由 SECDED ECC 诊断支持。 它被一个 64 位宽的数据总线接口（BTCM0 或者 BTCM1）连接至 Cortex-R4F CPU。 在这个 SECDED 机制中，一个 8 位代码字被用于存储在 64 位数据总线上计算出的 ECC 数据。

用于 BTCM SRAM 存取的 ECC 逻辑电路位于 Cortex-R4F CPU 内。 所有 BTCM 事务处理在数据有效载荷上具有 ECC。 ECC 评估由 CPU 内部的 ECC 控制逻辑电路完成。 这个机制在 CPU 和 SRAM 间的数据传输上提供端到端诊断。 检测到的不可校正的错误会导致处理器异常中断或者总线错误，这取决于请求主控。 检测到的能够校正的错误可以选择校正或者不校正此错误，这取决于校正功能是否被启用。 包括 ECC 错误的内存地址将被记录在 CPU 内。 更多信息，请见《*Cortex-R4 和 Cortex-R4F 技术参考手册*》此手册位于http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0363e/index.html。

可将错误检测事件从 CPU 输出到 SRAM 包装程序，然后从 SRAM 包装程序输出到 ESM。 默认情况下，这一功能性不启用并且必须由软件配置。 Cortex-R4F PMU 必须首先被设定为将事件输出到一个外部监控器。 然后，SRAM 包装程序必须被配置为将可校正的和不可校正的事件输出到 ESM。

用于 SRAM 的 ECC 逻辑电路在复位时被禁用并且可在 CPU 内进行配置。 诊断在系统控制协处理器中有用于检验、校正、和读取、修改以及写入功能的独立控制，这些控制必须由软件启用。 由于 ECC 诊断在 CPU 内部执行，所有它的运行状态可由锁步功能性持续检验，也可由 LBIST STC 测试。 强烈建议使用 SRAM ECC。 ECC 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.10.2 硬错误高速缓存和活锁

如果校正被启用，经校正的数据值被存储在一个单入口内部硬错误高速缓存，被重新写入到 SRAM，并从 SRAM 中重新取出。

一个单一指令和它的数据不能有多于一个可校正错误。 在检测到多于一个可校正错误的情况下，有可能会使硬错误高速缓存过载并使处理器处于一个不可用的活锁状态。 生成一个活锁的情况包括：

- 一个 64 位非对齐 32 位 Thumb-2 取指令中的两个单一位错误
- 指令数据有效载荷中的一个单一位错误之前的载入指令中的单一位错误 (LDR 或者 LDM)。

活锁由 ESM 标出并且通常要求尝试执行一个针对恢复的复位。 一个 SRAM 接口事务处理上的活锁可以是一个 SRAM 中的严重永久性故障的标志。

硬错误高速缓存和活锁功能性的使用是强制的。 这一特性在复位时被启用并且不能由软件禁用。

5.10.3 可校正 ECC 参数描述

SRAM 包装程序包括一个计算检测到的可校正 ECC 错误数量的功能。 当错误数量超过一个用户设定的阀值时，一个错误事件信号被发送给 ESM。

这个机制被默认启用并且必须由 SRAM 包装程序中的软件启用。 为了使这个功能正常运行，Cortex-R4F PMU 事件输出也必须被启用。 推荐使用可校正 SRAM ECC 参数描述特性。

5.10.4 BTCM 地址和控制总线奇偶校验

到 SRAM 的片载 BTCM 总线连接由地址和控制信号上的奇偶校验诊断支持。 此奇偶诊断由 CPU 生成并由 SRAM 进行评估。 检测到的错误由 SRAM 以信号告知 ESM 并且在 SRAM 包装程序内对错误地址进行捕捉。

这个诊断在复位时被启用。 这个诊断可通过对 SRAM 包装程序的 RAMCTRL 寄存器中的地址奇偶禁用键编程来禁用。 强烈建议使用 BTCM 地址和控制总线奇偶诊断。

5.10.5 SRAM 包装程序冗余地址解码

SRAM 包装程序包括一个用于检查地址解码逻辑中错误的诊断。 这个诊断在到功能地址解码逻辑的锁步中执行一个地址解码逻辑的检查工具副本。 SRAM 包装程序包括比较器逻辑来检测两个地址解码器输出间的差异。 任何检测到的不匹配将被信号传输至 ESM 并且此地址将被捕捉进一个本地寄存器。

复位之后，冗余地址解码逻辑诊断被激活。 这个诊断由软件触发的一个硬件自检支持。 自检运行期间，此比较功能被禁用。 强烈建议使用此复制的地址解码诊断。

5.10.6 按照逻辑地址，数据和 ECC 存储在多重物理组中

每一个逻辑 SRAM 字和与其相关的 ECC 代码被分开并存储于两个物理 SRAM 组中。 每个访问包含 72 位 - 64 位数据和 8 位 ECC 代码 这 72 位被分成相等的两部分，每个物理 SRAM 存储 32 位数据和 4 位 ECC 代码。图 8提供这个分区的图形视图。

这个系统配置提供一个针对物理 SRAM 组中地址解码故障的固有安全机制。 组寻址中的故障被 CPU 检测为一个 ECC 故障。 在 SRAM 中使用多种数据和 ECC 存储是强制的。 这个特性在复位时被启用且不能由软件禁用。

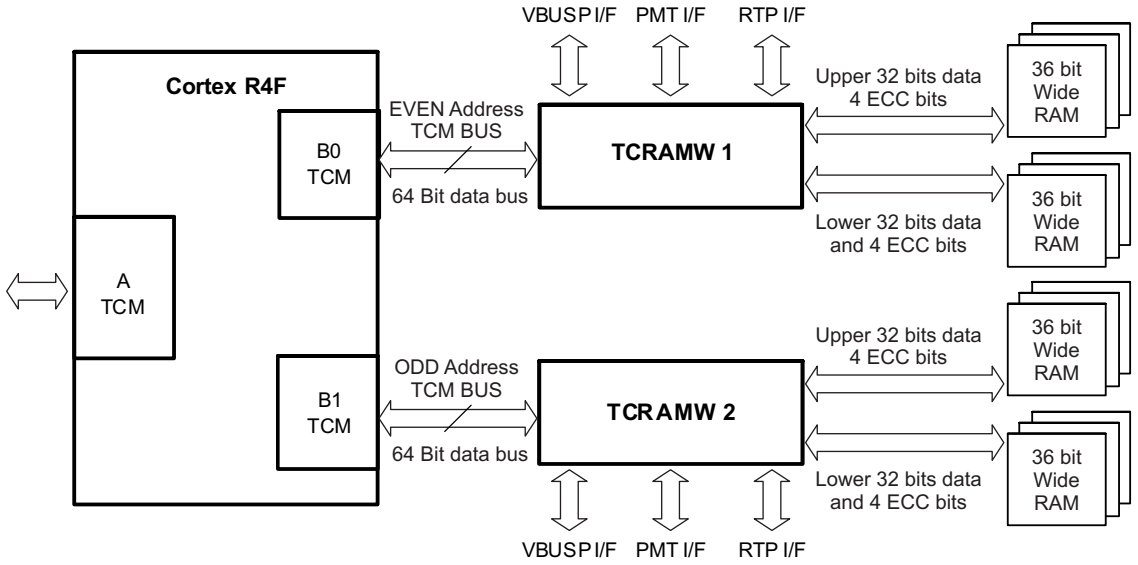


图 **8. CPU SRAM** 的程序块级实现

5.10.7 可编程内存 BIST (PBIST)

赫丘利斯系列架构支持使用一个硬件可编程内存 BIST (PBIST) 引擎。 在一个晶体管级上，这个逻辑电路被用于在已执行的 SRAM 上提供一个非常高的诊断覆盖。 这个逻辑电路采用与 TI 用来提供高质量制造测试的快速执行所使用的一样的测试设计 (DFT)逻辑电路和算法。 这一技术的效率以被证明远远高于 SRAM 的基于软件的测试，特别是对于具有复杂 CPU 的器件，在这样的器件中，寻址模式不会启用一个最优的基于软件的测试。

PBIST 测试必须由软件触发。 用户可以选择运行所有算法，或者根据可被分配给 PBIST 诊断的执行时间只运行这些算法的一个子集。 同样地，用户能够根据可被分配给 PBIST 诊断的执行时间，选择在一个 SRAM 或者一组 SRAM 上运行 PBIST。 PBIST 测试会破坏内存中的内容，正因如此，此测试通常只在 MCU 初始化时运行。 然而，当 CPU 可用时，用户可在任一时间启动这些测试。

由于此测试的高效率，PBIST 的执行会在每个时钟周期内引起比正常软件执行期间高很多的晶体管开关电平。 PBIST 内执行的软件控制使得用户能够在测试期间减少 SRAM 时钟。 这一特性使用户能够在流耗更高的快速执行或者流耗较小的较慢速执行之间做出折中的选择。

TI 已知的最有效 SRAM 测试要求在一个完全物理内存模块上进行测试并且会破坏之前的存储器内容。 如果要在运行期间执行 PBIST，建议在测试执行前将数据从将要测试的 SRAM 中复制到一个未经测试的内存中并在测试完成时恢复此数据。 当测试执行完成时，SRAM 可被用于正常运行。 SRAM 测试期间，器件逻辑电路的其余部分继续正常运行。 PBIST 检测到的任何故障会导致一个在 PBIST 状态寄存中标示出的错误。 也可将故障记录在 PBIST 逻辑电路中。

PBIST 逻辑包括对诊断正常运行进行测试的功能。 由于对诊断的测试时间是确定的，可通过 RTI（能检测一个故障）的软件编程来实现一个超时计数器，从而在预计的时间内完成此测试。 此外，通过运行一个使用 PBIST 引擎的校验和 ROM 测试也可测试包含 PBIST 算法的 ROM 内容。 错误强制可通过执行一个不指定目标内存的测试来完成，例如在 PBIST ROM 上执行一个 SRAM 读写测试。

强烈建议在器件初始化时使用 PBIST 逻辑。 在正常执行期间定期执行 MBIST 逻辑是可选的。 PBIST 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.10.8 SRAM 位复用

赫丘利斯架构中执行的 SRAM 模块执行一个位复用机制，这样的话，被存取用来生成一个逻辑 (CPU) 字的位在物理上不相邻。 这一系统配置有助于减少会导致逻辑多位故障的物理多位故障的可能性；相反的它们多表现为多个单一位故障。 由于 SECDED SRAM ECC 能够校正一个逻辑字中的单一位故障，这个机制提高了 SRAM ECC 诊断的有效性。 位复用是此架构的强制特性并且不能由软件更改。

5.10.9 SRAM 硬件 CRC-64

这个平台包括一个硬件 CRC，此校验执行 ISO CRC-64 标准多项式。 通过计算一个针对所有静态内容的 CRC 并将得出的值与一个之前生成的“极佳” CRC 相比较，此 CRC 模块能被用于测试静态内容的完整性。 读取到 CRC 的 SRAM 内容可由 CPU 或者 DMA 来完成。 结果比较、故障指示、和故障响应由管理此测试的软件负责。 由于大多数静态值被存储在闪存中，在 SRAM 的静态内容上执行一个 CRC 是可选的。 CRC 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.10.10 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.10.11 写入配置的软件回读

为了在 SRAM 包装程序中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将 SRAM 包装程序内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.10.12 注释

- 有两个 SRAM 包装程序，每个包装程序用于一个 BTCM 接口。 如果需要对诊断支持进行配置的话，请您确保在两个 SRAM 包装程序上都进行配置。
- 通过执行一个 SRAM 内容的 CRC 回读，同时在 SRAM 上启用 ECC，就有可能并行执行 SRAM 的两个诊断。
- 根据产品配置的不同，SRAM 模块也许会有唯一的电源引脚。 强烈建议如电源部分描述的那样在这些引脚上执行电压监控。
- 冗余地址解码不提供硬件容错。 这样的话，按照某些安全标准中冗余的定义，不认为此逻辑电路是完全冗余的。

5.11 2 级和 3 级（L2 和 L3）互连子系统

这个系统互连由一定数量的电桥、垫圈、通信、和路由逻辑电路组成，此系统互连将总线主控器件（DMA，CPU，TU 等）连接至 L2 受控器件和 L3外设。 虽然没有明确的安全目的，但是这个逻辑电路是一个安全通信传递中的媒介。

5.11.1 错误捕捉

L2 和 L3 互连子系统包括一些机制来检测和捕捉错误。 如果一个总线事务处理没有被解码为一个有效目标，诊断中的地址解码器使用一个总线错误对初始方进行响应。 逻辑电路也可检测特定事务处理的超时并且使用一个总线错误对事务处理初始方进行响应。

互连错误捕捉功能性默认被启用且不能被软件禁用。 这个安全机制的使用是强制的。 通过插入无效总线事务处理，可由软件对这些特性进行测试。

5.11.2 外设中央资源 (PCR) 访问管理

外设中央资源 (PCR) 提供两个能够限制到外设访问的安全机制。 根据 PCR 中的外设芯片选择，外设可被时钟选通。 这可被用于禁用未使用的特性，这样它们就不会干扰激活的安全功能。 此外，可对每一个外设芯片选择进行编程以限制基于事务处理优先级的访问。 这一特性可被用于将对于全部外设访问只限于特许操作系统代码。

复位后，这些安全机制被禁用。 软件必须配置且启用这些机制。 强烈建议使用这些机制。

5.11.3 内部/外部安全装置

一个内部或者外部安全装置可以提供一个事务的二级标示，此事务由于一个互连问题已经超时。 对于这些诊断的更多信息，请见 内部安全装置或者 外部安全装置。

5.11.4 信息冗余技术

信息冗余技术可由软件应用为一个 L2 和 L3 互连上的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 建议在 L2/L3 互连事务处理中使用信息冗余技术。

5.11.5 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.11.6 基本功能性的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 建议使用一个基本功能性报告定期软件测试。

5.11.7 写入配置的软件回读

为了在 PCR 中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将 PCR 内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.11.8 注释

- 在一个联网外设上执行端到端通信安全机制在 L2 和 L3 互连上提供了一个信息冗余诊断的间接形式。
- L2 L3 互连子系统中的唯一一个具有内存映射寄存器的模块为 PCR。

5.12 EFuse 静态配置

赫丘利斯平台器件通过一次性可编程 (OTP) EFus结构来支持一个特定功能性（诸如调整用于模拟宏的值）的制造时间配置。 在由一个自动载入功能执行的加电复位之后，EFuse 被自动读取。

5.12.1 自动载入自检

EFuse 控制器有一个自检逻辑电路，此电路在自动载入完成后自动执行。 错误由 ESM 标出。 此测试可随后由软件触发。 自动载入自检诊断的使用是强制的。

5.12.2 EFuse ECC

EFuse 采用一个 SECDED ECC 诊断来检测（并校正）不正确的配置值。 错误由 ESM 标出。 EFuse ECC 诊断的使用是强制的。 ECC 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.13 OTP 静态配置

赫丘利斯平台器件通过一次性可编程 (OTP) 闪存结构来支持一个特定功能性（诸如复位后的字节序和电源域的初始配置）的制造时间配置。 在由一个自动载入功能执行的热复位之后，OTP 配置的值被自动读取。

5.13.1 自动载入自检

OTP 自动载入控制器有一个自检逻辑电路，此电路在自动载入完成后自动执行。 错误由 ESM 标出。 此测试可随后由软件触发。 自动载入自检诊断的使用是强制的。

5.13.2 OTP 自动载入 ECC

OTP 自动载入控制器采用一个 SECDED ECC 诊断来检测（并校正）不正确的配置值。 错误由 ESM 标出。 OTP 自动载入 ECC 诊断的使用是强制的。 ECC 逻辑采用的循环校验提供了一个自检的固有电平（自动覆盖），可考虑将其应用在延迟故障诊断中。

5.13.3 注释

- 用于存储配置字的 OTP 闪存存储器可由 CPU 读取。

5.14 I/O 复用 (IOMM)

I/O 复用模式 (IOMM) 提供到器件引脚的内部模块 I/O 功能性的软件可配置映射。

5.14.1 针对控制寄存器的锁闭机制

IOMM 包含一个用于保护关键控制寄存器的二级锁闭机制。 为了改变引脚复用的配置，用户必须按照定义的顺序写入两个特定的 32 位值到“起始”寄存器。 当完成时，锁闭功能必须被复位。 当寄存器被锁闭时，写入访问将不会更新寄存器，它们也不会生成一个错误响应。

这个特性在复位后被启用。 通过使用解锁和不再重新锁闭，软件能够禁用此锁闭。 强烈建议使用针对控制寄存器的锁闭机制。

5.14.2 主控 ID 过滤

IOMM 检查所有进入的总线事务主控 ID。 只允许来自 CPU 的事务。 非法事务会导致一个违反总线主控的总线错误响应并且到 IOMM 的写入被丢弃。

这一特性在复位后备启用。 软件不能禁用这一特性。 总线 ID 过滤的使用是强制的。

5.14.3 错误捕捉

IOMM 能够捕捉地址和接收到的事务上的优先级错误。 试图访问 IOMM 芯片选择中的未生效的位置会导致一个 ESM 响应。 非特权模式中的事务也会生成一个 ESM 响应。

这一特性在复位后被启用。 软件不能仅用这个特性。 错误捕捉功能的使用是强制的。

5.14.4 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.14.5 使用 I/O 回送功能的软件测试

来自外设的模拟回送测试会导致信号穿过 I/O 引脚复用逻辑到达 I/O 焊垫并能够提供 I/O 引脚复用上的诊断覆盖。 模拟回送测试从模块到 I/O 单元的信号路径，此时输出驱动器被禁用。

I/O 回送在复位时不启用。 需要软件来配置和执行此诊断。 错误响应由软件管理。 强烈建议在启动时使用 I/O 回送机制。 定期使用 I/O 回送机制是可选的。

5.14.6 已写入配置的软件回读

为了在 IOMM 中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将 IOMM 内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.14.7 注释

- IOMM 的软件测试可与外设回送测试组合在一起。

5.15 矢量中断模块 (VIM)

矢量中断模块 (VIM) 被用于将外设中断连接至 Cortex-R4F CPU。 VIM 提供可编程中断优先级、屏蔽、和睡眠模式唤醒功能。 VIM 包括一个本地 SRAM，此 SRAM 被用于保持每个通道的中断句柄的地址。

5.15.1 VIM SRAM 奇偶校验

VIM SRAM 包括一个奇偶诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 VIM SRAM 奇偶校验特性。

5.15.2 VIM SRAM PBIST

VIM SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议复位后在 VIM SRAM 上执行 PBIST。 要获得这一诊断的更多信息，请见 PBIST。

5.15.3 VIM SRAM 位复用

VIM SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.15.4 VIM SRAM CRC-64 测试

VIM SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 VIM SRAM 的内容往往是静态的，建议使用这一诊断。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.15.5 VIM 运行的定期软件测试

依照 IEC 61508 中的指南，一个用于检测连续中断、无中断、和交叉中断的软件测试可以被实现。 这样的测试可包括指定期间内接收到的中断的预计平均数量的 PMU 参数描述，软件强制中断来检查 VIM 和 CPU 响应，或者来自 RTI 模块专门用于 VIM 测试目的的定期中断。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议使用对 VIM 运行的定期软件测试。

5.15.6 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.15.7 写入配置的软件回读

为了在 VIM 中确保内存映射控制寄存器的正确配置，强烈建议软件执行一个测试来确认所有控制寄存器写入的正确操作。 为了支持这个软件测试，强烈建议将 VIM 内存空间配置为一个使用 Cortex-R4F 内存保护单元的严格排序的、不可缓冲的内存区域。 这一配置在回读被启动之前确保寄存器写入完成。

5.15.8 内部和外部安全装置

一个内部或者外部安全装置可以提供一个事务的二级标示，此事务由于一个互连问题已经超时。 对于这些诊断的更多信息，请见 内部安全装置 外部安全装置。

5.15.9 注释

- 在对任一用于运行模式（遗留 IRQ 和 FIQ、遗留矢量化、或者完全矢量化硬件）的 VIM 软件诊断进行优化时应该小心。

5.16 实时中断 (RTI)

实时中断 (RTI) 模块提供针对器件的操作系统定时器。 OS 定时器被用于生成内部事件触发或者所需的中断来提供安全功能的定期运行。

5.16.1 使用第二个计数器作为诊断

RTI 模式包含至少两个上数计数器，此计数器可被用于提供操作系统时间记号。 当一个上数计数器被用作操作系统时基时，可使用第二个计数器作为第一个技术器的诊断，即通过软件对两个定时器中的计数器的值进行定期检查。 Cortex-R4F CPU 内部的 PMU CPU 周期计数器也可用于支持这样一个诊断。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 建议使用一个第二计数器来诊断 RTI 内的故障。

5.16.2 内部/外部安全装置

一个内部或者外部安全装置可以提供 RTI 模块内故障的标示。 对于这些诊断的更多信息，请见 内部安全装置 外部安全装置。

5.16.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.16.4 注释

- 当使用一个计数器作为操作系统时基计数器时，一个时钟源、比例因数等的多种配置可被用于减少共模故障的可能性。
- FlexRay 网络时间单元 (NTU) 可被用作用于操作系统定时器的时钟源。 这可被用于安全任务的网络同步。 如果 NTU 检测到与网络的同步损失，系统能够自动返回本地时钟源。

5.17 直接存储器存取 (DMA)

直接存储器存取 (DMA) 模块用于将数据从一个位置移到系统中的其它位置。 通常这一功能用于外设配置（闪存和 SRAM）和外设数据更新（外设缓冲器内存转移到 CPU SRAM 进行处理）。 为了提升系统总体性能，DMA 通常由操作系统用来从 CPU 卸载总线事务处理。 DMA 有一个用于通道控制信息的本地 SRAM。

5.17.1 内存保护单元 (MPU)

DMA 包括一个 MPU。 MPU 逻辑可被用于提供器件内存中软件任务的空间分离。 根据每一个任务的需求，DMA 驱动器控制 MPU 并改变 MPU 设置。 违反一个已设置的内存保护策略会导致一个 ESM错误。

复位时 MPU 不启用 软件必须启用、配置和测试 MPU。 强烈建议使用 MPU。

5.17.2 非特权总线主控访问

DMA 是一个非特权总线主控。 由于只有特权模式才能对 MCU 上的关键配置寄存器进行写入操作，这个机制通过 DMA 防止这些寄存器的无意配置。

这个安全机制的运行是连续的并且不用由软件变更。 可通过生成软件事务且检查器件响应来测试这个机制。 这个安全机制的使用是强制的。

5.17.3 信息冗余技术

使用 DMA 模块，可采用信息冗余技术。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 建议在 DMA 事务处理上执行信息冗余技术。

5.17.4 DMA SRAM 奇偶校验

DMA SRAM 包括一个奇偶校验诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 DMA SRAM 奇偶校验特性。

5.17.5 DMA SRAM PBIST

DMA SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议复位后在 DMA SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.17.6 DMA SRAM 位复用

DMA SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.17.7 DMA SRAM CRC-64 测试

DMA SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 DMA SRAM 的内容往往是静态的，建议使用这一诊断。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.17.8 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.17.9 基本功能性的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 建议使用一个基本功能性报告定期软件测试。

5.18 高端定时器 (N2HET)，HET 转移单元 (HTU)

N2HET 模块是一个具有输入/输出功能的可编程定时器。 N2HET 被执行为一个带有指令集（专门用于定时操作）的简单 RISC 处理器。 复杂输入可被捕捉并由 N2HET 进行预处理，随后由 CPU 处理。 输出生成通常为脉宽调制 (PWM)，但是也可以为简单通用输入/输出 (GIO) 类型信号。

每个 N2HET 有一个被称为 HTU 的专用微型 DMA 控制器。 HTU 提供了一个用于数据从 N2HET 和 CPU 内存移进移出的高带宽连接。

5.18.1 内存保护单元 (MPU)

HTU 包括一个 MPU。 MPU 逻辑可被用于提供器件内存中软件任务的空间分离。 根据每一个任务的需求，N2HET 驱动器控制 MPU 并改变 MPU 设置。 违反一个已设置的内存保护策略会导致一个 ESM 错误。

复位时 MPU 不启用 软件必须启用、配置和测试 MPU。 强烈建议使用 MPU。

5.18.2 信息冗余技术

信息冗余技术可被软件或系统应用为一个对 N2HET 运行的附件运行时间诊断。 有很多技术可被采用，诸如一个单一输入通道的多采样、两个或者更多通道的采样、和已写入输出的回读。 相对于在一个单一 N2HET 上执行一个功能和它的诊断，两个已执行的 N2HET 模块间的分离功能能够减少共模故障的可能性。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 N2HET 运行上使用信息冗余技术。

5.18.3 将 DCC 用作程序序列安全装置

在 N2HET 的此类设计中，指令内存在一个持续时间确定的环形回路中持续执行。 一个已知频率的被锁闭输出可在使用最小的软件开销的情况下在 N2HET 上生成。 每个 N2HET 有一个专用的通道，此通道被内部连接至两个 DCC 模块中的一个。 DCC 可以将 N2HET 时钟输出与一个已知时钟基准相比较以有效地在 N2HET 上执行一个程序序列。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。

强烈建议使用 DCC 作为一个 N2HET 上的程序序列。

5.18.4 第二 N2HET 的监控

为了使一个 N2HET 到第二 N2HET 的监控更加便利，N2HET 支持两个 N2HET 模块间的内部通道连接。 这一特性被支持为一个便利措施来限制器件级上功能通道的数量，此特性必须被用于诊断目的。 可使用例外的外部链接。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。

建议使用对 N2HET 运行的内部监控。

5.18.5 使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

N2HET 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 N2HET 功能性的任何测试应该包括 I/O 回路。

5.18.6 N2HET 和 HTU SRAM 奇偶校验

N2HET SRAM 包括一个奇偶校验诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 N2HET SRAM 奇偶校验特性。

5.18.7 N2HET 和 HTU SRAM PBIST

N2HET SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议在复位后在 N2HET SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.18.8 N2HET 和 HTU SRAM 位复用

N2HET SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.18.9 N2HET 和 HTU SRAM CRC-64 测试

N2HET SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 HTU SRAM 内容往往是静态的，而与之相对的 N2HET SRAM 的内容是动态，因此建议使用这一诊断并且这一诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.18.10 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.18.11 注释

- N2HET 上使用的信息冗余技术可被扩展至覆盖 HTU 总线主控操作。

5.19 多缓冲模数转换器 (MibADC)

MibADC 模块用于将模拟输入转换为数字值。 结果被存储在内部 SRAM 缓冲器内以用于之后的 DMA 或者 CPU 传递。 赫丘利斯器件系列产品执行两个带有共享通道的用于快速转换的模块（乒乓操作方法）。 使用双 ADC 转换器来执行两个通道的系统也许能够在应用中请求故障容错。

5.19.1 输入自检

赫丘利斯 MiADC 模块执行一个输入自检引擎，此引擎能够检测到 ADREFLO，ADREFHI 的短路或者开路输入。 软件必须配置、启用和评估这个诊断的结果。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 强烈建议使用输入自检机制。

5.19.2 转换器校准

赫丘利斯 MiADC 模块执行校准逻辑，此逻辑通常用于提升转换器准确性。 这个逻辑电路也可被用作一个安全机制。 来自校准逻辑电路的对已知基准值转换的软件比较能够提供一个对转换器功能性的诊断。 校准例程的重复执行可被用于检测应用期间的漂移。

软件必须配置、启用和评估这个诊断的结果。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 强烈建议在启动时使用转换器校准机制。 定期使用转换器校准机制是可选的。

5.19.3 信息冗余技术

信息冗余技术可由软件应用为一个 ADC 转换上的附加运行时间诊断。 有很多技术可被应用，例如使用共享通道和使用两个转换器的多重转换，在同一个转换器上使用多重通道的多重转换或者按照比较结果在同一通道上进行的多重转换。 对于已被转换值的过滤和处于预计范围内的真实性检验也可提升诊断的覆盖。

错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 强烈建议在 ADC 转换上使用信息冗余技术。

5.19.4 ADC SRAM 奇偶校验

MibADC SRAM 包括一个奇偶校验诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 MibADC SRAM 奇偶校验特性。

5.19.5 ADC SRAM PBIST

MibADC SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议复位后在 MibADC SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.19.6 ADC SRAM 位复用

MibADC SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.19.7 ADC SRAM CRC-64 测试

MibADC SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 MibADC SRAM 内容往往动态性更强，因此这个诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.19.8 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.19.9 注释

- MibADC 模块也许在一些文档中被称为 MibADC（多缓冲 ADC）。
- 应该如 电源中注释的那样对 ADC 模块电压进行监视。

5.20 多缓冲串行外设接口 (MIBSPI)

MibSPI 模块提供与 MibSPI 协议兼容的串行 I/O。 MibSPI 通信通常用于到智能传感器和传动器，串行存储器、和诸如安全器件的外部逻辑电路的通信。 MibSPI 模块包含内部 SRAM 缓冲器。 如果不被用于 MibSPI 通信，MibSPI 模块的 I/O 可被用于通用 I/O。

5.20.1 使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

MibSPI 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 MibSPI 功能性的任何测试应该包括 I/O 回路。

5.20.2 消息奇偶校验

赫丘利斯 MIBSPI 支持在由硬件发出的每一个 MIBSPI 消息数据的有效载荷中插入一个奇偶位。 硬件也支持进入的消息奇偶校验的评估。 检测到的错误生成一个到 CPU 的中断。 强烈建议使用这一特性。

5.20.3 信息冗余技术

信息冗余技术可由软件应用为一个针对 MIBSPI 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 替代的冗余技术可通过在系统中执行多重通道来实现。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 强烈建议在 MIBSPI 事务处理中使用信息冗余技术。

5.20.4 MIBSPI SRAM 奇偶校验

MIBSPI SRAM 包括一个奇偶校验诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 MIBSPI SRAM 奇偶校验特性。

5.20.5 MIBSPI SRAM PBIST

MIBSPI SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议在复位后在 MIBSPI SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.20.6 MIBSPI SRAM 位复用

MIBSPI SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.20.7 MIBSPI SRAM CRC-64 测试

MIBSPI SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 MIBSPI SRAM 内容往往动态性更强，因此这个诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.20.8 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.20.9 注释

- 在标准 SPI 模式中也可使用 MIBSPI。

5.21 串行外设接口 (SPI)

SPI 模块提供与 SPI 协议兼容的串行 I/O。 SPI 通信通常用于到智能传感器和传动器，串行存储器、和诸如安全器件的外部逻辑电路的通信。 如果不被用于 SPI 通信，SPI 模块的 I/O 可被用于通用 I/O。

5.21.1 使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

SPI 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 SPI 功能性的任何测试应该包括 I/O 回路。

5.21.2 消息奇偶校验

赫丘利斯 MIBSPI 支持在由硬件发出的每一个 MIBSPI 消息数据的有效载荷中插入一个奇偶位。 硬件也支持进入的消息奇偶校验的评估。 检测到的错误生成一个到 CPU 的中断。 强烈建议使用这一特性。

5.21.3 信息冗余技术

信息冗余技术可由软件应用为一个针对 SPI 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 替代的冗余技术可通过在系统中执行多重通道来实现。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 SPI 事务处理中使用信息冗余技术。

5.21.4 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.22 内置集成电路 (I2C)

I2C 模块提供一个与 I2C 协议兼容的多主控串行总线。

5.22.1 功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

5.22.2 信息冗余技术

信息冗余技术可由软件应用为一个针对 I2C 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 I2C 事务处理中使用信息冗余技术。

5.22.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.23 串行通信接口 (SCI)

SCI 模块提供针对诸如 UART 等典型异步串行通信接口 (SCI) 协议的串行 I/O 功能。 根据所使用的串行协议的不同，也许需要一个外部收发器。

5.23.1 使用 I/O 回路功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

SCI 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 SCI 功能性的任何测试应该包括 I/O 回路。

5.23.2 信息冗余技术

信息冗余技术可由软件应用为一个针对 SCI 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 SCI 事务处理中使用信息冗余技术。

5.23.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.24 本地互连网络 (LIN)

LIN 模块提供与 LIN 协议兼容的串行 I/O。 LIN 是一个低吞吐量时间触发协议。 这个模块可被配置成 SCI 模式并被用作一个通用串行接口。 一个外部收发器被用于 LIN 通信。

5.24.1 使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

LIN 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 LIN 功能性的任何测试应该包括 I/O 回路。

5.24.2 包括端到端安全状态恢复的信息冗余技术

信息冗余技术可由软件应用为一个针对 LIN 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。

为了提供对于 MCU 之外网络元件的诊断覆盖（线束、连接器、收发器），必须采用端到端安全状态恢复机制。 这些机制也可提供 MCU 内部的诊断覆盖。 可采用多种不同的机制，例如附加消息校验和、冗余传输、传输中的时间多样性等等。 大多数通用校验和被添加到一个传输的有效载荷部分以确保传输的正确性。除了任何协议级奇偶和校验和，这些校验和也被采用。 由于校验和由通信任一端的软件生成和评估，整个通信路径是安全的，实现端到端安全状态恢复。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员定义。 强烈建议使用这一机制。

5.24.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.24.4 注释

- 当在 SCI 模式中使用 LIN 模块，请参考 SCI 部分。

5.25 控制器局域网络 (DCAN)

DCAN 接口提供与基于事件的触发互连的中等吞吐量，与 CAN 协议兼容。 DCAN 模块要求一个外部收发器以在 CAN 网络上运转。

5.25.1 使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

DCAN 工具支持针对 I/O 的数字和模拟回路功能。 数字回路测试到模块边界的信号路径。 模拟回路测试从模块至 I/O 单元的信号路径，此时输出驱动器被禁用。 为了获得最佳的测试结果，对于 DCAN 功能性的任何测试应该包括 I/O 回路。

5.25.2 包括端到端安全状态恢复的信息冗余技术

信息冗余技术可由软件应用为一个针对 CAN 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。

为了提供对于 MCU 之外网络元件的诊断覆盖（线束、连接器、收发器），必须采用端到端安全状态恢复机制。 这些机制也可提供 MCU 内部的诊断覆盖。 可采用多种不同的机制，例如附加消息校验和、冗余传输、传输中的时间多样性等等。 大多数通用校验和被添加到一个传输的有效载荷部分以确保传输的正确性。除了任何协议级奇偶和校验和，这些校验和也被采用。 由于校验和由通信任一端的软件生成和评估，整个通信路径是安全的，实现端到端安全状态恢复。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员定义。 强烈建议使用这一机制。

5.25.3 DCAN SRAM 奇偶校验

DCAN SRAM 包括一个奇偶校验诊断，此诊断能够检测内存中的单一位错误。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 软件必须配置和启用这个特性。 强烈建议使用 DCAN SRAM 奇偶校验特性。

5.25.4 DCAN SRAM PBIST

DCAN SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议在复位后在 DCAN SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.25.5 DCAN SRAM 位复用

DCAN SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.25.6 DCAN SRAM CRC-64 测试

DCAN SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 DCAN SRAM 内容往往动态性更强，因此这个诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.25.7 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.26 FlexRay，FlexRay 传递单元 (FTU)

FlexRay 接口提供数据互连，此数据互连带有针对每一个被连接的通信结点预先定义的、时间触发的 (TDMA) 时间槽。 FlexRay 网络支持比旧版 CAN 网络更高的数据吞吐量并且与 FlexRay 协议兼容。 在典型汽车使用中，FlexRay 网络是用于与一个底盘和安全系统连接的骨干网络，这样信息可在安全系统间传递。

此器件提供两个 FlexRay 通道。 根据配置的不同，此器件能够连接两个不同的 FlexRay 网络并可作为网关，或者，如果在系统网络概念中要求多个时间槽，此器件可作为同一 FlexRay 网络上的两个节点。 需要一个外部收发器器件来提供物理层到 FlexRay 网络的连接。 集成了一个独立的 PLL 来生成 FlexRay 网络所需的频率。

赫丘利斯 FlexRay 模块包括 FlexRay 传递单元 (FTU) 中的专用 DMA 功能。

5.26.1 内存保护单元 (MPU)

FTU 包括一个 MPU。 MPU 逻辑电路可被用于提供器件内存中软件任务的空间分离。 根据每一个任务的需求，FlexRay 驱动器控制 MPU 并改变 MPU 设置。 违反一个已设置的内存保护策略会导致一个 ESM错误。

复位时 MPU 不启用 软件必须启用、配置和测试 MPU。 强烈建议使用 MPU。

5.26.2 非特权总线主控访问

FTU 是一个非特权总线主控。 由于只有特权模式才能对 MCU 上的关键配置寄存器进行写入操作，这个机制通过 FTU 防止这些寄存器的无意配置。

这个安全机制的运行是连续的并且不用由软件变更。 可通过生成软件事务且检查器件响应来测试这个机制。 这个安全机制的使用是强制的。

5.26.3 在收发器中使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

FlexRay 收发器的回路模式可对包括在物理层中的信号路径进行测试。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员来定义。 为了获得最佳的测试结果，对于 FlexRay 功能性的任何测试应该包括 I/O 回路。

5.26.4 包括端到端安全状态恢复的信息冗余技术

信息冗余技术可由软件应用为一个针对 FlexRay 通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。

为了提供对于 MCU 之外网络元件的诊断覆盖（线束、连接器、收发器），必须采用端到端安全状态恢复机制。 这些机制也可提供 MCU 内部的诊断覆盖。 可采用多种不同的机制，例如附加消息校验和、冗余传输、传输中的时间多样性等等。 大多数通用校验和被添加到一个传输的有效载荷部分以确保传输的正确性。除了任何协议级奇偶和校验和，这些校验和也被采用。 由于校验和由通信任一端的软件生成和评估，整个通信路径是安全的，实现端到端安全状态恢复。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员定义。 强烈建议使用这一机制。

5.26.5 使用两个 FlexRay 通道的 loo2 投票

赫丘利斯 FlexRay 模块有两个通道工具。 此两个通道可被用于冗余接收同一消息（这取决于网络定义）。FlexRay 消息缓冲器可被配置为从一个或者两个通道接收消息。 在一个单一缓冲器被用于两个通道的情况下，它存储从任一一个通道接收到的第一个语义正确消息。 为了增强通道分立，可在每个通道上配置一个缓冲器，然后可执行一个软件比较。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员来定义。 建议使用这一特性。

5.26.6 FlexRay 和 FTU SRAM 奇偶校验

FlexRay 和 FTU SRAM 包含可在内存中检测单一位错误的奇偶校验诊断。 当检测到一个奇偶错误时，ESM 被告知此错误。 这一特性在复位后被禁用。 对于每一个 SRAM，软件必须分别的配置和启用这个特性。 强烈建议使用 FlexRay 和 FTU 奇偶校验特性。

5.26.7 FlexRay 和 FTU SRAM PBIST

FlexRay 和 FTU SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议复位后在 FlexRay 和 FTU SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.26.8 FlexRay 和 FTU SRAM 位复用

FlexRay 和 FTU SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.26.9 FlexRay 和 FTU SRAM CRC-64 测试

FlexRay 和 FTU SRAM 内容可通过使用硬件 CRC-64 诊断进行定期测试。 由于 FTU SRAM 内容往往是静态的，而与之相对的 FlexRay SRAM 的内容是动态，因此建议使用这一诊断并且这一诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.26.10 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.26.11 注释

- FlexRay 通信控制器没有计算头文件 CRC 的能力。 主机为所有发送缓冲器提供头文件 CRC。
- 由于 FlexRay 传输的确定属性，这个接口首先选择发送定时敏感传感器数据并且执行多网路结点上安全任务的同步。

5.27 通用输入/输出 (GIO)

GIO 模块提供数字输入捕捉和数字输入/输出。 在这个块中没有处理功能。 GIO 通常用于静态的或者很少发生改变的输出，诸如收发器使能信号、报警光等。 GIO 也可被用于提供外部中断输入功能。

5.27.1 使用 I/O 检查的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

使用 I/O 检查的功能的软件测试 然而它有可能支持使用正常功能性的 I/O 检查。 为了实现这一功能，软件生成输出并回读和检验输入寄存器中的同一个值。 这个工具的功能性与其它模块中的模拟回路相似。 为了获得最佳的测试结果，对于 GIO 功能性的任何测试应该包括 I/O 回路。

5.27.2 信息冗余技术

信息冗余技术可由软件应用为一个对 GIO 功能的附加运行时间诊断。 可采用很多技术，诸如多重输入和使用一个输入通道的输出回读。 如果不被用于主要功能，来自很多其它外设的信号可被用作 GIO。 为多通道工具使用一个 GIO 模块信号和一个非 GIO 模块信号能够减少共模故障的可能性。

错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 GIO 功能上使用信息冗余技术。

5.27.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.27.4 注释

- 为了减少共模故障的可能性，用户应该考虑执行使用非邻近引脚的多重通道。

5.28 以太网

赫丘利斯平台包括一个以太网媒介访问控制器 (EMAC) 和物理层 (PHY) 器件管理数据输入/输出 (MDIO) 模块。 这些模块可实现赫丘利斯 MCU 到一个 10 和 100Mb 以太网的连接。 此以太网提供比 LIN，CAN，或者 FlexRay 更高的网络吞估量。

5.28.1 非特权总线主控访问

以太网模块是一个非特权总线主控。 由于只有特权模式才能对 MCU 上的关键配置寄存器进行写入操作，这个机制通过这个模块防止这些寄存器的无意配置。

这个安全机制的运行是连续的并且不用由软件变更。 可通过生成软件事务且检查器件响应来测试这个机制。 这个安全机制的使用是强制的。

5.28.2 在 PHY 中使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

大多数以太网 PHY 包含一个回路模块，此模块可实现对于包括 PHY 在内的信号路径测试。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员来定义。 为了获得最佳的测试结果，对于以太网功能性的任何测试应该包括 I/O 回路。

5.28.3 包括端到端安全状态恢复的信息冗余技术

信息冗余技术可由软件应用为一个针对以太网通信的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。

为了提供对于 MCU 之外网络元件的诊断覆盖（线束、连接器、收发器），必须采用端到端安全状态恢复机制。 这些机制也可提供 MCU 内部的诊断覆盖。 可采用多种不同的机制，例如附加消息校验和、冗余传输、传输中的时间多样性等等。 大多数通用校验和被添加到一个传输的有效载荷部分以确保传输的正确性。 除了任何协议级奇偶和校验和，这些校验和也被采用。 由于校验和由通信任一端的软件生成和评估，整个通信路径是安全的，实现端到端安全状态恢复。

错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员定义。 强烈建议使用这一机制。

5.28.4 EMAC SRAM PBIST

EMAC SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议在复位后在 EMAC SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.28.5 EMAC SRAM 位复用

EMAC SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.28.6 EMAC SRAM CRC-64 测试

EMAC SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 EMAC SRAM 内容往往动态性更强，因此这个诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.28.7 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.29 通用串行总线 (USB)

赫丘利斯系列扩展集架构包含通用串行总线 (USB) 功能。 支持一个全速主机和全速器件。 可支持 2 个主机或者 1 个主机和 1 个器件。

5.29.1 非特权总线主控访问

USB 模块是一个非特权总线主控。 由于只有特权模式才能对 MCU 上的关键配置寄存器进行写入操作，这个机制通过这个模块防止这些寄存器的无意配置。

这个安全机制的运行是连续的并且不用由软件变更。 可通过生成软件事务且检查器件响应来测试这个机制。 这个安全机制的使用是强制的。

5.29.2 在 PHY 中使用 I/O 回路的功能的软件测试

一个软件测试可被用于注入诊断错误并检验适当的错误响应。 这样一个测试可在启动时执行，或者定期执行。 必要的软件需求由系统集成人员执行的软件定义。 强烈建议使用基本功能性的引导时间软件测试。 使用一个基本功能性报告的定期软件测试是可选的。

大多数 USB PHY 包含一个回路模块，此模块可实现对于包括 PHY 在内的信号路径测试。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员来定义。 为了获得最佳的测试结果，对于以太网功能性的任何测试应该包括 I/O 回路。

5.29.3 信息冗余技术

信息冗余技术可由软件应用为一个针对 USB 事务处理的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 USB 事务处理中使用信息冗余技术。

5.29.4 USB SRAM PBIST

USB SRAM 可使用 PBIST 内存 BIST 引擎进行测试。 强烈建议在复位后在 USB SRAM 上执行 PBIST 测试。 要获得这一诊断的更多信息，请见 PBIST。

5.29.5 USB SRAM 位复用

USB SRAM 由一个内存设计实现，这样逻辑上邻近的位的物理位置不相邻。 这个安全机制的使用是强制的。 对于这一安全机制的更多细节，请见 SRAM 位复用。

5.29.6 USB SRAM CRC-64 测试

USB SRAM 内容可使用硬件 CRC-64 诊断进行定期测试。 由于 USB SRAM 内容往往动态性更强，因此这个诊断的使用是可选的。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。

5.29.7 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.30 外部存储器接口 (EMIF)

外部存储器接口被用于提供到芯片外内存或者支持一个内存接口的器件的访问。 提供对同步 (SDRAM) 和异步（NOR 闪存、SRAM）内存的支持。

5.30.1 信息冗余技术

信息冗余技术可由软件应用为一个针对 EMIF 事务处理的附加运行时间诊断。 可应用很多技术，例如已写入值的回读和与结果相比较的同一目标数据的多次读取。 错误响应、诊断的可测试性、以及任何必须的软件要求由系统集成人员所执行的软件来定义。 强烈建议在 EMIF 事务处理中使用信息冗余技术。

5.30.2 EMIF 内存 CRC-64 测试

被连接到 EMIF 的外部存储器的内容可使用硬件 CRC-64 诊断进行定期测试。 这个诊断对于静态存储器内容的诊断十分有用，但是对于应用中动态变化的内存内容的诊断作用就有所降低。 要获得这一诊断的更多信息，请见 SRAM 硬件 CRC-64。 强烈建议在启动时对非易失性内存使用此诊断并建议定期使用。 对于易失性内存，建议的做法与对内部 SRAM 采用的做法一致。

5.30.3 配置寄存器的定期回读

配置寄存器的定期回读能够为无意写入或者这些寄存器的混乱提供一个诊断。 错误响应、诊断的可测试性、以及任一所需的软件要求由系统集成人员所选择的软件来定义。 推荐使用配置寄存器的回读机制。

5.30.4 注释

- 为了实现更高完整性操作，来自外部存储器的安全数据可被传递或者复制到位于安全区域内的内部存储器。

5.31 JTAG 调试、跟踪、校准、和测试访问

赫丘利斯平台支持在 IEEE 1149.1 JTAG 调试端口上实现调试、测试、和校准功能。 物理调试接口被内部连接至一个 TI 调试复用器逻辑电路 (ICEPICK)，此电路对到测试、调试、和校准逻辑电路的访问进行仲裁。 为了实现最简单的制造板测试，边界扫面被并行连接至 ICEPICK 以支持不含前导码扫描序列的用法。

5.31.1 JTAG 端口的硬件禁用

JTAG 调试端口能够被物理禁用以防止已部署系统中的 JTAG 访问。 虽然其它替代系统配置也是可行的，但这个建议推荐的系统配置是为了保持测试时钟 (TCK) 到接地并保持测试模式选择 (TMS) 为高电平。 建议采用 JTAG 端口的硬件禁用。

5.31.2 使用 AJSM 的 JTAG 访问的锁存

赫丘利斯平台包含高级 JTAG 安全模块 (AJSM) 以支持已部署器件上调试访问的管理。 AJSM 可被用于设定一个到 OTP 闪存存储器的唯一访问密钥。 为了获得到基于 JTAG 的调试、跟踪和校准逻辑电路的访问权限，随后的调试访问必须使用正确的密钥对 AJSM 解锁。 解锁 AJSM 过程中的一个错误会导致无错误响应和没有对调试逻辑电路的访问权限。 强烈建议使用 AJSM 来锁存调试访问。

5.31.3 注释

- 即使系统被 AJSM 锁定，也可进行边界扫描访问。
- 一个安全装置可提供意外激活的标示。

5.32 Cortex-R4F 中央处理单元 (CPU) 调试和跟踪

赫丘利斯平台支持与 ARM CoreSight 标准兼容的 CPU 调试和跟踪。 每一个 CoreSight 元件可通过一个内存映射调试总线进行访问，此总线可由 CPU 或者 JTAG 端口进行访问。 CPU 调试和跟踪逻辑电路包含一个独立的调试总线主控 (AHB-AP)，CPU 内的调试单元、和嵌入式跟踪宏单元 (ETM-R4)。 不建议在安全操作和安全机制正在禁用这个逻辑电路期间使用这些模块。

5.32.1 禁用 JTAG 端口以限制功能访问

大多数调试和跟踪活动由一个外部调试工具启动，此调试工具使用 JTAG 端口将命令写入器件。 如 JTAG 调试/跟踪/校准/测试访问中所示，JTAG 端口可在生产硬件上被禁用。 强烈建议禁用 JTAG 端口以限制调试模块访问。

5.32.2 使用 AJSM 的JTAG 访问的锁存

赫丘利斯平台包含 AJSM 以实现对于已部署器件上调试访问的管理。 AJSM 可被用于设定一个到 OTP 闪存存储器的唯一的访问密钥。 为了获得到基于 JTAG 的调试、跟踪和校准逻辑电路的访问权限，随后的调试访问必须使用正确的密钥对 AJSM 解锁。 解锁 AJSM 过程中的一个错误会导致无错误响应和没有对调试逻辑电路的访问权限。 强烈建议使用 AJSM 来锁存调试访问。

5.32.3 阻止到内存映射调试的访问

可通过一个内存映射调试总线对 CoreSight 调试外设进行访问。 对于这一区域的访问可通过使用基于内存保护的总线主控进行阻止。 要获得与内存保护相关的更多信息，请见 CPU 内存保护单元 (MPU)。 强烈建议阻止到内存映射调试组件的访问。

5.32.4 CoreSight 调试逻辑密钥使能

为了开启内存映射 CoreSight 调试组件的运行，有必要在每一个调试模块中的解锁寄存器中写入一个已定义的 32 位密钥。 这个调试锁保护为限制非所需激活提供了一个额外的保护机制。 强烈建议使用调试模块解锁密钥。

5.32.5 注释

- 一个安全装置可提供意外激活的标示。

5.33 数据修改模块 (DMM)

数据修改模块 (DMM) 提供一个校准总线主控功能。 校准期间，DMM 被用作一个最小侵入式 DMA 以将校准数据复制到器件。 通过 JTAG 或者内存映射寄存器对 DMM 下达命令。

5.33.1 禁用 JTAG 端口以限制功能访问

大多数 DMM 活动由一个外部调试工具启动，此调试工具使用 JTAG 端口将命令写入器件。 如 JTAG 调试/跟踪/校准/测试访问中所示，JTAG 端口可在生产硬件上被禁用。 强烈建议禁用 JTAG 端口以限制 DMM 模块访问。

5.33.2 使用 AJSM 的JTAG 访问的锁存

赫丘利斯平台包含 AJSM 以实现对于已部署器件上调试访问的管理。 AJSM 可被用于设定一个到 OTP 闪存存储器的唯一的访问密钥。 为了获得到基于 JTAG 的调试、跟踪和校准逻辑电路的访问权限，随后的调试访问必须使用正确的密钥对 AJSM 解锁。 解锁 AJSM 过程中的一个错误会导致无错误响应和没有对调试逻辑电路的访问权限。 强烈建议使用 AJSM 来锁存调试访问。

5.33.3 阻止到内存映射调试的访问

DMM 外设可通过外设总线进行访问。 对于这一区域的访问可通过使用基于内存保护的总线主控进行阻止。 要获得与内存保护相关的更多信息，请见 CPU 内存保护单元 (MPU)。 强烈建议阻止到 DMM 控制寄存器的访问。

5.33.4 禁用 DMM 引脚接口

DMM 外设有一个器件级并行输入端口，此端口通常由一个外部工具或者随机存取内存跟踪端口 (RTP) 驱动。 为了生产使用，DMM 引脚接口可被禁用或者阻止数据输入。 一个可能的方法就是将 DMM 时钟输入驱动为低电平并将低电平有效 DMM 使能输入驱动为高电平。 强烈建议在生产环境中禁用 DMM 引脚接口。

5.33.5 注释

- 一个安全装置可提供意外激活的标示。

5.34 RAM 跟踪端口 (RTP)

RAM 跟踪端口 (RTP) 用于记录数据写入到内部 SRAM 中。 这个功能被使用在校准中以保持器件内存的远程镜像复制。

5.34.1 禁用 JTAG 端口以限制功能访问

大多数 RTP 活动由一个外部调试工具启动，此调试工具使用 JTAG 端口将命令写入器件。 如 JTAG 调试/跟踪/校准/测试访问中所示，JTAG 端口可在生产硬件上被禁用。 强烈建议禁用 JTAG 端口以限制 RTP 模块访问。

5.34.2 使用 AJSM 的JTAG 访问的锁存

赫丘利斯平台包含 AJSM 以实现对于已部署器件上调试访问的管理。 AJSM 可被用于设定一个到 OTP 闪存存储器的唯一的访问密钥。 为了获得到基于 JTAG 的调试、跟踪和校准逻辑电路的访问权限，随后的调试访问必须使用正确的密钥对 AJSM 解锁。 解锁 AJSM 过程中的一个错误会导致无错误响应和没有对调试逻辑电路的访问权限。 强烈建议使用 AJSM 来锁存调试访问。

5.34.3 阻止到内存映射调试的访问

RTP 外设可通过外设总线进行访问。 对于这一区域的访问可通过使用基于内存保护的总线主控进行阻止。 要获得与内存保护相关的更多信息，请见 CPU 内存保护单元 (MPU)。 强烈建议阻止到 RTP 控制寄存器的访问。

5.34.4 禁用 RTP 引脚接口

RTP 外设有一个器件级并行输出端口，此端口通常被连接至一个外部工具或者用于驱动一个 DMM 的输入。 为了生产使用，RTP 引脚接口可被禁用以阻止数据输出。 一个可能的方法就是驱动 RTP 时钟输入为低电平并驱动低电平有效的 RTP 使能输入为高电平。 建议在生产环境中禁用 RTP 引脚接口。

5.34.5 注释

- 一个安全装置可提供意外激活的标示。

5.35 参数覆盖模块 (POM)

参数覆盖模块 (POM) 被用于将闪存访问重新定向到一个不同的内部或者外部存储器。 校准期间使用这一功能来测试代码或者数据的升级部分而不需要执行一个耗时的闪存擦除和编程操作或者代码重建操作。 POM 被执行为一个 CoreSight 兼容外设。

5.35.1 禁用 JTAG 端口以限制功能访问

大多数 POM 活动由一个外部调试工具启动，此调试工具使用 JTAG 端口将命令写入器件。 如 JTAG 调试/跟踪/校准/测试访问中所示，JTAG 端口可在生产硬件上被禁用。 强烈建议禁用 JTAG 端口以限制 POM 模块访问。

5.35.2 使用 AJSM 的JTAG 访问的锁存

赫丘利斯平台包含 AJSM 以实现对于已部署器件上调试访问的管理。 AJSM 可被用于设定一个到 OTP 闪存存储器的唯一的访问密钥。 为了获得到基于 JTAG 的调试、跟踪和校准逻辑电路的访问权限，随后的调试访问必须使用正确的密钥对 AJSM 解锁。 解锁 AJSM 过程中的一个错误会导致无错误响应和没有对调试逻辑电路的访问权限。 强烈建议使用 AJSM 来锁存调试访问。

5.35.3 阻止到内存映射调试的访问

可通过一个内存映射调试总线对 POM 进行访问。 对于这一区域的访问可通过使用基于内存保护的总线主控进行阻止。 要获得与内存保护相关的更多信息，请见 CPU 内存保护单元 (MPU)。 强烈建议阻止到内存映射 POM 寄存器的访问。

5.35.4 CoreSight 调试逻辑密钥使能

为了启用 POM 的运行，需要向 POM 内的一个解锁寄存器写入一个已定义的 32 位密钥。 这个调试锁保护为限制非所需激活提供了一个额外的保护机制。 强烈建议使用 POM 解锁密钥。

5.35.5 注释

- 一个安全装置可提供意外激活的标示。

6 您安全开发中的下几个步骤

TI 对于您的安全开发并不仅仅限于这本安全文档。 客户可选择多种类型的支持方式，诸如：

- 随时在线访问包括安全文档和应用报告在内的赫丘利斯文档：http://www.ti.com/hercules
- 使用 TI 互连社区（E2E 论坛）与 TI 专家和其它赫丘利斯开发人员探讨问题和关心的事项：http://www.ti.com/hercules-support。
- 赫丘利斯维基网页提供对于很多常见问题的解答：http://www.ti.com/hercules-wiki。
- 参加由 TI 赫丘利斯专家教授的培训课程：http://www.ti.com/herculestraining

我们随时欢迎您对于安全手册的反馈和参与，并且可以通过点击本文档每页底部的反馈链接将您的意见在线提交给我们。

Appendix A 建议的安全特性用法总结

表 2提供了5 节中注释的安全概念建议的总结并按照器件分区进行组织。 每个建议都被指定了一个唯一的标识符以在需求管理中提供帮助。 对于每一个安全特性或者诊断，建议通过以下的简化形式进行注释：

- M --> 强制应用
- ++ --> 强烈推荐
- + --> 建议
- O --> 可选

此外，还提供了对于每一个器件分区的可能潜在的诊断方法和安全特性以及诊断组合列表。 对于每一个安全特性或者诊断的详细信息，请见5 节。

表 2. 安全特性和诊断的总结

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
电源	PWR1	电压监控器 (VMON)	M	外部电压监视器
	PWR2	外部电压监视器	++	电压监视器 (VMON)
电源管理模块 (PMM)	PMM1	锁步 PSCON	M	PSCON 锁步自检
	PMM2	特权模式访问和程序序列控制寄存器	M	寄存器配置和错误响应的软件测试
	PMM3	静态配置寄存器的定期软件回读	+	CPU 锁步
	PMM4	已写入配置的软件回读	++	CPU 锁步
时钟	CLK1	LPOCLKDET	++	DCC，ECLK，安全装置
	CLK2	PLL 滑动检测器	++	DCC，ECLK，安全装置
	CLK3	双时钟比较器 (DCC)	++	DCC 自动覆盖，ECLK，安全装置
	CLK4	通过 ECLK 的外部监控	O	LPOCLKDET，PLL 滑动检测器，DCC，安全装置
	CLK5A	内部安全装置 - DWD	O	外部安全装置，安全装置配置和错误响应的软件测试
	CLK5B	内部安全装置 - DWWD	+	外部安全装置，安全装置配置和错误响应的软件测试
	CLK5C	外部安全装置	++	内部安全装置，安全装置配置和错误响应的软件测试
	CLK6	静态时钟配置寄存器的定期软件回读	+	CPU 锁步
	CLK7	已写入配置的软件回读	++	CPU 锁步
复位	RST1	热复位的外部监控	O	安全装置
	RST2	最后复位的软件检查	++	CPU 锁步
	RST3	软件热复位生成	O	CPU 锁步
	RST4	复位引脚上的毛刺脉冲过滤	M	安全装置
	RST5	状态影子寄存器的使用	++	CPU 锁步
	RST6	外部安全装置	++	外部安全装置配置和错误响应的软件测试
	RST7	静态配置寄存器的定期软件回读	+	CPU 锁步
	RST8	已写入配置的软件回读	++	CPU 锁步
系统控制	SYS1	优先模式访问和多位使能密钥	M	寄存器配置和错误响应的软件测试
	SYS2	已写入配置的软件回读	++	CPU 锁步
	SYS3	静态配置寄存器的定期软件回读	+	CPU 锁步
错误信令模块 (ESM)	ESM1	静态配置寄存器的定期软件回读	+	CPU 锁步
	ESM2A	错误路径报告的引导时间软件测试	++	CPU 锁步
	ESM2B	错误路径报告的定期软件测试	+	CPU 锁步
	ESM3	状态影子寄存器的使用	++	CPU 锁步
	ESM4	已写入配置的软件回读	++	CPU 锁步

表 2. 安全特性和诊断的总结 (continued)

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
Cortex-R4F 中央处理单元 (CPU)	CPU1	锁步比较	M	CCM（锁步）自检，LBIST
	CPU2A	LBIST STC 引导时间执行	++	LBIST 自动覆盖
	CPU2B	LBIST STC 的定期执行	O	LBIST 自动覆盖
	CPU3	MPU	++	CPU 锁步，LBIST
	CPU4	使用 PMU 的在线参数描述	O	CPU 锁步，LBIST
	CPU5A	内部安全装置 - DWD	O	外部安全装置，安全装置配置和错误响应的软件测试
	CPU5B	内部安全装置 - DWWD	+	外部安全装置，安全装置配置和错误响应的软件测试
	CPU5C	外部安全装置	++	内部安全装置，安全装置配置和错误响应的软件测试
	CPU6	无效操作和指令陷阱	++	CPU 锁步，LBIST
	CPU7	已写入配置的软件回读	++	CPU 锁步
初级闪存和 1 级 (L1) 互连	FLA1	闪存数据 ECC	++	CPU 锁步，LBIST，ECC 自动覆盖
	FLA2	硬错误高速缓存和活锁	M	CPU 锁步，LBIST
	FLA3	闪存包装程序地址 ECC	++	闪存包装程序 ECC 逻辑，ECC 自动覆盖的软件测试
	FLA4	地址奇偶校验	++	CPU 锁步，LBIST
	FLA5A	闪存存储器内容的引导时间硬件 CRC 检查	++	CRC 自动覆盖
	FLA5B	闪存存储器内容的定期硬件 CRC 检查	+	CRC 自动覆盖
	FLA6	闪存阵列中的位复用	M	ECC，CRC 测试
	FLA7	闪存扇区保护	++	CRC 测试
	FLA8	静态配置寄存器的定期软件回读	+	CPU 锁步
	FLA9	已写入配置的软件回读	++	CPU 锁步
闪存仿真 EEPROM (FEE)	FEE1	FEE 数据 ECC	++	EEPROM 仿真闪存包装程序，ECC 自动覆盖的软件测试
	FEE2A	FEE 内存内容的引导时间硬件 CRC 检查	++	CRC 自动覆盖
	FEE2B	FEE 内存内容的定期硬件 CRC 检查	+	CRC 自动覆盖
	FEE3	FEE 阵列中的位复用	M	ECC，CRC 测试
	FEE4	FEE 扇区保护	++	CRC 测试
	FEE5	静态配置寄存器的定期软件回读	+	CPU 锁步
	FEE6	已写入配置的软件回读	++	CPU 锁步
SRAM 和 1 级 (L1) 互连	RAM1	数据 ECC	++	CPU 锁步，LBIST，ECC 自动覆盖，PBIST
	RAM2	硬错误高速缓存和活锁	M	CPU 锁步，LBIST
	RAM3	可校正的 ECC 参数描述	+	CPU 锁步，LBIST
	RAM4	地址和控制奇偶校验	++	CPU 锁步，LBIST，PBIST
	RAM5	冗余地址解码	++	ECC，PBIST
	RAM6	存储在多重物理组中的数据和 ECC	M	ECC，自动覆盖，PBIST
	RAM7A	RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	RAM7B	RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	RAM8	SRAM 阵列中的位复用	M	ECC
	RAM9	SRAM 内容的定期硬件 CRC 检查	O	CRC 自动覆盖
	RAM10	静态配置寄存器的定期软件回读	+	CRC 自动覆盖
	RAM11	已写入配置的软件回读	++	CPU 锁步

表 2. 安全特性和诊断的总结 (continued)

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
2 级和 3 级（L2 和 L3）互连	INC1	错误捕捉	M	基本功能性和错误响应的软件测试
	INC2	PCR 访问管理	++	基本功能性和错误响应的软件测试
	INC3A	内部安全装置 - DWD	O	外部安全装置，安全装置配置和错误响应的软件测试
	INC3B	内部安全装置 - DWWD	+	外部安全装置，安全装置配置和错误响应的软件测试
	INC3C	外部安全装置	++	内部安全装置，安全装置配置和错误响应的软件测试
	INC4	信息冗余	+	CPU 锁步
	INC5	静态配置寄存器的定期软件回读	+	CPU 锁步
	INC6A	基本功能性的引导时间软件测试	++	CPU 锁步
	INC6B	基本功能性的定期软件测试	+	CPU 锁步
	INC7	已写入配置的软件回读	++	CPU 锁步
EFuse 静态配置	EFU1	引导时间自动加载自检	M	自检自动覆盖
	EFU2	E-fuse（电子熔丝）ECC	M	自动载入自检
一次可编程 (OTP) 闪存静态配置	OTP1	引导时间自动载入自检	M	自检自动覆盖
	OTP2	E-fuse ECC	M	自动载入自检
输入/输出 (I/O) 复用 (IIOMM)	IOM1	锁闭控制寄存器	++	基本功能性和错误响应的软件测试
	IOM2	主控 ID 过滤	M	基本功能性和错误响应的软件测试
	IOM3	错误捕捉	M	基本功能性和错误响应的软件测试
	IOM4	静态配置寄存器的定期回读	+	CPU 锁步
	IOM5A	使用带有模拟 I/O 回路的外设的功能的引导时间软件测试	++	CPU 锁步
	IOM5B	使用带有模拟 I/O 回路的外设的功能的定期软件测试	O	CPU 锁步
	IOM6	已写入配置的软件回读	++	CPU 锁步
矢量中断模块 (VIM)	VIM1	VIM SRAM 数据奇偶校验	++	PBIST
	VIM2A	VIM RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	VIM2B	VIM RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	VIM3	VIM SRAM 阵列中的位复用	M	PBIST，奇偶校验
	VIM4	VIM SRAM 内容的定期硬件 CRC 检查	+	PBIST
	VIM5	VIM 功能性的定期软件测试	++	CPU 锁步
	VIM6	静态配置寄存器的定期软件回读	+	CPU 锁步
	VIM7	已写入配置的软件回读	++	CPU 锁步
	VIM8A	内部安全装置 - DWD	O	外部安全装置，安全装置配置和错误响应的软件测试
	VIM8B	内部安全装置 - DWWD	+	外部安全装置，安全装置配置和错误响应的软件测试
	VIM8C	外部安全装置	++	内部安全装置，安全装置配置和错误响应的软件测试
实时中断 (RTI) 操作系统定时器	RTI1	使用二级自由运行计数器的 loo2 软件投票	+	CPU 锁步，PMU 周期计数器
	RTI2A	内部安全装置 - DWD	O	外部安全装置，安全装置配置和错误响应的软件测试
	RTI2B	内部安全装置 - DWWD	+	外部安全装置，安全装置配置和错误响应的软件测试
	RTI2C	外部安全装置	++	内部安全装置，安全装置配置和错误响应的软件测试
	RTI3	静态配置寄存器的定期软件回读	+	CPU 锁步

表 2. 安全特性和诊断的总结 (continued)

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
直接内存访问 (DMA)	DMA1	针对总线主控访问的内存保护单元	++	DMA MPU 配置和错误响应的软件测试
	DMA2	非特权总线主控访问	M	总线主控功能和错误响应的软件测试
	DMA3	信息冗余	+	CPU 锁步
	DMA4	DMA SRAM 数据奇偶校验	++	PBIST
	DMA5A	DMA RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	DMA5B	DMA RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	DMA6	DMA SRAM 阵列中的位复用	M	PBIST，奇偶校验
	DMA7	DMA SRAM 内容的定期硬件 CRC 检查	+	PBIST
	DMA8	静态配置寄存器的定期软件回读	+	CPU 锁步
	DMA9A	基本功能性的引导时间软件测试	++	CPU 锁步
	DMA9B	基本功能性的定期软件测试	+	CPU 锁步
高端定时器 (N2HET)，此定时器包含 HET 转移单元 (HTU)	HET1	用于总线主控访问的内部保护单元	++	HTU MPU 配置和错误响应的软件测试
	HET2	信息冗余	++	CPU 锁步
	HET3	将 DCC 用作程序序列安全装置	++	DCC 自动覆盖
	HET4	第二 N2HET 的监控	+	CPU 锁步
	HET5A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	HET5B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	HET6	N2HET/HTU SRAM 数据奇偶校验	++	PBIST
	HET7A	N2HET/HTU RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	HET7B	N2HET/HTU RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	HET8	N2HET/HTU RAM 阵列中的位复用	M	PBIST，奇偶校验
	HET9	N2HET/HTU SRAM 内容的定期硬件 CRC 检查	O	PBIST
	HET10	静态配置寄存器的定期回读	+	CPU 锁步
多缓冲模拟数字转换器 (MibADC)	ADC1	引导时间自检	++	CPU 锁步
	ADC2A	引导时间转换器校准	++	CPU 锁步
	ADC2B	定期转换器校准	O	CPU 锁步
	ADC3	信息冗余技术	++	CPU 锁步、ADC 转换器校准、ADC 自检
	ADC4	MibADC SRAM 数据奇偶校验	++	PBIST
	ADC5A	MibADC RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	ADC5B	MibADC RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	ADC6	MibADC RAM 阵列中的位复用	M	PBIST，奇偶校验
	ADC7	MibADC SRAM 内容的定期硬件 CRC 检查	O	PBIST
	ADC8	静态配置寄存器的定期软件回读	+	CPU 锁步
多缓冲串行外设接口 (MibSPI)	MSP1A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	MSP1B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	MSP2	信息奇偶校验	++	CPU 锁步
	MSP3	信息冗余技术	++	CPU 锁步
	MSP4	MibSPI SRAM 数据奇偶校验	++	PBIST
	MSP5A	MibSPI RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	MSP5B	MibSPI RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	MSP6	MibSPI RAM 阵列中的位复用	M	PBIST，奇偶校验
	MSP7	MibSPI SRAM 内容的定期硬件 CRC 检查	O	PBIST
	MSP8	静态配置寄存器的定期软件回读	+	CPU 锁步
串行外设接口 (SPI)	SPI1A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	SPI1B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	SPI2	消息中的奇偶校验	++	CPU 锁步
	SPI3	信息冗余技术	++	CPU 锁步
	SPI4	静态配置寄存器的定期软件回读	+	CPU 锁步

表 2. 安全特性和诊断的总结 (continued)

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
内部集成电路 (I2C)	IIC1A	功能的引导时间软件测试	++	CPU 锁步
	IIC1B	功能的定期软件测试	O	CPU 锁步
	IIC2	信息冗余技术	++	CPU 锁步
	IIC3	静态配置寄存器的定期回读	+	CPU 锁步
串行通信接口 (SPI)	SCI1A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	SCI1B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	SCI2	信息冗余技术	++	CPU 锁步
	SCI3	静态配置寄存器的定期回读	+	CPU 锁步
本地互连网络 (LIN)	LIN1A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	LIN1B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	LIN2	包含端到端安全状态恢复的信息冗余技术	++	CPU 锁步
	LIN3	静态配置寄存器的定期软件回读	+	CPU 锁步
控制器局域网 (DCAN)	CAN1A	使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	CAN1B	使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	CAN2	包含端到端安全状态恢复的信息冗余技术	++	CPU 锁步
	CAN3	DCAN SRAM 数据奇偶校验	++	PBIST
	CAN4A	DCAN RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	CAN4B	DCAN RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	CAN5	DCAN RAM 阵列位复用	M	PBIST，奇偶校验
	CAN6	DCAN SRAM 内容的定期硬件 CRC 检查	O	PBIST
	CAN7	静态配置寄存器的定期软件回读	+	CPU 锁步
包含 FlexRay 转移单元 (FTU) 的 FlexRay	FRY1	用于总线主控访问的内部保护单元	++	DMA MPU 配置和错误响应的软件测试
	FRY2	非特权总线主控访问	M	总线主控功能和错误响应的软件测试
	FRY3A	在 PHY 中使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	FRY3B	在收发器中使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	FRY4	包含端到端安全状态恢复的信息冗余技术	++	CPU 锁步
	FRY5	使用两个 FlexRay 通道的 loo2 投票	+	CPU 锁步
	FRY6	FlexRay 和 FTU SRAM 数据奇偶校验	++	PBIST
	FRY7A	FlexRay 和 FTU RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	FRY7B	FlexRay 和 FTU RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	FRY8	FlexRay 和 FTU RAM 阵列的位复用	M	PBIST，奇偶校验
	FRY9	FlexRay 和 FTU SRAM 内容的定期硬件 CRC 检查	O	PBIST
	FRY10	静态配置寄存器的定期软件回读	+	CPU 锁步
通用输入/输出(GIO)	GIO1A	使用 I/O 检查的功能的引导时间软件测试	++	CPU 锁步
	GIO1B	使用 I/O 检查的功能的定期软件测试	O	CPU 锁步
	GIO2	信息冗余技术	++	CPU 锁步
	GIO3	静态配置寄存器的定期软件回读	+	CPU 锁步
以太网	ETH1	非特权总线主控访问	M	总线主控功能和错误响应的软件测试
	ETH2A	在 PHY 中使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	ETH2B	在 PHY 中使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	ETH3	包含端到端安全状态恢复的信息冗余技术	++	CPU 锁步
	ETH4A	以太网 RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	ETH4B	以太网 RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	ETH5	以太网阵列中的位复用	M	PBIST，奇偶校验
	ETH6	以太网 SRAM 内容的定期硬件 CRC 检查	O	PBIST
	ETH7	静态配置寄存器的定期回读	+	CPU 锁步

表 2. 安全特性和诊断的总结 (continued)

器件分区	唯一标识符	安全特性或者诊断	特性建议	可能潜在诊断
通用串行总线 (USB)	USB1	非特权总线主控访问	M	总线主控功能和错误响应的软件测试
	USB2A	在 PHY 中使用 I/O 回路的功能的引导时间软件测试	++	CPU 锁步
	USB2B	在 PHY中使用 I/O 回路的功能的定期软件测试	O	CPU 锁步
	USB3	信息冗余技术	++	CPU 锁步
	USB4A	USB RAM 的引导时间 PBIST 检查	++	PBIST 自动覆盖
	USB4B	USB RAM 的定期 PBIST 检查	O	PBIST 自动覆盖
	USB5	USB 阵列中的位复用	M	PBIST，奇偶校验
	USB6	USB SRAM 内容的定期硬件 CRC 检查	O	PBIST
	USB7	静态配置寄存器的定期回读	+	CPU 锁步
外部存储器接口 (EMIF)	EMF1	信息冗余技术	++	CPU 锁步
	EMF2A	外部存储器的引导时间硬件 CRC 检查	++	CRC 自动覆盖
	EMF2B	外部存储器的定期硬件 CRC 检查	O	CRC 自动覆盖
	EMF3	静态配置寄存器的定期软件回读	+	CPU 锁步
联合技术行动组 (JTAG) 调试/跟踪/校准访问	JTG1	JTAG 端口的硬件禁用	+	影响程序流的意外激活的安全装置检测
	JTG2	使用 AJSM 的JTAG 访问的锁存	++	影响程序流的意外激活的安全装置检测
Cortex-R4F 中央处理单元 (CPU) 调试和跟踪	DBG1	JTAG 端口的硬件禁用	+	影响程序流的意外激活的安全装置检测
	DBG2	使用 AJSM 的JTAG 访问的锁存	++	影响程序流的意外激活的安全装置检测
	DBG3	使用 MPU 来阻止到内存映射调试的访问	++	影响程序流的意外激活的安全装置检测
	DBG4	使用 CoreSight 调试逻辑密钥使能系统配置	++	影响程序流的意外激活的安全装置检测
数据修改模块 (DMM)	DMM1	JTAG 端口的硬件禁用	+	影响程序流的意外激活的检测
	DMM2	使用 AJSM 的JTAG 访问的锁存	++	影响程序流的意外激活的安全装置检测
	DMM3	使用 MPU 来阻止到内存映射调试的访问	++	影响程序流的意外激活的安全装置检测
	DMM4	禁用 DMM 引脚接口	++	影响程序流的意外激活的安全装置检测
RAM 跟踪端口 (RTP)	RTP1	JTAG 端口的硬件禁用	+	影响程序流的意外激活的安全装置检测
	RTP2	使用 AJSM 的JTAG 访问的锁存	++	影响程序流的意外激活的安全装置检测
	RTP3	使用 MPU 来阻止到内存映射调试的访问	++	影响程序流的意外激活的安全装置检测
	RTP4	禁用 RTP 引脚接口	++	影响程序流的意外激活的安全装置检测
参数覆盖模块 (POM)	POM1	JTAG 端口的硬件禁用	+	影响程序流的意外激活的安全装置检测
	POM2	使用 AJSM 的JTAG 访问的锁存	++	影响程序流的意外激活的安全装置检测
	POM3	使用 MPU 来阻止到内存映射调试的访问	++	影响程序流的意外激活的安全装置检测
	POM4	使用 CoreSight 调试逻辑密钥使能系统配置	++	影响程序流的意外激活的安全装置检测

Appendix B 开发接口协定

一个开发接口协定 (DIA) 用于在一个客户和供货商间对于在开发一个实用安全系统方面管理共有责任达成协议。 在定制开发方面，DIA 是一个在开发过程早期在客户和供货商间执行的关键文档。 由于赫丘利斯系列是一款商用、现货 (COTS) 产品，TI 已经在这个部分准备了一个标准 DIA，描述了TI 能够为客户开发提供的支持。 对于定制 DIA 的要求应首先请您当地的 TI 销售办公室进行处理。

B.1 安全经理的任命

德州仪器 (TI) 已经开发了赫丘利斯 MCU，在整个芯片设计、上市、和批量生产的过程中，一名或者几名开发专业安全经理参与其中。 批量生产后的安全管理由独立的专门负责生产和运转事务的安全经理支持。 安全管理责任在产品停产时将继续。

B.2 安全声明周期的定制

TI 已经修改了 IEC 61508:2010 和 ISO 26262:2011 的安全生命周期以最好的匹配一个环境安全元件的需要 (SEooC)。 修改的活动已经与来自 exida 和 Yogitech 的输入共同执行以确保最先进的技术和方法的应用。

定制的安全周期的关键元件包括：

- 系统级设计、安全概念、和需求的假定
- 组合定性和定量或者相似安全分析技术包含 TI 和 Yogitech 所知的芯片故障模式和诊断技术的汇总
- 基于多重工业标准以及 TI “真实世界” (real-world) 可靠数据的故障评估
- Yogitech 的代表技术发展水平的用于已宣称诊断范围验证的故障注入方法的应用
- 两个公司通过针对 IEC 61508 多重安全开发和参与 ISO 26262 国际工作组的过程中所获得的经验和教训的整合

图 9用图例显示了这些针对微控制器的覆盖在 TI 的标准 QM 开发流程顶部的活动。

Phase 0 Business Opportunity Prescreen	Phase 1 Program Planning	Phase 2 Create	Phase 2.5 Validate, Sample, and Characterize	Phase 3 Qualify	Phase 4 Ramp or Sustain
Determine if safety process execution is necessary	Define SIL/ASIL capability	Execute safety design	Validate safety design in silicon	Qualification of safety design	Implement plans to support operation and production
Execute development interface agreement (DIA) with lead customers and suppliers	Generate safety plan	Qualitative analysis of design (FMEA and FTA)	Release safety manual	Release safety case report	Update safety case report (if needed)
	Initiate safety case	Incorporate findings into safety design	Release safety analysis report	Update safety manual (if needed)	Periodic confirmation measure reviews
	Analyze system to generate system level safety assumptions and requirements	Develop safety product preview	Characterization of safety design	Update safety analysis report (if needed)	
	Develop component level safety requirements	Validation of safety design at RTL level	Confirmation measure review	Confirmation measure review	
	Validate component safety requirements meet system safety requirements	Quantitative analysis of design (FMEDA)			
	Implement safety requirements in design specification	Incorporate findings into safety design			
	Validate design specification meets component safety requirements	Validation of safety design at gate/layout level			
	Confirmation measure review	Confirmation measure review			

图 **9.** 安全生命周期的赫丘利斯修改

B.3 TI 执行的活动

DIA 所覆盖的 TI 微控制器产品有开发为环境安全元件的硬件组件。 这样，TI 的安全活动集中在那些与实用安全和硬件组件开发管理相关的方面。 系统级架构、设计、和安全分析不在 TI 活动的范围之内，而是由 TI 客户负责。

表 3. 由 TI 执行的活动与 SEooC 客户执行的活动间的关系

安全生命周期活动	TI 执行	SEooC 客户执行
实用安全的管理	支持	支持
端设备和项目的定义	不支持	支持
端设备/项目的危险和风险分析	不支持	支持
端设备安全概念的开发	做出的假定	支持
子系统、硬件组件、和软件组件的端设备需求分配	做出的假定	支持
MCU 安全要求的定义	支持	不支持
MCU 架构和设计执行	支持	不支持
MCU 级别安全分析	支持	不支持
MCU 级别验证和确认	支持	否
将 MCU 整合进端设备	提供的支持	支持
端设备级别安全分析	不支持	支持
端设备级别验证和确认	不支持	支持
端设备级别安全评估	提供的支持	支持
端设备批量生产	不支持	支持
生产中安全事务的管理	提供的支持	支持

B.4 将被交换的信息

在一个定制开发中，在 IEC 61508 和 ISO 26262 中有一个例外，即所有与工作产品相关的开发文档都提供给客户。 在一个 COTS 产品中，这个方法是不可持续的。 TI 已经将大多数关键开发项目总结进一系列文档，这些文档或者公开提供给客户，或者在保密协定 (NDA) 下提供给客户。 为了保护在特定安全文档中披露的私有和敏感信息，NDA 是必需的。

表 4概括了产品安全文档，TI 可将这些文档提供给客户以帮助他们开发安全系统。

表 4. 产品安全文档

可交付使用的名称	内容	机密性	供货
安全产品预览	产品开发和产品架构的安全考虑概述。 在公开产品声明之前发布。	需要 NDA	由于安全手册的可获得性，在上市后从文档中删除。
安全手册	针对产品的安全特性的用户指南，包括系统级使用假定	公开的，无需 NDA	可供货
《用于 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要》(SPNU521)	按照系统级 ISO 26262 和/或者 IEC 61508 的 FIT 速率和器件安全标准汇总	需要 NDA	可供货
《用于 RM48x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要》(SPNU522)			
《用于 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的详细安全分析报告》(SPNU523)	所有可用安全分析 - FMEA, FTA, FMEDA, ... -的结果采用允许使用定制标准进行计算的记录格式	需要 NDA	开发中
《用于 RM48x 赫丘利斯 ARM 安全微控制器的详细安全分析报告》(SPNU527)			
安全案例报告（更多信息，请与您的 TI 销售代表联系）	产品符合 ISO 26262 和/或者 IEC 61508 标准的摘要	需要 NDA	开发中

表 4. 产品安全文档 (continued)

可交付使用的名称	内容	机密性	供货
安全案例数据库（更多信息，请与您的 TI 销售代表联系）	符合 ISO 26262 和/或者 IEC 61508 标准的逐条细节	需要 NDA	开发中

B.5 对安全活动负责的参与方

TI 采用一个交叉功能方法进行安全相关开发。 与安全相关的活动由多位程序经理、安全经理、应用工程师、设计工程师、和其他开发和产品工程职能部门执行。 每个人参与每一项活动的细节、他们的工作分工、能力证明等等将按照安全案例数据库中的标准要求的那样被永久保存。

B.6 目标值的通信

由于赫丘利斯 MCU 产品被开发成环境安全元件，在 MCU 设计期间，没有系统开发人员参与，为 MCU 开发提供目标标准。 在开发开始时，TI 对可能正确的 MCU 级安全 目标值做出假设并设计产品以满足这些目标值。 《*针对 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要*》(SPNU521) 和《*针对 RM48x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要*》(SPNU522) 和详细安全分析报告可被用于评估已实现的安全标准。 系统集成人员负责确定 TI 组件是否适合在系统中使用。

B.7 支持过程和工具

TI 使用多种工具和相应的数据格式用于内部和外部文档。 和 SEooC 客户共享的与安全文档相关的工具和数据格式在表 5中进行了注释。

表 5. 产品安全文档工具和格式

可交付使用的名称	创建工具	输出格式
安全产品预览	Microsoft® 字	Adobe™ PDF
安全手册	XML	Adobe PDF
《*用于 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要*》(SPNU521)	XML，Microsoft Excel®	Adobe PDF
《*用于 RM48x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要*》(SPNU522)		
《*用于 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的详细安全分析报告*》(SPNU523)	XML，Microsoft Excel	Adobe PDF，Microsoft Excel 2003
《*用于 RM48x 赫丘利斯 ARM 安全微控制器的详细安全分析报告*》(SPNU527)		
安全案例报告（更多信息，请与您的 TI 销售代表联系）	IBM® DOORS®，XML	Adobe PDF
安全案例数据库（更多信息，请与您的 TI 销售代表联系）	IBM DOORS，XML，Microsoft Excel	Adobe PDF，Microsoft Excel 2003

B.8 供货商危险和风险评估

IEC 61508 和 ISO 26262 下的危险和风险评估针对系统抽象级。 当开发一个环境硬件组件时，系统工具未知。 因此，TI 并未执行一个系统危险和风险分析。 作为替代措施，TI 对提供给组件设计的危险和风险分析的结果进行了假设。 系统集成人员负责确定 TI 组件是否适合在系统中使用。

B.9 实用安全概念的创建

IEC 61508 和 ISO 26262 下的实用安全概念针对系统抽象级。 当开发一个环境硬件组件时，系统工具未知。 因此，TI 不能生成一个系统实用安全概念。 作为替代措施，TI 对系统实用安全概念的输出做出假设并将这个数据提供给组件设计。 系统集成人员负责确定 TI 组件是否适合在系统中使用。

Appendix C 修订历史记录

由于下列的技术改变，这个文档的版本已经从 SPNU511 修改为 SPNU511A。

表 **6.** 修订

位置	添加、删除、和编辑
1 节	安全文档的更新说明表明出安全分析报告和安全案例报告已经被分至多个文档中
全篇文档	对于 ISO 26262 的更新参考表明为 2011 年 11 月发布的 IS 版本而非 FDIS 版本
Appendix B	删除了部分 3.5；相关数据已为最新Appendix B- DIA
5 节	进行了更新，与《用于 TMS570LS31x/21x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要》(SPNU521) 和 3) 《用于 RM48x 赫丘利斯 ARM 安全微控制器的安全分析报告摘要》保持一致
5.2 节 - 5.4 节	添加了针对 PMM，时钟，和复位的建议
5.6 节	为 ESM 创建了全新的部分
5.7 节 - 5.11 节	添加了针对 CPU，闪存，FEE，SRAM，L2/L3 的建议
5.14 节 - 5.15 节	添加了针对 IOMM，VIM 的建议
5.17 节 - 5.18 节	添加了针对 DMA，N2HET 的建议
5.20 节 - 5.29 节	添加了针对 MibSPI， SPI， I2C， SCI， LIN， DCAN， FlexRay， GIO， 以太网， USB 的建议
5.32 节 - 5.35 节	添加了针对 CPU 调试/跟踪，DMM，RTP，ROM 的建议
Appendix A	添加了全新的部分对来自5 节的建议进行了汇总。
Appendix B	针对通用 SEooC DIA 的新部分，已将之前版本中的部分 3.5 作为开始
Appendix C	新部分已被添加至修订历史中

Functional Safety Disclaimer for Safety Critical Solutions

TI's safety critical solutions, including integrated circuits, software and tools help TI's customers create end products that may be used in appropriately designed safety-critical applications to comply with functional safety standards or requirements.

Buyers represent and agree that they have all the necessary expertise to design, manage and assure effective system-level safeguards to anticipate, monitor and control system failures in safety-critical applications. Buyers agree and accept sole responsibility to meet and comply with all applicable regulatory standards and safety-related requirements concerning their systems and end-products which use TI's safety-critical applications. Buyers will fully indemnify TI and its representatives against any damages arising out of the use of TI products in safety-critical applications.

TI integrated circuits are not authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

重要声明

德州仪器(TI) 及其下属子公司有权在不事先通知的情况下, 随时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，并有权随时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 保证其所销售的硬件产品的性能符合TI 标准保修的适用规范。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。除非政府做出了硬性规定，否则没有必要对每种产品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 产品或服务的组合设备、机器、流程相关的TI 知识产权中授予的直接或隐含权限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，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。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售TI 产品或服务时，如果存在对产品或服务参数的虚假陈述，则会失去相关TI 产品或服务的明示或暗示授权，且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

TI 产品未获得用于关键的安全应用中的授权，例如生命支持应用（在该类应用中一旦TI 产品故障将预计造成重大的人员伤亡），除非各方官员已经达成了专门管控此类使用的协议。购买者的购买行为即表示，他们具备有关其应用安全以及规章衍生所需的所有专业技术和知识，并且认可和同意，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在关键安全应用中使用其产 品及TI 产品所需的所有法律、法规和安全相关要求。此外，购买者必须全额赔偿因在此类关键安全应用中使用TI 产品而对TI 及其 代表造成的损失。

TI 产品并非设计或专门用于军事/航空应用，以及环境方面的产品，除非TI 特别注明该产品属于“军用”或“增强型塑料”产品。只 有TI 指定的军用产品才满足军用规格。购买者认可并同意，对TI 未指定军用的产品进行军事方面的应用，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

TI 产品并非设计或专门用于汽车应用以及环境方面的产品，除非TI 特别注明该产品符合ISO/TS 16949 要求。购买者认可并同意，如果他们在汽车应用中使用任何未被指定的产品，TI 对未能满足应用所需要求不承担任何责任。

可访问以下URL 地址以获取有关其它TI 产品和应用解决方案的信息：

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

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