ICS 43.040
CCS T 35

中华人民共和国国家标准

GB/T 34590.11—XXXX

道路车辆 功能安全 第11部分：半导体应用指南

Road vehicles—Functional safety—Part11:Guidelines on applications to semiconductors

(ISO 26262-11:2018,Road vehicles—Functional safety—Part11:Guidelines on applications of ISO26262 to semiconductors,MOD)

（征求意见稿）

（本草案完成时间：2021年4月1日）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

国家市场监督管理总局
国家标准化管理委员会　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

GB/T 34590-XXXX《道路车辆 功能安全》分为以下部分：

——第 1 部分：术语；

——第 2 部分：功能安全管理；

——第 3 部分：概念阶段；

——第 4 部分：产品开发：系统层面；

——第 5 部分：产品开发：硬件层面；

——第 6 部分：产品开发：软件层面；

——第 7 部分：生产、运行、服务和报废；

——第 8 部分：支持过程；

——第 9 部分：以汽车安全完整性等级为导向和以安全为导向的分析；

——第 10 部分：指南；

——第 11 部分：半导体应用指南；

——第 12 部分：摩托车的适用性。

本部分为GB/T 34590-XXXX的第11部分。

本部分修改采用ISO 26262-11：2018 《道路车辆 功能安全 第11部分：ISO 26262对半导体的应用指南》。

本部分与ISO 26262-11：2018的技术性差异及其原因如下：

——关于规范性引用文件，本部分做了具有技术性差异的调整，以适应我国的技术条件，调整的情况集中反映在第 2 章“规范性引用文件”中，具体调整如下：

——用修改采用国际标准的 GB/T 34590-XXXX.1-XXXX 代替 ISO 26262-1:2018；

本文件做了下列编辑性修改：

——将国际标准中的“本国际标准”改为“本文件”；

——删除国际标准的前言；

——修改国际标准的引言及其表述。

本文件由中华人民共和国工业和信息化部提出。

本文件由全国汽车标准化技术委员会（SAC/TC114）归口。

本文件起草单位：

本文件主要起草人：

引　言

ISO 26262是以IEC 61508为基础，为满足道路车辆上电气/电子系统的特定需求而编写。

GB/T 34590修改采用ISO 26262，适用于道路车辆上由电子、电气和软件组件组成的安全相关系统在安全生命周期内的所有活动。

安全是道路车辆开发的关键问题之一。汽车功能的开发和集成强化了对功能安全的需求，以及对提供证据证明满足功能安全目标的需求。

随着技术日益复杂、软件和机电一体化应用不断增加，来自系统性失效和随机硬件失效的风险逐渐增加，这些都在功能安全的考虑范畴之内。GB/T 34590通过提供适当的要求和流程来降低风险。

为了实现功能安全，GB/T 34590-XXXX（所有部分）：

a) 提供了一个汽车安全生命周期（开发、生产、运行、服务、报废）的参考，并支持在这些生命周期阶段内对执行的活动进行剪裁；

b) 提供了一种汽车特定的基于风险的分析方法，以确定汽车安全完整性等级（ASIL）；

c) 使用 ASIL 等级来定义 GB/T 34590 中适用的要求，以避免不合理的残余风险；

d) 提出了对于功能安全管理、设计、实现、验证、确认和认可措施的要求；及

e) 提出了客户与供应商之间关系的要求。

GB/T 34590针对的是电气/电子系统的功能安全,通过安全措施(包括安全机制)来实现。它也提供了一个框架，在该框架内可考虑基于其它技术（例如，机械、液压、气压）的安全相关系统。

功能安全的实现受开发过程（例如，包括需求规范、设计、实现、集成、验证、确认和配置）、生产过程、服务过程和管理过程的影响。

安全问题与常规的以功能为导向和以质量为导向的活动及工作成果相互关联。GB/T 34590涉及与安全相关的开发活动和工作成果。

图1为GB/T 34590的整体架构。GB/T 34590基于V模型为产品开发的不同阶段提供参考过程模型：

——阴影”V”表示 GB/T 34590.3-XXXX、GB/T 34590.4-XXXX、GB/T 34590.5-XXXX、GB/T 34590.6-XXXX、GB/T 34590.7-XXXX 之间的相互关系；

——针对摩托车应用：

- GB/T 34590.12-XXXX，第8章对应GB/T 34590.3-XXXX；
- GB/T 34590.12-XXXX，第9章和第10章对应GB/T 34590.4-XXXX

——以“m-n”方式表示的具体章条中，“m”代表特定部分的编号，“n”代表该部分章的编号。

示例：“2-6”代表 GB/T 34590.2-XXXX 的第 6 章。

——阴影”V”表示 GB/T 34590.3-XXXX、GB/T 34590.4-XXXX、GB/T 34590.5-XXXX、GB/T 34590.6-XXXX、GB/T 34590.7-XXXX 之间的相互关系；

——对于摩托车：

- GB/T 34590.12-XXXX的第8章支持GB/T 34590.3-XXXX；
- GB/T 34590.12-XXXX的第9章和第10章支持GB/T 34590.4-XXXX。

——以“m-n”方式表示的具体章条中，“m”代表特定部分的编号，“n”代表该部分章的编号。

示例： “2-6”代表 GB/T 34590.2-XXXX 的第 6 章。

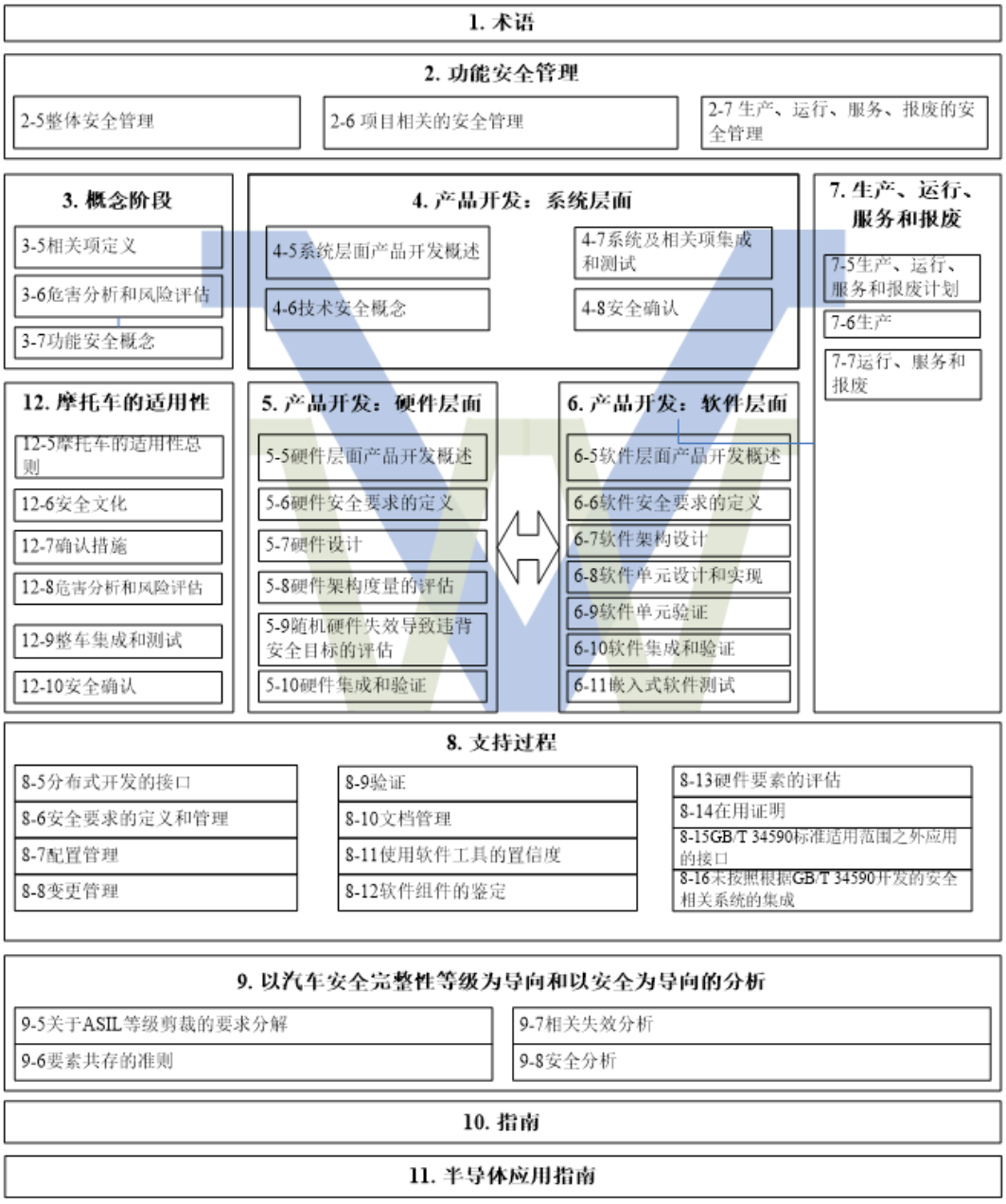


图1　GB/T 34590-XXXX 概览

道路车辆 功能安全 第11部分：半导体应用指南

1 范围

本文件适用于安装在除轻便摩托车外的量产道路车辆上的包含一个或多个与安全相关的电子电气系统。

本文件不适用于特殊用途车辆上特定的电子电气系统，例如，为残疾驾驶者设计的车辆。

其他专用的安全标准可作为本文件的补充，反之亦然。

已经完成生产发布的系统及其组件或在本文件发布日期前开发的系统及其组件不适用于本文件。于在本文件发布前完成生产发布的系统及其组件进行变更时，仅修改的部分需要按照本文件开发并进行安全生命周期的裁剪。未按照和按照本文件正在进行开发的系统进行变更时，仅修改的部分需要按照本文件开发并进行安全生命周期的裁剪。

本文件针对由电子电气安全相关系统的故障行为而引起的可能的危害，包括这些系统相互作用而引起的可能的危害。本文件不针对与触电、火灾、烟雾、热、辐射、毒性、易燃性、反应性、腐蚀性、能量释放等相关的危害和类似的危害，除非危害是直接由电子电气安全相关系统的故障行为而引起的。

本文件提出了安全相关的电子电气系统进行功能安全开发的框架，应将此框架内的功能安全活动整合到企业的整体开发体系中。本文件规定了为实现产品功能安全的技术开发要求，也规定了组织应具备相应功能安全能力的开发流程要求。

本文件不涉及电气/电子系统的标称性能。

本文件只具有资料性特性，包含了GB/T 34590其他部分针对半导体开发的可能解释。关于可能的解释，该内容并非详尽无遗，即为了满足GB/T 34590的其他部分中定义的要求，其他解释也是可能的。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 34590.1-XXXX 道路车辆 功能安全 第1部分：术语（ISO 26262-1:2018，MOD）

3 术语和定义

GB/T 34590.1-XXXX界定的术语和定义适用于本文件。

4 半导体组件及其分区

4.1 如何考虑半导体组件

4.1.1 半导体组件开发

如果半导体组件的开发作为一个符合GB/T 34590的相关项开发的一部分，它需基于硬件安全要求开发，此要求源于相关项的顶层安全目标，并通过技术安全概念产生。如果分配给该相关项的目标包括为满足硬件架构度量而设定的相关失效模式的诊断覆盖率，随机硬件失效概率度量（PMHF）或对违反安

全目标的每个原因的评估（ECC）：在这种情况下，半导体组件只是要素之一。 如 GB/T 34590.5-XXXX [66]，8.2 的示例所述，为了促进分布式开发，可以通过在相关项层面获得SPFM，LFM和PMHF的目标值或将EEC应用于硬件元器件层面，从而将这些目标值分配给半导体组件本身。半导体组件的安全分析是根据GB/T 34590.5-XXXX,7.4.3和GB/T 34590.9-XXXX [70] 的第8章中所定义的要求和建议进行的。

注：如果要素未按照GB/T 34590开发，则可以参考GB/T 34590.8-XXXX [69]第13章中的要求。

半导体组件可以按照SEooC来开发，如 GB/T 34590.10-XXXX [61]中所述。 在这种情况下，开发是基于对半导体组件使用条件的假设（使用假设或AoU，参见4.4）完成的，然后在下一个更高的集成层面上，考虑这些将要使用半导体组件的相关项的安全目标导出的半导体组件的要求，进行验证。

该部分的表述和方法都是假定这个半导体组件是一个SEooC，但是如果半导体组件未被视为一个SEooC，则所表述的方法（例如，半导体组件的失效率的计算方法）仍然有效。 考虑半导体组件自身而实施这些方法时，应当给出适当的假设。 第4.4条描述了如何在系统层面或要素层面调整和验证这些方法和假设。 在半导体组件自身层面， GB/T 34590.2-XXXX [63]，GB/T 34590.5-XXXX，GB/T 34590.6-XXXX [67]，GB/T 34590.7-XXXX [68]，GB/T 34590.8-XXXX 和GB/T 34590.9-XXXX的要求（例如，与安全分析，相关失效分析，验证等相关）可以应用。

4.2 半导体组件划分

如图2所示并且按照GB/T 34590.1-XXXX,3.21中的定义，半导体组件可分为数个元器件： 整个半导体层级可视为一个组件，第二层级（例如CPU）可视为一个元器件，下一层级（例如CPU寄存器组）可视为子元器件，直到基础子元器件（其内部寄存器和相关逻辑）。

注：详细程度（例如，是否停止于元器件层面或下至子元器件或基础子元器件层面）以及基础子元器件的定义（例如触发器，模拟晶体管）可取决于安全概念、分析的阶段以及使用的安全机制（在半导体组件内部或在系统层面或要素层面）。

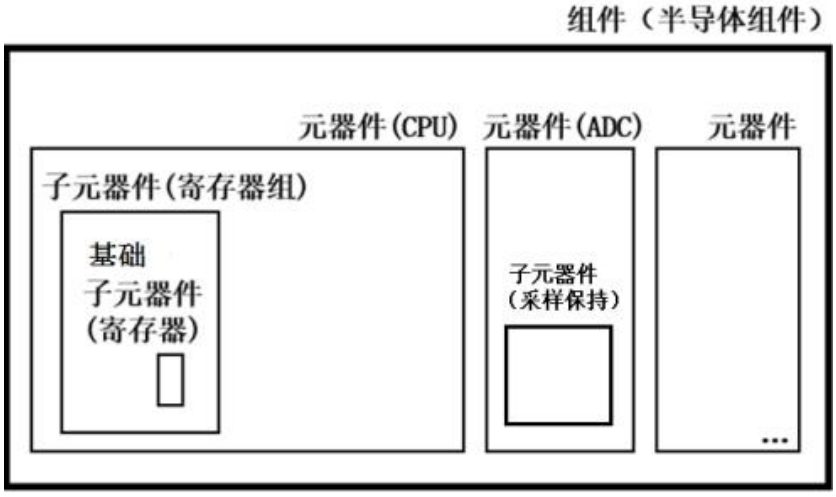


图2 半导体、元器件和子元器件

4.3 关于硬件的故障、 错误和失效模式

4.3.1 总述

如下图3所示， 集成电路的随机硬件故障和失效模式相互关联。

注1：失效模式可以是抽象的，也可根据具体实施情况而裁剪，例如与组件、元器件或子元器件的引脚相关。

通常情况下，失效模式在本部分中被描述为功能失效模式。 也可进一步表征失效模式的特性。

示例：附录 A 中给出了数字电路失效模式的示例。

在本部分中描述的故障和错误与给定半导体组件的物理实现有关。

注2：故障、 错误和失效的术语是按照 GB/T 34590.1中的定义而使用的，即故障会产生错误从而会导致失效的产生。 在许多可靠性建模标准中，故障和失效这两个术语可互换使用。

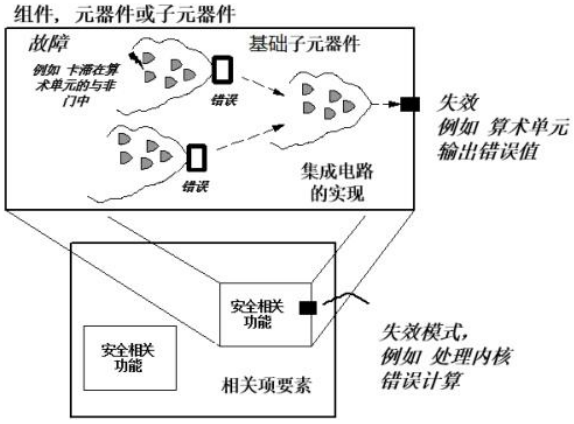


图3　硬件故障和失效模式的关系

4.3.2　故障模型

故障模型是物理故障的抽象表达。

失效模式的分布与图3中所示的故障模型相关联。

示例：如果某个失效模式 X%是由于卡滞故障和 Y%是由于短路而造成的，并且如果安全机制仅以 Z%的覆盖率覆盖卡滞故障，则，声明的诊断覆盖率为 X%×Z%。

在半导体组件的背景下，基于工艺和电路实现来确定相关的故障模型。

注1：有关数字组件故障模型的更多详细信息，请参见5.1.2；对于存储器，参见5.1.3。

注2：通常情况下，由于故障数量和所需的详细程度，无法单独评估每个可能的物理故障。

4.3.3　失效模式

失效模式采用与安全概念和相关的安全机制相符的详细程度来描述。

示例1：如果一个 CPU 具有硬件锁步安全机制，则可将 CPU 功能视为整体来定义失效模式。

示例2：如果 CPU 具有基于结构化软件的硬件测试作为安全机制，则 CPU 功能的失效模式可被更详细地定义，因为软件测试将以不同的失效模式覆盖率来覆盖不同的失效模式。

示例3：附录 A 中给出了数字失效模式不同详细程度的示例。

如果适用，可用关键字定义失效模式。

示例4：关键字的示例包括：程序流执行错误、数据损坏、访问非预期位置、死锁、活锁、指令执行错误。

在特殊情况下，更接近物理实现的失效模式可能会更有帮助。

示例5：模拟失效模式（表 36）。

有证据表明，识别出的失效模式与电路实现故障模型之间存在关联， 这确保将所有失效模式分配给组件的元器件/子元器件的同时，所有相关的元器件/子元器件至少有一种失效模式。

注：目标是确保在电路实现与已列出的失效模式之间没有差距。

4.3.4　失效模式下的基础失效率的分布

基础失效率（参见4.6）分布在失效模式中。这种分布的准确性与分析的详细程度以及对可用的有关安全机制的考虑是一致的。

示例1：如果 CPU 具有硬件锁步安全机制，则详细的 CPU 失效模式分布不是必需的。

示例2：如果 CPU 有一个基于结构化软件的硬件测试，则分布将会被更详细地定义，因为这种方式有可能足够精确地估计失效模式的诊断覆盖率。

在以规定精度来计算分布时，若无可用数据，则失效率在失效模式中均匀分布，或应提供带有相关论据的专家判断。

注：对分布进行敏感度分析，以评估对诊断覆盖率和定量安全分析结果产生的影响。

4.4 关于使半导体组件安全分析适应系统层面

使半导体组件安全分析适应系统层面，可通过以下方式完成：

——将半导体组件的详细失效模式转换为在系统层面分析期间所需的高层面失效模式，如图 4 所示；

图4　自下而上方法推导系统层面失效模式的示例

注1：通过结合自上而下（例如：FTA）和自下而上的方法（例如： FMEA），可确定详细的半导体组件失效模式并将它们整合到组件层面。

注2：从低抽象层面开始，可为半导体组件提供定量和精确的失效分布，否则将基于定性分布的假设。

注3：如4.2中所述，必要的详细程度可取决于分析的阶段和所使用的安全机制。

——通过在元器件层面、组件层面或系统层面或相关项层面采取措施，可以提高在元器件层面或子元器件层面计算出的诊断覆盖率； 或

示例1：一个半导体组件包含一个 ADC，此 ADC 没有由硬件实现的安全机制。 在组件自身层面，认为诊断覆盖率为零。 在系统层面，ADC 被包含在一个闭环中，其故障可通过基于软件的一致性检查来探测。 在这种情况下，由于在系统层面实施的安全机制，该子元器件的诊断覆盖率得到增加。

——在元器件或子元器件层面计算得出的诊断覆盖率可能可以在某些特定的假设（“使用假设”或 AoU）下被计算出来。

注：在系统层面，可呈现不同的安全机制或失效屏蔽。 当能给出正当理由时，可在安全分析中考虑这一点。

示例2：半导体组件包含一个存储器，ECC 修正其每个单一错误并通知 CPU。 在组件自身层面，假设已实现软件驱动程序来处理此事件。在系统层面，出于性能原因，此软件驱动程序无法实现，因此该假设并未满足。 因此，通过对半导体组件的编程，将错误更正标志直接发送到外界。

4.5 知识产权（IP）

4.5.1 关于 IP

4.5.1.1 理解 IP

在这一条中，IP是指可复用的逻辑设计单元或物理设计单元，作为一个元器件或组件集成到设计中。术语“IP集成商”指负责将来自一个或多个源的IP设计集成到具有安全要求的设计中的组织。 术语“IP

供应商”用于指负责设计或开发IP的组织。 IP集成商和IP供应商可以是单独的各方，也可以是同一公司或同一公司的不同组织。

根据GB/T 34590的要求，为基于IP的设计确定了四种可能的方法。 这些方法如图5所示。 IP集成商通常根据对IP供应商提供的信息以及IP的成熟度的考量来选择方法。

示例：如果无法从 IP 供应商处获得支持信息，可能的方法可以仅限于“在用证明”论据（如果适用）。若“在用证明”不适用，那么 IP 在安全架构中的作用将被区别对待，例如： 使用多样化冗余来降低系统性和硬件随机失效的风险。

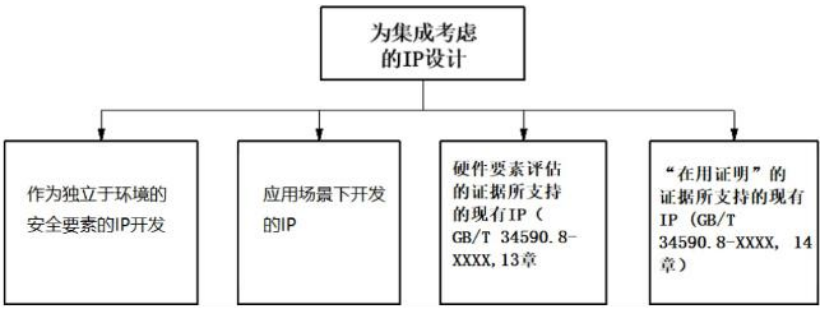


图5　在安全相关设计中使用 IP 可能用到的方法

IP可以是具有预定义功能集的现有设计。 在这种情况下，IP集成商有责任确定支持设计的安全概念所需的功能集。 IP也可根据议定的安全要求进行设计。 在这种情况下，IP集成商确定了支持设计的安全概念所必要的IP要求。

注1：此条中的指导可适用于新开发的IP、修订的IP和现有的未修改的IP。

注2：如GB/T 34590.2-XXXX,6.4.5.7中定义，通常的方法是假设可能的目标用途。 该方案在GB/T 34590.10 [61]中被描述为SEooC。 SEooC的开发依赖于识别由IP集成商验证的假设用例和安全要求。

4.5.1.2　IP 的类型

表1中列出了常用的IP类型。该表并未涵盖所有可能的IP类型。 本部分考虑了应用于半导体设计的IP的物理和模型表示类型。

表1　IP 类型

IP 类型	描述
物理表示	一个完整的芯片布局描述，包含特定单元库的标准单元实例或者目标制造过程中的模拟单元。 **示例：**A / D转换器宏，PLL宏。
模型表示	根据硬件描述语言（HDL）如Verilog或VHDL，或模拟晶体管级电路原理图来进行的设计描述。 模型表示中的逻辑设计被综合为由基本单元组成的门阵列，随后是布置和布线以实现半导体设计。 模拟电路原理图组件，例如晶体管，二极管，电阻器和电容器，映射到目标技术库组件中，然后进行布局和布线，以实现半导体设计。 **示例：**处理器或存储器控制器设计转换而不映射到特定工艺，运算放大器晶体管级示意图。
注1：物理表示IP也称为“硬IP （hard IPs）”。 **注2：**模型表示IP也称为“软IP（soft IPs）”。 **注3：**分类适用于通用IP设计，包括数字，模拟，混合信号，PLD，传感器和转换器。	

注1：逻辑设计形式的IP也是可配置的。 在这种情况下，配置选项由IP集成商指定。

示例1：配置选项用于定义接口总线宽度、存储器大小和是否选用故障探测机制。

注2：IP也可以使用专用工具（存储器编译器、C到HDL编译器、片上网络生成器）生成。 在这种情况下：

——软件工具的可信度可以通过使用 GB/T 34590.8-XXXX，第 11 章中描述的方法来证明，此方法的裁剪基于在已生成 IP 上执行的验证量；

——在适用时，IP 集成商或 IP 供应商执行必要的验证活动以确保生成 IP 的正确性（例如 DIA 中的协议）；

——提供下列章节所列的必要工作成果；及

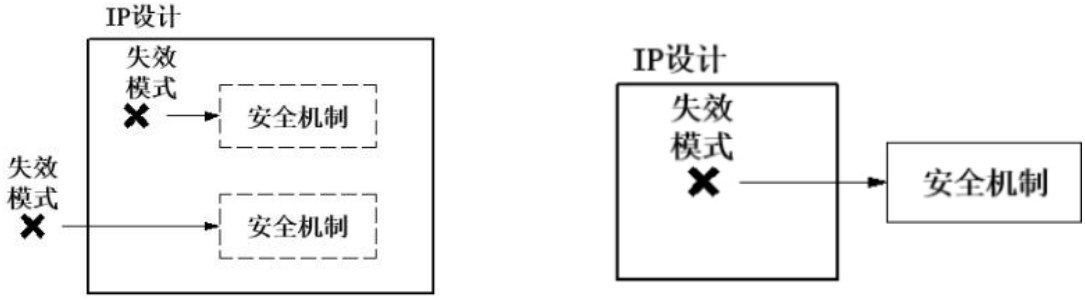
——IP 集成商验证了 IP 在这种场景中的正确集成。

4.5.2 IP 的类别和安全要求

通常情况下，可以基于对安全要求的分配来确定两类IP：没有分配安全要求的IP和分配有一个或多个安全要求的IP。当IP没有分配安全要求时，除非在安全分析时被识别出，否则GB/T 34590对此没有要求附加的考虑。在非安全相关IP与安全相关要素共存的情况下，可使用相关失效分析来评估是否免于干扰。对于相关失效分析指南，请参见GB/T 34590.9-XXXX 第7章以及本部分4.7中的附加指南。

如果为IP分配了一个或多个安全要求，则GB/T 34590的要求依然适用。 特别是GB/T 34590.2-XXXX，GB/T 34590.4-XXXX [65]，GB/T 34590.5-XXXX，GB/T 34590.8-XXXX和GB/T 34590.9-XXXX的要求通常被裁剪以适用于IP设计。 以下内容为分配了安全要求的IP提供了指导，以及如何在有和没有集成安全机制的情况下考虑这些IP要求。

安全相关的IP可以基于安全机制的集成进行进一步分类。图6中图解说明了两种可能的情况，其中，子图（a）说明了集成了安全机制的IP，而子图（b）说明了没有集成安全机制的IP。



a) b)

图6 IP 类型与分配的安全要求

注1：IP安全机制可包括，探测IP自身的失效模式，以及探测IP外部的失效模式。

注2：IP内实施的安全机制可以针对一组已定的失效模式，提供完整或部分诊断覆盖率。也可能仅由IP执行失效模式探测，而失效模式控制由IP外部的组件提供。

IP供应商负责提供IP开发期间提出的使用假设，以便IP集成商检查与安全要求的一致性。

为了将IP集成到安全相关的硬件环境中，IP的硬件功能最初可以通过提供基于假设的安全要求的安全机制来开发，该安全机制旨在控制给定的失效模式。 在这种情况下，GB/T 34590.2-XXXX、GB/T 34590.4-XXXX、GB/T 34590.5-XXXX、GB/T 34590.6-XXXX（在基于软件的安全机制覆盖硬件失效的情况下）、GB/T 34590.8-XXXX以及GB/T 34590.9-XXXX的要求，适用时可以在IP的开发过程中用于安全机制的设计。

示例1：具有内置总线监控器的总线“结构”，包括故障探测和通知逻辑（例如，中断信号）。

示例2：带有监控（欠压和过压探测）、保护（限流或热保护）和自诊断（内置自检的监控和保护电路）的电压调节器

或者，可以在没有假定安全要求、或没有特定的用于探测和控制故障的安全机制的情况下，进行IP的开发。

示例3：无内置总线监控器或错误报告逻辑的总线“结构”。

示例4：没有监控、保护或内置监控或保护电路诊断的电压调节器。

定义于GB/T 34590.9-XXXX，第8章中的安全分析可应用于IP。定性安全分析和在某些情况下的定量安全分析可以提供给IP集成商，以证明安全机制控制已知失效模式的能力，或提供失效模式及其分布的信息。类似地，可以提供相关失效分析以证明所需的独立性或免于干扰。

注3：IP供应商根据特定的实施假设，将有关失效模式分布的示例信息计入安全分析结果。 与安全机制相关的文档可以与IP的其他安全相关文档一起提供。这些信息也可以合并成单个的安全手册或安全应用说明，如5.1.11（用于数字组件），5.2.6（用于模拟或混合信号组件），5.3.6（用于PLD）和5.5.6（用于传感器/转换器）中所述。

注4：如4.6中所述，基础失效率取决于在集成电路里IP的实际实现（包括IP技术）和集成电路的使用条件。因此，对于负责按照实际用例重新计算失效率的IP集成商来说， 基础失效率仅能作为参考。

注5：该信息可包含在现有文档中（例如集成指南、技术参考文档、应用说明）。

IP集成商可以在执行安全要求时向IP供应商请求附加信息。 IP供应商可以通过提供有关用于避免系统性故障的措施的信息以及安全分析结果来支持该请求。 安全分析结果可用于支持对集成IP的硬件度量的评估，也可用于证明免于干扰和独立性。

由于IP将被集成到与安全相关的设计中，因此考虑共存对于确保被集成的IP不会对其他安全相关功能产生不利影响非常重要。 为了声明免于干扰，可以使用如GB/T 34590.9-XXXX，第6章和GB/T 34590.9-XXXX，第7章中所述的相关失效分析，以及本部分4.7中的附加指南。

如果IP集成商判定，使用提供的IP无法满足安全要求，则可以按照GB/T 34590.8-XXXX,5.4.4中的所述对供应商提出更改请求，当IP是SEooC时，还需参考标准GB/T 34590.10-XXXX [61]。或者，IP集成商可以采取符合安全要求的其他措施，如集成层面的附加安全机制。 安全机制可以用硬件、软件或两者的结合来实现。如果合规开发缺少证据，GB/T 34590.8-XXXX，第13章和GB/T 34590.8-XXXX，第14章可以提供替代方法以论证符合性。

IP集成商负责与所分配的安全要求和安全机制有关的每次集成，以及相关的验证和测试活动（如果适用）。

注6：IP供应商负责确保交付符合指定的属性，并避免在生成的IP中有系统性故障。此外，IP供应商提供支持信息，以允许IP集成商进行集成活动。

4.5.3 IP 生命周期

4.5.3.1 介绍

在IP生命周期中，避免和探测系统性故障是为了确保最终设计适合在具有一个或多个分配安全要求的应用中使用。在硬件设计的背景下，GB/T 34590.5-XXXX，第7章中提供了避免和探测系统性故障的要求。 在本部分中，5.1.9（对于数字组件），5.2.5（对于模拟或混合信号组件），5.3.5.3（对于PLD）和5.5.5（对于传感器和转换器）提供了进一步的指导。该指南可用于确定在IP开发期间，用于避免和探测系统性故障的一般方法。

对于具有可编程行为的IP，则可参考GB/T 34590.4-XXXX,6.4.6.5，同样可参考 5.3中描述的指南。

IP集成商负责集成所提供的IP。 对于集成活动，应考虑IP使用假设，以及描述IP集成指导原则。当使用假设无法实现或无效时，集成IP的设计根据GB/T xxx 第8章所述的变更管理对其影响进行分析和考虑。图7提供了一个基于SEooC开发的生命周期示例，如GB/T 34590.10-XXXX [61]中已有描述。

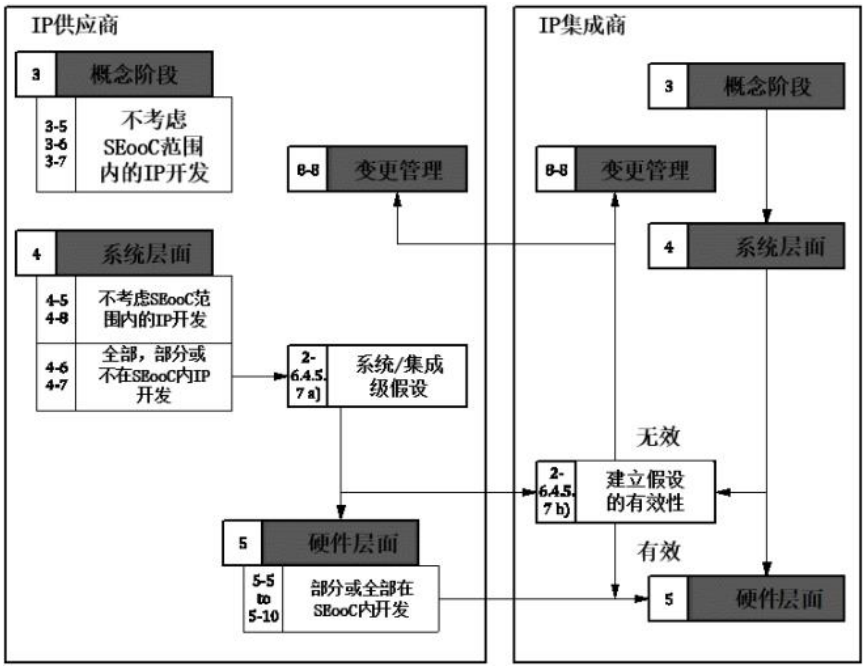


图7　当 IP 作为 SEooC 对待时 IP 的生命周期

注1：如图7中所示的参考文献与GB/T 34590-XXXX有关。

注2：在图7中，GB/T 34590.5-XXXX，第10章只有部分是由IP供应商负责，因为许多相关要求不适用于IP供应商，例如ESD测试。

DIA可以定义将由IP供应商提供的工作成果（如4.5.4中所列），以支持IP集成商进行IP集成活动。

4.5.3.2　IP 作为 SEooC

在开发SEooC IP的时候，根据GB/T 34590.2-XXXX， 6.4.5.7中的描述裁剪适用的安全活动。这种针对SEooC开发的裁剪并不意味着可以省略任何一个安全生命周期的步骤。如果在SEooC开发期间某些步骤被推迟，则可以在相关项开发期间完成这些步骤。

如果SEooC ASIL等级能力（参见GB/T 34590.1-XXXX,3.2）和由IP集成商指定的ASIL等级要求不匹配,则IP集成商可以采用IP外部附加的安全机制。也应将关于避免系统性失效的附加安全措施考虑在内。可以使用GB/T 34590.9-XXXX第5章中定义的ASIL等级分解，前提是可以表明存在冗余和独立的要求，并考虑了集成IP的系统性失效避免和控制方法。

SEooC是基于预期功能和使用场景(包括外部接口)的假设开发的。这些假设的设置方式可以将SEooC集成到其中的组件的超集，从而可以在以后的多个不同设计中使用SEooC。这些假设的有效性是在集成SEooC的实际组件的场景下建立的。在这种情况下，被开发为SEooC的IP通常可以配置为针对许多不同的设计。 配置可以在综合之前，综合之后，或通过熔丝、激光切割、闪光或任何其他编程方式来完成。在这种情况下，IP供应商提供有关测试和验证活动所涉及的IP配置的信息。

示例：用于确定互连总线的宽度，内部高速缓存大小，中断数，存储器映射的配置选项。

注1：IP配置与软件配置数据不同：因此GB/T 34590.6-XXXX，附录C不能直接地应用于IP。

注2：IP集成商执行必要的验证活动，以确保生成的IP的正确性； 在后续章节中所列出的必要的工作成果应是可用的； 并且，IP集成商验证IP在这种场景下的正确集成。

4.5.3.3　特定场景下进行 IP 设计

当在某场景下开发IP时，IP供应商根据GB/T 34590.2-XXXX,6.4.5.1中的描述裁剪安全活动。对基于场景的设计，IP供应商可以在了解安全要求的情况下开发IP。

示例：在系统层面的一个特定安全要求的场景下设计的模拟组件。

4.5.3.4 通过硬件要素评估的 IP 应用

如果没有SEooC或场景信息可用于IP，如GB/T 34590.8-XXXX，第13章中所述的硬件要素的评估可被用于增加对IP的信心。用于评估硬件要素的预知活动可以应用于没有预先可用的支持信息的IP（如4.5.5中所述）。

4.5.3.5 通过“在用证明”论证的 IP 应用

如果用以避免系统故障的证据不可用，则在GB/T 34590.8-XXXX第14章中所述的“在用证明”论据可以为IP集成商提供一种方法，表明符合GB/T 34590。

关于“在用证明”论证的有效性的条件可能是受限的。确保GB/T 34590.8-XXXX，14.4.5.3中所述的有效的现场监控程序的实施是具有挑战性的，因为通常来自于包含IP的设计的现场反馈有限，或IP配置有所差异。

4.5.4 IP 工作成果

4.5.4.1 IP 工作成果清单

工作成果示例在5.1.11（用于数字组件），5.2.6（用于模拟或混合信号组件），5.3.6（用于PLD）和5.5.6（用于传感器和转换器）中描述。后续给出了用于IP设计的有关工作成果内容的通用性指导。

注：DIA（参见GB/T 34590.8-XXXX，第5章）可用于定义哪些文档可供IP集成商使用，以及所包含的详细程度。

4.5.4.2 安全计划

对于分配了一个或多个安全要求的IP，安全计划是根据GB/T 34590.2-XXXX,6.4.6 中的要求制定的。 可以使用单个计划或多个相关计划。详细计划应纳入GB/T 34590.8-XXXX中描述的适用的支持过程，包括配置管理、变更管理、影响分析和变更请求、验证、文档管理和软件工具鉴定。

4.5.4.3 分配给 IP 设计的安全要求

如GB/T 34590.5-XXXX第6章中定义，硬件安全要求可以分配到IP设计。

示例：根据对在 IP 中安全机制要求的描述，允许在适当的集成层面验证要求。 集成和测试规范可以与定义于技术安全概念中的要求相关联。

4.5.4.4 IP 设计的硬件设计验证和验证评审

对于逻辑设计形式的IP设计，定义设计验证的准则，特别是针对环境条件（振动，EMI等）的准则通常是不可能的，因为物理特性高度依赖于由IP集成商完成的设计的物理实现。

注：对于用作数字逻辑设计的IP，可通过使用在5.1.9中列出的技术进行硬件设计验证。

验证报告包括用于验证IP设计的活动结果。 验证可以按照在GB/T 34590.8-XXXX第9章的描述进行，包括验证活动的计划、执行和评估。

4.5.4.5 安全分析报告

IP设计可采用GB/T 34590.9-XXXX第8章中的安全分析的要求。基于GB/T 34590.5-XXXX，表2，选择合适的安全分析方法。

对于定性分析，供应商提供已识别的IP的失效模式，以支持其集成。

对于定量分析，如GB/T 34590.9-XXXX,8.4.10中所定义的，所包含的数据支持硬件架构度量的评估和由于随机硬件故障导致违背安全目标的评估。

示例：数据包括估计的失效率和失效模式分布信息。

注1：对于IP的逻辑设计，比如寄存器传输级（RTL），定量分析取决于关于失效率和失效模式分布的假设，因而不能代表实际的物理设计。IP集成商验证针对特定实现的假设和定量安全分析结果。

注2：在评估度量时，可以考虑嵌入在IP中的安全机制及其预期的失效模式覆盖率（在适用于给定IP的等级上）。若IP可配置，则安全分析可包括关于配置选项对失效模式分布影响的信息。

注3：分析配置选项对安全机制的实施和诊断覆盖率的影响。

可以定义通过IP内部和外部特征的组合实现的安全机制，以及在IP外部实施的安全机制。 这些附加的安全机制可依赖于针对SEooC设计的使用假设，其可以在如GB/T 34590.2-XXXX,6.4.5.7中所述的适当层面进行确认。

4.5.4.6 相关失效分析

IP的相关失效分析可以按照GB/T 34590.9-XXXX，第7章所述进行。本部分的4.7中包含有关如何将相关失效分析应用于半导体器件的附加指导。

4.5.4.7 认可措施

实施认可措施的结果包括与IP开发流程相关以及关于避免系统性故障的证据和论据。GB/T 34590.2-XXXX，表1中描述了认可措施。对于半导体IP，通常的认可措施报告包括：

——安全计划的认可评审；

——安全分析的认可评审；

——安全档案的完整性的认可评审；及

——功能安全审核和评估报告。

5.1.9（数字组件）、5.2.5（模拟或混合信号组件）、5.3.5.3（PLD）和5.5.5（传感器和转换器）中包括了适用于避免系统性故障的IP开发活动的技术示例。

4.5.4.8 开发接口协议

GB/T 34590.8-XXXX，第5章中对分布式开发的要求适用于IP设计。 DIA定义了针对IP设计的用于交换的工作成果，以及IP供应商和IP集成商之间的安全相关的角色和责任。

4.5.4.9 集成文档集

集成文档集可以包括作为SEooC开发的IP的安全手册或安全应用说明。集成文档集还可以包含以下信息：

——IP 开发的生命周期裁剪的描述；

——IP 的使用假设，包括例如：

- IP的假定安全状态；
- 关于最大故障处理时间间隔和多点故障探测间隔（MPFDI）的假设（如适用）；
- 关于IP集成环境的假设，包括接口；及
- 建议的IP配置。

——安全架构的描述，包括：

- 故障探测和控制机制；
- 故障报告能力；
- 自检能力和关于潜伏故障的自检的附加要求，如果适用；
- 故障恢复机制，如果适用；及

- 配置参数对上述相关项的影响，如果适用。

——用于支持 IP 安全机制，以及探测后以控制失效所需的硬件 – 软件接口；

——用于探测 IP 组件故障的基于软件的测试例程的规范，如果适用。 这也可供作源代码或二进制库；

——IP 安全分析结果的描述；及

——IP 认可措施的描述。

IP集成商可以以正式的形式识别每一个与安全机制相关的硬件特征，以便可以在IP集成商层面上完成与硬件安全要求的映射，并且能够识别由IP集成商负责的集成验证和确认活动。

注1：IP安全机制的要求，以一种可以追溯到IP集成商要求的方式被指定。

注2：对于没有可用于故障探测的特定功能的IP，提供使用假设足以符合IP集成商的要求。

对于在某场景下进行开发的IP，通常提供类似的文档。

注3：对于在应用场景下的IP，不需要使用假设，因为IP是根据已有的应用场景信息进行设计的。

4.5.4.10 工作成果对 IP 类别的适用性

在4.5.4.1至4.5.4.9中描述的工作成果的适用性取决于4.5.2中描述的IP的分类。对于没有集成安全机制的知识产权：

——安全分析报告仅限于 IP 的失效模式分布。 由于没有集成的安全机制，因此没有针对硬件度量的估计。IP 集成商需要失效模式分布，以便能够在集成层面执行安全分析；

——集成文档集（非特定的工作成果，而是如 4.5.4.9 中所述的信息集合）受限于对 IP 集成环境假设的描述，包括接口；

——它通常不包括相关失效的分析。

4.5.5 黑盒 IP 的集成

在一些开发中，IP集成商可能遇到需要集成未完全公开内容的IP的情况。从IP集成商的角度来看，将要集成的IP是一个“黑盒”。

示例1：IP 集成商的客户需要使用其专利逻辑，例如特定通信接口，定时器外设或类似逻辑。

示例2：IP 集成商被要求集成竞争对手逻辑，以利于执行多源供应协议。

黑盒IP集成有多种方式，包括但不局限于：

——预硬化，或以门级层面布局或晶体管层级移交；

——以加密的网表形式，只有受信任的工具才能对其进行有效的解析；及

——以打乱的 RTL 源码形式，（其中有意义的变量名称被替换为随机字符串，并且任何解释性说明被移除）。

注1：黑盒集成方法也适用于从IP供应商无法获得可用信息的情况。

当黑盒IP被集成时，IP供应商、IP集成商和IP集成商客户之间的责任分工可以通过如GB/T 34590.8-XXXX，第5章中描述的开发接口协议来定义。

示例3：如果 IP 集成商被要求使用黑盒 IP，例如由于客户的要求，DIA 可以定义由客户负责评估并接受在安全相关场景中使用黑盒 IP 的适用性。

开发接口协议还可包括在GB/T 34590.2-XXXX,6.4.5.7中所述的裁剪安全活动的细节以及在整个供应链中的文档交换。

示例4：开发接口协议可以指定集成详细信息由 IP 供应商以集成指南的形式提供，该集成指南还包含一组确认测试。这些测试可用于确认合适的集成。

除非IP专门针对汽车市场开发，否则特定证据有可能是不存在的。 在这种情况下， 接受可用证据的责任可以在开发接口协议中定义。

示例5：按照其他功能安全标准如 IEC 61508：2010 [14]等开发的 IP。

注2：在这种情况下，有关开发生命周期和用于开发IP的相关过程的信息可用于执行差距分析，以评估在GB/T 34590中使用的IP的适用性。

IP集成商并不总是有足够的数据来评估黑盒IP的基础失效率。 由于这会影响定量分析的结果，因此开发接口协议可以明确IP供应商、IP集成商和IP集成商的客户之间的责任，以估算基础失效率。 黑盒IP安全分析的责任可以用类似的方式定义。

注3：黑盒IP在硬件开发中的集成与在软件开发中相似，例如开发人员将来自第三方供应商的单元软件作为编译过的目标代码进行集成。 因此，黑盒IP在硬件开发中的集成商可以在5.1.9.1中找到方法和技术，包括与GB/T 34590.6-XXXX的适用表的关联。

在黑盒IP需要安全机制的情况下，IP集成商无法获得足够的信息来实现IP之外的安全机制。 开发接口协议规定了在这些情况下对此类安全机制的要求。

4.6 半导体的基础失效率

4.6.1 基础失效率评估的总则说明

4.6.1.1 简介

本条提供有关如何计算和使用基础（或原始）失效率的说明、指南和示例。 按照GB/T 34590.5-XXXX，基础失效率是计算定量安全分析和度量的主要输入。

注：GB/T 34590.5-XXXX中的定量安全分析侧重于随机硬件失效并排除系统性失效。 因此，在GB/T 34590中使用的基础失效率仅针对随机硬件失效。也可参见4.6.1.3。

可用于基础失效率评估的每种技术都需要考虑失效机理的假设。由不同的基础失效率评估技术造成的结果差异通常是由于缺乏对同一组失效机理的考量。如果不对一组共同的失效机理进行协调，那么将不同技术应用于同一组件的结果就不具有可比性。

示例1：例如，可通过考虑相同的失效机理和相同的应力源来进行协调。

半导体的失效机理取决于电路类型、实现工艺和环境因素。 随着半导体技术的快速发展，公认的失效率行业来源难以跟上现有技术的发展水平，尤其是深度亚微米工艺技术。 因此，考虑JEDEC（联合电子设备工程委员会）、国际设备和系统路线图（IRDS）以及SEMATECH / ISMI可靠性委员会等行业组织的出版物有助于全面了解半导体技术的现状。

示例2：JEDEC 发布了数份文档，这些文档可为理解特定的失效机理并估算失效率提供参考帮助：

——参考文献[16] 总结了许多不同的、已被广泛理解和被业界接受的硅和封装失效机理；它还可以用于提供失效模式的物理学模型，用于对已识别的失效机理的失效率进行评估；

——参考文献 [53] 提供了关于开发可靠性评估方法的指南，该指南基于特定应用使用模型（任务剖面）；及

——参考文献 [17] 总结了许多与暴露在自然发生的辐射源有关的瞬态故障机理，并提供了如何通过实验获得易受软错误影响的失效率的指南。

4.6.1.2 定量目标值和可靠性预测

由于随机硬件失效导致的在相关项层面违反每个安全目标的最大概率的定量目标值（PMHF）有时会被误认为是可靠性预测的输入。 如GB/T 34590.5-XXXX,9.4.2.2，注1中所述，这些定量目标值不具有绝对意义，但在比较现有设计和新设计时非常有用。 它们旨在提供可用的设计指南并提供证明设计符合安全目标的可用证据。 因此，这些值不能够在可靠性预测中“按原样”使用。

4.6.1.3 系统性失效和随机失效之间的差异

GB/T 34590区分了系统性失效和随机失效。 绝大多数用于基础失效率评估的可用技术旨在提供可靠性评估但不做出这种区分。由于包含估算系统性失效的因子，这些技术的结果可能过于保守。 例如，基于现场失效观测的评估技术通常不具有适当的样本大小或观测质量以区分系统性失效和随机失效。

类似地，在GB/T 34590的背景下（例如，在参考文献[9]中定义的πpm和π过程因子）使用将系统性功能作为基础失效率计算的一部分的模型可能具有挑战性。

4.6.1.4 失效恢复机制的影响

一个值得关注的点是对用于增强可用性的诊断的处理。这可能导致基础失效率与诊断的混合，而GB/T 34590.5-XXXX要求将它们分开以进行度量计算。

示例：考虑在许多最先进的汽车功能安全电子中使用的常见 SEC-DED（单错误纠正-双错误探测）ECC。所报告的具有 SEC DED ECC 的 SRAM 的 MTTF（平均失效间隔）不能考虑导致可纠正错误的故障-从而综合了基础失效率和诊断的影响，而计算 GB/T 34590.5-XXXX 度量时二者是分开的。

4.6.1.5 非恒定失效率的考量

许多标准化模型都利用“浴盆曲线”简化模型，它假设供应商已经有效地筛选了“早期寿命”（早期夭折）缺陷，并假设“磨损”（寿命终止）失效机理，例如在有效的任务寿命期间，电迁移、时间相关的介质击穿，热载流子、或负偏置温度的不稳定性将以可忽略的速率有效地发生。

在某些情况下，可靠性模型的失效率分布不符合“浴盆曲线”简化模型的恒定失效率。 非恒定失效率的使用与GB/T 34590.5-XXXX中所述的硬件架构度量的计算不兼容。

一种可能性是通过使用恒定失效率的近似值来简化非恒定失效率的分布。

示例1：恒定失效率是在可靠性模型失效率分布的最大失效率下保守地假设而来。

示例2：根据分布，可以限制产品的运行寿命，从而得到更加合适的恒定失效率的近似值。 这种情况通常适用于寿命终止机理在整体失效率分布中占主导地位时。

注1：如果使用指数模型，当失效率目标被超过时，在产品寿命内到达浴盆曲线末端就是一个系统性的问题了。在GB/T 34590.5-XXXX第8章和第9章的硬件度量内不对其是否可接受进行评估。它是单独被评估，例如按照AEC-Q100的集成电路鉴定结果[62]。

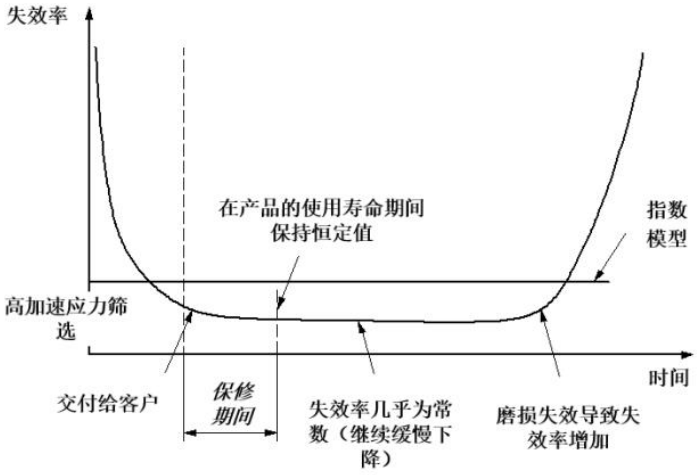


图8 浴盆曲线—失效率随时间的演化

注2：在图8中，真实的浴盆曲线可以通过“在产品的使用寿命期间保持恒定值”来近似，或者通过置信水平为70%的指数模型计算。

如果整体失效率分布是多个故障模型的集成结果，如 5.1.7.2所荐考虑瞬态故障，将失效模式的分离，通过使用不同的（但恒定的）失效率近似，来分别评估各个失效模式的影响，从而简化安全分析。

4.6.1.6 用于基础失效率评估的技术和来源

有许多不同的技术可用于评估基础失效率。这些技术通常归纳如下：

——从实验测试中得出的失效率，例如：

- 温度、偏置和运行寿命测试（TBOL），也称为高温工作寿命（HTOL）测试或延长寿命测试（ELT），用于内在产品运行可靠性，
- 可靠性测试芯片 和/或 片上测试结构，用于评估硅技术的内在可靠性，
- 基于暴露于辐射源的软错误测试，或

注1：JEDEC标准，例如JESD89 [17]，为软错误测试提供了指导。

- 用于筛选的加速测试的收敛特性。

——由现场事故观测得出的失效率，例如，对于作为现场失效返回的材料分析；

注2：对于永久性故障：半导体行业提供的数据可以基于（随机）失效的数量除以等效设备小时数。这些数据是从现场数据或加速寿命测试（如定义于JEDEC和AEC等标准中）中获得，在假定是恒定失效率（随机失效，指数分布）情况下，该数据按照比例缩放到任务剖面（例如温度，开/关周期）。 这些数字可以用作估算失效率的输入，可供作为基于采样统计置信水平的最大失效率使用。

——通过应用行业可靠性数据手册估算出的或从其中推导并结合专家判断得出的失效率；

示例1：IEC 61709[15]，SN 29500[38]或FIDES指南[9]。

示例2：如4.6..2.1.1中描述的电子组件的可靠性预测模型（原IEC TR 62380）。

注3：实际得出的失效率预计会低于由这些方法推导出的失效率。

示例3：如GB/T 34590.5：XXXX,8.4.3，注6和7所述，通过失效物理学的方法进行可靠性评估。

——由国际设备和系统路线图（IRDS）维护的文件，例如国际半导体技术路线图（ITRS [41]），提供了针对每一代软错误率的预测值，从而使得当技术数据可用时该信息可用于第一次评估并且可被精确化。

4.6.1.7 关于基础失效率计算假设的文件

当计算基础失效率时，供应商提供描述所做假设和支持依据的文件。

示例：假设可以是：

——已选择的方法来计算失效率（例如，工业来源或现场数据），

——假定的任务剖面，

——使用的失效率数据的置信水平（例如，当使用现场数据或基于测试的数据时），

——应用于失效率数据的任何缩放或修正，

——如何将非运行时间和焊点考虑进去，或

——从现场数据（Weibull 模型或指数模型）导出的用于失效率的模型。

集成商可以在要素层面或相关项层面使用此信息来评估、理解、判断、比较和可能地协调来自不同供应商和组件的失效率。

4.6.1.8 瞬态故障量化

如5.1.2中所述，软错误是瞬态故障的典型示例。

由内部或外部α、 β、中子或 γ 辐射源引发的软错误引起的瞬态故障是随机硬件失效，可以使用测量数据支持的概率方法进行量化。

由EMI或串扰引起的瞬态故障未被量化。 即使它们可以产生与其他瞬态故障相同的效果，它们也主要与系统原因有关。在设计阶段，通过适当的技术和方法可以避免这些问题（例如，在组件开发后端进行串扰分析）。

GB/T 34590.5-XXXX,8.4.7，注2指出，当由于相关原因，比如，所使用的工艺时，应该考虑瞬态故障。因此根据故障的影响并当其可适用时，可以在安全分析中考虑它们。 瞬态故障和永久性故障的分析是分开进行的。 这适用于定性或定量分析。

如果它易受软错误影响，特别是对于直接或诱导的阿尔法粒子和中子，则研究每个基础子元器件类型（例如触发器、锁存器、存储器要素，模拟器件）。 对这些现象的敏感性取决于半导体前端技术和裸片上表面的材料，包括封装，例如，模具合成物和焊料（覆晶）会影响软错误率。

示例1：阿尔法粒子的基础失效率可受封装类型的影响，例如，低阿尔法（LA）或超低阿尔法（ULA）发射半导体装配材料。

在文献[2]和[22]中，根据技术和运行频率等因素，考虑了单粒子翻转（SEU）、多比特翻转（MBU）和单粒子瞬态（SET）等瞬态故障模型。

注1：单粒子闩锁（SEL），单粒子烧毁（SEB）和单粒子栅穿（SEGR）等破坏性单粒子效应不被视为瞬态故障，因为这些故障会造成永久性影响。

注2：更多有关数字故障模型的详细信息，见5.1.2。

JESD89[17]被认为是与半导体中阿尔法粒子和地球宇宙射线引起的软错误的测量和报告有关的主要参考文献。 在这种情况下，软错误的基础失效率会与计算或测量的条件一起提供。

注3：中子粒子通量、海拔高度、温度和电源电压等条件与软错误的瞬态失效率估计有关。 JESD89[17]用于理解这些条件。

GB/T 34590.5-XXXX,8.4.3，注2指出，在应用选定的工业来源时，为避免人为减少计算出的基础失效率，以下的考虑是必要的：任务剖面、考虑运行条件时失效模式的适用性、或失效率的单位（每个运行小时或每个日历小时）。

示例2：在软错误的情况下，仅考虑车辆的运行时间来降低基础失效率，会导致每小时平均概率人为过度的降低。

注4：如果半导体供应商提供修正软错误率，有关修正因子的信息则可用于比如安全手册中，如5.1.11（对于数字组件），5.2.6（对于模拟或混合信号组件），5.3.6（对于PLD）以及5.5.6（对于传感器和换能器）中定义的。

此外，软错误的基础失效率是在不考虑“架构脆弱性因子”或ECC等安全机制影响对其修正的情况下提供的。

注5：架构脆弱性因子（AVF）是设计结构中的故障将导致功能最终输出中出现可见错误的概率，例如在参考文献[25]中对处理器设计的描述。

注6：在考虑安全故障的数量时，会考虑脆弱性因子，如5.1.7.2所述。

4.6.1.9 组件封装失效率的注意事项

在评估硬件组件失效率时，半导体提供商考虑与硅裸片，壳/封装（例如外壳）和连接点（例如引脚）有关的失效。连接到电路板的连接点（例如焊点）之间的连接被认为是电路板失效，并且通常在系统层面或要素层面的安全分析期间由系统集成商考虑。

注1：按照参考文献 [59]，在如图9中描述的模型中计算出的封装失效率 $\lambda_{封装}$ 对应于封装本身内部的故障模型（包括例如裸片和引线框架之间的连接），但它还包括与封装连接点和电路板（焊点）之间的连接有关的失效率。

注2：在SN 29500-2中计算出的硬件组件的失效率包括与裸片和封装相关的故障模型，但不同于在4.6.2.1.1中描述的模型，它不包括封装连接点和电路板之间的连接失效率，它在SN 29500-5中被单独处理。

注3：FIDES指南提供由热循环引起的封装（外壳）和焊点单独的失效率。

注4：实际上，封装连接点和电路板之间连接的失效率取决于许多因素，这些因素涉及电路板的特定设计以及如何将电路板封装在保护外壳内。 随着电子组件和电路板材料技术共同的迅速发展，这些因素也在不断变化。

4.6.1.10 考虑上电时间和下电时间

按照GB/T 34590.5-XXXX,8.4.3，注2，在应用选定的工业来源时，应适当考虑以下因素以避免人为减少计算出的基础失效率：

——任务剖面；

——失效模式对于运行条件的适用性；及

——失效率单位（每运行小时或每日历小时）。

该基础失效率与任务剖面一起被提供。 如果在任务剖面中定义了上电和下电时间，则可考虑将它们用于计算应力因子，如4.6.2.1.1（τ_{on} 和 τ_{off}）和SN 29500（π_w）中描述的方法所述。

4.6.2 永久性基础失效率的计算方法

4.6.2.1 使用或基于行业来源的永久性基础失效率的计算

4.6.2.1.1 电子组件的可靠性预测模型（前 IEC TR 62380）

本部分采用前IEC/TR 62380[40]作为电子组件可靠性预测模型的基础。

在本条中使用的数学模型，如图9所示。

$$\lambda = \left(\underbrace{\left[\lambda_1 \times N \times e^{-0.35\alpha} + \lambda_2\right] \times \left[\frac{\sum_{i=1}^{x}(\pi_t)_i \times \tau_i}{\tau_{on} + \tau_{off}}\right]}_{\lambda_{die}} + \underbrace{\left[2.75\times10^{-3} \times \pi_\alpha \times \left(\sum_{i=1}^{z}(\pi_n)_i \times (\Delta T_i)^{0.68}\right) \times \lambda_3\right]}_{\lambda_{package}} + \left[\frac{\pi_1 \times \lambda_{EOS}}{\lambda_{overstress}}\right] \right) \times 10^{-9}/h$$

图9 可靠性预测的数学模型

在图9描述的模型中，几个参数用于确定失效率：

——每种技术使用的每个晶体管的单个参数（λ_1）。λ_1 的值供给不同类型的集成电路系列，如图 10 所示；

——与所掌握的工艺有关的参数（λ_2,），且无论集成要素的数量如何，对整个组件都有效，如图 10 所示；

——与硬件组件的晶体管数量有关的参数（N）；

——与制造年份或工艺发布/更新年份与参考年份（1998 年）之间的差异相关的参数（α）；

——与硬件组件的运行和非运行阶段相关的参数（τ_i、τ_{on}和τ_{off}）；

——与温度应力因子[$(\pi_t)_i$]相关的参数，适用于组件的裸片部分；

——与集成电路可能暴露于电气过应力有关的参数（π_1 和 λ_{EOS}）如图 11 所示；

——与硬件组件的温度循环的次数和幅度相关的参数（n_i 和 ΔT_i），如图 11 所示；

——与电路板的热系数和封装材料的热系数之间不匹配有关的参数（α_S 和 α_C），如图 11 和图 12 所示；及

——与封装相关的参数（λ_3），或者作为封装类型及其引脚编号 S 的函数（如图 14 所示），或者作为用于表面装配的集成电路的封装对角线 D 的函数（如图 15 所示）。

可以基于工艺技术和设计所使用的电路类型来完成参数的选择。

注1：图10中，“实际数量”对应于晶体管的实际数量，不考虑这些晶体管的尺寸大小。

注2：为了计算整个器件的数字组件裸片失效率，应使用等效门数。 通过将等效门数乘以每个门代表的晶体管数来计算有效等效晶体管的数量。当计算由CMOS数字逻辑引起的微控制器裸片失效率时，模块的每个数字逻辑的贡献（例如CPU，CAN，定时器，FlexRay，SPI）都包含在N中。

注3：考虑到摩尔定律以及器件失效率几乎稳定的事实，引进了工艺成熟度修正因子。 如果每个晶体管的失效率保持不变，那么按照摩尔定律，失效率会增加。这一点没有被观察到。 因此，当改变工艺节点时，晶体管失效不能保持恒定。可用的一种选择是使用生产日期，为了显示工艺技术发生改变，另一种选择可以使用将这一特定技术节点首次引入的年份而非其制造年份。 为了表达相对于芯片供应商的独立性，可以使用来自ITRS的年份[41]。

注4：除非半导体供应商提供更精确的数据，对于模拟元器件或主要基于模拟工艺技术构建的数字组件，可以使用图10的“线性电路”条目。

注5：若有足够的理由支持，可以使用与所考虑的工艺相关的数据来代替上述参数，以便更加准确地评估基础失效率。

缩写	类型	N代表晶体管数	λ_1(FIT)	λ_2(FIT)
硅:MOS:标准电路（3）				
	数字电路，微核，DSP	4个/门	3.4×10^{-4}	1.7
	线性电路	实际数量	1.0×10^{-2}	4.2
	数字/线性电路(Telekom, CAN, CNA, RAMDAC等)	实际数量	2.7×10^{-4}	20
	存储器：			
ROM	只读存储器	1个/比特	1.7×10^{-7}	8.8
DRAM/VideoRAM/AudioRAM	动态随机存储器	1个/比特	1.0×10^{-7}	5.6
高速SRAM, FIFO	静态随机存储器- 先进先出寄存器;(“混合”MOS)	4个/比特	1.7×10^{-7}	8.8
低功耗SRAM	静态随机存储器-低功耗；(CMOS)	6个/比特	1.7×10^{-7}	8.8
Double access SRAM	双访问静态RAM	8个/比特	1.7×10^{-7}	8.8
EPROM,UVPROM,REPROM	电可编程，UV可清除—只读存储器			
OTP	可编程一次EPROM	1个/可编程点	2.6×10^{-7}	34
FLASH	电可编程可清除 (块)(1)			
EEPROM,flash EEPROM	电可编程可清除 (字)(2)	2个/可编程点	6.5×10^{-7}	16
（1）全存储器阵列或块的字节可清除 （2）字节块可清除 （3）MOS包括CMOS, HCMOS, NMOS等技术				
硅：MOS：专用集成电路（ASIC）				
	标准单元，完全自定义	4个/门	1.2×10^{-5}	10
	门阵列	4个/门	2.0×10^{-5}	10

	用户可编程逻辑器件：			
LCA(基于RAM)	由外部存储器电配置的逻辑单元阵列	40个/门	4.0×10^{-5}	8.8
PLD (GAL, PAL)(2)	电可编程可清除（与/或 阵列）	3对/网节点	1.2×10^{-5}	16
CPLD (EPLD,MAX,FLEX,FPGA等)	电可编程（互联宏单元阵列）	100个/宏单元	2.0×10^{-5}	34
（1）或4000每宏单元；（2）EEPROM, EPROM或抗熔技术				
硅：双极电路（1）				
	数字电路	3个/门	6.0×10^{-4}	1.7
	线性电路（FET和其他）	实际数量	2.2×10^{-2}	3.3
	MMIC	实际数量	1.0	3.3
	线性/数字电路，低电压（< 30伏特）	实际数量	2.7×10^{-3}	20
	线性/数字电路，高电压（= 30伏特）	实际数量	2.7×10^{-2}	20
	存储器-可编程阵列-门阵列：			
SRAM	静态随机存储器	2.5个/比特	3.0×10^{-4}	1.7
PROM,	可编程只读存储器	1.2个/可编程点	1.5×10^{-4}	32
PLD(PAL)	单次电可编程逻辑阵列（与/或 阵列）	1.6个/网格点	1.5×10^{-4}	32
	门阵列	3个/门	1.0×10^{-3}	10
双极包括：TTL, MTTL, LSTTL, FET, JFET, BCL, 等技术				
硅：双极MOS电路（BICMOS）				
	数字电路	4个/门	1.0×10^{-6}	1.7
	线性/数字电路低电压(< 6伏特)	实际数量	2.7×10^{-4}	20

	线性/数字电路，高电压(= 6伏特）和智能电源	实际数量	2.7×10^{-3}	20
SRAM	静态随机存储器	4个/比特	6.8×10^{-7}	8.8
	门阵列	4个/门	6.4×10^{-5}	10
砷化镓				
数字	与常导通晶体管	5个/门	2.5	25
数字	与常关闭和常导通晶体管	3个/门	4.5×10^{-4}	16
MMIC	低噪声或低能（< 100毫瓦）微波电路	实际数量	2.0	20
MMIC	能量（>100毫瓦）微波电路	实际数量	4.0	40

图10 λ_1和λ_2在集成电路中的值

工艺结构	温度因子π_t
MOS BIMOS（低电压）	$e\left[A\left(\frac{1}{328}-\frac{1}{273+t_j}\right)\right]$ A=3 480；(E_a=0.3 eV)
双极 BICMOS（高电压）	$e\left[A\left(\frac{1}{328}-\frac{1}{273+t_j}\right)\right]$ A=4 640；(E_a=0.4 eV)
砷化镓 数值化	$e\left[A\left(\frac{1}{373}-\frac{1}{273+t_j}\right)\right]$ A=3 480；(E_a=0.3 eV)
砷化镓 MMIC	$e\left[A\left(\frac{1}{373}-\frac{1}{273+t_j}\right)\right]$ A=4 640；(E_a=0.4 eV)
t_j = 结温 （℃）	

接口电路典型的计算值			λ_{EOS} FIT	π_I
功能	电气环境			
接口	电脑		10	1
	电信	转换	15	1
		传输，接入，用户卡	40	1
		用户设备	70	1
	轨道，付费电话		100	1
	民用航空（机载计算器）		20	1
	电源供给，整流器		40	1
无接口	所有电气环境		-	0

影响因子π_α的数学表达	$\pi_\alpha=0.06\times(\lvert\alpha_S-\alpha_C\rvert)^{1.68}$
电路板与封装间热膨胀系数的不匹配	$\lvert\alpha_S-\alpha_C\rvert$
α_S	见 图12
α_C	

影响因子$(\pi_n)_i$的数学表达	$n_i\le8\,760$ 循环次数/年	$(\pi_n)_i=n_i^{0.76}$
	$n_i>8\,760$ 循环次数/年	$(\pi_n)_i=1.7\times n_i^{0.76}$
n_i：每年幅度为ΔT_i 循环数		
针对开/关阶段	$\Delta T_i=\left[\frac{\Delta T_i}{3}+(t_{ac})_i\right]-(t_{ac})_i$	
针对永久工作阶段，存储或者休眠状态	在任务剖面的第 i 阶段时，ΔT_i=每个t_{ac}变化周期的平均值	

图11 温度和过压因子

线性热膨胀系数	材料类型	值（ppm/℃）
α_S（电路板）	环氧玻璃（FR4, G-10）	16
	PTFE玻璃（聚四氟乙烯）	20
	弹性电路板（聚酰亚胺芳纶）	6.5
	铜/殷钢/铜（20/60/20）	5.4
α_C（组件）	环氧树脂（塑料封装）	21.5
	氧化铝（陶瓷封装）	6.5
	合金（金属封装）	5

图12　α_s和α_c的热膨胀系数

气候类型	t_{ae}夜间	t_{ae}日天	t_{ae}日天/夜间均值	ΔT_i白天/夜间
全世界	5　℃	15　℃	14　℃	10　℃
法国	6　℃	14　℃	11　℃	8　℃

图13　气候

缩写	材料类型	描述		引脚号：S	λ_3（FIT）
SO,SOP:1.27mm 间距	环氧树脂	塑料翼型（L形）小外形，宽度：3.8-7.5 mm		4 到 40	$=0.012\times S^{1.65}$
功率 SO	环氧树脂	同小外形带热下沉			同SO
SOJ:1.27mm 间距	环氧树脂	塑料J形小外型，宽度：10.16 mm		28 到 44	$=0.023\times S^{1.5}$
VSOP:0.76 mm 间距	环氧树脂	L形极小外形，宽度：10.16 mm		40 到 56	$=0.011\times S^{1.47}$
SSOP: 0.65 mm 间距	环氧树脂	缩L形小外形，宽度：10.16 mm		8 到 56	$=0.013\times S^{1.35}$
TSSOP: 0.65 mm 间距	环氧树脂	薄L形小外形，宽度：4.1-6.1 mm		8 到 38	$=0.011\times S^{1.4}$
TSOP Ⅰ: 0.55 mm 间距	环氧树脂	薄L形小外形，宽度：11.8 mm		18 到 32	$=0.54\times S^{0.4}$
TSOP Ⅰ: 0.5 mm 间距	环氧树脂	薄L形小外形，宽度：18.4 mm		18 到 32	$=1.0\times S^{0.36}$
TSOP Ⅱ: 0.8 mm 间距	环氧树脂	薄L形小外形，宽度：10.16 mm		28 到 54	$=0.04\times S^{1.2}$
TSOP Ⅱ: 0.65 mm 间距	环氧树脂	薄L形小外形，宽度：10.16 mm		34 到 60	$=0.042\times S^{1.1}$
TSOP Ⅱ: 0.5 mm 间距	环氧树脂	薄L形小外形，宽度：10.16 mm		34 到 60	$=0.075\times S^{0.9}$
TSOP Ⅱ: 0.4 mm 间距	环氧树脂	薄L形小外形，宽度：10.16 mm		34 到 60	$=0.13\times S^{0.7}$
PLCC: 1.27 mm 间距	环氧树脂	J形塑封芯片载体，所有		20 到 84	$=0.021\times S^{1.57}$
CLCC: 1.27 mm 间距	氧化铝	陶瓷引线（无引线）芯片载体，所有			同PLCC
MQUAD: 1.27 mm 间距	合金	金属扁平封装（PLCC 引脚设计）；所有			同PLCC
PQFP, TQFP	环氧树脂	塑封（薄）扁平封装，L形，大小被定义在右边	5 ×5 mm^2	32 到 40	1.3
			10 ×10 mm^2	40 到 60	4.1
			14 ×14 mm^2	60 到 68	7.2

			14 ×20 mm²	68 到 110	10.2
			28 ×28 mm²	110 到 225	23
			32 ×32 mm²	225 到 280	29
			40 ×40 mm²	280 到304	42
ED QUAD, 功率QUAD	环氧树脂	同 PQFP 带 热 下 沉（裸露金属块）			同PQFP
CQFP, CERQUAD	氧化铝	陶瓷扁平封装			同PQFP
MQFP, MQUAD	合金	金属扁平封装			同PQFP
PBGA	环氧树脂	塑料球栅阵列封装 - pas > 1mm, 大小定义在右边	13.5 ×15 mm²	64 到 80	11.4
			17.4 ×19 mm²	80 到 160	16.6
			23 ×23 mm²	160 到 280	26.6
			35 ×35 mm²	280 到 400	51.3
SBGA	环氧树脂	缩BGA-pas 1mm-corps 42.5 ×42.5 mm²		580	71
SBGA	环氧树脂	缩BGA-pas 1mm-corps 27 ×27 mm²		672	33
CBGA	氧化铝	陶瓷			同PBGA
PDIL	环氧树脂	塑封双列直插		8 到 64	$=9+0.09\times S$
CDIL, CERDIP	氧化铝	陶瓷双列直插		8 到 64	$=9+0.09\times S$
PPGA	环氧树脂	塑封陈列引脚		40 到 160	$=9+0.09\times S$
CPGA	氧化铝	陶瓷陈列引脚		40 到 160	$=9+0.09\times S$

图14　针对集成电路λ_3的值作为 S 的函数

封装类型	示例	λ_3（单位：FIT）
两行连接封装	SO、SOP、SOJ、VSOP、SSOP、TSSOP、TSOP I、TSOP II等	$=0.024\times D^{1.68}$（1）
外围连接封装	PLCC、CLCC、MQUAD、PQFP、CQFP、MQFP等	$=0.048\times D^{1.68}$（2）
矩阵连接封装	PBGA、CBGA、SBGA;、 μBGA、CSP等	$=0.073\times D^{1.68}$（3）
有环氧树脂滴的裸片	COB（片上芯片）	$=0.048\times D^{1.68}$（4）

注（1）：$D=\left[\left(\left(\frac{S}{2}-1\right)\times(\text{间距})\right)^2+(\text{宽度})^2\right]^{\frac{1}{2}}$

注（2）：$D=\left[\left(\left(\frac{S}{4}-1\right)\times(\text{间距})\right)^2+(\text{宽度})^2\right]^{\frac{1}{2}}$

注（3）：$D=\left[(\text{长度})^2+(\text{宽度})^2\right]^{\frac{1}{2}}$

注（4）：D 指区域对角线

图15　表面安装的集成电路封装的λ_3的值

一旦生成了组件裸片的基础失效率（FIT），就会根据热效应和运行时间来明确修正因子。 修正因子基于以下因素确定：

——组件裸片的结温的计算基于：

- 组件裸片的功率消耗；及
- 封装热阻，基于封装类型、封装的引脚数和气流；

——定义从 1 到 Y 个使用阶段的应用剖面，每个阶段由应用“运行时间”组成，其作为总器件寿命的百分比，和环境温度；及

示例：于可能的汽车使用工况的两个示例：“电机控制”和“乘客仓”，如图 16 所示。

任务剖面阶段	温度. 1		温度. 2		温度. 3		比率开/关		2夜间开始		4白天开始		不使用汽车	
应用类型	$(t_{ac})_1$ ℃	τ_1	$(t_{ac})_2$ ℃	τ_2	$(t_{ac})_3$ ℃	τ_3	$\tau_{开}$	$\tau_{关}$	n_1 周期/年	ΔT_1 ℃/周期	n_2 周期/年	ΔT_2 ℃/周期	n_3 周期/年	ΔT_3 ℃/周期
电机控制	32	0.020	60	0.015	80	0.023	0.058	0.942	670	$\frac{\Delta T_j}{3}+55$	1340	$\frac{\Delta T_j}{3}+45$	30	10
乘客舱	27	0.006	30	0.046	85	0.006	0.058	0.942	670	$\frac{\Delta T_j}{3}+30$	1340	$\frac{\Delta T_j}{3}+20$	30	10

图16　汽车任务剖面的示例

——活化能和每种技术类型的频率以完成阿伦尼乌斯方程。

注6：与所考虑的产品相关的数据，如封装热特性，制造工艺，阿伦尼乌斯方程等，可用于代替上述一般因素以更准确地估算基础失效率。

4.6.2.1.1.1　如何结合 λ1 和 λ2

关于图9中所描述的方法，存在多个选项，关于在将不同技术的电路要素（CPU，存储器等）实施在同一器件上的情况下如何结合 λ_1 和 λ_2。

在其中一个选项中，每个电路要素继承各自技术的 λ_1 和 λ_2，因此基本上将 λ_1 和 λ_2 相加，如表2所示。

注：λ_2 的值是加权的，例如各个电路要素的晶体管数量如公式（1）所示。

$$\lambda_{裸片}=\sum_{要素}\left(\lambda_{1,要素1}\times N_{要素}\times e^{-0.35\times\alpha}+\frac{N_{要素}}{N_{全部}}\times\lambda_{2,要素}\right)\times\frac{\sum_{i=1}^{y}\left(\pi_{t,要素}\right)_i}{\tau_{开}+\tau_{关}} \quad \text{(1)}$$

式中：

在此示例中，假设基于CMOS技术的MCU消耗0.5W功率。数字组件裸片采用144引脚方型扁平式封装，并通过自然对流进行冷却。MCU暴露在“电机控制”温度剖面下。由此引起的结温的增加 ΔTj为26,27 ℃。对于阿伦尼乌斯方程，假设活化能为 0.3eV。使用图9中的模型，这导致修正因子（即 $\lambda_{裸片}$ 的第二因子）为 0.17。

表2　总和 λ2 为数字组件示例

电路要素	λ_1(FIT)	N(晶体管)	α	λ_2(FIT)	基本失效率	温度修正因子	有效失效率
50 k 门 CPU	3.4×10^{-6}	200000 (4 晶体管/门)	10	1.7	1.73	0.17	0.06
16 kB SRAM	1.7×10^{-7}	786 432(6 晶体管/bit 用于低功耗SRAM)	10	8.8	8.80	0.17	1.18
						裸片失效率 (FIT)	1.25

作为替代方法，可以使用单个（保守）λ_2 的最大值的等式（2）作为代表值（见表3）：

$$\lambda_{芯片}=\sum_{要素}\left(\lambda_{1,要素}\times N_{要素}\times e^{-0.35\times\alpha}\times\frac{\sum_{i=1}^{y}\left(\pi_{t,要素}\right)_i\times\tau_i}{\tau_{开}+\tau_{关}}\right)+max\left(\lambda_{2,要素}\right)\times\frac{\sum_{i=1}^{y}Max\left(\pi_{t,要素}\right)_i\times\tau_i}{\tau_{开}+\tau_{关}}\cdots\cdots(2)$$

表3　最大值为λ₂的混合信号示例

电路要素	λ1(FIT)	N(晶体管)	α	基础失效率不包括λ2(FIT)	λ2(FIT)	温度修正因子	有效失效率(FIT)
数字电路	1.0×10^{-6}	28000	10	8.5×10^{-4}	1.7		
线性/低压数字电路(<6 V)	2.7×10^{-4}	30000	10	0.25	20		
裸片失效率 (FIT)				0.25	Max(20、1.7) = 20	0.17	3.44

在以下示例中，集成电路由三个要素组成，其组成与图10中相应的 λ1 和 λ2 值如表4所示。

表4　IC 的组成

要素1	数字电路	λ_1 [FIT]	1.00×10^{-6}	N	100000
		λ_2 [FIT]	1.70		
要素2	线性电路低电压 LV	λ_1 [FIT]	2.70×10^{-4}	N	5000
		λ_2 [FIT]	20		
要素3	线性电路高电压 HV	λ_1 [FIT]	2.70×10^{-3}	N	2000
		λ_2 [FIT]	20		

使用电机控制剖面（图16）作为任务剖面，制造年份为2018年，与 λ_1 的相关的裸片失效率项可使用公式（3）到（8）来计算。

$$\lambda_{1,要素1}\times N_{要素1}\times e^{-0.35\times\alpha}=(1.0\times10^{-6}\times100000)\times e^{-0.35\times(2018-1998)}=9.12\times10^{-5}FIT\cdots(3)$$

$$\left(\pi_{t,要素1}\right)_1\times\tau_1=e^{\left[3480\left(\frac{1}{328}-\frac{1}{273+32}\right)\right]}\times0.020=8.99\times10^{-3}\cdots\cdots(4)$$

$$\left(\pi_{t,要素1}\right)_2\times\tau_2=e^{\left[3480\left(\frac{1}{328}-\frac{1}{273+60}\right)\right]}\times0.015=1.76\times10^{-2}\cdots\cdots(5)$$

$$\left(\pi_{t,要素1}\right)_2\times\tau_2=e^{\left[3480\left(\frac{1}{328}-\frac{1}{273+85}\right)\right]}\times0.023=5.60\times10^{-2}\cdots\cdots(6)$$

$$\sum_{i=1}^{3}\left(\pi_{t,要素1}\right)_i\times\tau_i=8.25\times10^{-2}\cdots\cdots(7)$$

$$\lambda_{1,要素1}\times N_{要素1}\times e^{-0.35\times\alpha}\times\sum_{i=1}^{3}\left(\pi_{t,要素1}\right)_i\times\tau_i=7.53\times10^{-6}FIT\cdots\cdots(8)$$

模拟计算为其他要素提供了如下结果：

$$\sum_{i=1}^{3}\left(\pi_{t,要素2}\right)_i\times\tau_i=8.25\times10^{-2}\cdots\cdots(9)$$

$$\lambda_{1,要素2}\times N_{要素2}\times e^{-0.35\times\alpha}\times\sum_{i=1}^{3}\left(\pi_{t,要素2}\right)_i\times\tau_i=1.02\times10^{-4}FIT\cdots\cdots(10)$$

$$\sum_{i=1}^{3}\left(\pi_{t,要素3}\right)_i\times\tau_i=1.01\times10^{-1}\cdots\cdots(11)$$

$$\lambda_{1,要素3} \times N_{要素3} \times e^{-0.35\times\alpha} \times \sum_{i=1}^{3}\left(\pi_{t,要素3}\right)_i \times \tau_i = 4.96 \times 10^{-4} FIT \quad \cdots\cdots (12)$$

$$\sum_{要素=1}^{3}\left[\lambda_{1,要素} \times N_{要素} \times e^{-0.35\times\alpha} \times \sum_{i=1}^{3}\left(\pi_{t,要素}\right)_i \times \tau_i\right] = (7.53 \times 10^{-6} + 1.02 \times 10^{-4} + 4.96 \times 10^{-4}) FIT = 6.05 \times 10^{-4} FIT \quad \cdots\cdots (13)$$

对于 λ_2相关的裸片失效率项，得到：

$$Max\left(\lambda_{2,要素}\right) = \left(\lambda_{2,要素2}\right) = \left(\lambda_{2,要素3}\right) = 20FIT \quad \cdots\cdots (14)$$

$$Max\left[\sum_{i=1}^{y}\left(\pi_{t,要素}\right)_i \times \tau_i\right]_{要素} = \sum_{i=1}^{3}\left(\pi_{t,要素3}\right)_i \times \tau_i = 1.01 \times 10^{-1} \quad \cdots\cdots (15)$$

$$Max\left(\lambda_{2,要素}\right) \times Max\left[\sum_{i=1}^{y}\left(\pi_{t,要素}\right)_i \times \tau_i\right]_{要素} = 20 \times 1.01 \times 10^{-1} FIT = 2.01FIT \quad \cdots\cdots (16)$$

这导致整体裸片失效率：

$$\lambda_{裸片} = 6.05 \times 10^{-4} FIT + 2.01FIT = 2.01FIT \quad \cdots\cdots (17)$$

为了简化计算，如果用户可以确定其产品与如图10中所列出的集成电路系列类型之一之间相匹配，那么如下表5所示，用户可以直接应用如图9所示的失效率计算方法。

表5　具有匹配设备类型的数字组件示例

电路要素	λ1(FIT)	N(晶体管)	α	λ_2(FIT)	基本失效率(FIT)	温度修正因子	有效失效率(FIT)
50 k 门 CPU	3.4×10^{-6}	200000 (4 晶体管/门)	10	1.7	1.80	0.17	0.31
16 kB SRAM		786432 (6 晶体管/bit 用于低功耗SRAM)					
裸片失效率 (FIT)							0.31

4.6.2.1.1.2　温度修正

图9中的模型使用以下的参数来计算温度修正因子ΔT ：

——$(\pi_t)_i$：i^{th} ：与集成电路任务剖面的第 i 个结面温度相关的温度因子；

——τ_i：i^{th} ：集成电路任务剖面的第 i 个结面温度的工作时间比率；

——$\tau_{开}$：集成电路的总工作时间比率，$\tau_{开} = \sum_{i=1}^{y}\tau_i$ ；

——$\tau_{关}$：集成电路存储（或休眠）的时间比率；

——$\tau_{开} + \tau_{关} = 1$.

为了计算保守温度修正因子，非工作状态时间比率 $\tau_{关}$可以设置为0，由此而产生对应于保守温度修正因子$\delta_{T,保守}$的δ_T的微修订版本：

$$\delta_{T,保守}=\frac{\sum_{i=1}^{y}(\pi_t)_i}{\tau_{开}} \quad (18)$$

表2，表3和表4中，在考虑时间比率τ开和τ关之后再计算修正因子。 在上述数字组件示例（表5）中，将τ关设置为零从而给出的修正因子为2.91，对此有效失效率的值从0.31变为5.24 FIT。

4.6.2.1.1.3 封装基础失效率计算

图9中所计算的封装失效率λ封装对应于封装内部的失效模式（其包括例如裸片和引线框架之间的连接），但它也包含了与封装连接点和电路板（焊点）之间连接相关的失效率，如参考文献[54]所示，其大约占整个λ封装失效率（FIT）的20％。半导体供应商可以使用80％的λ封装值来分配硬件组件的封装失效率（FIT）。

如4.6.2.1.1中所述，封装失效率的计算参考了以下参数：

—— π_α：与安装基板和封装材料之间的热膨胀系数差异相关的影响因子；

—— $(\pi_n)_i$:与封装处年度内温度变化循环次数相关的第 i 个影响因子，幅值为ΔTi；

—— ΔT_j：任务剖面中第 i 个温度幅值变化范围；及

—— λ_3：集成电路封装的基础失效率。

表6 基本失效率计算示例

封装类型	ΔT_j (° C)	S (引脚的数量)	D (mm)	π_α	λ_3 (FIT)	温度循环修正	有效失效率 (FIT)
PQFP 144	26.27	144	26.58	1.05	11.87	6009	206
封装失效率包括封装与电路板之间的焊点(FIT)							206
总封装失效率不包括封装与电路板之间的焊点(FIT)							166

影响因子 π_α可用图11所示的公式进行计算，其中，αs，αc分别是基板和组件的线性热膨胀系数。在此示例中，假设FR4作为安装基板和塑料封装，由图12查得αs= 16 和 αc = 21.5。

对于温度变化循环次数/年≤8760的汽车使用工况，参数$(\pi_n)_i$可用图11中所提供的公式进行计算，其中，n_i ：幅值为 ΔT_i.的温度循环次数。

为了计算失效率（FIT）中的λ3，可使用如图15所示的外围连接封装的计算公式，其中宽度值为20 mm，管脚中心距为0.5 mm。使用图16中所示的“电机控制”温度剖面，这会导致没有焊点的封装总失效率为：

λ封装=166 FIT。

假设封装失效率在引脚间均匀分布，可得一个引脚的失效率为：

λ引脚=1.15 FIT

注1：示例中的封装是144引脚方型扁平式封装（QFP），并通过自然对流冷却。 其功耗为0.5 W，从而导致结温的增加 ΔT_j 是 26,27 ° C。D和λ3的值可通过使用图15 基于以下参数来计算：管脚中心距= 0.5mm 和宽度= 20mm。

注2：并非所有封装都包含在图14或图15所示的表中。在这种情况下，专家的判断可用于评估封装对总体失效率产生的影响。

示例1：封装失效率的评估是基于器件封装和系统的印刷电路板的结构和热特性的知识。

注3：所有引脚采用相等概率可在此示例中使用，但不能用于所有情况。

示例2：在球形焊点阵列封装（BGA）中，某些位置可以具有比其他位置更高概率的分布。

4.6.2.1.1.4 由电气过应力导致的失效率示例

由于电气过应力导致的整个器件的失效率可以用图9所示的公式计算。如果器件直接连接到外部环境，即器件是个接口，π_I 等于1； 如果器件不是接口，即它并没有直接连接到外部环境，π_I 等于零。

图11提供了在各种各样的电气环境下不同的λ_{EOS}，然而，汽车的电气环境并没有被给出。取而代之的是，可以选择“民用航空电子设备（机载计算器）”：

λ_{EOS}=20 FIT。

这会导致由整个器件的电气过应力引起的失效率是：

若器件与外部环境直接连接， $\lambda_{过应力}$= 20 FIT；或

在其他任何情况下， $\lambda_{过应力}$ = 0 FIT。

预测电气过应力对器件的影响是意义重大的。 若无特殊影响论证，那$\lambda_{过应力}$可被加到$\lambda_{裸片}$中以增加整个器件的总的裸片失效率。

注：电气过应力可被视为系统性失效模式，并且在计算硬件随机失效度量时，电气过应力可降至0 FIT。

4.6.2.1.2 SN 29500

SN 29500采用了表查找方法，并给出了在特定参考条件下的失效率的预期值。通过使用产品类型，技术和晶体管数量视为输入，即可在表格中查到对应值。 如果集成电路是在不同于参考条件的情况下进行操作的，则应参考其运行条件来进行计算。此计算中考虑了温度、电压和漂移（对于模拟要素）的影响。对于运行条件的计算的温度部分，应使用修订的阿伦尼乌斯方程。

4.6.2.1.2.1 半导体组件的计算示例

应用SN 29500计算失效率所需的参数：

——N：等效晶体管的数量；

——λ_{ref}：硬件组件的基础失效率，基于工艺技术；

——ΔT_j：结温的增加值；及

——硬件组件的任务剖面。

注1：若在SN 29500-2：2010的失效率系列表1、2或3中未列出等效晶体管数N，用户可以使用内插法或外插法确定等效 λ_{ref} 和 $\theta_{vj,1}$（虚拟结温）的值。

示例：对于 SN 29500-2：2010 中表 2 定义的“微处理器和外设、微控制器和信号处理器”系列，可使用以下的内插法示例来确定 λ_{ref} 和 $\theta_{vj,1}$ 的值。

假设有一个具有500K门的微控制器，则， λ_{ref}的计算可通过以下步骤来完成：

——第一步：通过使用温度相关因子 π_T ，将表 2 中的 λ_{ref} 值转换为相同的虚拟参考温度 $q_{vj,1}$ ，以 90° C 为例：

$$\pi_T = \frac{A\times e^{Ea1\times z}+(1-A)\times e^{Ea2\times z}}{A\times e^{Ea1\times z_{ref}}+(1-A)\times e^{Ea2\times z_{ref}}} \quad (19)$$

SN 29500－2:2010, 表 2 CMOS							
(19)	1k	10k	100k	1M	10M	100M	$\theta_{vj,1}$
λ_{ref} (50° C)	25						50 ° C

λ_{ref} (60 ° C)		30					60 ° C
λ_{ref} (80 ° C)			50				80 ° C
λ_{ref} (90 ° C)				80	120	150	90 ° C
π_T (90 ° C)	5,18	3,47	1,53	1	1	1	—
λ_{ref} (90 ° C)	130	105	76	80	120	150	FIT

—— 第二步：在 90° C 下对 λ_{ref} 使用线性内插以获得所需的复杂度，即 500 K 晶体管：

λref (90 ° C)							
门	1k	10k	100k	1M	10M	100M	$\theta_{vj,1}$
λref (90 ° C)	130	105	76	80	120	150	FIT

$$\lambda_{\mathrm{ref}\left(500\ \mathrm{K}@90\ ^{0}\mathrm{C}\right)} = \lambda_{\mathrm{ref}\left(100\ \mathrm{K}@90\ ^{0}\mathrm{C}\right)} + \left(500\ \mathrm{K} + 100\ \mathrm{K}\right) \times \frac{\lambda_{\mathrm{ref}\left(1\ \mathrm{M}@90\ ^{0}\mathrm{C}\right)} - \lambda_{\mathrm{ref}\left(100\ \mathrm{K}@90\ ^{0}\mathrm{C}\right)}}{1\ M - 100\ K} = 76 + 400\ K \times \frac{\left(80 - 75\right)}{900\ K} = 78,2\ \mathrm{FIT} \quad (20)$$

——第 3 步：线性内插 $\theta_{vj,1}$ 以得出所需的复杂度，即 500 K 晶体管：

$$\theta_{vj,1(500\ \mathrm{K})} = \theta_{vj,1(100\ \mathrm{K})} + \left(500\ \mathrm{K} - 100\ \mathrm{K}\right) \times \frac{\theta_{vj,1(1\ \mathrm{M})} - \theta_{vj,1(100\ \mathrm{K})}}{1\ \mathrm{M} - 100\ \mathrm{K}} = 80 + 400\ \mathrm{K} \times \frac{\left(90 - 80\right)}{900\ \mathrm{K}} = 84,4\ ^{\circ}\mathrm{C} \quad (21)$$

——第 4 步和最后一步：使用温度相关因子 π_T 将 $\lambda_{ref(500K@90\ ^\circ C)}$ 转换为 $\theta_{vj,1(500K)}$：

$$\pi_{T(90\ ^\circ C \Rightarrow 84.4\ ^\circ C)} = 0.79 \quad (22)$$

$$\lambda_{ref(500K@84.4^\circ C)} = \lambda_{ref(500K@90^\circ C)} \times \pi_{T(90^\circ C \Rightarrow 84.4\ ^\circ C)} = 78.2 \times 0,79 = 62\ \mathrm{FIT} \quad (23)$$

SN 29500－2:2010, 表 2 CMOS							
门	1k	10k	100k	1M	10M	100M	$\theta_{vj,1}$
λ_{ref} (50 ° C)	25						50 ° C
λ_{ref} (60 ° C)		30					60 ° C

λ_{ref} (80 ° C)			50				80 ° C
λ_{ref} (90 ° C)				80	120	150	90 ° C

注2：关于任务剖面的值仅是示例。在电控单元（ECU）内所有半导体的要求应与各自ECU的规范一致。

4.6.2.1.2.2 无非运行阶段的半导体组件示例的失效率计算

对于在之前章节中描述的数字组件示例，在具有500k到5百万个晶体管的CMOS技术中，可在90° C参考温度条件下得到80 FIT。表7和表8中列出了以下参数：

——A：常数；

——Ea1、Ea2：恒定活化能，单位为电子伏。

表7 SN 29500 的失效率计算示例所需的参数

N(晶体管)	技术和系列	λ_{ref} (FIT)	ΔT_j(° C)	温度依变参考 (Z_{ref}) (1/eV)	A	E_{a1}(eV)	E_{a2}(eV)
986432(数字 + SRAM)	CMOS、微处理器	80	26.27	5.11	0.9	0.3	0.7

假设每年工作500小时并使用图16中定义的电机控制任务剖面，可以得到表8的结果。

表8 SN 29500 的数字组件失效率计算示例

环境温度 θ_U(° C)	工作时长(h)	结面温度 $\theta_{j,2}$(° C)	依变因子Z(1/eV)	温度依变因子 π_T(θ_u)
32	172.4	58.27	2.04	0.27
60	129.3	86.27	4.77	0.85
85	198.3	111.27	6.87	2.51
			总温度依变因子 π_T	1.31
			整硬件组件有效失效率 (FIT)	105

4.6.2.1.2.3 具有非运行阶段的半导体组件示例的失效率计算

在考虑非运行阶段时，4.6.2.1.1小节和SN 29500中描述的模型之间存在差异。在4.6.2.1.1中描述的模型中，非运行时间默认包含在产品的任务剖面当中，而对于SN 29500中的模型，其仅默认考虑运行时间。正如4.6.2.1.1.2中所述，计算失效率的另一种备选方法是将 $\tau_{关}$时间设置为零。

以相似的方式，在计算失效率时，运行和非运行阶段也可以在SN 29500中被考虑。这是通过使用在SN 29500-2：2010，4.4章节中描述的应力因子 π_{ω}来完成的。使用图16中所定义的电机控制任务剖面和10.5 ° C的平均温度，可得应力因子值应为 0.06。 将计算出的应力因子应用于数字组件示例失效率当中，可以得到表9的结果。

表9 SN 2950 有/没有非运行阶段失效率计算

N(晶体管)	技术和系列	λ_{ref}(FIT)	λ 没有非运行阶段 (FIT)	压力因子	λ 与非运行阶段 (FIT)
986432 (数字 + SRAM)	CMOS、微处理器	80	104.65	0.06	6.3

注：非运行平均温度是从在图13中所定义的全球平均夜间和白天温度（分别为5° C和15° C）获得的，此时设夜间和白天之间占比为50%。

4.6.2.1.2.4 将 SN 29500 整体失效率分为裸片和封装失效率的方法

如SN 29500的维护人员所述，使用SN 29500计算出的基础失效率仅对整个硬件组件有效，并且不提供划分出封装失效率和裸片失效率的方法。因此，如果需要，可以根据专家意见来估计裸片失效率和封装失效率的比率。

示例：作为专家判断的示例，可以使用经 4.6.2.1.1 中的方法确定的或者基于其他行业来源的相同的比率来评估 SN 29500 基础失效率中的封装和裸片失效率的分配。其中，这些工业来源提供此类数据或可用的现场数据统计。

4.6.2.1.3 FIDES 指南

以下是使用在FIDES指南[9]中详述的方法所支持的定量分析所需的硬件失效率的评估示例。根据FIDES指南，半导体的失效率模型将器件的失效率视为和以下相关的因子：

——物理影响（$\lambda_{物理}$）；

——工艺影响（$\pi_{工艺}$）；及

——器件制造影响（π_{PM}）。

第一个是附加的构成项，包括针对可靠性的物理及技术影响因子。第二个是乘法项，对于包括器件在内的产品的开发、制造和使用过程的质量和技术控制。第三个因子代表比如生产场地的质量和供应商的经验，若这些因子都与系统问题有关，那么将 $\pi_{工艺}$和 π_{PM} 设置为1。

物理影响包括由于使用条件引起的应力加速因子和包含器件在内的产品的应用所固有的诱导（即非预期的过应力）乘法项。然而为了简单起见，在当前的示例中该诱导乘法因子被设置为1。当实际应用它时，其值应基于位置、使用控制和对组件过应力的敏感度来确定。

在集成电路的FIDES指南中使用的模型包括以下物理应力系列：

——热；

——温度循环；

——机械；及

——湿度；

注：为了使示例简单，以下计算不包括与机械和湿度相关的失效模式。这些附加的失效模式应在实际应用中被考虑。

为了计算数字组件裸片和封装基础失效率(即在应用运行条件的修正之前),有必要考虑以下要素：

——λ_{OTH}，与器件类型和工艺技术相关的基础失效率；及

——与封装类型相关的物理应力参数 a 和 b。

这些因子通过使用FIDES而相结合。参数选择可以基于工艺技术、电路类型和设计使用的封装。与微处理器、微控制器、DSP和SRAM以及具有144个引脚的PQFP封装相关的值均可用。

下面的表10和表11展示了用于基于CMOS技术的MCU的定量示例的失效率的计算，此MCU消耗功率为0.5W。数字组件裸片采用144引脚方型扁平式封装并且使用自然对流和低导电板冷却。

表10 UTE FIDES裸片基础失效率

电路要素	$\lambda 0_{TH}$ (FIT)
50 k gate CPU	0,08
16 kB SRAM	0,06
总和	0,13

表11 UTE FIDES封装基本效率

封装	$\lambda 0_{TCy\_Case}$			$\lambda 0_{TCy\_Solderjoints}$		
	a	b	$\lambda 0_{TCy\_Case}$ (FIT)	a	b	$\lambda 0_{TCy\_Solderjoints}$ (FIT)
144 引脚 PQFP	12.41	1.46	0.01	10.80	1.46	0.03

一旦生成了数字组件裸片和封装的基础失效率，就会根据热效应和运行时间来应用修正因子。修正因子应考虑到：

——数字组件裸片的结温，它可基于以下公式进行计算：

- 数字组件裸片的功耗；及
- 封装热阻，基于封装类型、封装引脚数和气流。

——定义1到Y使用阶段的应用剖面，每个阶段由应用程序“运行时间”、“循环时间”、“循环增量温度”和“循环最高温度”以及“环境温度”组成。

注：相较参考文献[40]中的剖面，在模型中使用的剖面会提供更多或者其他的参数。

首先，如表12中所示的简化任务剖面示例被考虑。

表12 简化任务剖面示例

			热	热循环			
阶段	开/关	$T_{年阶段}$ (小时)	$T_{环境}$ (° C)	$\Delta T_{循环}$ (° C)	θ_{cy} (小时)	$N_{cy-年}$ (小时)	$T_{最大循环}$ (° C)
不运作日程	关	720	15	10	24.0	30	20

晚间运作	开	168	60	55	0,25	670	60
白天运作	开	335	60	45	0.25	1340	60
停止运作	关	7.538	15	10	22.5	30	20

具有修正因子的裸片基础失效率可如表13所示进行计算。

表13　裸片基础失效率与温度修正因子

电路要素	λ_{0TH} (FIT)	温度修正因子	有效失效率
50 k 门 CPU	0.08	5.79	0.43
16 kB SRAM	0.06	5.79	0.32
总和	0.13		0.75

为了评估这些修正因子，使用FIDES中描述的参数和公式，将结温（即由自加热引起的ΔT_j）计算为18K（见表14）。

表14　封装基础失效率和温度循环修正因子

	λ_{TCy_case}			$\lambda_{TCy\_焊点}$		
封装	$\lambda 0_{TCy\_case}$ (FIT)	循环修正	有效失效率	$\lambda 0_{TCy\_焊点}$ (FIT)	循环修正	有效失效率
144 引脚 PQFP	0.01	130	0.75	0.03	10	0.28

随后，表15中所示详细任务剖面示例被考虑，修正因子列于表16中。

表15　详细任务简介示例

			热	热循环			
阶段	开/关	$T_{年阶段}$ (小时)	$T_{环境}$ (° C)	$\Delta T_{循环}$ (° C)	θ_{cy} (小时)	$N_{cy-年}$ (小时)	$T_{最大循环}$ (° C)
不运作日程	关	720	14	10	24.0	30	19
晚间运作	开	117	32	22	0.0	670	32

白天运作	开	58	32	18	0.0	1340	32
满载运行	开	201	85	53	1.0	335	85
高速运行	开	131	60	28	4.0	30	60
停止运作日程	关	7.532	14	10	23.0	30	19

表16　有效失效率

电路要素	λ_{0TH} (FIT)	温度修正因子	有效失效率 (FIT)
50 k门 CPU	0.08	12.44	0.93
16 kB SRAM	0.06	12.44	0.68
总和 (FIT)	0.13		1.61

为了评估这些修正因子，使用FIDES中描述的参数和公式，计算由于自加热引起的结温即 ΔT_j为18K。如表17所示，组件封装失效率为0.25FIT。在表17中的焊点失效率值仅作为信息给出，不被视为封装失效率的一部分。

表17　封装和焊点失效率

	λ_{TCy_case} (FIT)			$\lambda_{TCy\_焊点}$ (FIT)		
封装	$\lambda 0_{TCy\_case}$ (FIT)	循环修正	有效失效率	$\lambda 0_{TCy\_焊点}$ (FIT)	循环修正	有效失效率
144 引脚　PQFP	0.01	42	0.25	0.03	4	0.12

4.6.2.2　使用现场数据统计计算永久性基础失效率

由于很难得到合适的评估，所以谨慎使用现场数据统计非常重要。对现场反馈进行全面彻底分析，并将分析结果用于定量评估。 特别是当评估以下主题时：

——现场反馈如何处理已知的质量问题；

——什么类型的信息可用于真实的任务剖面；及

——现场监测过程的有效性如何。

由于根据现场数据计算失效率的方法论对产生的失效率的置信水平有影响，因此半导体供应商会考虑以下几点：

——现场数据收集系统需到位，如 GB/T 34590.2-XXXX， 7.4.2.3 中的注意事项中所要求；

——该方法的目标不是尽可能接近真实的失效率，而是提供一个失效率值，它有高的置信度，其值高于实际失效率；

——系统性故障的主要来源只有在系统性故障的来源得到减轻的情况下才能从现场统计中去除；

示例1：一个关于系统性故障主要来源的示例是 EOS。

注1：关于系统性故障来源的减轻证据需文档化。

——因为半导体供应商无法注意到现场的所有失效，所以可以将修正因子（CF）应用于总返回数。 该因子取决于许多参数，例如应用和用于评估基于现场的失效率的器件数量；

注2：半导体供应商根据现场反馈评估失效率，并提供依据。

——对应于温度应力或热循环应力效应的加速因子 AF 可以被分别通过使用可用的、经验证过的热应变或脆性断裂模型来计算。

示例2：Coffin-Manson 或 Englemaier-Clech 方法。

——现场产品的总运行时间可以通过使用产品任务剖面来进行评估，若任务剖面可用。 也可通过评估现场花费的小时数量，（例如每年平均 500 小时，其中标准偏差 145 小时）来考虑驾驶员对汽车使用的可变性；及

——文档化现场数据的任务剖面，并在定量评估中适当考虑。

4.6.2.2.1 指数模型方法

指数模型通常可用于从现场反馈确定恒定失效率。 在该模型中，x 2（卡方）统计函数提供了失效率的良好评估。对于失效率， 建议使用具有至少70%置信水平的单边上区间评估的区间估计器，而不是使用点估计器。 这意味着有70%的概率，失效率的实际值低于该值。 失效率可通过以下公式计算得出：

$$\mathrm{FIT} = \frac{\chi^2_{CL;2n+2} \times 10^9}{2 \times \text{cumulative operational hours} \times \mathrm{AF}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (24)$$

其中：

n：失效次数乘以修正因子；

CL:置信水平值（通常为 70%）；

AF:加速因子。

注：加速因子用于使失效率值从一个任务剖面适应另一个任务剖面，如4.6.2.2.2所述。

4.6.2.2.2 硬件组件失效率的计算示例

在本条中，通过指数模型方法的使用给出了使用现场数据统计的裸片失效率计算的示例。 在此示例中，假设半导体供应商正在收集现场三种产品的统计数据，如下表18所示。

表18 任务剖面和等效结温 $T_{j,eq}$

T_j（° C）	芯片1阶段持续时间（小时）	T_j（° C）	芯片2阶段持续时间（小时）	T_j（° C）	芯片3阶段持续时间（小时）

−20	1000	−25	100	−20	500
10	2000	10	500	15	800
30	1500	35	10000	45	6000
45	6000	55	8000	80	4200
70	1000	90	1000	100	600
100	1300	100	200	120	300
130	200	120	200	150	100
$T_{j,eq}$ (° C)	55.1	$T_{j,eq}$	51.4	$T_{j,eq}$	67.4
总持续	13 000	总持续	20 000	总持续	12500

注1：从温度应力角度看，任务剖面等效温度$T_{j,eq}$对应于和整个任务剖面具有相同效果的温度。 $T_{j,eq}$可以通过使用阿伦尼乌斯方程进行计算。在上述例子中，假设活化能E_a为0.3 eV。

注2：如果不同器件的器件运行时间有相同的参考温度Tref，则可以将它们相加在一起。 在此示例中，T_{ref}为55℃，在T_{ref}温度下的等效器件小时数可通过使用活化能E_a为0.3eV的阿伦尼乌斯方程进行计算。

注3：如表19所示，在参考温度T_{ref}下，每平方毫米的失效率是根据失效总数和裸片面积小时总数并使用 x^2 统计函数来计算的。在此示例中，使用了70%的置信水平。

表19 计算参考温度 T_{ref} 下每 mm^2 的失效率

产品名称	裸片大小mm^2	任务剖面等效温度 $T_{j,eq}$ (℃)	设备总运行时间（百万设备小时）	阿伦尼乌斯加速因子	等效运行时间在 T_{ref} 为 55°C (百万设备小时)	等效裸片面积小时 在T_{ref}为55°C （百万mm^2小时）	保修期内的Nb失效	CF = 5的Nb失效
芯片1	30	55.1	7000	1.00	7 022.67	210680	1	5
芯片2	25	51.4	10200	0.89	9 066.96	226674	1	5
芯片3	50	67.4	5000	1.47	7 359.25	367963	2	10
总裸片面积小时数						805317	总Nb失效	20
FIT/mm^2 在T_{ref} 为55 ℃						0.029		

如下图17所示，根据现场数据统计得出的在T_{ref}温度下的每平方毫米的失效率可用于计算在设计下的目标产品的失效率（见表20）。

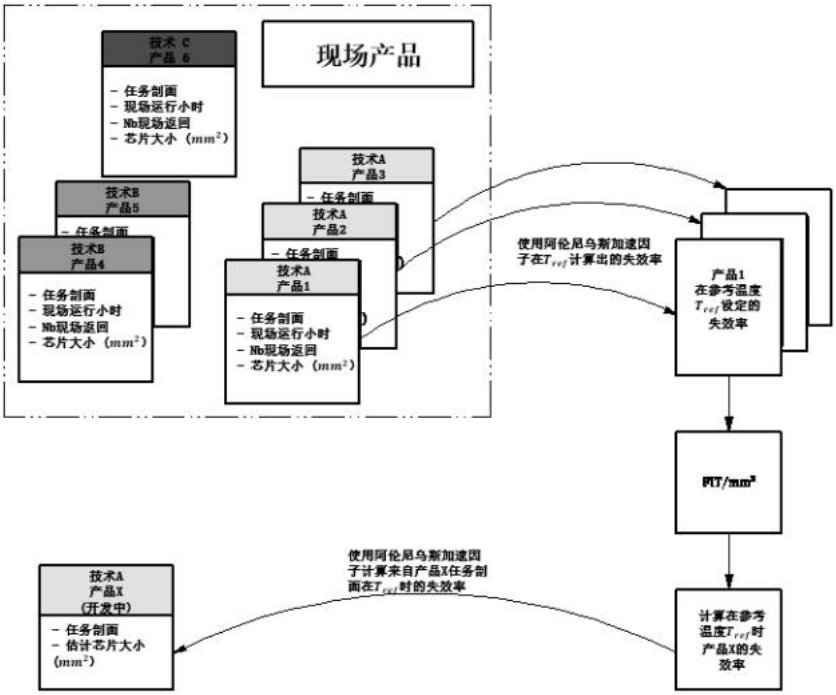


图17　采用现场数据统计的裸片失效率计算方法

表20　最终芯片失效率计算

	任务剖面等效温度 $T_{j,eq}$ (°C)	裸片大小 (mm²)	FIT/mm² at T_{ref}	阿伦尼乌斯加速因子	FIT/mm² 在等效温度 $T_{j,eq}$	裸片基础失效率
设计下的目标芯片	75	23	0.03	1.84	0.05	1.22

注4：计算封装失效率时采用相同的方法，但计算加速因子时，采用Coffin-Manson或Norris-Landzberg模型（如参考文献[15] 5.2.7.10条“失效模式”，参考文献[16] 5.14条和参考文献[9] 2.5.1“失效物理学和模型”小节）来代替阿伦尼斯模型。 图18概述了通过现场数据统计计算封装失效率的方法。

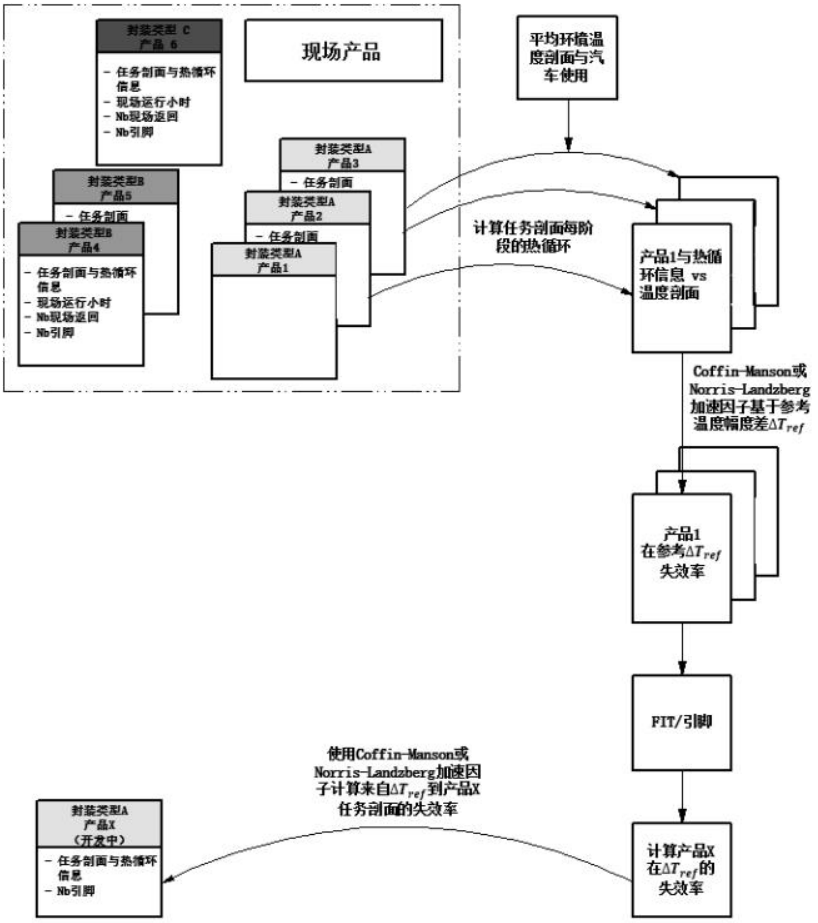


图18　采用现场数据统计的封装失效率计算方法

注5：如果现场数据分析不能区分裸片和封装（例如在SN 29500 [38]中的示例），那么阿伦尼乌斯定律可通过使用如图18所示的任务剖面温度和参考温度T_{ref} 来计算硬件组件（裸片和封装）的失效率。

4.6.2.3　使用加速寿命试验计算基础失效率

启用加速因子，使寿命测试中的温度修正到最大运行温度。该计算使用了阿伦尼乌斯方程，其活化能为0.7eV。 建议评估和验证与所需失效机理相关的活化能。

从样本中得到的故障数量在具有一定置信水平的χ2分布函数中使用，以获得在整个测试群体中可能发生的总故障数量。

在确定器件寿命时还需要考虑电压的加速度。CMOS的计算是通过考虑栅极氧化层厚度，并将应力测试电压修正至寿命运行电压来计算的。

$$AF_v = \exp(\beta) \times [V_t - V_o] \quad \cdots\cdots(25)$$

其中,

AF_v:电压加速度因子；

V_o:典型运行条件下的栅极电压（单位为伏特）；

V_t:加速测试条件下的栅极电压（单位为伏特）；

β:电压加速系数（单位为 1 / 伏特）。

4.6.2.4 失效率分配方法

前面的条目详细阐述了几种确定半导体组件的基础失效率的方法。根据所述方法，整体半导体组件失效率可由单个值或封装失效率和裸片失效率的组合值而得到。在安全分析期间，半导体组件失效率被分配给组成半导体组件要素的失效模式。

可以使用不同的分配方法：

——组件的裸片元器件的组成部分（即数字模块、模拟模块和存储器）的失效率分布。 可以考虑两种方法来提取或获取分布：

- 第一种方法是使用裸片失效率或整个硬件组件失效率（如果没有分成封装和裸片的失效率）除以硬件组件裸片面积来获得每平方毫米的失效率。通过将与分析中失效模式相关的元器件或子元器件面积乘以在每平方毫米的失效率来完成失效率分布；及
- 第二种方法是基于基础失效率和基础子元器件所建立的。这是通过评估在分析中与失效模式相关的每个元件、子元件或者基础子元器件的等效门数（或晶体管数量）来完成的。

——封装的失效率分布。 只有当封装的失效率可用时才可以得到。在这种情况下，对于与安全相关的引脚，可以使用每个引脚的失效率来完成失效率的分配，该失效率是通过将封装失效率分配给封装的总引脚数所得到的（安全相关与否）。

注：所使用的方法的选择可以基于所分析的电路的布局（或计划的布局），或者基于硬件要素之间失效模式的共享方式。

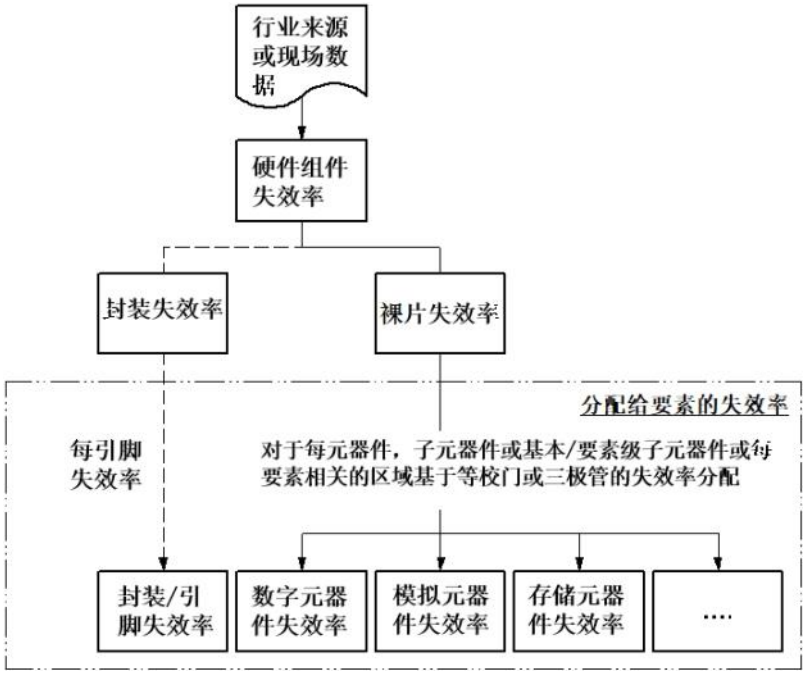


图19 失效率分配

4.6.2.5 MCM 的基础失效率

多芯片模块（MCM）的基础失效率需要经过仔细评估。 如果使用行业来源（或使用4.6.2.1.1中描述的模型）来评估失效率，则应提供论据以证明该行业来源的适用性或定制化。

4.7 半导体相关失效分析

4.7.1 相关失效分析介绍

本条的目的是为以下方面提供指导：识别和分析给定要素之间可能的共因失效和级联失效，评估其违反安全目标（或得出的安全要求）的风险，并在必要时定义安全措施以减轻此类风险。这样做是为了评估潜在的安全概念缺陷，并提供证据证明由ASIL等级分解产生的独立性的要求（参见GB/T 34590.9-XXXX第5章）或在共存分析期间识别的免于干扰的要求（参见GB/T 34590.9-XXXX第6章）得到了满足。

本条适用的范围是在一个硅裸片内硬件要素之间以及硬件与软件要素之间的相关失效分析。考虑的要素通常是硬件要素及其安全机制（在GB/T 34590.5中定义）。

分析范围，分析方法和必要的安全措施可取决于给定要素的性质（例如，仅软件要素，仅硬件要素或硬件和软件相结合的要素）以及所涉及的安全要求的性质（例如， 失效安全）。

如GB/T 34590.1-XXXX中3.30所定义的，相关失效引发源（DFI）是导致多个要素通过耦合因子失效的单一根本原因。考虑到不同的系统性、环境性和随机硬件问题，本条提供了相关失效引发源列表作为相关失效分析的起点（表21至表26）。一些随机硬件相关失效引发源，例如共享资源或被考虑要素的干扰要素，一旦确定了相关性，就可以在标准安全分析中被考虑，并将其归类为残余故障、单点故障或多点故障（GB/T 34590.5-XXXX的9.4 .2.4，注1）。相关失效分析以定性的方式处理那些在标准安全分析中无法处理的相关失效引发源。

示例：由于随机硬件故障或系统性故障，干扰要素有能力破坏其它硬件要素的资源：例如 DMA（直接存储器访问外设）向错误的地址写入，并静默地破坏与安全相关的数据。

相关失效引发源列表还包含一些用于解决这些问题的典型安全措施。可以根据安全要求的性质来决定需要的安全措施。要求可以是尽量减少现场相关失效的发生，也可以是确保相关失效不违反安全目标。

4.7.2 相关失效分析与安全分析之间的关系

根据GB/T 34590.5-XXXX, 7.4.3所做的安全分析，可以识别出相关失效分析相关的要素。

这些可以是：

——双点失效情况，例如：

- 功能及其安全机制（包括故障响应路径 - 实施故障响应所需的要素和/或任务链）； 及
- 功能冗余（例如，两个电流驱动器或两个模数转换器）。

——属于半导体基础构造的共享要素的单点（残余）失效情况，如：

- 时钟生成；
- 嵌入式稳压器；及
- 上述要素使用的任何共享硬件资源。

安全分析主要侧重于识别单点故障和双/多点故障，以评估GB/T 34590.5度量目标，并在需要时定义安全机制改进度量。相关失效分析通过确保安全机制的有效性不受相关失效引发源的影响来补充安全分析。如GB/T 34590.5-XXXX中7.4.3所述，首先安全分析可用于支持硬件设计的定义，其次可用于硬件设计的验证。 相关失效分析可以在硬件设计的定义中应用(例如，为已识别的共享要素指定安全机制)，也可以验证定义中所做的假设已实现并达到预期的有效性。

4.7.3 相关失效场景

在图20中，要素A和要素B是由于某个外部根本原因而导致潜在失效的要素。这个根本原因可能与随机硬件故障或系统性故障有关。

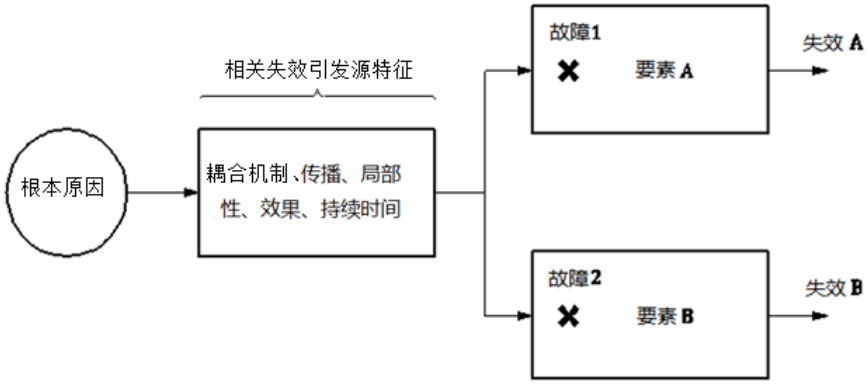


图20　相关失效及其相关失效引发源的原理示意图

与随机硬件故障相关的典型情况可能包括共享资源失效或单一物理性根本原因。 针对这些情况，按照GB/T 34590.5-XXXX第8和第9章，失效率可以被量化并且可以在安全分析中被考虑。

注1：在这种情况下，相关失效引发源产生的风险在定量安全分析中进行评估。 因此，不需要再单独论证。

与系统性故障相关的典型情况可能包括环境故障、开发故障等。针对这些情况，通常不可能进行定量分析。另外，根本原因可能位于所考虑的半导体要素的内部或外部，例如通过信号或电源接口传播到半导体要素内。

图20提到一个耦合机制，用来描述由给定的根本原因造成干扰的一些典型特性。这些特性可以帮助定义减轻措施，也可以定义用于验证减轻措施有效性的恰当模型（参见4.7.5.2）。 现介绍如下：

a)　耦合机制：它描述了根本原因引起干扰的方式。已知的耦合机制有：

传导性耦合(电耦合或热耦合)，发生在当发射源与接收器之间的耦合路径是通过与导体(例

——如传输线、导线、电缆、印刷电路板（PCB）走线或金属外壳)直接接触而形成的情况下；及

——近场耦合，当发射源和接收器间隔距离较短（通常小于一个波长）时发生的耦合。严格地说，“近场耦合”可以分为电感应耦合和磁感应耦合两种。通常将电感应耦合称为容性耦合，将磁感应耦合称为感性耦合：

- 容性耦合，当两个相邻导体（通常相距小于一个波长）之间存在一个变化的电场时发生的耦合，并引起两个导体之间电压的变化；及
- 感性耦合或磁耦合，当两个相邻导体（通常相距小于一个波长）之间存在变化的磁场时发生的耦合，并引起接收导体上的电压变化。

——机械耦合，当机械力或应力通过物理介质从发射源传递到接收器时发生的耦合。

示例：在 MEMS 中，具有特定共振和波形的冲击可能导致加速度计中的梳状结构粘住（也称为粘滞）。有关详细信息，见 5.5.3.2。

——辐射耦合或电磁耦合， 当发射源和接收器相隔很远（通常超过一个波长）时发生的耦合。发射源和接收器就像无线电天线：发射源发射或辐射电磁波，电磁波在两者之间的空间里传播，然后被接收器接收。

——传播介质：描述的是干扰通过半导体要素传输的耦合路径。通常可以是：

- 信号线；
- 时钟网络；

- 供电网络；
- 衬底材料；
- 封装；及
- 空气。

b) 局部性：描述了干扰影响多个要素或仅限于单个要素的可能性。 在后一种情况下，假定受影响的要素产生错误的输出，该输出传播到与其连接的多个要素（级联影响）；

c) 时序：描述有关传播延迟的干扰特性（例如，用于温度梯度的传播）、或其时间表现，如周期性（例如，在电源上存在纹波噪声的情况下）等。

为了说明上述特性，在图21和图22中给出了两个示例。

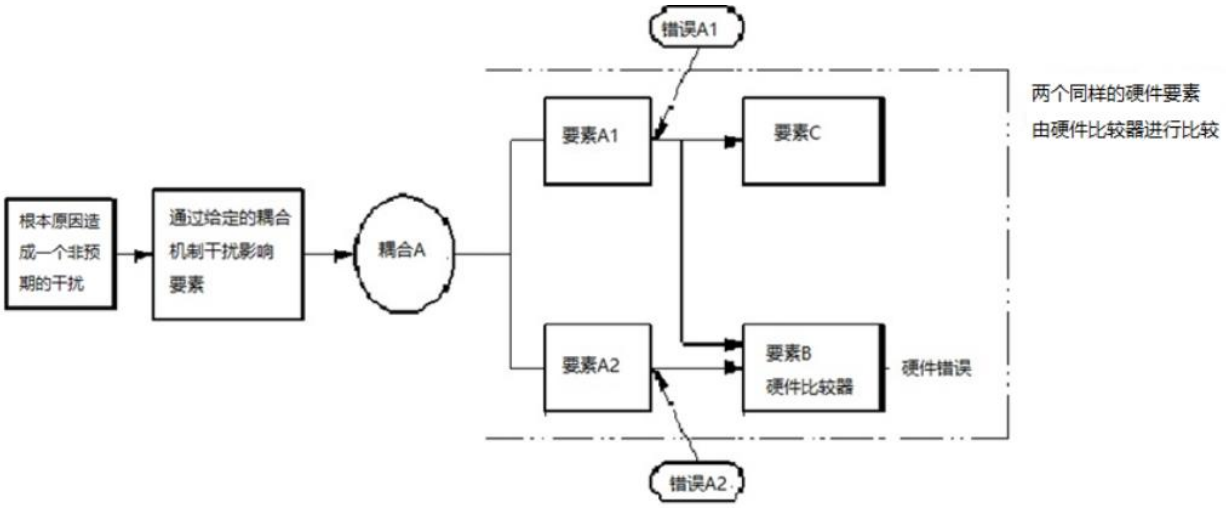


图21　物理耦合引起的相关失效

在图21中，要素A1的结果被要素C用来实现安全功能。要素A1和要素A2作为冗余要素，由要素B硬件比较器进行比较，当不匹配时(失效A1或失效A2)，“硬件错误”信号被激活。在此示例中，如果要素A1和要素A2都受到同一根本原因导致的故障的影响，则要素A1和要素A2可以产生相同的错误输出(错误A1和错误A2)。在比较要素A1和要素A2时，要素B无法区分这种可能发生的相关失效。

注2：为简化起见，假设要素B本身不受干扰影响。 假设要素B处在正常运行状态，进一步假设只要错误A1和错误A2在时间或空间上存在一定的差异，就可以控制相关失效情况。这种差异可能是干扰传播到两个要素的方式不同造成的（例如，信号毛刺的不同传播延迟，该信号毛刺采用不同的物理路径到达要素A1和要素A2的边界），或可能是干扰影响的结果不同造成的（例如，如果影响是信号时序的违反，它会对要素A1和要素A2的相应逻辑产生不同的影响）。

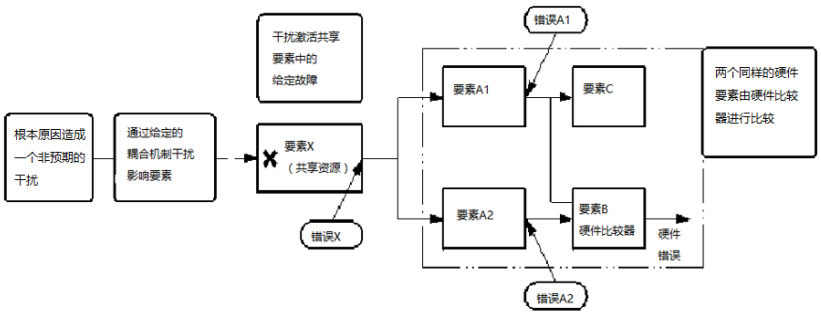


图22　资源共享导致的相关失效

图22扩展了图21，其中受到自身外部根本原因导致的故障的影响，造成了要素X的错误输出，该错误输出同时传播到要素A1和要素A2，从而进一步导致了要素A1和要素A2产生错误输出。要素x是“共享资源”作为相关失效引发源的代表。

4.7.4 级联失效与共因失效之间的区别

相关失效分析针对的是共因失效和级联失效。 虽然在某些情况下这种区分是必要的（例如GB/T 34590.9-XXXX，第7章），但在其他情况下（例如半导体器件），在给定的失效场景中，级联失效和共因失效之间的确切区分不一定可行或有用。 在这种情况下，两种失效场景不作进一步区分。

如果相关失效分析的重点是按照GB/T 34590.9-XXXX第7章的要求，提供两个给定要素(例如要素A和要素B)之间免于干扰(共存)的证据，可以采用以下方法:

——识别可能对要素 B 产生影响的要素 A 的失效模式；

——识别这些失效模式是否会由于要素 B 失效而导致有可能违反安全目标；

——如有必要，定义适当的安全措施以降低风险（例如，为 DMA 指定安全机制，以监控 DMA 生成的地址）；及

——如有必要，切换要素的角色重复此分析。

4.7.5 相关失效引发源和减轻措施

4.7.5.1 相关失效引发源和相关减轻措施清单

可以使用以下相关失效引发源分类：

——共享资源的失效；

——单个物理性根本原因；

——环境类故障；

——开发类故障；

——制造类故障；

——安装类故障；及

——服务类故障。

注1：相关失效引发源的其他分类是有可能的。

对于每类相关失效，都提供了可能的措施。

注2：列出的措施是可能的解决方案示例，但并非详尽列表。它们的有效性取决于几个因素，包括电路类型和工艺，这意味着它们对可能的相关失效引发源的有效性会有所不同。 因此，建议提供证据证明所声称的有效性。 一些措施本身不足以适当地降低风险。 在这种情况下，可以选择不同措施的适当组合。

措施分为：

——防止运行期间相关失效发生的措施；及

——无法防止相关失效发生，但可防止其违反安全目标的措施。

注3：由软件引起的相关失效引发源不在此相关失效引发源列表中。正确的软件开发由GB/T 34590.6-XXXX规定。相关失效分析的结果会影响软件要素的ASIL等级分配。

注4：汽车行业维修服务通常通过更换整个ECU或传感器模块来实现。半导体元件通常不提供维修服务。因此，半导体元器件的维修服务故障通常不是相关失效引发源。

表21 由于共享资源的随机硬件故障导致的相关失效引发源

相关失效引发源示例	共用时钟要素的失效（包括 PLL，时钟树，时钟使能信号等）；

	共用测试逻辑和共用调试逻辑的失效，其中共用测试逻辑包括 DFT（面向测试的设计）信号和扫描链等，共用调试逻辑包括调试路由网络（该网络提供对模拟或数字信号的访问或使能数字寄存器的读取）和跟踪信号（同步跟踪一个或多个信号的机制，例如由触发器或跟踪时钟控制，然后读取结果）； 电源要素的失效，包括电源分配网络，共用电压调节器，共用参考源（例如带隙，偏置发生器和相关网络）； 非同步电源开启，可能会导致闩锁或高冲击电流等影响； 共用复位逻辑的失效，包括复位信号； 共享模块中的失效（例如:RAM、闪存、ADC、定时器、DMA、中断控制器、总线等）。
防止相关失效违反安全目标的措施	对共享资源的专用独立监控（例如，时钟监控，电压监控，存储器 ECC，配置寄存器内容上的 CRC，测试或调试模式的信号）； 针对软错误或选定冗余功能的选择性加固； 对共享资源在启动时或后运行或运行期间进行自检； 影响的多样化（例如，主核和检测核之间的时钟延迟，多样化的主核和检测核，不同的关键路径）； 间接探测共享资源的失效（例如，在共享资源发生失效的情况下会失效的功能循环自检）； 使用特殊传感器进行间接监控（例如用作共因失效传感器的延迟线）。
防止在运行期间发生相关失效的措施	故障避免措施（例如保守的规范），共享资源内的功能冗余（例如多个过孔/触点）； 故障诊断（例如识别和隔离或重新配置/替换失效的共享资源的能力，相应的设计规则）； 专用生产测试（例如，能够发现复杂故障的 SRAM 下线测试）； 用来减少共享资源数量或范围的独享资源； 降低敏感性的自适应措施（例如电压/工作频率降低）。

表22　由于随机物理性根本原因导致的相关失效引发源

相关失效引发源示例	短路（例如：局部缺陷、电迁移、过孔迁移、接触迁移、氧化物分解）； 闩锁 (Latch up)； 串扰（衬底电流，容性耦合）； 局部过热，例如由于电压调节器或输出驱动器缺陷导致；
防止相关失效违反安全目标的措施	影响的多样化（例如，主核和检测核之间的时钟延迟，多样化的主核和检测核，不同的关键路径）；

	间接探测（例如，对可能由于物理性根本原因导致失效的功能进行循环自检）或使用特殊传感器进行间接监控（例如用作共因失效传感器的延迟线）。
防止在运行期间发生相关失效的措施	专门的生产测试； 故障避免措施（例如物理分离/隔离，相应的设计规则）； 单芯片上的物理分离。

表23　由于环境条件导致的系统性相关失效引发源

相关失效引发源示例	温度 振动 压力 湿度/凝露 腐蚀 电磁干扰 外部施加的过压 机械应力 磨损 老化 水和其他液体侵入
防止相关失效违反安全目标的措施	影响的多样化（例如：主核和检测核之间的时钟延迟、多样化的主核和检测核、不同的关键路径） 直接监控环境条件（例如：温度传感器）或间接监控环境条件（例如：用作相关失效传感器的延迟线）
防止在运行期间发生相关失效的措施	故障避免措施（例如：保守的规格/鲁棒性设计） 物理分离（例如：裸片与裸片外部的局部热源的距离） 降低敏感性的自适应措施（例如：电压/工作频率降低） 限制访问频率或限制子元器件允许的运行周期（例如：指定 EEPROM 的写周期数） 半导体封装的鲁棒性设计

表24　由于开发故障导致的系统性相关失效引发源

相关失效引发源示例	需求错误 规格错误 实施错误，即功能的不正确实施

	避免串扰的设计措施缺乏或不足 闩锁预防措施缺乏或不足 配置错误 布局错误，例如错误的走线，例如区块的过度冗余，绝缘不足，分离或隔离不足，EMI 屏蔽不足 裸片中功耗元器件发热引起的温度 温度梯度造成敏感测量电路的不匹配
防止相关失效违反安全目标的措施	监控器（例如：协议检查器）
防止在运行期间发生相关失效的措施	设计流程符合 GB/T 34590 多样性（根据相关失效引发源，多样性可以是实施/功能/架构的多样性或开发多样性）

表25　由制造故障引起的系统性相关失效引发源

相关失效引发源示例	与流程、程序和培训相关 控制计划和特殊特性监控的故障 与软件刷写和下线编程有关（例如错误的版本或错误的编程条件，协议或时序） 掩模错位 下线修整或熔断不正确（例如，标定系数或自定义设置的激光修整，OTP 或 EEPROM 编程）
防止相关失效违反安全目标的措施	无
防止在运行期间发生相关失效的措施	专门的生产测试 符合 GB/T 34590（见 4.9） 多样性（根据相关失效引发源，多样性可以是实施/功能/架构的多样性或开发多样性）

表26　由于安装故障导致的系统性相关失效引发源

相关失效引发源示例	与线束走线有关 与元器件的互换性有关 相邻相关项或元器件或要素的失效。 （例如，将数据传送到输入连接接口的错误配置，或驱动输出上的错误负载） PCB 连接错误

	配置错误（例如空闲的内存使用）
防止相关失效违反安全目标的措施	无
防止在运行期间发生相关失效的措施	专用安装测试 符合 GB/T 34590（见 4.9） 多样性（根据相关失效引发源，多样性可以是实施/功能/架构的多样性或开发多样性）

4.7.5.2 减轻措施验证

本条介绍了对控制或避免相关失效的有效性进行评估的示例性方法。这些方法可基于：

——使用已知原理的分析方法；

示例1：参考文献[4]和类似提供的分析方法可用作评估所提供的解决相关失效的安全措施有效性的基础。

——使用文档化的测试协议进行流片前仿真，以提供针对已识别的相关失效引发源的鲁棒性证据；

示例2：测试协议允许时钟或电源干扰的仿真，EMI 仿真等。仿真可以基于不同的抽象级别（基于目标故障模型）并使用适当的故障注入技术来产生预期的干扰。

——流片后鲁棒性测试（例如 EMI 测试，老化测试，加速老化测试，电气压力测试）；及

——有书面依据的专家判断。

可以使用组合的措施，例如，参考文献[24]，[21]和类似文献提供一种基于分析、故障注入及专家判断的综合方法，其可用作评估针对相关失效所提供的安全措施有效性的基础。

注1：结果和论据被文档化及被证明。

注2：按照GB/T 34590.9-XXXX, 7.4.2，注释中的说明，在IEC 61508-2：2010 [14]中使用beta因子以量化耦合效应通常不被认为是足够可靠的方法。

评估的详细程度与相关失效引发源的类型、声明的安全措施及应用相称。

如GB/T 34590.9-XXXX, 7.4.7中的示例所述，多样性是可用于预防、减少或探测共因失效的措施。如果多样性被作为控制或避免相关失效的方法，则需提供依据来证明所实施的多样性的层级与目标相关失效引发源相称。

示例3：可通过分析方法和故障注入的组合来提供理由（例如参考文献[24]中所述）。有关故障注入的详细信息，请参见 4.8。若使用隔离或分离作为控制或避免相关失效的方法，需提供理由来证明已实施的隔离或分离程度与目标相关失效引发源相称。

示例4：仿真可用于提供证据证明两个分离块之间的距离足以避免目标相关失效引发源。

4.7.6 相关失效分析工作流程

相关失效分析工作流程的目的是确定必要的主要活动，以理解所实施的确保安全要求达成的安全措施的运行，并验证它们符合独立性或免于干扰的要求。

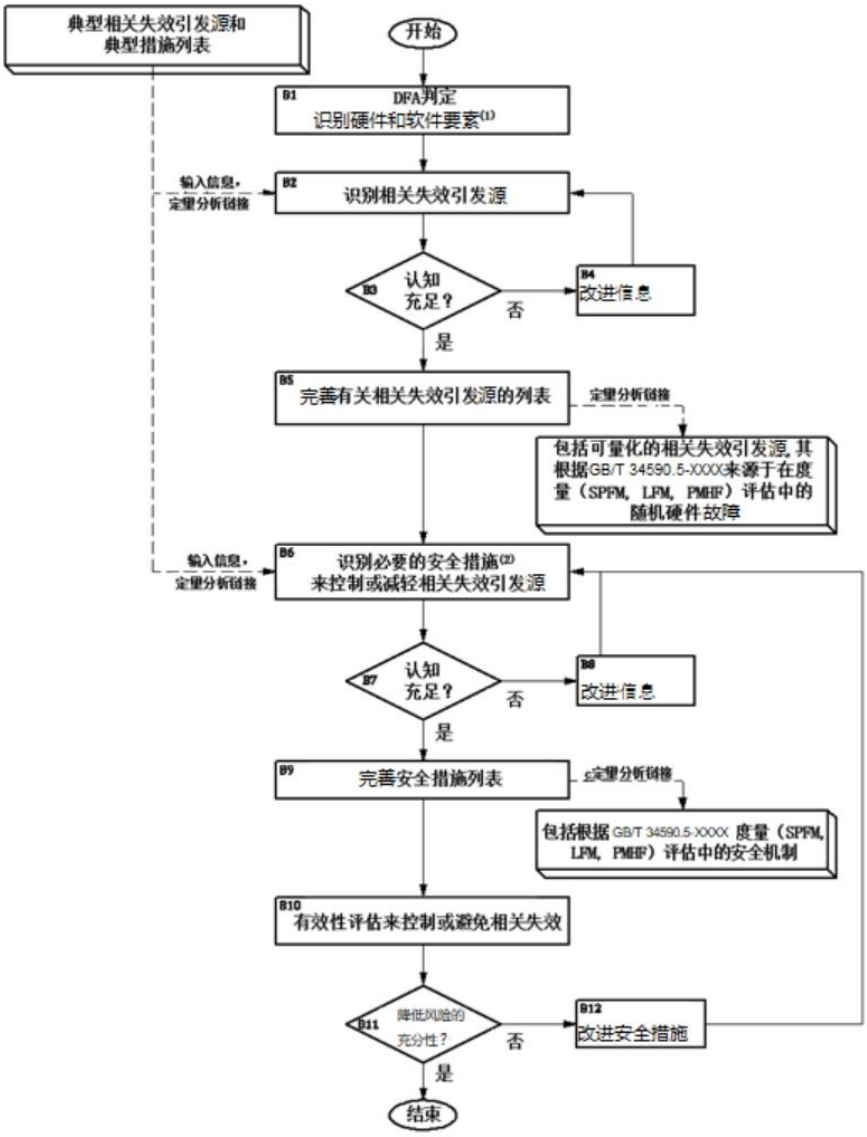


图23　相关失效分析工作流程

注1：固件和在可编程硬件要素上运行的任何微代码，无论它们是否被归类为CPU，都可以被视为软件要素。

注2：安全措施可以是表明失效与相关失效分析不相关的活动。

4.7.6.1　相关失效分析决策和软硬件要素的识别（B1）

按照GB/T 34590.9-XXXX，第7章，每当半导体要素被要求具有独立性或免于干扰时，相关失效分析必须被实施，例如：

——分配给硬件或软件要素的诊断功能；

——硬件或软件要素的相似或不相似冗余；

——硬件组件或元器件上的共享资源（例如：时钟，复位，电源存储器，模数转换（ADC），输入输出接口（I/O），测试逻辑）；

——在共享硬件上执行多个软件任务；

——共享软件功能（例如：输入输出（I/O）程序，中断处理，配置，数学库或其他库函数）；及

——由系统层面或要素层面的ASIL等级分解得到的独立性要求会影响集成电路（IC）的不同要素，其中相关失效分析需提供关于设计中的充分独立性或潜在的共同原因导致进入安全状态的证据（见GB/T 34590 XXXX,第5章）

此步骤的输入为：

——技术安全要求，特别是由系统层面安全概念产生的独立性和免于干扰的要求；

——架构描述，包括框图，流程图，故障树，状态图，硬件分区，软件分区；及

——安全措施。

此步骤的重点是分析架构，并识别可能受上述任何情况影响的每对或每组要素，并评估架构描述是否足够详细，以获取整体设计相关性。此步骤产生的结果是每对或每组要素的列表，这些要素可能受到相关失效，及独立性要求或免于干扰要求的影响。

4.7.6.2 相关失效引发源的识别（B2）

此步骤基于先前的架构分析，其目标是检查得到的独立性要求或免于干扰要求的完整性，并细化分解直到识别完所有导致相关失效的不同引发源。

4.7.5.1中提供的典型相关引发源列表可以用来证明，除了那些从架构推导出的相关失效之外，已知的相关失效是否可被应用。如果在定量分析期间识别了相关失效机理，则进行进一步的交叉检查。

此步骤的结果是对前一步骤列表内容的完善。

4.7.6.3 关于所识别的相关失效引发源影响的现有信息所提供之认知的充分性（B3和B4）

此步骤验证现有文档是否为之前步骤中评估的每个相关失效引发源提供了充分的认知。如果需要附加的信息来判断某个DFI对目标架构的有效性，则需要添加这些信息，并根据更新的描述完成相关失效引发源的识别(步骤2)。

注：建议采用层级分析法，以便分析能在合适的细节程度上进行。例如，顶层视角揭示了哪些是共享的资源。 然后，包含了硬件子元器件及其安全措施的分解视图可用于识别在设计层面的相关性。

4.7.6.4 完善有关的相关失效引发源列表（B5）

根据所提供的信息，已识别的相关失效分析相关要素、独立性要求和实现安全要求的关联相关失效引发源的列表被完善(例如，通过评审)。

按照GB/T 34590.5-XXXX第8章和第9章的规定，可以将来自完善后列表且由随机硬件故障引起的相关失效纳入到所要求度量的定量分析中。

4.7.6.5 用于控制或减轻相关失效引发源的必要安全措施的识别（B6）

为满足独立性要求或免于干扰要求，增设必要的安全措施，以减轻与目标架构有关的相关失效的影响。

4.7.5.1中提供了相关失效引发源示例列表和已知有效措施。最后文档化了所需的安全措施。

注1：对于由随机硬件故障引起的相关失效，定量分析的结果可用于识别那些与实现目标度量有关的失效，按照GB/T 34590.5-XXXX，第8章和第9章。

注2：如果可量化的随机硬件失效被识别为可能的相关失效引发源（例如，共用的振荡器输出的时钟信号对于数字内核的时间约束条件来说太快；由于供电电压调节器故障而造成供给到内部供电端的过压），那么这些失效在定量分析时要予以考虑（见GB/T 34590.5-XXXX, 9.4.3.2，注1）。对于随机硬件故障无法量化的情况（例如，由时钟树中的故障造成的时序效应的影响；在要素与其安全机制之间的热耦合效应； 由于某个区块的故障引

起的衬底电流），需要定性地持续进行减轻措施的评估和确定（参见GB/T 34590.9-XXXX, 7.4.2）。

4.7.6.6 关于已定义的减轻措施的现有信息所提供之认知的充分性（B7 和 B8）

此步骤验证现有文档是否提供了足够的认知来分析前一步骤中所引入安全措施的有效性。如果现有的信息被认为不足以进行适当的评估，可以在相关失效引发源减轻措施定义中增加更多的细节。

4.7.6.7 安全措施列表的完善（B9）

根据更新的文件，已定义的用以减轻相关失效的安全措施的列表被完善（例如，通过评审）

注1：对于纳入定量分析的安全措施（见B5），其效果也可以进行量化评估。

注2：为减轻相关失效引发源而引入的附加安全措施，无论是由定量评估还是定性评估引入，都会改变芯片面积，从而影响芯片各元器件的失效率分布。 因此，定量分析通常会更新。

4.7.6.8 控制或避免相关失效的有效性评估（B10）

用以减轻或避免相关失效而引入的安全措施的有效性需得到验证。按照GBT34590：5第10章，可应用的验证方法与用于避免或减轻随机硬件失效影响或系统性失效影响的安全措施的验证方法相同。以下方式会有用：

——故障树分析（FTA）、事件树分析（ETA）、失效模式和影响分析（FMEA）；

——故障注入仿真；

——基于工艺鉴定测试的特定设计规则应用；

——针对例如器件电压等级或距离的过设计；、

——针对温度剖面或供电和输入过压的压力测试；

——EMC 和 ESD 测试；及

——专家判断。

注1：结果和论据都被文档化并证明。

按照GB/T 34590.5-XXXX，第8和9章，定量分析中包括用于实现安全措施的要素。

注2：当引入的安全措施也可能是相关失效的主体时，新引入的相关失效通过（重新）执行DFA流程来评估其避免或减轻的程度。

注3：如果有类似措施的经验证明可以减轻相关失效，鉴于认为结果具有可转移性，则可以用它来判断所分析措施的有效性。

注4：在分析过程中，可以考虑硬件和软件之间可能的关系（参见GB/T 34590.6-XXXX，第6章）。

4.7.6.9 评估风险降低的充分性并在需要时改进定义的措施（B11 和 B12）

为了完结相关失效分析，需要完成对相关失效的剩余风险的评估。 如果减轻效果被认为不充分，则安全措施需要进一步改进（B12），并对其有效性重新进行评估。

对于残余风险可被量化的情况，它们可在定量分析中进行核算（如果尚未按照B5和B9定量分析途径进行分析）。例如，在一个功能及其安全机制受到相关失效影响的情况下，安全机制的失效模式覆盖率会因为未减轻的相关性而降低。

注：如果达到了定量分析的目标度量，硬件要素会受到被识别为相关失效引发源的故障的影响，即使没有为其分配安全措施，从随机硬件故障的角度看，也可认为风险足够低。相同要素系统性的相关失效引发源在相关失效分析中，用定性的方法处理，可以导出与定量分析结果相独立的安全机制的定义。

4.7.7 相关失效分析示例

按照本条，相关失效分析的详细示例在本部分的附录B中描述。

4.7.8 软件要素和硬件要素之间的相关失效

通常单独考虑硬件和软件相关失效。如果处理硬件的安全机制在软件中实现，则会共同考虑硬件和软件的相关失效。

示例1：基于软件的 CPU 自检与独立的硬件看门狗相结合，以便当 CPU 出现故障时，或由 CPU 的自检功能探测到，或由看门狗捕捉到。

示例2：在 EGAS 概念[55]中，第二层软件监视第一层软件。两个软件要素都可以运行在同一硬件要素上。第一层和第二层已经彼此互不相同，这有助于减少违反安全目标的相关故障。为了进一步降低由于在硬件中的相关故障导致违反安全目标的可能性，引入了附加的安全措施，例如，程序流程监控和 CPU 自检，以解决 CPU 中的相关失效，RAM 模块中第二层软件重要变量的取反冗余存储以及独立的问答看门狗，以确保相关的软件模块已被执行。

4.8 故障注入

4.8.1 总则

当安全概念涉及半导体组件时，半导体组件层面的故障注入是一种已知的方法（见参考文献[30]、[31]、[32]、[33] 、[21]），它可用于支持生命周期的若干活动。

特别地，对于半导体组件，故障注入可用于：

——支持硬件架构度量的评估；及

——安全机制的诊断覆盖率的评估。

注1：如果在合理的时间内使用合理的资源无法获得准确的结果，那么可能的方法是将故障注入限定在一定范围（例如仅在IP模块层级上进行故障注入），或者仅使用分析法或组合使用分析法与故障注入。

示例1：使用故障注入来验证基于软件的硬件测试或基于硬件的安全机制（如硬件内置自检）所能达到的诊断覆盖率。

——诊断时间间隔和故障响应时间间隔的评估；及

——故障影响的确认。

示例2：使用故障注入来评估在特定输入环境中，某个故障导致 IP 输出出现可观察错误的概率，例如，如参考文献[25]所描述的，用来计算瞬态故障的架构脆弱性因子。

——根据安全机制的要求，支持其流片前验证，包括其探测故障和控制故障影响(故障响应)的能力。

示例3：故障注入被用于引起一个错误，以触发基于硬件的安全机制，并验证其在相关软件层级上的正确响应。

示例4：在安全机制的流片前验证期间，使用故障注入来验证特定的临界情况 。

示例5：在安全机制集成期间使用故障注入以验证互连性。

4.8.2 故障注入的特性或变量

关于故障注入，以下信息有助于进行验证计划：

——故障模型的描述和选择理由，以及相关的抽象层级；

——安全机制的类型，包括所需的置信度水平；

——观测点和诊断点；

——故障点、故障列表；及

——故障注入期间使用的工作负载。

特别地，验证计划描述并证明：

——故障模型和相关的抽象层级：

- 正如以下各章所阐述的DFA、数字、模拟和PLD，故障注入可以在适当的层级上进行，具体取决于所考虑的故障模型、具体的半导体技术、可行性、可观察性以及用例；及

注1：根据目的不同，故障注入可以在不同的抽象层级（例如：半导体组件顶层、元器件或子元器件层面、RTL等）实施。这时需要提供抽象层级的选择理由。

示例1：抽象层级的选择也可取决于拟通过故障注入模拟的故障的性质：卡滞故障可以在门级网表进行注入，而位翻转在 RTL 层级注入就可以了。

注2：抽象层级的选择还取决于所需要的精确度。

示例2：通过在门级网表中注入端口故障或网络故障来评估 CPU 基于软件的硬件测试的诊断覆盖率，具有高置信度水平。

- 观察故障影响（观测点）的层级和观察安全机制反应（诊断点）的层级。

示例3：为了验证奇偶校验电路的诊断覆盖率，可以在元器件或子元器件层级设置观察点和诊断点。

示例4：为了验证不同输入输出之间回路的诊断覆盖率，它们可在顶层设置。

注3：如果顶层故障注入不可行（例如，由于待测半导体组件的复杂性），可以通过在仿真环境中创建安全机制的模型，以在元器件层级或子元器件层级实施故障注入。观察点和诊断点可相应地进行设置。这时需要提供证据证明所使用的模型充分反映了安全机制的安全属性。

示例5：在带有看门狗的微控制器的完整 RTL 表示中，看门狗被一个功能上等效的模型所代替。

——故障注入方法。根据目的、可行性和可观察性，故障注入可以通过不同的方法实现；

示例6：在故障点已知的地方直接注入故障、通过形式化的方法注入故障、通过仿真注入故障、通过辐射注入故障。

——根据要被验证的失效模式来考虑即将注入故障的位置（故障点）和 数量（故障列表）。

注4：如果符合指定的目的、置信度水平、安全机制的类型/性质、选择标准等，可以使用抽样因子来减少故障列表。

注5：选择标准包括（例如参考文献[57] 和 [58]）：样本量n（例如仿真或分析了多少个故障和时间点）；样本的分析结果p（例如被安全机制检测到的卡滞故障的比率）；“期望的置信度” α；误差范围（置信度区间）CI，有时用d值表示，则误差范围为p±d；统计独立性。需要为这些选择提供理由。

示例7：在验证双核锁步的诊断覆盖率时，相关的故障可以局限于被比较的 CPU 输出以及相关的故障位置。

示例8：在验证基于软件的硬件测试的诊断覆盖率时，CPU 的内部故障是相关的。

注6：类似故障收敛的技术也可用来将故障总体减少至主要故障

——故障注入控制，与各自安全分析中的相关声明有关；及

示例9：故障注入控制包括注入的故障类型、瞬态故障的持续时间、仿真运行中注入的故障数量、故障发生的时间和位置以及安全机制预期动作的观察窗口。

——故障注入期间使用的测试台架（工作负载）。根据具体目的，测试台架可以来自电路的功能测试套件，或来自与预期用例相似的测试台架。

示例10：为了验证双核锁步比较器的完整性和正确性，可以使用基本工作负载，例如，只激励 CPU 的一部分（如输出）。

示例11：使用一组由功能测试套件导出的激励来验证非对称冗余机制的诊断覆盖率。

示例12：应考虑使用与预期用例类似的工作负载来验证瞬态故障的 F_{safe}（见参考文献[61]）。

示例13：对于 CPU 的基于软件的硬件测试，测试本身是主要的工作负载。

4.8.3 故障注入结果

故障注入的结果可用于验证4.8.1中列出的安全概念和基本假设（例如安全机制的有效性，诊断覆盖率和安全故障数量）。

注1：应维护故障注入的证据用于功能安全审核期间的检查。

注2：仿真的故障和安全分析中识别的故障（例如开路故障）之间并不总是存在确切的对应关系。在这种情况下，安全分析的完善可以基于其他代表性故障的结果（例如，5.1.10.2 中所提到的N-detect测试）。

4.9 生产和运行

4.9.1 关于生产

GB/T 34590.7-XXXX 第5章和第6章的首要目标是开发和维护安装在道路车辆中安全相关的要素或相关项的生产过程。

半导体产品通常采用标准化生产流程，如晶圆加工和裸片装配操作。为特定的产品或封装开发生产流程是可能的，但这比使用标准化流程要少见。通常不可能将过程流中的不同步骤识别为与安全相关或不相关，因此所有步骤都被认为是与安全相关的。

半导体产品通常使用目标工艺制程和相关的器件模型库来设计，这些模型库体现了采用该制程的器件的电气特性。通过遵循一系列标准化的制造工艺（如扩散、氧化沉积、离子注入、晶片装配），要素设计可以以一定的工艺制程来实现，其中每项制造工艺通常都有适当的方法来减轻风险，例如过程FMEA和控制计划。在产品开发过程中使用的器件模型库体现了以该工艺制程制造的器件(如晶体管、电阻、电容器)。通过遵循符合质量标准的受控半导体制造工艺，可以满足要素的安全相关生产要求。产品和流程也都通过制造测试来验证。制造测试根据要素的电气规范来评估要素的性能。制造工艺的性能根据流程控制计划中制定的流程控制规范来进行评估。该测试流程有助于确保所制造的要素符合其要求，包括硬件安全要求。

4.9.2 生产工作的成果

通过满足符合标准的质量管理体系如IATF 16949:2016[51]的要求，可以满足GB/T 34590.7-XXXX，第5章和第6章的要求。如果半导体供应商或分包商拥有符合类似标准的质量管理体系，则可以部分或完全复用现有的工作成果，以满足GB/T 34590.7-XXXX，第5章和第6章的要求。

示例1：生产控制计划的安全相关内容(参见 GB/T 34590.7-XXXX, 5.5.2)可以部分或完全复用质量管理体系生产控制计划的内容。

示例2：控制措施报告(见 GB/T 34590.7-XXXX, 6.5.1)可以部分或完全复用质量管理体系控制措施报告的内容。

4.9.3 关于服务(维护和维修)和报废

通常，在GB/T 34590范围内，半导体组件没有维护或报废要求，并且不可维修。因此，安全计划通常会裁剪掉与维护、维修和报废相关的工作成果，因为它们不属于半导体要素的范围。

在DIA中可以包含半导体供应商和客户在服务和报废方面的预期共识。

4.10 分布式开发中的接口

GB/T 34590.8-XXXX第5章描述了对相关项和要素进行分布式开发的流程并分配了职责。本条的目的是阐明涉及半导体的分布式开发方面的“供应商”一词。

如果半导体开发方作为供应商参与分布式开发，则其应遵守GB/T 34590.8-XXXX第5章的要求。关于安全相关的开发责任,客户(即一级供应商或半导体集成商)负责将半导体开发方作为供应商进行管理。在此背景下，可由半导体开发方执行的GB/T 34590.8-XXXX，第5章的工作成果包括但不限于：

——开发接口协议(GB/T 34590.8-XXXX, 5.5.2)；及

——供应商安全计划(GB/T 34590.8-XXXX, 5.5.3)。

半导体开发方也可以是分布式开发中的客户。半导体开发方的供应商可以来自半导体开发方组织的内部或外部。在所有这些情况下，半导体开发方负责管理其供应商的安全相关开发责任。供应商提供的符合GB/T 34590.8-XXXX第5章的工作成果，将成为半导体开发方安全证据的一部分。在此情况下，按照GB/T 34590.8-XXXX第5章，可由半导体开发方执行的工作成果包括但不限于：

——开发接口协议(GB/T 34590.8-XXXX， 5.5.2)；

——供应商选择报告(GB/T 34590.8-XXXX， 5.5.1)；及

——功能安全评估报告(GB/T 34590.8-XXXX, 5.5.4)。

与安全相关的分布式开发的最低层级是安全责任终止的层级。可能有些较低层级的供应商没有安全责任，例如制造材料的供应商。这些较低层级的供应商可以按照GB/T 34590.XXXX范围之外的要求进行约束，例如质量管理体系的要求。

4.11 认可措施

按照GB/T 34590.2-XXXX 6.4.10、6.4.11和6.4.12对半导体进行认可评审、功能安全审核和功能安全评估。

这些章节对半导体的适用性是根据评估半导体器件的应用场景而裁剪的。如果半导体器件是作为SEooC开发的，则可以按照GB/T 34590.10-XXXX[61]中的指导方针进行裁剪。在知识产权方面，可以按照本部分4.5中的指导原则进行裁剪。

一般来说，每一项关于相关项层面的安全认可评审都将被裁剪掉，因为它们通常不属于半导体供应商的范围。

注：可以通过检查列表支持裁剪。

示例：功能安全审核可以借助流程安全审核(PSA)进行裁剪。PSA 是按照检查表执行的。PSA 检查表基于安全计划，并根据半导体器件评估的应用场景列出了哪些活动和工作成果是必需的。如果识别到差距，就采取措施来弥补这些差距。PSA 按照 GB/T 34590.X-XXXX 表 1 所列的功能安全审核所需的独立性等级执行。

4.12 硬件集成与验证说明

下表27和表28展示了如何将GB/T 34590.5-XXXX 表10和表11应用于半导体中。

注1：这些表是一个起点，可以根据适当的理由针对特定的用例进行修改。

表27 导出半导体层级硬件集成测试的测试用例的方法

方法	半导体级解释
需求分析	将相关的安全要求分配给半导体器件。在半导体行业，这通常在芯片流片前验证（仿真级）和流片后验证（硅级）期间完成。
内部和外部接口分析	与集成电路的集成和集成电路IOs相关的每个流片前或流片后验证活动都可以声明依照了此条目
等价类的生成和分析	按照同质特征组选择测试台架
边界值分析	标准验证技术
基于知识或经验的错误猜测法	例如，在外部分析中识别的潜在设计关注点，例如：设计FMEA

功能的相关性分析	标准验证技术
共因失效的共有限制条件、序列及来源分析	例如：时钟、功率、温度、EMI测试
环境条件和运行用例分析	例如：温度循环、SER实验、HTOL测试
标准（如果存在）	例如：CAN、I2C、UART、SPI等标准
重要变量的分析	例如：PVT(工艺偏差，电压，温度)、特性测试

表28　验证半导体级硬件安全要求实施的完整性和正确性的硬件集成测试

方法	半导体级解释
功能测试	可以通过流片前验证技术覆盖
电气测试	可以通过流片后验证技术进行覆盖，仅限于可以在该级别验证的硬件安全要求
故障注入测试	见4.8

关于表27，“测试用例”一词在系统和半导体组件之间的应用有所不同。半导体组件的测试方法有两种:

——流片后验证侧重于正确集成和避免系统性故障，适用于一小部分器件；及

——生产测试关注的是生产过程中可能发生的故障。最先进的生产测试采用结构化测试。生产测试应用于所有被生产的器件。这与“生产”章节有关，不属于硬件集成验证的范围。

注2：在此应用场景下，术语“测试用例”是指确认测试用例，用于测试设计的功能和电气行为。用于生产测试的测试结构和测试设备也有助于流片后验证。

表27所列的几种方法通常是半导体测试过程的标准方法，除非另有说明，否则它们直接与在规定工作范围（例如电压、温度、频率）内的数据手册技术规范的验证有关。通常，等价类方法和错误猜测方法与半导体硬件测试的相关性较低，因此不太常用。

5　特定半导体技术和应用案例

5.1　数字组件和存储器

5.1.1　关于数字组件

数字组件包括微控制器、片上系统（SoC）器件和专用集成电路（ASIC）等组件的数字电路部分。

5.1.2　非存储数字组件故障模型

常用数字故障模型的列表包括（例如，参考 [56]、[60]）：

——永久性故障，详细描述如下文；及

- 卡滞故障：电路中的故障特征为不管输入激励如何变化，节点保持在逻辑高（1）或逻辑低（0）的状态；
- 开路故障：通过将一个节点破坏为两个或多个节点，从而改变节点数量的电路故障；
- 桥接故障：意外连接的两个信号。根据所采用的逻辑电路，可能导致“线或”或者“线与”的逻辑功能。通常仅限于设计中物理上相邻的信号；及
- 单粒子硬错误（SHE）：由单次辐射事件导致运行的不可逆变化，通常与器件中一个或多个要素的永久性损坏（如栅极氧化物破裂）有关

——瞬态故障，详情如下：

- 单粒子瞬态脉冲（SET）：由于单个高能粒子穿过，造成集成电路某节点瞬时电压漂移（例如，电压尖峰）；
- 单粒子翻转（SEU）：由高能粒子穿过引发的信号所造成的软错误；
- 单比特位翻转（SBU）：单粒子造成的单个存储单元翻转；
- 多单元翻转（MCU）：单粒子引起集成电路中的多个比特位同时失效。错误位通常（但不总是）在物理上相邻；及
- 多比特位翻转（MBU）：两个或多个由单粒子引起的同一个半字节、字节或字中的比特位错误。多比特位翻转不能通过简单的纠错码（ECC）进行校正（例如，单比特位错误校正）。

注1：单粒子瞬态脉冲（SET）、单粒子翻转（SEU）、单比特位翻转（SBU）、多单元翻转（MCU）和多比特位翻转（MBU）通常表示为“软错误”。

注2：当与特定工艺相关时，考虑转换故障和类似的时序相关现象。

注3：某些故障模型可能与其他故障模型具有相同的影响，因此可以通过相同的安全机制进行检测。需要提供适当的理由来证明这种对应关系。

示例：针对卡滞故障而设计的安全机制也能检测桥接故障或开路故障，这些故障随时间的推移将表现为卡滞故障

注4：表29包括与存储器有关的其他故障模型。

5.1.3 存储器详细故障模型

存储故障模型可能因存储架构和存储技术而有所差异。半导体存储器的典型故障模型如表29 所示。该表并不完备，也可以根据其他已知故障或结合实际应用进行调整。

注1：通常，在典型的压力条件下，只能激活所列存储故障模型的一个子集，而其他故障模型则可以在下线测试设备处激活。根据测试条件，需要提供证据来表明存储器测试的有效性

注2：如一些出版物（例如参考文献[47]）所示，不同存储器之间实际的缺陷分布可能不同。因此，可以基于帕累托（pareto）故障模型来更改先前的故障模型列表以及与目标诊断覆盖率的关系。

表29 存储器要素故障模型

要素	故障模型
闪存（NAND、嵌入式）	卡滞、其他故障模型[a]、软错误模型
只读存储器、一次性可编程存储器、电编程熔丝	卡滞、其他故障模型[a]
电可擦可编程只读存储器	卡滞、其他故障模型[a]

嵌入式随机存储器	卡滞、其他故障模型[a]、软错误模型
动态随机存储器	卡滞、其他故障模型[a]、软错误模型

[a] 例如，卡滞开路故障（SOFs），某种耦合故障。 基于存储器结构，例如，寻址故障（AF）、寻址延迟故障（ADF）、转换故障（TFs）、邻域模式敏感故障（NPSFs）、感应晶体管缺陷（STDs）、字线擦除干扰（WED）、位线擦除干扰（BED）、字线编程干扰（WPD）、位线编程干扰（BPD）。这些故障模型是针对随机存储器的。但对于嵌入式闪存或与非型闪存（NAND FLASH），相同的故障模型依然有效，即使是由不同的现象引起的（参见参考文献[48]、[49]和[50]）。例如，卡滞开路故障（SOFs），某种耦合故障。 基于存储器结构，例如，寻址故障（AF）、寻址延迟故障（ADF）、[b]转换故障（TFs）、邻域模式敏感故障（NPSFs）、感应晶体管缺陷（STDs）、字线擦除干扰（WED）、位线擦除干扰（BED）、字线编程干扰（WPD）、位线编程干扰（BPD）。这些故障模型是针对随机存储器的。但对于嵌入式闪存或与非型闪存（NAND FLASH），相同的故障模型依然有效，即使是由不同的现象引起的（参见参考文献[48]、[49]和[50]）。

5.1.4 数字组件的失效模式

本条举例说明了如何根据数字组件的功能规范来描述其失效模式。

作为分类的示例，对于要素的任何功能，可以将要素失效模型化为：

——功能遗漏：需要时功能未执行（FM1）；

——功能误启动：不需要时功能执行（FM2）；

——功能时序：功能执行时序错误（FM3）；及

——功能值：功能提供不正确的输出（FM4）。

失效模式可以适用到任何逻辑功能。在安全分析（GB/T 34590.9-XXXX 第8章）中，通过根本原因影响分析来增强失效模式的描述，以了解失效模式如何传递到其他元器件或子元器件。

通常，IP模块的失效模式可以按不同的抽象层级，并基于模块的无故障功能和有故障行为的不同角度进行描述。失效模式集合的选择会影响到安全分析的可行性、工作量和可信度。合理的、目标导向的失效模式集合定义的准则是：

——失效模式允许将底层技术故障映射到失效模式，如 4.3 所述；

——失效模式支持所使用的安全机制的诊断覆盖率的评估；及

——各失效模式在理想情况下是分隔的，即每个初始故障理想情况下只会导致一个特定的失效模式。

注：在建议的抽象级别上，多种失效模式可能由相同的物理性根本原因引起。

示例：FM3（时序）和 FM4（值）可能是由卡滞故障或影响某些内部逻辑功能的软错误引起的。假如 FM3 和 FM4 由具有不同诊断覆盖率能力的不同安全机制控制，应对失效模式分布的安全概念则更为鲁棒。

附录A提供了如何使用数字失效模式进行诊断覆盖率评估的示例。

5.1.5 常规数字模块的失效模式定义示例

表30 包含常规IP模块的示例性、非约束性失效模式定义。

表30 数字组件失效模式示例

元器件/子元器件	功能	失效模式应考虑的方面[a]

中央处理单元（CPU）	按照给定的指令集架构执行给定的指令流。	CPU_ FM1：给定指令流未执行（完全遗漏） CPU_ FM2：非预期指令流被执行（误启动） CPU_FM3:指令流执行时间错误（过早/过晚） CPU_FM4：指令流结果不正确 如有必要，可将CPU_FM1进一步细化为： — CPU_FM1.1：由于程序计数器挂起，给定的指令流未执行（完全遗漏） — CPU_FM1.2：由于指令取指挂起，给定的指令流未执行（完全遗漏）
CPU 中断处理电路（CPU_INTH）	按照中断请求执行中断服务程序（ISR）	CPU_INTH_FM1: ISR未执行（遗漏/太少） CPU_INTH_FM2：非预期ISR执行（误启动/太多） CPU_INTH_FM3：延迟的ISR执行（过早/过晚） CPU_INTH_FM4：不正确的ISR执行（见CPU_INTH_FM1/2/4）
CPU存储器管理单元(CPU_MMU)	存储器管理(MMU)通常执行两种功能： —将虚拟地址转换为物理地址 —控制存储器访问权限。	CPU_MMU_FM1:地址转换未被执行 CPU_MMU_FM2：未请求时的地址转换 CPU_MMU_FM3：延迟的地址转换 CPU_MMU_FM4：转换的物理地址不正确 CPU_MMU_FM5：非预期的阻止访问 CPU_MMU_FM6：非预期的允许访问 CPU_MMU_FM7：延迟的访问
中断控制单元（ICU）	按照基于硬件或基于软件的中断事件以及预期服务质量（例如，优先级），向给定的CPU发送中断请求。中断控制器可以服务于多个CPU。	ICU_FM1: 对CPU的中断请求丢失 ICU_FM2：无触发事件时，向CPU请求中断 ICU_FM 3：中断请求过早/过晚 ICU_FM 4：中断请求发送错误数据

直接内存访问	数据传送：当请求传送数据时，将数据从源地址移到目标地址，并通知数据传输完成。 被传送的数据集称为一个消息。	DMA_FM1：请求的数据传送未发生。消息未按预期发送到目标地址。 DMA_FM2：无请求的数据传送。 DMA_FM3：数据传送过早/过晚。 DMA_FM4：输出不正确
（第一级抽象）		
总线和互连（内部通讯）	按照预期的服务质量，将从给定总线主机发起的总线事务传递到目标地址（TXFR）。 事务是由总线协议定义的一组给定数据。	BUS_TXFR_FM1:请求的事务未送达 BUS_TXFR_FM2：无请求地发送事务 BUS_TXFR_FM3：在错误的时间发送事务 BUS_TXFR_FM4：以错误的数据发送事务
带SDRAM控制器的外部同步动态随机存储器	易失性存储器根据SDRAM控制器的输入命令，按照给定行和列地址，获取（读取）或存储（写入）数据。	SDRAM_ RW_FM1：给定的写入/读取访问未执行（遗漏） SDRAM_RW_FM2：非预期的写入/读取访问被执行（误启动） SDRAM_ RW_FM3：写入/读取访问结果不正确（过早/过晚） SDRAM_ RW_FM4：写入/读取访问结果不正确
或（第二级抽象）		
带SDRAM控制器的外部同步动态随机存储器	SDRAM控制器提供行地址，以便在选定的分区上执行读写运行。	SDRAM_ RA_FM1：给定的行地址未被访问（遗漏） SDRAM_ RA_FM2：非预期的行地址被访问（误启动） SDRAM_ RA_FM3：行地址结果延迟（过早/过晚） SDRAM_ RA_FM4:行地址结果不正确
带SDRAM控制器的外部同步动态随机存储器	SDRAM控制器提供列地址，以访问数据进行读写运行。	SDRAM_ CA_FM1：给定的列地址未被访问（遗漏） SDRAM_ CA_FM2：非预期的列地址被访问（误启动） SDRAM_ CA_FM3：列地址结果延迟（过早/过晚） SDRAM_CA_FM4:列地址结果不正确

带SDRAM控制器的外部同步动态随机存储器	SDRAM控制器提供命令（例如，激活、写入、读取、预充电、刷新…）以获取用于读或写运行的数据。	SDRAM_IN_FM1：给定指令未执行（遗漏） SDRAM_IN_FM2：非预期的指令被执行（误启动） SDRAM_IN_FM3：指令结果延迟（过早/过晚） SDRAM_IN_FM4：错误的指令结果
带SDRAM控制器的外部同步动态随机存储器	SDRAM数据路径提供对存储器阵列的读/写数据。	SDRAM_DW_FM1：给定数据字未执行（遗漏） SDRAM-DW-FM2：非预期的数据字被执行（误启动） SDRAM_DW_FM3：数据字结果延迟（过早/过晚） SDRAM_DW_FM4：数据字结果不正确
带闪存控制器的外部闪存	非易失性存储器按照闪存控制器的输入命令，将数据提取（读）或存储（写）到给定的地址。	FLASH_RW_FM1：给定的写入/读取访问未执行（遗漏） FLASH_RW_FM2：非预期的写入/读取访问被执行（误启动） FLASH_RW_FM3：写入/读取访问结果延迟（过早/过晚） FLASH_RW_FM4：写入/读取访问结果不正确
（第一级抽象）		
带SRAM控制器的静态随机存储器	为变量和/或常量提供存储空间。 分析是在考虑了存取控制逻辑后进行的，从硬件要素发起命令的角度来看，该逻辑称为SRAM控制器。 典型地，命令是读取、写入或可能是读取-修改-写入。	SRAM_RW_FM1：给定命令未执行（遗漏） SRAM_RW_FM2：非预期的命令被执行（误启动） SRAM_RW_FM3：命令结果延迟（过早/过晚） SRAM_RW_FM4:命令结果不正确
带SRAM控制器的静态随机存储器	静态随机存储器硬宏（HM）：按照SRAM控制器的输入命令提供数据或存储数据到给定的地址。	SRAM_HM_FM1:来自SRAM控制器的命令未执行（遗漏） SRAM_HM_FM2：非预期访问SRAM，例如，由瞬态故障引起的非预期访问 SRAM_HM_FM3：SRAM命令延迟（过早/过晚），例如，由内部时序生成导致的延迟 SRAM_HM_FM4:最终SRAM数据损坏或写入到错误位置

带eFLASH控制器的嵌入式闪存（eFLASH）	非易失性存储器（NVM）存储程序代码和数据常量。 编程和擦除功能。擦除挂起和恢复运行以中断正在进行的擦除运行。	eFLASH_E_FM1：编程或擦除未被执行。 eFLASH_E_FM2：未请求而被执行的编程或擦除操作。 eFLASH_E_FM3：编程或擦除时间不正确 eFLASH_E_FM4：编程或擦除的内容错误。
	非易失性存储器（NVM）存储程序代码和数据常量。 读取功能	eFLASH_R_FM1：读取访问未被执行。 eFLASH_R_FM2：未请求的读取访问。 eFLASH_R_FM3：读取访问时间不正确。 eFLASH_R_FM4：读取访问得到错误的内容。
数据一致性	一致性是由独立于底层架构的一致不变量定义的。本示例选择的不变量基于参考文献[52]。	基于该主题的复杂性，失效模式仅仅是可以导致给定地址的不一致状态的情况的几个示例。 COHERENCY_FM1：写入存储器A未执行（遗漏）。存储器被看做是由一致性参与方更新的。这种失效模式导致存储器A处于非一致状态。 COHERENCY_FM2：非预期写入存储器A（误启动）。这种情况可能与多个内核试图写入同一位置的情况有关。 COHERENCY_FM3：存储器A的更新（写入）延迟（过早/过晚）。一种可能的情况是，某合法写操作被延迟，但一致性协议内的其他参与方认为地址内容是一致的。 COHERENCY_FM4：存储器A的内容已损坏。这可能是由错误的写入命令（参见例如SRAM）或存储要素中的缺陷造成的。
通信外围设备（COM） 可适用于CAN、Flexray、以太网、SPI	按照接口协议将软件提供的数据传输到外部接口。 按照接口协议接收和处理外部接口提供的数据。并通知软件数据可用。 被传输的数据集称为消息。	COM_TX_FM1：请求的消息未被传输 COM-TX-FM2：未请求时，消息被传输 COM-TX-FM3：消息被传输过早/过晚 COM-TX-FM4：有错误值的消息被传输 COM_RX_FM1：传入消息未被处理 COM_RX_FM2：未请求时，消息被传输 COM_RX_FM3：消息被传输过早/过晚 COM_RX_FM4：有错误值的消息被传输

信号处理加速器	加速器（如GPU、DSP）按照给定的代码和/或配置，从数据源（如传感器数据）获取高带宽信号并对其进行处理（如算术处理）。通常这样做是为了减轻通用CPU的工作负荷，而CPU只能以较低效率执行那些任务。通常，这些处理需要满足实时性要求。	SP_FM1：处理停滞，没有输出或输出定值（服务遗漏） SP_FM2：未请求的输出或中断（服务意外启动） SP_FM3:输出结构性损坏，例如，帧损坏（服务时间） SP_FM4：输出结构正常，但数据错误（服务值）
[a] 失效模式可能由永久性随机硬件故障和瞬态随机硬件故障引起。		

5.1.6 数字组件的定性和定量分析

如 GB/T 34590.9-XXXX第8章所示，在概念和产品开发阶段，在合适的抽象层面上进行定性和定量安全分析。对于数字组件：

——定性分析有助于找出数字组件的失效模式。一种可行的方法是使用从数字组件模块框图中获得的信息和从本部分中获得的信息；

注1：附录A 给出了如何定义数字组件失效模式的示例。

注2：定性分析包括4.7 所示部分的相关失效分析。

——采用下列组合进行定量分析：

- 逻辑块级构造；
- 从数字组件寄存器传输级（RTL）描述（获取功能信息）和门级网表（获取功能和结构信息）中获得的信息；
- 用来评估子功能间潜在不确定的交互的信息（相关失效，见4.7）；
- 布局信息：仅在最后阶段可获取；
- 用以验证某些特定故障模型（如桥接故障）的诊断覆盖率的信息（见5.1.2）。这仅适用于某些情况，如元器件和其相应安全机制间进行比较的情况；及
- 有理由支持的专家判断，和对系统级措施有效性的仔细考虑。

注3：GB/T 34590.5-XXXX，附录D作为出发点，用声明的有合适理由支持的诊断覆盖率去评估这些安全机制的诊断覆盖率。

注4：定量分析所用信息可在数字组件开发阶段逐步获得。因此，相关分析可以根据最新信息重复进行。

示例1：在定量分析的初始阶段，前期可测试性设计（DFT）和前期布局门级网表可能是存在的，随后使用后期可测试性设计（DFT）和后期布局门级网表重复进行分析。

注5：每当进行定量分析时，分析的准确性都会计入其结果中。有效性论证表明结果的可信度，同时对结果进行适当修正（例如，保护带），以确保高度确定性。有关计算和验证（在这种情况下，使用故障注入）的可信度的讨论，见 5.1.10。

——由于数字组件的元器件和子元器件可在单个物理组件中实现，因此相关失效分析和独立性或免于干扰分析都是针对数字组件的重要活动。有关详细信息，见 4.7。

注6：相关失效分析是在定性的基础上进行的，因为不存在普遍的和足够可靠的方法来量化此类失效。

示例2：在设计初期启动相关失效的评估。应定义设计措施以避免和揭示相关失效的潜在来源或者探测相关失效对“片上系统”安全性能的影响。在最终设计阶段进行布局确认。

5.1.7 数字组件的定量分析说明

5.1.7.1 如何考虑数字组件的永久性故障

失效率计算的通用性要求和建议在GB/T 34590.5-XXXX中进行了定义，并在本部分4.6中针对半导体组件进行了裁剪。

按照GB/T 34590.5-XXXX附录E中给出的示例，数字组件永久性故障的失效率和度量可以通过以下方式计算：

——根据需要将数字组件分为层级（元器件、子元器件或基础子元器件）；

注1：对确定元器件的独立性假设在相关失效分析过程中可得到验证。

注2：必要的详细程度（例如，是否只到元器件层面，或是否深入到子元器件或基础子元器件层面）取决于分析的阶段和所用的安全机制（在数字组件内部或系统层面或要素层面）。

示例1：对于具有硬件锁步安全机制的 CPU，分析将 CPU 功能作为一个整体来考虑，而锁步比较器则需要更详细的描述。

示例2：对于基于结构化软件的硬件测试的 CPU，由于软件测试将以不同的失效模式覆盖率来覆盖不同的失效模式，因此需要对失效模式进行更详细的定义。

示例3：元器件或子元器件的失效率计算的准确性置信度与详细程度成比例：低详细程度用于概念阶段的分析是合适的，而更高详细程度则用于开发阶段的分析。

注3：由于现代数字组件（成百上千个元器件和子元器件）的复杂性，为了保证分析的完整性，用自动工具支持划分过程是很有帮助的。需要谨慎处理以保证跨模块边界的数字组件层面分析。如果寄存器传输级（RTL）可用，则依照寄存器传输级的结构层级进行分区。

——如 4.6.2.4 所述，可以使用以下两种方法之一计算每个元器件或子元器件的失效率：

——如果给出了整个数字组件裸片（即不包含封装和键合）的总失效率（以 FIT 为单位），则可以假设元器件或子元器件的失效率等于元器件或子元器件的占用面积（即与门、触发器以及互连有关的相应面积）除以数字组件芯片的总面积，再乘以总的失效率。或者

注4：对于带有功率级的混合信号芯片，该方法限于各个域内部应用，因为数字域的总失效率可能与模拟域和功率域有差异。有关详细信息，请参见5.2。

示例4：如果一个 CPU 面积占整个数字组件裸片面积的 3%，则可以认为其失效率等于总数字组件裸片失效率的 3%。

——如果给出了基础失效率，比如数字组件的基本子元器件（如门电路）的失效率，则认为元器件或子元器件的失效率等于基本子元器件的数量乘以其失效率的乘积的最后总和。

注5：有关如何得出基础失效率值的示例，见4.6。

——通过将故障分为安全故障、残余故障、可探测的双点故障和潜伏的双点故障，来完成评估；及

示例5：在 CPU 内实现的调试单元的某些部分是安全相关的（因为 CPU 本身是安全相关的），但它们本身不会导致直接违反安全目标，或者它们的发生不会显著增加违反安全目标的概率。

——确定元器件或子元器件的残余故障和潜伏故障的对应的失效模式覆盖率。

示例6：通过将子元器件划分为更小的子元器件，并对每个更小的子元器件，计算其安全机制的预期覆盖能力，可以计算出某个失效率相关的失效模式覆盖率。例如，计算 CPU 寄存器组失效的失效模式覆盖率时，可以通过将寄存器组分成较小的子元器件，每个子元器件与特定寄存器（例如 R0、R1，…）相关，并计算对应安全机制的失效模式覆盖率，例如，针对每个相应底层失效模式的失效模式覆盖率。

注6：相关失效可能影响安全机制的有效性。4.7中列出了适当的需要考虑的措施。

注7：由于现代数字组件（百万门）的复杂性，故障注入方法有助于计算，并用于验证安全故障数量，尤其是失效模式覆盖率。详细解释参见4.8和5.1.10。故障注入不是唯一的方法，其他方法也是可能的，如5.1.10所述。

5.1.7.2 如何考虑数字组件的瞬态故障

5.1.7.2.1 瞬态故障失效率

如GB/T 34590.5-XXXX中8.4.7注2所述，当瞬态故障与所用技术相关时，要考虑这些瞬态故障。可以通过给它们指定并确认一个特定的“单点故障度量”目标值，或通过一个定性理由来处理这类瞬态故障。

注：为所选流程给出理由。

当使用定量方法时，使用瞬态故障的基础失效率计算每个元器件或子元器件的瞬态故障失效率。

由于随机访问存储器RAM中存储元件的数量和密度，导致瞬态故障的失效率可能显著高于与处理逻辑电路或数字组件的其他元器件相关的失效率。因此，按照GB/T 34590.5-XXXX，8.4.7，注1推荐的，分别计算RAM存储器和数字组件其他元器件的度量，可能会有所帮助。

5.1.7.2.2 瞬态故障分类

对于瞬态故障，大部分和安全故障有关。为了证明所估算的安全瞬态故障的数值，需要提供关于结果和用于得出结果的假设的依据。

注1：依据可由4.8所述的故障注入得出，也可基于电路架构或应用的论据得出。

示例1：存储安全相关常量的寄存器中的故障（即，仅写入一次但在每个时钟周期读取的值，如果错误，则违反安全目标）永远不会是安全的。反之，例如寄存器每隔 10 ms 进行写入，但仅在写入后 1ms 用于安全相关计算一次，寄存器中的随机瞬态故障将导致 90%的安全故障，因为在剩余的 90%时钟周期中，该寄存器中的故障不会导致违背安全目标。

注2：如GB/T 34590.5-XXXX，8.4.7注2中所述：瞬态故障可通过单点故障度量进行处理。就潜伏故障而言，不考虑瞬时故障。由于根本原因迅速消失（根据瞬态定义），因此不会计算瞬态潜伏故障的失效模式覆盖率。此外，假设在大多数情况下，影响将迅速消除，例如，在第二个故障可能导致多点故障发生之前，通过随后的下电循环消除由瞬时故障改变的触发器或存储器单元的错误状态。在特殊情况下，这一假设可能无效，需要采取附加措施，并根据具体情况加以解决。

注3：瞬时故障包含在受影响的子元器件内，如果未在逻辑上连接，则不会无意间扩散到其他子元器件。

注4：GB/T 34590.5-XXXX附录D表D.3至D.10中定义的安全机制的某些覆盖率值仅对永久性故障有效。这一重要的区别可以在相关的安全机制描述中找到，其中描述了针对瞬态故障应如何考虑覆盖率值。

示例2：随机访问存储器（RAM） 跨步测试（见表 33）覆盖率的典型值被评为高。然而在相关描述中（5.1.13.7）写明了该类型的测试对软错误的探测是无效的。因此，例如，随机访问存储器（RAM） 跨步测试对瞬态故障的覆盖率为零。

5.1.8 定量分析示例

附录C 给出了定量分析的示例。

5.1.9 数字组件设计过程中，检测或避免系统性失效的技术或措施示例

有关硬件架构和详细设计的一般要求和建议分别在GB/T 34590.5-XXXX、7.4.1和GB/T 34590.5-XXXX、7.4.2中定义了。此外，与硬件验证相关的要求在 GB/T 34590.5-XXXX，7.4.4中给出。

数字组件是基于标准化的开发流程进行开发的。如何提供证据，证明在数字组件开发过程中，是否采取了足够的措施来避免系统性失效，可以参考以下两个示例的方法：

——使用表 31 所采用的检查列表；及

——使用类似产品的现场数据，并且该产品与目标器件是采用相同流程进行开发的。

另外，可以考虑以下通用准则：

——每项设计活动、测试安排和功能仿真所用的工具以及仿真结果，都有文档记录；

——每项活动及其结论，都有验证，比如通过仿真、等效检查、时序分析或对技术约束的检查

——设计实现过程中，使用可重复性和自动化措施（基于脚本、自动化工作和设计实现流程）；及

注：这意味着能够冻结工具版本，以便在未来根据法规要求实现可重复性。

——对于第三方软核和硬核，使用经过确认的宏块。如果可行，应符合宏核提供方定义的每个约束和过程。

表31　数字组件开发过程中，为符合 GB/T 34590.5 要求所采用的技术或措施的示例

GB/T 34590.5-XXXX 要求	设计阶段	技术/措施	目的
7.4.1.6 模块化设计特性	设计入口	结构化描述与模块化	电路功能的描述以易于阅读的方式构建，即，通过描述就可以直观地理解电路功能，而无需模拟仿真工作。
7.4.1.6 模块化设计特性		设计描述使用硬件描述语言（HDL）	使用硬件描述语言（例如 VHDL 或 Verilog）在上层进行功能描述，如寄存器传输级 RTL。
7.4.4 硬件设计验证		硬件描述语言 HDL　仿真	通过仿真针对 VHDL 或 Verilog 描述的电路进行流片前验证。
7.4.4 硬件设计验证		形式验证	通过静态形式验证针对 VHDL 或 Verilog 中描述的电路进行流片前验证。
7.4.4 硬件设计验证		需求驱动验证	所有功能和安全相关要求均被验证，并通过规范和验证计划之间的追溯关系来展示。
7.4.4 硬件设计验证		模块级流片前验证	“自下而上”流片前验证，比如基于断言的流片前验证，即通过运行时的属性检查来验证以 VHDL 或 Verilog 描述的电路，其中属性是以某些模型或断言语言来定义。
7.4.4 硬件设计验证		顶层流片前验证	全电路验证。
7.4.2.4 鲁棒性设计原则		限制使用异步构造	避免综合过程中的典型时间异常，避免仿真和综合过程中因建模或设计的可测试性不足而产生的模糊性。 对于特定类型的电路，如复位逻辑或极低功耗微控制器，这并不排除异步逻辑可能是有用的：在这种情况下，目的是提出额外的注意事项以处理和验证这些电路。
7.4.2.4 鲁棒性设计原则		主要输入的同步与亚稳定的控制	避免因信号建立和保持时间违反，而导致电路行为不确定。

7.4.4 硬件设计验证		功能和结构覆盖率驱动的验证（以百分比表示验证目标的覆盖率）	功能测试期间，针对所用的验证场景进行定量评估。覆盖率的目标等级被定义和证明。
7.4.2.4 鲁棒性设计原则		遵守编码准则	严格遵守编码风格，可以保证电路代码句法和语义正确。
7.4.4 硬件设计验证		使用代码检查工具	使用代码检查工具自动验证编码规则（“编码样式”）。
7.4.4 硬件设计验证		仿真结果归档	为验证特定电路功能，成功仿真用到的每个数据需要归档。
7.4.4 硬件设计验证	综合	时序约束检查，或传输延迟的静态分析（STA – 静态时序分析）	在综合过程中，对是否满足时序约束进行验证。
7.4.4 硬件设计验证		门级网表与参考模型的比对（形式等效检查）	对综合出的门级网表进行功能等效性检查。
7.4.1.6 模块化设计特性		综合约束、结果和工具归档	生成最终的门级网表的最优综合所需的每个定义的约束均需要归档。
7.4.1.6 模块化设计特性		基于脚本的过程	结果的可重复性，综合循环的自动化。
7.4.2.4 鲁棒性设计原则		对于使用不足 3 年的工艺技术，需要留有足够的时间裕度	即使工艺和参数有强烈的波动，也能保证所实现的电路功能的鲁棒性。
7.4.1.6 模块化设计特性（可测试性）	测试插入和测试模式生成	可测试性设计（取决于测试覆盖率的百分比）	为达到生产测试和在线测试的高测试覆盖率，应避免不可测或不易测的结构
7.4.1.6 模块化设计特性（可测试性）		基于所达到的测试覆盖率百分比，采用 ATPG（自动测试模式生成）时测试覆盖率的证明	在生产测试过程中，通过综合测试模式（扫描路径、BIST），可确定所期望的测试覆盖率。 覆盖率的目标等级和故障模型被定义和证明。
7.4.4 硬件设计验证		为检查时序约束，在测试插入后对门级网表的仿真，或对传播延时的静态分析（STA）	在测试插入期间，对所达到的时序约束的验证。

7.4.4 硬件设计验证		测试插入后的门级网表与参考模型的比较（形式等效检查）	测试插入后，对门级网表进行功能等效性检查。
7.4.4 硬件设计验证	布置、布线、布局生成	为检查时序约束，在布局后对门级网表的仿真，或对传播延时的静态分析（STA）	在后端设计期间，对所达到的时序约束的验证。
7.4.4 硬件设计验证		供电网络分析	表明供电网路的鲁棒性和相关安全机制的有效性。示例：内部调节器（IR）电压降测试。
7.4.4 硬件设计验证		在测试插入前和测试插入后，对门级网表进行跨时钟域检查	在功能或测试模式下，避免跨时钟域冲突。
7.4.4 硬件设计验证		对布局后的门级网表与参考模型的比较（形式等效检查）	在后端设计后，对门级网表进行功能等效性检查。
7.4.4 硬件设计验证		设计规则检查（DRC）	工艺设计规则的验证。
7.4.4 硬件设计验证		布局与原理图对比检查（LVS）	布局验证。
7.4.5 生产、运行、服务和报废 9.4.1.2、9.4.1.3 专用措施	芯片生产中安全相关特殊特性	确定生产测试过程中可达到的测试覆盖率	在生产测试期间，根据数字组件安全相关的部分进行测试覆盖率评估。
7.4.5 生产、运行、服务和报废 9.4.1.2、9.4.1.3 专用措施		检测和排除早期失效的措施的确定	确保所选用的技术工艺生产出来的芯片的鲁棒性。例如，对于栅极氧化层的完整性（GOI）：高温/高压下运行（老化）、大电流运行、电压压力测试等。其他例子还包括电迁移（EM），应力迁移和负偏置温度不稳定性（NBTI）测试。

7.4.5 生产、运行、服务和报废 10 硬件集成与验证	硬件要素评估	确定和执行鉴定测试，如掉电测试、高温工作寿命（HTOL）测试和功能测试用例	对于具有集成掉电探测功能的数字组件，测试数字组件功能以验证数字组件的输出被设置为定义的状态（例如通过在复位状态下停止微控制器的操作），或者当由掉电探测监控的任何电源电压达到正确操作定义的下边界时，由其他方式（例如通过使能安全状态信号）指示欠压状态。 对于没有集成掉电探测功能的数字组件，测试数字组件功能以验证当电源电压从标称值降至零时，数字组件是否将其输出设置为定义的状态（例如通过在复位状态下停止数字组件的操作）。否则，需要定义应用假设，并考虑外部措施。

5.1.9.1 寄存器传输级 RTL 设计中检测或避免系统性失效的原则、技术或措施

可以考虑软件开发（参见 GB/T xxxxx.6）中使用的一些原则、技术或措施，以减轻寄存器传输级RTL设计过程中的系统性失效。

由于软件开发与使用寄存器传输级RTL进行硬件设计的不同，不可以直接应用GB/T 34590.6-XXXX的内容，需要对其作适当的裁剪并采纳RTL硬件设计的具体需求。

示例1：静态代码分析的类似效果（参见 GB/T 34590.6-XXXX，表 7，条目 1h），可通过代码检查工具进行代码规则（“编码样式”）的自动验证来实现。

示例2：GB/T 34590.6-XXXX 表 7、GB/T 34590.6-XXXX 表 8 和 GB/T 34590.6-XXXX 表 9 中所列方法的类似效果，可通过使用功能和结构覆盖率驱动的验证（验证目标的覆盖率以百分比表示）和基于属性的形式化方法来实现。

注1：关于功能测试期间所采用的验证场景的定量评估，覆盖率的目标等级可以基于：语句覆盖率、块覆盖率、条件/表达式覆盖率、分支/判定覆盖、切换覆盖率和有限状态机（FSM）覆盖率。

注2：在高层级综合流的情况下，如采用OpenCL、C-to-HDL流或基于模型的方法进行的开发，与GB/T 34590.6 要求的交互会更适用。

5.1.10 使用故障注入仿真进行验证

5.1.10.1 总则

如4.8所述，针对半导体组件，故障注入是一种有效的方法。这对于数字电路尤其如此，对于其特定的故障模型来说,在硬件层面上进行单粒子翻转的故障插入测试是不可行的,甚至是不可能的。因此，使用设计模型（例如，在门级网表中完成的故障注入）进行故障注入，有助于完成验证步骤。

注1：故障注入既可用于永久性故障（如卡滞故障），也可用于瞬态故障（如单粒子翻转）。

注2：故障注入只是可行的验证方法之一，还可采用其他可行的方法。

利用设计模型进行故障注入，可以成功地用于辅助安全故障的验证、及其数量和失效模式覆盖率的计算。

示例1：注入故障并利用指定的观测点，以确定故障是否引起了可测量的效果。而且，它还可用于辅助计算和验证失效模式覆盖率的值，即注入能够产生可测量效果的故障，并确定这些故障是否在最大故障处理时间间隔内被安全机制检测或控制。

对通过故障注入进行计算和验证的置信度的评估可基于：

——用于激励被测电路的测试台的质量和完整性；

注3：测试台的质量和完整性根据其激活被测电路的能力进行测量。也可以根据测试台的功能覆盖率进行测量。

——故障注入活动的完整性，是以被覆盖的故障场景占所有可能场景的比率来衡量的；

注4：场景包括故障地点、故障发生、故障持续时间等。

——电路表示的详细程度；及

示例2：门级网表适用于针对永久性故障（如卡滞故障）的故障注入。基于硬件加速器的方法有助于最大化测试执行速度。对于卡滞故障，寄存器传输级 RTL 也是可接受的方法，前提是需要说明门级相关性。

示例3：寄存器传输级 RTL 建模适用于单粒子翻转 SEU 瞬时故障的注入。仿真模型也是单粒子翻转 SEU 瞬态故障的可接受的方法，前提是用 RTL 或门级模型已证明了适当的关联性。

——提供所仿真的安全机制的详细信息。

5.1.10.2 关于卡滞之外其他故障模型的验证

5.1.2条表明，可考虑除卡滞之外的故障模型。

示例1：简化非卡滞故障验证的一种合适方法是提供证据，证明卡滞开路/桥接故障的故障分布是所有故障模型总量中非常有限的一部分，即远低于卡滞 0/1 故障量。

示例2：在某些情况下，硬件安全机制可以更有效地检测每种故障，并且更容易被验证，例如 使用 N-detect 方法进行验证。而另一方面，在基于软件的安全机制处理随机硬件失效的情况下，由于在运行时，上下文可能在后续测试执行间发生变化，对于模式的丰富性，通过 N-detect 技术很难获得高置信度。对于这种情况，可采用替代方案（例如，参考文献[39]）。

如果正确运用，从卡滞仿真中得出的方法（如N-detect测试，参见文献[35]至[37]）也可用于验证非卡滞故障模型。

示例3：由于不需要穷尽，因此根据可能的影响（例如比较器）或统计基础，可以将非卡滞故障模型分析用于所选数字组件子元器件的子集。

示例4：对于 N-detect 测试，“正确运用”意思是通过模式集（即模式丰富性）保证对同一故障的 N 种不同探测。N 的范围在 5 到 10 之间。

注：根据布局分析，故障注入可用于在特定位置注入桥接故障（见 5.1.2），或验证相关失效的影响，如注入时钟和复位故障。

5.1.11 数字组件安全文档示例

提供给系统集成商工作产出物的必要信息，包括需求假设的文档、与SEooC外部设计相关的假设条件，以及可用的工作产出物。

在此基础上，SEooC数字组件的安全文档可以包括开发接口协议（DIA）中指定的以下文件或其中部分文件：

——与数字组件有关的安全档案，见 GB/T 34590.2-XXXX，6.5.4；

——数字组件的安全计划，见 GB/T 34590.2-XXXX，6.5.3；

——当适用时，参见 GB/T 34590.8 中的其他计划，如配置管理计划、变更管理计划、影响分析和变更需求计划、验证计划、文档管理计划和软件工具的鉴定计划；

——GB/T 34590.2 中所述，与执行安全计划适用步骤有关的证据；

——GB/T 34590.5 中所述的硬件规范，例如硬件安全要求规范、软硬件接口（HSI）规范和硬件设计规范；

——与执行验证计划和其他计划中适用步骤有关的报告，参见 GB/T 34590.5 和 GB/T 34590.8，例如硬件安全要求验证报告、硬件设计验证报告，以及硬件集成和验证报告；及

——安全分析相关的报告，如 GB/T 34590.5 、GB/T 34590.8 和 GB/T 34590.9 中所述，包括硬件安全分析报告、数字组件应对随机硬件失效的架构有效性的评审报告、随机硬件失效导致违背安全目标评估的评审报告和相关失效分析结果。

注1：开发接口协议（DIA）规定了哪些文件可以给到数字组件用户，以及文件内容的详细程度。

可以考虑以下信息：

——为数字组件裁剪的生命周期描述；适用的工作产出物列表（描述数字组件生命周期哪些工作产出物适用）；

——数字组件安全架构的描述，包括对数字组件功能的抽象描述和安全机制的描述；

——根据数字组件预期使用的使用假设（AoU）的描述，包括：数字组件安全状态的假设，最大故障处理时间间隔和多点故障检测间隔（MPFDI）的假设，数字组件应用场景的假设，含其外部接口；

——数字组件配置的描述，以及失效被探测后用于控制失效的相关硬件和/或软件流程的描述；

——开发接口协议（DIA）定义了在系统/相关项层面，以下哪些报告是需要的：

- 硬件安全分析报告；
- 应对随机硬件故障的数字组件架构的有效性报告；
- 因为随机硬件失效导致违背安全目标的评估报告；及
- 相关失效分析结果。

——功能安全评估流程的描述，认可措施清单及独立性等级描述，避免数字组件系统性失效的流程的总结。

注2：这份文档可记录在数字组件的名为“安全手册”或“安全应用说明”的文档中。

5.1.12 数字组件和存储器安全机制示例

注：该条对GB/T 34590.5-XXXX 附录D中的数字半导体组件作扩展补充说明。

对于存储器，可以应用表 32和表 33。

表32 非易失性存储器

安全机制/措施	参见技术概述	可实现的典型诊断覆盖率	备注
奇偶校验位	5.1.13.6	低	—
使用错误探测纠错码(ECC)监控存储器	5.1.13.1	高	有效性取决于冗余位数。可用于修正错误
改进的校验和	5.1.13.2	低	取决于在测试区域内的错误位的数量和位置
存储器签名	5.1.13.3	高	—
存储块复制	5.1.13.4	高	—

表33 易失性存储器

安全机制/措施	参见技术概述	可实现的典型诊断覆盖率	备注

随机访问存储器（RAM）模式测试	5.1.13.5	中	对卡滞失效具有高覆盖率。对链接失效无覆盖。适合在中断保护下运行
随机访问存储器（RAM）跨步测试	5.1.13.7	高	对链接单元的覆盖率取决于写和读的次序。通常该测试不适合运行时执行
奇偶校验位	5.1.13.6	低	—
使用错误探测纠错码(ECC)监控存储器	5.1.13.1	高	有效性取决于冗余位数。可用于修正错误
存储块复制	5.1.13.4	高	共因失效模式可以降低诊断覆盖率
运行校验和/ CRC	5.1.13.8	高	签名的有效性取决于与要保护的信息块长度相关的多项式。在校验和计算过程中，需要注意用于确定校验和的值不会改变。 如果返回的是随机数据模式，那么可能性就是校验和最大值的倒数

对于一般数字逻辑，可以应用表 34。

表34　组合逻辑和顺序逻辑

安全机制/措施	参见技术概述	可实现的典型诊断覆盖率	备注
通过软件实现的自检	GB/T 34590.5-XXXX, D.2.3.1	中	—
硬件支持的自检（单通道）	GB/T 34590.5-XXXX, D.2.3.2	中	根据测试的有效性，有可能达到更高的覆盖率。门级是此测试的适当级别。

对于片上互连，可以应用表 35。

表35　片上通信

安全机制/措施	参见技术概述	可实现的典型诊断覆盖率	**备注**
单位硬件冗余	GB/T 34590.5-XXXX, D.2.5.1	低	—

多位硬件冗余（包括ECC）	GB/T 34590.5-XXXX, D.2.5.2	中	多位冗余通过适当的数据、地址和控制线的交错，且如果与一些完全冗余相结合（例如，提供给仲裁），可以达到高的诊断覆盖率。
完全硬件冗余	GB/T 34590.5-XXXX, D.2.5.3	高	共因失效模式可以降低诊断覆盖率
使用测试模式进行检查	GB/T 34590.5-XXXX, D.2.5.4	High 高	取决于模式类型

5.1.13 数字组件和存储器技术概述

5.1.13.1 使用错误探测纠错码(ECC)监控存储器

注1：本部分的表32和表33中引用了此技术/措施。

目的：探测每个字（典型的为32位、64位或128位）中的每个单个位失效、每个双位失效、某些三位失效和某些全位失效。

描述：存储器的每个字元扩展若干冗余位，产生汉明距离至少为4的改进汉明码。每次读取字元时，检查冗余位可以确定是否发生了损坏。如果发现差异，则会生成失效消息。

通过计算数据字及其地址关联的冗余位，该过程还可用于检测寻址失效。否则对于寻址失效，探测的概率取决于返回的随机数据的ECC位数（例如，地址线开路或地址线短路到另一个地址线，导致返回的是两个单元的平均值）。如果寻址错误导致选中完全不同的单元，则很可能只达到更低覆盖率，如果不提供对地址解码器故障的保护，则覆盖率甚至只能为0％。

对于随机访问存储器RAM单元写使能失效，如果存储单元无法初始化，ECC可以达到高覆盖率。如果写使能失效在初始化后影响整个存储单元，则覆盖率为0%。

5.1.13.2 改进的校验和

注：本部分的表32中引用了此技术/措施。

目的：检测所有一位失效。

描述：校验和由合适的算法产生，该算法计算时使用内存块中的每个字。校验和可以作为附加字段存储在只读存储器ROM中，或者将附加字段加入到存储块中，确保校验和算法产生预定值。在稍后的存储器测试中，使用相同的算法再次产生校验和，并将结果与存储的值或定义的值进行比较。如果发现差异，将生成失效消息（参见参考文献[34]）。如果返回随机结果，漏探测的概率为$1/(2^{校验和的长度})$。如果某些数据干扰可能性更高，则校验和可以达到比随机结果更好的探测率。

5.1.13.3 存储器签名

注1：本部分的表32中引用了此技术/措施。

目的：探测每个一位失效和大部分的多位失效。

描述：使用比如循环冗余校验（CRC）算法，将存储块的内容压缩（使用硬件或软件）成一个或多个字节。典型的CRC算法将块的全部内容视为字节串行或位串行数据流，在该数据流上使用多项式生成

器执行连续多项式除法。除法的余数代表压缩的存储内容——这就是存储器的“签名”，并被存储起来。在以后的测试中，将再次计算签名，并与已存储的签名进行比较。如果存在差异，将生成失效消息。

CRC在探测突发错误方面特别有效。签名的有效性取决于被保护的信息块长度有关的多项式。如果返回随机结果，漏探测的概率为$1/(22^{校验和的长度})$（参见参考文献[34]）。

注2：基于当前技术，针对超过4K的存储器通常不考虑使用8位CRC。

5.1.13.4 存储块复制（例如，利用硬件或软件进行比较的双存储器）

注：部分的表32和表33中引用了此技术/措施。

目的：检测所有位失效。

描述：在两个存储器中复制地址空间。第一个存储器以正常方式工作。第二个存储器包含同样的信息并且同第一个并行存取。比较它们的输出，当探测到有差异时就生成失效信息。取决于存储器子系统的设计，在两个存储器中的一个存储相反的数据能够提高诊断覆盖率。如果在两个存储块中存在共同的失效模式（例如共同的地址线、共同的写入准许）或存储单元的物理位置使逻辑上远离的单元却物理相邻，那么诊断覆盖率会被降低。

5.1.13.5 随机访问存储器 RAM 模式测试

注1：本部分的表33中引用了此技术/措施。

目的：探测主要的静态位失效。

描述：将位模式及其补码写入存储器的单元。

随机访问存储器RAM位置通常被单独测试。存储单元内容被保存后，将全0写入单元。然后通过读回0值来验证单元内容。 通过将全1写入单元并读回内容，来重复该过程。如果从1到0的转换失效是所关注的失效模式，则可以执行附加的全0数据的写和读。最后，恢复单元的原始内容（参见文献[34] 第4.2.1节）。该测试在检测卡滞和转换失效方面是有效的，但不能检测大部分软错误、寻址故障和链接单元故障。

注2：在每个独立位置的测试过程中，此测试经常在中断禁止的情况下进行。

注3：因为测试的执行包括读取刚刚写入的值，优化编译器倾向于优化该测试。如果仍采用优化编译器，好的设计实践是通过汇编级代码检查来验证此测试代码。

注4：一些RAM可能发生失效以致对上一个存储单元的访问操作值返回作为当前单元的读取值。若这是一种可能的失效模式，诊断可同时测试两个位置，首先对第一个位置写入一个0，随后对第二个位置写入一个1，然后验证从第一个位置读取的是否为0。

5.1.13.6 奇偶校验位

注：本部分的表 32和表 33中引用了此技术/措施。

目的：探测字中（典型的有8位、16位、32位、64位或128位）单个位或奇数个位失效。

描述：把存储器的每个字都扩展1位（奇偶校验位），此位给每个字补齐偶数个或奇数个逻辑“1”。每次读字时都将检查它的数据奇偶性。如发现1的个数有误，则生成失效信息。应这样选择偶校验或奇校验，即“0字”（全0）或“1字”（全1）中的哪一个在失效事件中更不期望发生，则不选择那个字码。

当数据字及其地址一起用于计算奇偶校验位时，奇偶校验也可用于探测寻址失效。否则，对于寻址失效，对随机返回的数据的探测存在50%的概率（例如，地址线开路或地址线和其它地址线短路，使得返回的是两个存储单元的平均值）。如果寻址错误导致选择完全不同的存储单元，则覆盖率为0％。

对于RAM单元写使能失效，如果该单元不能被初始化，奇偶校验能够探测50%的失效。如果写使能失效在初始化之后影响到整个单元，覆盖率为0%。

5.1.13.7 随机访问存储器（RAM） 跨步测试

注1：本部分的表 33中引用了此技术/措施。

目的：探测主要的持续位失效、位转换失效、寻址失效和链接单元失效。

描述：将0和1序列以特定模式写入存储器单元，并以特定顺序加以验证。

当跨步要素是在处理另一个单元之前应用于存储阵列中每一个单元的一个有限操作序列，跨步测试由跨步要素的有限序列构成。例如，一个操作可能是写一个0到一个单元中，写一个1到一个单元中，从单元中读取预期的0，从单元中读取预期的1。如果预期的“1”没有被读出，失效就会被探测出来。对链接单元的覆盖程度取决于写/读的顺序。

参考文献[34]，第4章，列出了许多不同的跨步测试设计以探测不同的RAM失效模式：卡滞故障、转换故障（不能从1转换到0或从0转换到1但不包括两者都发生）、地址故障和链接单元故障。这些类型的测试对于软错误的探测无效。

注2：这些测试通常在初始化或关机时运行

5.1.13.8 运行校验和/CRC

注：本部分的表 33中引用了此技术/措施。

目的：探测RAM中的单个位失效和某些多位失效

描述：借助合适的算法来生成校验和，此算法使用了存储器块中的每个字位。校验和可作为附加字位存储在RAM中。由于存储块更新时，RAM校验和/CRC通过移除旧数据值并在存储区域加入新数据值也会被加以更新。校验和/CRC为数据块周期性地进行计算并与存储的校验和/CRC进行对比。如果发现有差异，就产生失效信息。如果返回一个随机结果，则探测失败的概率是1/(2校验和长度/CRC长度)。诊断覆盖率可能会随存储器的增大而降低。

5.2 模拟/混合信号组件

5.2.1 关于模拟和混合信号组件

如4.2中所述，半导体组件由元器件和子元器件构成。如果在要素（组件、元器件或子元器件）中处理的信号不限于数字状态，则该要素被视为模拟要素。该准则适用于和物理世界连接的所有测量接口，包括传感器、执行器输出和电源。

模拟组件的每个要素都是模拟的，不包含数字要素。混合信号组件包含至少一个模拟要素和一个数字要素。由于模拟要素和数字要素要用不同的方法和工具来进行设计、布局、验证和测试，因此建议明确区分模拟和数字模块。这样会导致组件配置的多样化，从以模拟模块为主但包含数字支持模块的组件（例如可配置稳压器或自动调零放大器），到像微控制器这样仅具有少量混合信号外设（例如模数转换器和锁相环）的组件。图24显示了典型混合信号组件(包括示例元器件和子元器件)的层次结构。

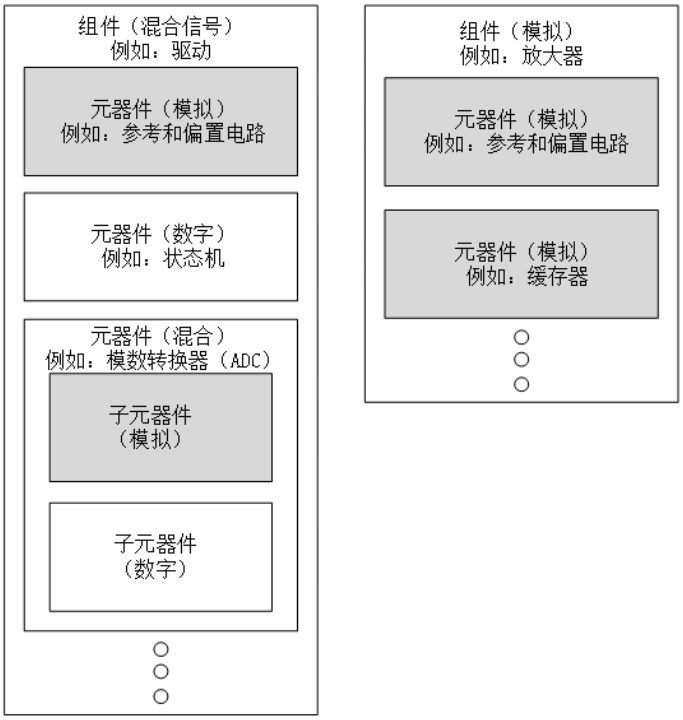


图24　模拟和混合信号组件的通用层次结构

为了简化安全分析，可以将一个混合信号组件分解为模拟要素和数字要素。 模拟要素的边界可以通过其功能及其相关的故障模型和失效模式来定义。此外，每个具有免于干扰或独立性要求的要素（例如，冗余路径或功能和其相应的诊断功能）可由元器件或子元器件的边界区分开。

将混合信号要素（组件或元器件）划分为子要素（元器件或子元器件）时，还可以考虑其他准则：

——信号流；

示例1：混合信号控制回路可由反馈 ADC，数字调节器和输出驱动器组成。

——关联性；

示例2：参考和偏置电路信号可用于多个模拟模块，振荡器信号可用于多个数字或混合信号模块。

——不同的技术；

示例3：高压开关是采用 DMOS 晶体管，而栅极门驱动器可以采用传统的 MOS 器件。

注：区分这些元器件的一个好处是它们可能具有不同数量级的失效率或不同的故障模型。

——不同的供电域；及

示例4：反馈 DAC 可以采用与其他混合信号输出驱动器不同的电源。

——其他划分准则。

示例5：按频率划分，例如高频与低频子元器件。

分析的详细程度取决于相关的安全要求、安全机制以及提供安全机制独立性证据的需要。 更细的颗粒度划分不一定能给安全分析带来显著益处。

5.2.2　模拟和混合信号组件以及失效模式

5.2.2.1　关于失效模式

硬件要素的失效模式取决于其功能。 失效模式分布取决于硬件要素的实现。

注1：“实现”既指实际电路设计，也指所使用的目标工艺。

失效模式的分类取决于分配给集成要素的系统的功能和安全要求。特定的失效模式是否会导致违背安全要求，取决于集成的方式。表36列出了模拟和混合信号元器件或子元器件可能存在的失效模式。该表可用于扩展GB/T 34590.5-XXXX附录D中包含的失效模式列表。

表36中的失效模式以及元器件和子元器件作为一般参考，可以根据具体情况进行调整。 模拟电路的失效模式可以通过使用4.3.2中提到的关键字来导出。

可以根据特定的实现细节，或根据所需的分析颗粒度水平，来调整特定项目中实际使用的失效模式列表（添加或删除失效模式）。

需要说明的是，失效模式的相关性取决于要分析的功能的具体内容， 包括但不限于表36中列出的内容。

示例1：稳压器常见的失效模式有过压和欠压，可采用 5.2.4.2 所述的过压和欠压(OV/UV)监控器来探测这些失效模式。

除了上述示例中提到的常见的失效模式外，重要的是识别每个相关的失效模式，以执行完整的和彻底的分析。

示例2：如果将稳压器用作传感器电源或 ADC 参考电源，那么即使在 OV/UV 阈值范围内，对电压输出稳定性和精度造成影响的失效模式也是至关重要的。 使用适当的措施，可以减轻在 OV/UV 阈值内的输出电压精度不足和输出电压振荡。可以使用独立的 ADC（内部或外部），按照要求的精度对稳压器输出电压进行周期性的测量，以探测这些失效模式。

示例3：如果将稳压器用于对电源电压纹波有严格要求的射频（RF）模块的电源，则防止由输入电压变化引起的调节电压的波动是一个重要特性，即电源抑制比（PSRR）。

在OV/UV范围内的输出电压振荡（即纹波）和影响调制电压的尖峰等失效模式可能是相关的。5.2.4.8中描述的低通滤波器可用于减轻这些失效。

示例4：用作 MCU 内核供电的稳压器，如果在启动（上电）期间对因浪涌电流超过稳压器负载电流和/或电流限制造成的输出电压跌落很敏感，则过快的启动时间可导致严重的失效。可以使用适当的稳压器软启动功能来减轻这种失效。

如果失效模式被归类为非安全相关，则在安全分析中应提供合理的理由来支持这种分类。

由于实现的多样性，表36没有给出关于列出的失效模式的定量影响的任何指示，即失效模式分布 。半导体供应商有责任提供这些定量数据。5.2.3.3给出了一个示例。

注2：虽然已知单一的物理性根本原因可以导致不止一种失效模式，但每种失效模式的分布之和为100%的假设是合理的，这是定量分析的前提。

表36　模拟和混合信号元器件及子元器件的可能失效模式

元器件/子元器件	简要描述	失效模式
稳压器和功率级		

稳压器（线性、开关型等。）	硬件元器件/子元器件，其将电源的电压维持在使用该电压的要素可容许的规定范围内	输出电压高于规定范围的高阈值（即过压 - OV） 输出电压低于规定范围的低阈值（即欠压 - UV） 输出电压受尖峰影响[b] 启动时间不正确（即：在预期范围之外） 输出电压精度太低，包括漂移[c] 输出电压在规定的范围内振荡 输出电压受快速振荡[a]影响，超出规定范围，但平均值在规定范围内 静态电流（即保证稳压器内部电路正常工作需要的电流）超过最大值
电荷泵，升压稳压器	硬件元器件/子元器件，使用开关技术和电容储能要素转换和有选择地调节电压，并在变化的输入电压下保持恒定的输出电压。	输出电压高于规定范围的高阈值（即过压：OV） 输出电压低于规定范围的低阈值（即欠压：UV） 输出电压受尖峰影响[b] 不正确的启动时间（即超出预期范围） 静态电流（即保证稳压器内部电路正常工作需要的电流）超过最大值
高边/低边(HS/LS)驱动器	在单一方向上向负载施加电压的硬件元器件/子元器件： - 用于将负载连接到高压轨的高边驱动器 - 用于将负载连接到低压轨的低边驱动器。	HS / LS驱动器卡滞在ON或OFF状态 HS / LS驱动器浮空（即开路，三态） HS / LS驱动器导通时阻抗过高 HS / LS驱动器关断时阻抗过低 HS / LS驱动器导通时间太快或太慢 HS / LS驱动器关断时间太快或太慢
半桥驱动器或 全桥（H桥）驱动器	可以通过任一方向在负载上施加电压的硬件元器件/子元器件。 半桥驱动器由两个驱动器（一个高边和一个低边驱动器）构成。 一个H桥（或全桥）驱动器由四个驱动器（两个高边和两个低边驱动器）构成	HS/LS驱动器卡滞在ON或OFF状态 HS/LS驱动器浮空（即开路，三态） HS/ LS驱动器导通时阻抗过高 HS/LS驱动器关断时阻抗过低 HS/LS驱动器导通时间太快或太慢 HS/LS驱动器关断时间太快或太慢 “死区时间”太短（在关闭高边驱动器并打开低边驱动器，或关闭低边驱动器并打开高边驱动器时） “死区时间”太长

高边/低边预驱动器	硬件元器件/子元器件 驱动用作高边或低边驱动器的外部FET的栅极。	HS/LS预驱动器卡滞在ON或OFF状态 HS/LS预驱动器输出电压/电流过高或过低 HS/LS驱动器浮空（即开路，三态） HS/LS预驱动器斜率太缓或太陡
模数转换器和数模转换器[d]		
N位数模转换器 (DAC)[d]	硬件元器件/子元器件 将以“N位”编码的数字数据转换为模拟信号（电压或电流）。	输出卡滞（即高或低） 输出浮空（即开路） 偏移误差（不包括输出上的卡滞或浮空，低分辨率） 单调转换的线性误差（不包括输出上的卡滞或浮空，低分辨率） 满量程增益误差（不包括输出上的卡滞或浮空，低分辨率） 非单调转换 不正确的建立时间（即超出预期范围） 输出信号的振荡[a]，包括漂移[c]
N位模数转换器 (N-bit ADC)[d]	硬件元器件/子元器件 将连续时间和连续幅度的模拟信号(如电压值)转换为“N位”编码的离散时间和离散幅度的数字信号。	一路或多路输出卡滞（即高或低） 一路或多路输出浮空（即开路） 精度误差（即误差超过LSB） 偏移误差（不包括输出上的卡滞或浮空，低分辨率） 无单调转换特性（即给定两个输入模拟电压V1> V2，相应的数字值为D1 <D2） 满量程误差（不包括输出上的卡滞或浮空，低分辨率） 单调转换曲线的线性误差（不包括输出上的卡滞或浮空，低分辨率） 不正确的建立时间（即，超出预期范围）
振荡器和时钟发生器		

振荡器	硬件元器件/子元器件 产生周期性振荡信号， 它可以用作数字电路中的时钟。	输出卡滞（即高或低） 输出浮空（即开路） 不正确的输出信号摆幅（即超出预期范围） 不正确的输出信号的频率（即，超出预期范围，适用时包括谐波，例如EMC辐射） 不正确的输出信号的占空比（即，超出预期范围） 输出频率漂移[c] 输出信号抖动过大
锁相环(PLL)	硬件元器件/子元器件 以控制振荡器来产生方波信号，该方波信号与输入或参考信号保持恒定的相位角（即锁定）。 它可以用作数字电路中的时钟。	输出卡滞（即高或低） 输出浮空（即开路） 不正确的输出信号频率（即，超出预期范围，包括可用时的谐波，比如 EMC 辐射） 不正确的输出信号的占空比（即，超出预期范围） 输出频率漂移[c] 输出信号抖动过大 失锁状态（即相位误差，输出时钟与输入时钟不同步，但不会导致错误的频率和占空比） 输出信号中缺少脉冲 输出信号中出现额外的脉冲
通用器件		

运算放大器和缓冲器	将直流耦合高增益电压放大器与差分输入集成在一起的硬件元器件/子元器件，通常是单端输出。	输出卡滞（即高或低） 输出浮空（即开路） 输出电压的不正确增益（即超出预期范围） 输出电压的不正确偏移（即超出预期范围） 不正确的输出动态范围（即超出预期范围） 不正确的输入动态范围（即超出预期范围） 输出电压精度太低，包括漂移[c] 输出电压受尖峰影响[b] 输出电压振荡[a] 输出电压的稳定时间过短
模拟开关	硬件元器件/子元器件，能够根据数字控制信号的电平切换或传送模拟信号。 通常使用“传输门”实现。	输出卡滞（即高或低） 输出浮空（即开路或三态） 偏移太高以至影响输出信号 控制信号和输出信号之间的电阻或容性耦合，包括串扰 输出信号的衰减 影响输出信号的漂移[c] 影响输出信号的尖峰[b]，例如在开关切换时
电压/电流比较器	硬件元器件/子元器件 将输入模拟信号与预定阈值（即电压或电流常数值）进行比较，并在输出端产生二进制信号； 输出取决于输入信号和阈值之间哪个更高，并且当差值具有相同的极性时，输出保持不变。	电压/电流比较器在需要时未触发 电压/电流比较器错误触发 输出卡滞（即高或低） 输出浮空（即开路） 输出振荡[a]

采样&保持	硬件元器件/子元器件 对连续变化的模拟输入信号的电压进行采样，并在指定的最短时间内将保持其采样值	输出卡滞（即高或低） 输出浮空（即开路） 不正确的采样导致和输入信号相关的输出信号的增益/偏移误差 输出电压的不正确增益（即超出预期范围） 输出电压的不正确偏移（即超出预期范围） 不正确的输出动态范围（即超出预期范围） 不正确的输入动态范围（即超出预期范围） 保持阶段输出电压精度过低，包括漂移[c] 保持阶段输出电压受尖峰影响[b] 保持阶段输出电压振荡[a] 在保持时间内输出不够准确
模拟多路复用器	硬件元器件/子元器件由多个模拟输入信号、多个控制输入信号和一个输出信号组成。	输出卡滞（即高或低） 输出浮空（即开路） 不正确的通道选择 导致输出信号过高的偏移 输入通道和输出信号之间的电阻或容性耦合，包括串扰 选择器和输出信号之间的电阻或容性耦合，包括串扰 不正确的输出动态范围（超出预期范围） 输出信号的衰减 影响输出信号的漂移[c] 影响输出信号的尖峰[b]（例如在开关转换时）
电压参考	硬件元器件/子元器件 产生恒定的DC（直流）输出电压，而不受外部条件如温度，气压，湿度，电流需求或时间变化的影响。	输出卡滞（即高或低） 输出浮空（即开路） 不正确的输出电压值（即超出预期范围） 输出电压精度过低，包括漂移[c] 输出电压受尖峰影响[b] 输出电压在预期范围内振荡[a] 不正确的启动时间（即超出预期范围）

无源网络	硬件元器件/子元器件 由无源器件网络（电阻器和电容器）组成，提供特定的低通传递函数功能	输出卡滞（即高或低） 输出浮空（即开路） 不正确的输出动态范围（即超出预期范围） 不正确的输出信号的衰减（即超出预期范围） 不正确的建立时间（即超出预期范围） 影响输出信号的漂移[c] 影响输出信号的振荡[a]（由于串扰，耦合或寄生效应产生） 影响输出信号的尖峰[b]（由于串扰，耦合或寄生效应产生）
电流源（包括偏置电流发生器）	硬件元器件/子元器件 输出或输入与其上的电压无关的电流（即参考电流）。它通常包括多条分路，它们被分配到需要参考或偏置电流的其他电路。	一路或多路输出卡滞（即高或低） 一路或多路输出浮空（即开路） 不正确的参考电流（即超出预期范围） 参考电流精度过低，包括漂移[c] 参考电流受尖峰影响[b] 参考电流在预期范围内振荡[a] 一个或多个支路电流超出预期范围，而参考电流是正确的 一个或多个支路电流精度过低，包括漂移[c] 一个或多个支路电流受尖峰影响[b] 一个或多个支路电流在预期范围内振荡[a]

[a] 振荡是由内部失效引起的元器件/子元器件的不稳定性，例如，调节回路失效，比较器的低或负迟滞等。振荡包括任何重复的电压和电流变化（即周期性脉冲）。

[b] 尖峰是输出电压或电流的非重复变化，即由于负载跳跃等引起的脉冲。

[c] 漂移是参数（即电流、电压、阈值等）在超出预期电路规范范围的缓慢而连续的变化。缓慢变化意味着比最大故障处理时间间隔慢。 例如，漂移涵盖浮空或卡滞开路的失效模式。

[d] ADC或DAC的几种失效模式可分为两大组：静态误差和绝对精度（总）误差。 静态误差是在转换静态（DC）信号时影响转换器精度的误差，可以通过四个术语完全描述：偏移误差，增益误差，积分非线性和微分非线性。

注1： 每个术语可以以LSB单位表示，或者有时以满量程范围（FSR）的百分比表示。 例如，8位转换器的$^{1}/_{2}$LSB误差对应于0.2%FSR。

注2： 绝对精度（总）误差是模拟值与理想中间值之差的最大值。 它包括偏移，增益和积分线性误差，以及ADC的量化误差。

5.2.2.2 关于瞬态故障

如GB/T 34590.1-XXXX，3.173章节中所定义的，瞬态故障是发生一次后随之消失的故障。 单粒子翻转（SEU）和单粒子瞬态（SET）等软错误被定义为瞬态故障（见5.1.2）。 GB/T 34590.5-XXXX，8.4.7章节中规定，当表明瞬态故障与所使用技术相关时，需对其加以考虑，可通过定量的方法来解决，该方法定义并验证其满足特定目标的“单点故障度量”值，或通过定性理由来解决，该理由是基于对其内部安全机制覆盖这些瞬态故障的有效性的验证。

在地表模拟电路中，瞬态故障是由α粒子或中子撞击或电磁干扰引起的，例如功率瞬变和串扰。 它们可能导致SEU甚至SET（也叫模拟单粒子瞬变，ASET），例如运算放大器，比较器或参考电压电路中的瞬态脉冲。

由于模拟技术的固有特性（在设计中考虑了瞬态及噪音影响），它对瞬态故障的敏感性比数字电路低几个数量级。因此，可以仅针对其数字元器件进行影响分析（例如， Σ-Δ ADC的数字抽取滤波器）

但在某些情况下，如ADC转换周期的前期（参见参考文献[28]）或PLL（参见参考文献[20]）或差分开关电容电路（参见参考文献[10]）， 软错误发生率可能很高。 在这些情况下，需要进行更详细的分析并确定适当的对策（参见参考文献[1]）。

对于混合信号组件，数字元器件中软错误的影响如5.1.7.2所述。

注：通过在模拟电路中进行辐射测试来评估软错误率并不是一项简单的任务。 在这种情况下，主要通过对模拟元器件更详细的分析来评估。

5.2.3 安全分析相关说明

5.2.3.1 总则

5.1章节中给出的示例和指南可用于模拟或混合信号组件。后续的章节介绍了一些要求对模拟或混合信号组件进行附加说明的主题。

5.2.3.2 分析的颗粒度

对模拟要素进行安全分析的一个关键点是正确识别分析的颗粒度。一方面，较小的颗粒度有益于更好的理解失效模式与失效模式分布。 另一方面，较大的颗粒度能够对安全机制进行明确的分配。模拟要素通常用于与物理对象进行交互，因此考虑机械特性并相应地区分失效模式也有所帮助。

如GBT 34590 9：XXXX第8章所述，在概念和产品开发阶段，是在适当的抽象层级上进行定性和定量的安全分析， 因此可以根据分析的目标调整抽象层级。 定性分析更适合用来识别失效模式，而定量分析则用来量化其失效率和分布。

示例：使用窗口型电压监控器监控线性稳压器。 电压监控器位于稳压器的输出端，能够检测过压情况。 如果输出值超出定义的阈值，则认为输出值有问题，例如： 1,2 V±0,12 V。如果分析侧重于调节器的输出，则可以相对容易地区分失效类型（例如，在允许的范围内失效则为安全，过压或欠压则为安全相关 ）并量化电压监控器提供的保护。 但是，很难量化在度量计算中所涉及的每种失效类型的可能性。 如果将分析深入到稳压器内部，例如，关注带隙故障，则更容易分析稳压器每次失效的传递和可能性，但不能简单地量化外部电压监控器对带隙本身提供的保护。

在安全分析中，安全机制的类型可以促成颗粒度级别的选择。如果处理模拟特征的安全机制位于系统或要素层面，则往组件层面的细化可能导致过于复杂的分析。失效模式分布的量化可能需要更高的颗粒度。 例如，与将失效率均等分布在线性稳压器的失效模式上相比，将失效率均等分布在构成线性稳压器的模块（例如带隙，缓冲器，驱动器等）上，会让结果更准确。 关于术语，根据4.2中描述的分类，线性稳压器被认为是元器件，而带隙、缓冲器、驱动器等被认为是子元器件。

5.2.3.3 导出模拟组件的失效模式分布

模拟组件的失效分布取决于电路实现和目标工艺。 每个供应商都提供在分析中使用的失效模式分布的详细信息。

示例1：在初始分析中，可用统一的失效模式分布进行分析，例如， 如果定义了五种失效模式，则每种失效模式按20%分配。 在5.2.3.5的示例中使用的统一失效模式分布。

示例2：可以基于面积考虑每种失效模式的更详细分布； 如果确定该电路面积或造成该失效模式的根本原因的电路面积为5%，则分配给它的失效模式分布为5%。

要按照具体的电路实现及其物理面积来判断适用的失效模式及失效模式分布的详细程度，并相应文档化。

5.2.3.4 关于安全故障

GB/T 34590.10 [61]中指出安全故障可能是以下两类中的一类故障：

——n> 2的所有n点故障，除非安全概念表明它们是与安全要求相关，或

——不会导致违反安全要求的故障。

模拟组件具有连续的信号区域的特征，因此，当在系统中使用时，要考虑公差。模拟功能的公差作为分配给该模拟组件的安全要求的一部分，可以比模拟组件本身的实际公差受到更少的约束。 因此，导致参数失效或漂移但仍保持在这些公差范围内的这部分失效模式是安全的。 因此，模拟组件具有容忍故障的固有能力。 这些故障是安全故障。

示例1：电阻用于限制流过特定分支的电流。 电阻的精度失效使其阻值增加（例如 50%）但不妨碍电流限制功能，是安全故障。

根据所考虑的特定安全要求，要素的特定故障可以具有不同的分类。 更多详细信息，请参阅GB/T 34590.5 。

根据系统配置和安全要求，某些失效模式是与安全要求无关的，即它们不能违反安全要求。 在这种情况下，这些失效模式可归类为安全，这有助于硬件安全度量计算，能增加安全故障的失效率。

示例2：输出驱动器可以用输出斜率控制来限制造成电磁干扰（EMI）的输出值上升和下降的时间。 如果此斜率与违反安全目标无关，则在这个斜率控制时的失效将会是安全故障。

示例3：如果电压调节器仅用于数字电路，则影响稳定性和影响在过压/欠压（OV/UV）阈值以内的输出电压准确性的失效模式可被归类为安全的。

5.2.3.5 模拟组件的定量分析示例

附件D中描述了模拟组件定量分析的详例。

5.2.3.6 相关性失效分析

如GB/T 34590.9-XXXX,7.4.2中的“注”所述，相关性失效分析是在定性分析的基础上进行的，因为没有通用的和足够可靠的方法来量化这种失效。

4.7中报告的步骤也适用于模拟和混合信号组件。 在相关性失效分析中，有些方面可以在处理模拟组件、元器件或子元器件时被清楚地考虑到。

模拟电路本质上对不同区块或功能之间的噪声和干扰敏感。 所以，出于功能原因会使用隔离和分离的方法（例如通过使用屏障和/或保护环或将电路放置在特定距离以外或分离电源分配甚至地层）来保证结构上足够的独立性。 实际上，基板、电源和诸如偏置、时钟或复位之类的全局信号通常被认为

是干扰源，并且要特别注意减少它们的影响。出于功能原因，通常遵循这种良好的设计实践有利于避免相关失效。

模拟电路对导致器件行为不匹配的工艺变化非常敏感。 为了确保两个区块具有“相同”的传递函数，如冗余部分，设计和物理布局的对称性是关键因素。 在这种情况下，需要特别注意确保两个区块的布局完全相同，包括方向，对称放置，布线等；因此，多样性设计并不总是用来避免模拟电路共因失效的有效解决方案。

由此，相关失效引发源通常通过确保隔离或分离的技术来解决，而不是通过将影响差异化的技术来解决。

在其他情况下，多样化设计仍然是探测或避免相关失效的有效技术。 例如，在双通道方案中，使用两种不同的ADC架构（例如，逐次逼近型ADC和Σ－Δ ADC）可以显着降低常见共因失效的发生概率。

5.2.3.7 架构度量计算验证

本条涉及安全分析验证的一个特定部分：验证硬件架构安全度量，特别是安全故障的比例和失效模式覆盖率。

可能的方法包括：

——基于工程方法的专家判断。前提是任何数据，无论是定性的还是定量的，都有理由和相关论据支持，并相应文档化；

注1：在某些情况下，这些论据可以从负责保证此参数的硬件要素的功能特性中得出。 该功能特性的目的是避免系统性失效，而不是硬件随机失效，但在某些情况下，它可以作为证据来证明特定失效模式的覆盖率：这种情况下， 安全机制的目标是100％探测一种或多种失效模式，并由设计保证这个探测能力。

示例1：5.2.4.2 中描述的电压监控器是用于检测影响稳压器的过压和欠压失效模式的典型安全机制。 如果在硬件设计验证期间，电压监控器的功能特性表明：

——对于任何调节电压超出规范中的规定范围，并且其时长足够导致负载硬件电路异常的事件，能被电源监控器检测到；及

——任何导致在规范中规定范围内的调节电压变化的事件，无论多久都不会影响调节器负载硬件电路的正确行为；

那么，这些特征可以用作论据来声称100％探测到所涉及的失效模式。

——如 4.8 中所述，在开发阶段的故障注入仿真是验证与硬件安全要求相关的安全机制实现的完整性和正确性的有效方法。 使用设计模型的故障注入可以成功用于协助验证。该方法可应用于模拟和混合信号组件；及

注2：在特定环境下，故障注入活动可仅限于故障或失效中那些被判断为关键的子集。 最关键的失效模式是在考虑过它们的分布、它们声称的安全故障数量、它们声称的探测水平以及保证这些检测水平的安全机制或安全要求之后确定的。

——上述方法的组合，即通过故障注入来支持专家判断时，仅通过故障注入方法对判断为更关键和/或更可处理的案例提供论证和证据。

5.2.4 安全机制示例

下表列出了常用模拟安全机制示例的非详尽列表，这些机制补充了GB/T 34590.5-XXXX，附录D中包含的信息。

一些模拟安全机制具有数字输出信号，该信号用于控制对失效的反应并使组件进入安全状态。 在许多情况下，该信息需要被存储以便可以通过数字接口进行通信。 其他模拟安全机制通过控制或抑制故障来避免违反安全要求，它们没有与数字域的接口。

为了符合GB/T 34590.5-XXXX,8.4.8，下表中描述的安全机制，作为会导致违反安全目标的双点故障，可能需要附加的措施来检测会影响它们的故障。

表37至表40中所列举示例并非详尽，可以使用其他技术。

注1：不可能为传感器/转换器的诊断覆盖率提供通用指南，因为诊断覆盖率很大程度上取决于具体的技术，电路类型以及用例。

注2：需要提供证据以支持声称的诊断覆盖率。

表37 电源

安全机制/措施	参见技术概览	备注
过压和欠压监控	5.2.4.2	通常是模拟电路，其输出锁存在数字内核中。
电压钳位（限制）	5.2.4.3	通常用于抑制电压瞬变或尖峰。
过流监控	5.2.4.4	通常是模拟电路，其输出锁存在数字内核中。
电流限制	5.2.4.5	通常是具有到模拟控制回路的反馈的模拟电路（例如，用于关断调节器主通路要素）。
上电复位	5.2.4.6	在电源轨和/或时钟信号稳定之前，使电路保持在已知的初始状态的功能模块

表38 模拟 I/O

安全机制/措施	参见技术概览	备注
上拉/下拉电阻	5.2.4.1	通常用于输入信号,以避免由于引脚失效或外部引脚互连失效引起的浮空情况。
滤波	5.2.4.8	模拟或数字电路，通常用于抑制高频信号变化，如模拟过压和欠压监控电路的输出。

表39 其他模拟组件

安全机制/措施	参见技术概览	备注
模拟看门狗	5.2.4.7	通常是单稳态电路，用于监控振荡器的正常运行。
热监控	5.2.4.9	通常是模拟电路，其输出锁存在数字内核中，或反馈给模拟电路控制环路（例如，关闭受影响的电路）。
AADC 监控	5.2.4.11	模拟电路，通常由数字电路控制和评估。

模拟BIST	5.2.4.10	通常是由数字电路控制的模拟电路来验证模拟安全机制的正确功能，例如欠压/过压监控，限流保护和热保护电路。

表40　模数转换器

安全机制/措施	参见技术概览	备注
ADC衰减探测	5.2.4.12	通常是由数字电路控制的模拟电路通过测量已知且稳定的信号值来验证ADC转换路径。
ADC通道卡滞探测	5.2.4.13	通常是由数字电路控制的模拟电路通过测量已知且稳定的信号值来验证ADC转换路径。

5.2.4.1　上拉/下拉电阻

目的：定义电路节点的默认电压。

描述：在驱动信号变为断开/高阻抗的情况下，可以通过由电阻器从电路节点连接到电源电压或接地来确定默认电压。 常用于I/O引脚。

示例：当设备/模块的输入引脚处于未驱动或断开状态时，此引脚电平为未知电压。可以通过上拉到 I/O 电源电压（或模块电源电压）的上拉电阻或接地的下拉电阻，将输入保持在已知电压电平。 电路本身可以是无源电阻器或像电流镜一样的有源电路。

5.2.4.2　过压&欠压监控

目的：尽可能早地检测调节电压是否超出规定范围。

描述：调节电压通过一对差分输入与低和/或高模拟参考电压进行比较，此参考电压表示了指定工作范围的极限。 当调节电压超出指示故障的定义电压窗口时，监控器输出将改变状态。

示例：窗口比较器用于监控低压差（LDO）稳压器的输出，其参考电压设置为调节器的最小和最大电压电平。

5.2.4.3　电压钳位（限制器）

目的：防止电路节点的电压超过可以安全承受的最大电压。

描述：电压钳将电路节点的正/负电压限制到系统和/或设备的工艺能力可接受的水平。 电压钳位可以是偏置的或无偏的。 无偏钳位通常使用齐纳二极管来定义参考电压，而偏置钳位使用电压源结合专用二极管（齐纳，肖特基）来定义可接受的电压电平。电压钳位通常用于防止瞬态事件。

示例：ESD 保护电路是通常在 I/O 引脚上应用的专用电压钳位。它的设计能使内部电路远离 I/O 引脚上的高压静电放电能量，以确保内部电路在 ESD 事件期间不会暴露于过高的电压下。

5.2.4.4　过流监控

目的：尽早检测到输出电流超过某个值。

描述：过电流监控的实施可能有所不同。 具有MOS输出器件的稳压器电路的典型方法是添加与调节器主FET并联的感测FET，这个与主FET电流成比例的感测FET电流流过检测电阻，检测电阻两端的电压降由电压监控器放大和监控。

注：过电流监控控器的输出是数字输出，用作对模拟电路控制回路的反馈，并且/或者锁存在与控制/状态监控电路接口的数字核中。

5.2.4.5 限流器

目的：将输出电流限制在最大水平，以保持输出设备的安全工作区域并防止超过最大电气耐受范围。

描述：一种闭环系统，使用来自电流监控器的负反馈来减少对输出设备的驱动，从而限制输出电流。

5.2.4.6 上电复位

目的：保持系统输出处于已知状态（通常为关闭），直到内部节点在上电或电源复位条件下稳定。

描述：通常，将基于带隙的参考电压与衰减的电源电压进行比较，以便检测确保正确的运行的最小指定电源电压。当衰减的电源电压超过参考电压时，通常需要迟滞 设计以防止振荡。

示例：欠压监控器是用于检测和驱动上电复位的机制。

5.2.4.7 模拟看门狗

目的：监控振荡器的正常运行。

描述：通常用单稳态电路（单次触发）实现，该电路在振荡器的每个周期复位。 如果在单稳态电路定义的指定时间段内没有发生振荡器转换，则产生故障信号。

5.2.4.8 滤波器

目的：为避免可能导致失效的瞬态干扰

描述：滤波器可以多种方式用作安全机制。

示例1：旁路电容可用于抑制电压瞬变。 RC 时间常数用于评估可能违反安全目标的故障持续时间是否在最大故障处理时间间隔内。

示例2：数字去毛刺电路可用于滤除模拟电压比较器输出的电平移位。

5.2.4.9 热监控器

目的：探测电路温度超过规定限值的情况。

描述：通常，将PTAT（与绝对温度成比例的）电压与基于带隙的与温度无关的参考电压进行比较。当PTAT电压超过参考电压时，比较器将产生故障信号。

5.2.4.10 模拟内置自检测（模拟 BIST）

目的：验证诊断电路的正确运行，增加对潜伏故障的探测。

描述：模拟BIST的实现按照要验证的诊断功能而变化。 模拟BIST通常通过将电流或电压注入诊断电路来使诊断电路进入和退出故障场景，以确保诊断电路可以正常切换到故障和非故障状态。

5.2.4.11 ADC 监控

目的：通过数字转换测量模拟信号，该数字转换的输出在数字内核中进行处理/评估，作为独立/冗余的模拟信号监控器

描述：对精度要求较高的关键模拟信号通过独立的ADC（例如，位于组件外部或至少由独立源偏置）转换为数字量。 然后由CPU或等效数字机器处理数字代码，以便确定原始模拟信号在精度、静态和动态行为方面是否具有所需的性能。 采样频率和ADC的分辨率以及数字处理定义了可以检测哪些失效模式以及其检测精度。

5.2.4.12 ADC 衰减探测

目的：检测模拟信号到其数字解析的错误转换。

描述：在每个转换循环中，要素在选择和不选择衰减的情况下执行内部$V_{中}$电压的转换。转换结果分别存储在单独的SPI字段中。 通过将衰减结果除以非衰减结果的数学运算来验证衰减因子在指定范围内。

5.2.4.13 ADC 通道卡滞探测

目的：检测影响ADC转换输入信号的卡滞故障

描述：该要素提供带有串联电阻RPOST的多路复用器通道，仅在转换测试电压通道（$V_{高}$、$V_{低}$、$V_{中}$）时被选择，其他时候RPOST被旁路。 选择RPOST的值使得在后缓冲多路选择器内的卡滞通道将一个或多个测试电压通道拉出预期的电压范围。

示例：每个软件循环，MCU 通过 SPI 读取对于 $V_{高}$、$V_{低}$、$V_{中}$组件 ADC 通道的 ADC 转换结果，并将它们和固定的探测阈值进行比较。

5.2.5 在开发阶段避免系统性故障

模拟和混合信号组件是基于标准化开发流程开发的。

有关硬件架构和详细设计的一般要求和建议在GB/T 34590.5-XXXX第7章中定义。

如果出现以下情况，5.1.9中的指南可应用于模拟和混合信号组件：

——表 31 由表 41 替代；及

——如果可行，使用第三方确认的宏块并遵守宏内核提供者定义的每个约束和过程，仅限于硬核。

注：在开发过程中考虑磨损和老化，并进行适当的验证和确认程序。

表41 避免模拟和混合信号组件系统性失效的措施示例

GB/T 34590.5-XXXX 各章条	设计阶段	技术/措施	目的
6.5.1 硬件安全要求规范	规范	使用适当的需求管理工具	简化识别和跟踪硬件要素的安全要求。

6.5.2 硬件/软件接口规范		使用模型描述关键要素的硬件/软件接口	T降低误解的风险并确保硬件和软件设计之间的一致性。
7.5.1 硬件设计规范		使用适当的工具将需求分配给硬件设计	简化识别和跟踪硬件要素的设计规范。
7.4.1.6 模块化硬件设计的特性	设计	使用模块化，层级和简单的设计	电路功能的描述以易于理解的方式构造。 即，无需仿真也可以通过其描述直观地理解电路功能
7.4.1.6 模块化硬件设计的特性		使用原理图的硬件设计	原理图输入是模拟电路中常用的方法。
7.4.4 硬件设计的验证		关键要素的行为模型仿真	行为模型是设计的简化模型。 模拟电路的行为建模允许在早期设计阶段评估功能（例如，以证明设计概念）和减少仿真时间。
7.4.4 硬件设计的验证		晶体管级仿真	当仿真时间可行时，晶体管级仿真是用于验证和确认模拟电路的专用关键功能的方法。
7.4.4 硬件设计的验证		通过设计评审和/或工具完成安全工作区域（SOA）检查	模拟电路由具有不同电流/电压能力的器件组成。SOA检查确保每个器件按照其工艺在其特定的运行区域内安全地工作。
7.4.4 硬件设计的验证		角点仿真（即工艺流程和环境条件扩散）	为了确保模块级的功能，执行考虑了过程参数和环境条件的扩展的仿真。
7.4.4 硬件设计的验证		大多数敏感块的蒙特卡罗仿真	为了确保关键电路的模块级功能，使用统计方法（即蒙特卡罗仿真）仿真片上流程扩展的影响。
7.4.4 硬件设计的验证		关键要素的混合模式仿真	为了确保关键要素的正确性，例如 模拟到数字接口，模拟/数字闭环控制，数字电路在模拟域中仿真。
7.4.4 硬件设计的验证		需求驱动验证	验证所有功能和安全相关要求。 通过规范和验证计划之间的可追溯性显示
7.4.4 硬件设计的验证		可测试性设计	设计和布局中包含特定的硬件结构（例如测试模式，多路复用器），以便测试那些原本无法测试的电路节点并改善测试覆盖率
7.4.2.4 鲁棒性设计原则		应用原理图设计指南	人工检查

7.4.4 硬件设计的验证		原理图检查器的应用	例如，在互连上或根据功能选择适当的设备时执行自动检查。 例如SOA（安全操作区域）检查器
7.4.4 硬件设计的验证		仿真结果的文档化	成功仿真所需的每个数据的归档，以验证指定的电路功能
7.4.4 硬件设计验证		原理图设计检查或走查	设计评审通常包括检查或走查。
7.4.4 硬件设计的验证		硬核的应用和确认（重用的原理图设计和/或布局）	使用已经过验证的原理图或布局。
7.4.4 硬件设计的验证		针对晶体管级描述验证行为模型（如果使用）	通过仿真对行为模型和晶体管级原理图设计进行交叉检查
7.4.4 硬件设计的验证		从关键要素布局中提取寄生参数的网表仿真	由模拟仿真器仿真的后批注网表
7.4.4 硬件设计的验证	设计	对关键要素，对照原理图网表来验证从布局中提取寄生参数的网表	根据仿真结果检查后批注的网表，以便考虑寄生布局效应。
7.4.4 硬件设计的验证		布局检查或走查（避免噪声和敏感网络之间的串扰;避免信号路径具有最小宽度;使用多个触点/过孔连接多个层）	模拟电路的布局主要是手动完成的（相对于模拟模块，自动化非常有限），因此布局检查至关重要。 设计评审通常包括布局检查或走查。
7.4.4 硬件设计的验证		设计规则检查（DRC）	模拟电路的布局主要是手动完成的（相对于模拟模块，自动化非常有限），因此设计规则检查比数字域更重要。
7.4.4 硬件设计验证		布局与原理图对比检查（LVS）	模拟电路的布局通常是手动完成的（与模拟模块相比，自动化非常有限），因此检查布局与原理图比在数字域中更为重要。
7.4.4 硬件设计验证	硬件设计验证	通过硬件原型开发	通过原型（例如测试芯片，电路板）验证已实现的功能，可以检查设计评审不充分的硬件设计的特定点。
6.5.3 硬件安全要求验证报告	验证	硬件安全要求验证报告	提供与硬件规范，完整性和正确性一致的证据

10.5.1 硬件集成验证活动	硬件集成验证	验证组件层面设计实现的完整性和正确性。	执行组件测试，生成报告
7.4.5 生产，运行，服务和报废 9.4.1.2， 9.4.1.3 专用措施	芯片生产中安全相关特殊特性	确定生产测试的可实现测试覆盖率	在生产测试期间评估组件的安全相关方面的测试覆盖率。
7.4.5 生产、运行、服务和报废 9.4.1.2， 9.4.1.3 专用措施		确定检测和剔除早期故障的措施	确保制造组件的鲁棒性。 在大多数过程中，栅极氧化物完整性（GOI）检查是关键的早期寿命失效机理。 有多种方法可以筛选早期GOI失效，包括高温/高压运行（Burn-In），高电流运行和电压应力，但是如果GOI不是过程中早期寿命失效的主要原因，这些方法可能没有任何作用。
7.4.5 生产、运行、服务和报废 10 硬件集成和验证	硬件要素的评估	定义和执行质量测试，如欠压测试，高温工作寿命（HTOL）测试和功能测试用例 与生产、运行、服务和报废有关的要求的规范 硬件集成和验证报告	对于具有集成掉电探测的模拟组件，探测组件功能以验证模拟电路的输出被设置为定义的状态（例如，通过在复位状态下停止模拟电路的运行）或者当由掉电探测的任何电源电压达到正常运行区间所定义的低边界时，以另一种方式（例如通过发出安全状态信号）发出欠压状态信号。 对于没有集成掉电探测的模拟组件，测试模拟功能以验证模拟电路是否在电源电压从标称值下降到零时将其输出设置为定义状态（例如，通过在复位状态下停止模拟电路的运行）。 否则，定义使用假设并考虑外部措施。

5.2.6 模拟/混合信号组件的安全文档示例

由于其功能的特定性质，模拟和混合信号组件主要在分布式开发中开发。

5.1.11中针对数字组件公布的指南可用作要交换的安全工作成果的参考，但是，需要适应不同的开发方法。

——组件制造商和终端用户之间的 DIA 指定了各方可以提供哪些文件以及各方之间的工作分担；及

——安全要求规范定义了组件的预期功能。 至关重要的是，最终用户应按照 GB/T 34590.8-XXXX 第 6 章仔细编制此类规范，以确保分布式开发中的每个供应商都能理解正确的功能。 关于组件要素的使用以及预定义的片上/片外安全机制的识别描述对于在系统或要素层面进行适当的安全分析是重要的（例如，针对每个安全目标，允许将故障分类为安全故障、潜在违反安全目标的故障等）。

注1：如果组件是脱离背景开发的，那么源自技术安全概念的要求要被用途假设所取代。

下面列出了描述模拟和混合信号组件能力的文档：

——根据 GB/T 34590 的适用要求进行检查的结果，包括认可措施报告（如果适用）；

——达成共识的安全分析结果；

注2：这些可能是组件的原始失效，对于不同的安全要求，它们的分配和诊断覆盖率由指定的安全机制或完整的FMEA提供。

——有关失效率计算的信息（例如晶体管数量）；及

——关于组件在其预期用途方面的任何使用假设的描述。

注3：这可以合并到模拟或混合信号组件的“安全手册”或“安全应用说明”中。

5.3 可编程逻辑器件

5.3.1 关于可编程逻辑器件

5.3.1.1 总则

如图25所示，可编程逻辑器件PLD可看做可配置I/O、非固定功能（它由逻辑块和用户存储器组成，通过相关配置技术对其配置）、链接逻辑块的信号路由功能和固定逻辑功能的组合。

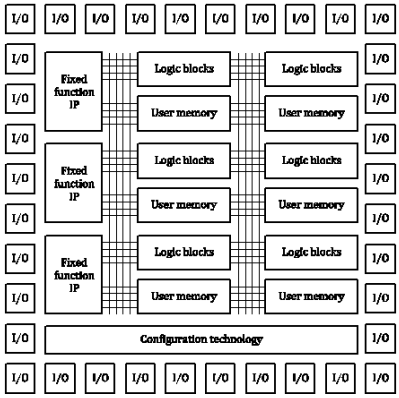


图25 PLD 通用框图

非固定逻辑功能可包括，但不限于：简单逻辑门、多路复用器、反相器、触发器和存储器，以实现更复杂的功能，如数字信号处理功能。信号路由能力，可以实现从简单的点对点的方案，到具有灵活路由可能性和时钟选项的复杂的总线互连。PLD器件在用户存储的实现上可能有所不同。有些器件提供的有限的内存容量，而另一些器件提供的本地或全局存储结构，可用于更广泛的不同的应用场合。更复杂的器件还可以实现一些固定功能，如CPU、存储控制器、加密模块和其他功能，从而释放设计资源给用户配置。时钟、电源和复位电路是固定功能。如果要实现单个或多个实例，由PLD设计的决定。

可编程逻辑PLD的一个共同特点是，用户可以按照特定应用需求，来配置它们得到（相应的）功能。器件的设计或配置可以使用不同的工具来完成，从非常简单的，到完整开发包，如时序分析和设计优化，以支持复杂功能。一旦用户设计完成，就可以将其编程到器件中。器件一次性编程，亦或多次重新编程，由不同技术来支持。这些方法可以通过提供易失性或非易失性功能来进一步区分。这在框图中由标记为“配置技术”的块表示。

注：闪存（可重复编程）或反熔丝（可编程）等非易失性技术的安全相关功能可与静态随机访问存储器SRAM等易失性技术不同。

5.3.1.2 关于可编程逻辑器件 PLD 类型

表42 给出了常用PLD类型的非详尽列表。

表42 常用 PLD 类型

类型	描述
可编程阵列逻辑（PAL）	一次性可编程器件对每个输出实现积之和逻辑。
门阵列逻辑（GAL）	与PAL类似的功能，具有可多次编程的特点。
复杂可编程逻辑器件（CPLD）	具有与PAL类似功能的非易失性器件，有更高集成率，更复杂反馈路径。
现场可编程门阵列（FPGA）	主要是对非常复杂的逻辑，路由和存储功能易失的实现。

5.3.1.3 可编程逻辑器件 PLD 的功能安全生命周期裁剪

5.3.1.3.1 概述

图26 描述了使用GB/T 34590.10的相同方法，如何裁剪可编程逻辑器件PLD的功能安全生命周期。

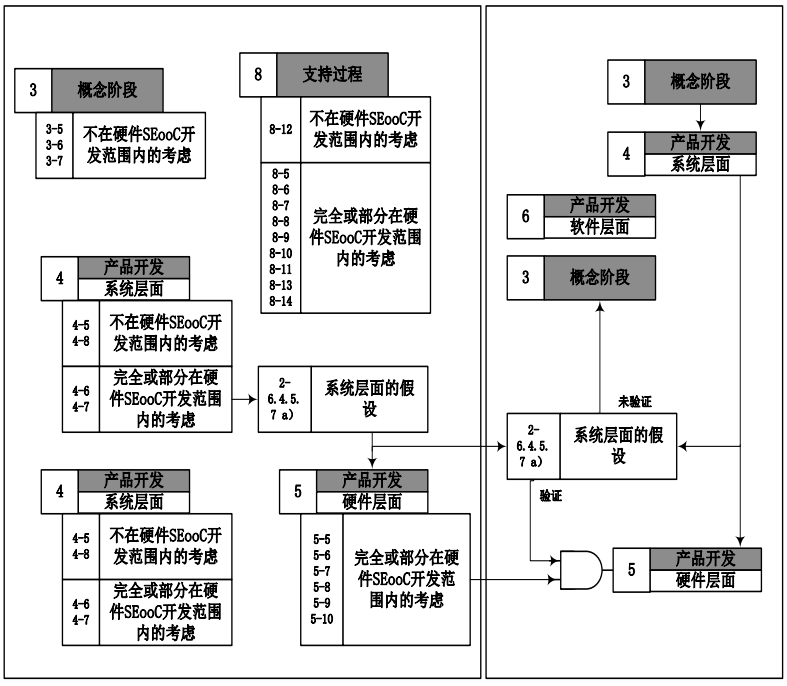


图26 SEooC 可编程逻辑器件 PLD 硬件开发

注1：图26中所示的参考文献与GB/T 34590有关。

注2：在本部分的上下文中，PLD制造商指的是开发PLD并负责PLD制造的组织。 PLD用户是指为PLD开发程序或将其应用到实际应用的组织。

注3：本部分4.5章节中考虑了PLD的IP模块供应商。

注4：虽然图26中未显示GB/T 34590的每个章节，但这并不意味着它们不适用。

针对PLD制造商或PLD用户，以下章节给出了关于GB/T 34590某些特定部分的示例。

5.3.1.3.2 GB/T 34590.2 （功能安全管理）

通常，GB/T 34590.2 匹配到合适的水平，对于PLD制造商和PLD用户来说是适用的。

示例1：GB/T 34590.2-XXXX,6.4.2.1 要求在相关项开发启动时，指定项目经理。 对于 PLD 制造商而言，这意味着在 PLD 开发的起始阶段，也应该任命项目经理。

示例2：按照 GB/T 34590.2-XXXX,6.4.6.5，安全计划包括相关项层面计划，如 GB/T 34590.3 [64]，第 6 章中给出的危害分析和风险评估的计划。由于危害分析和风险评估仅在相关项层面进行，所以此要求对于可编程逻辑器件 PLD 级别的安全计划并不可行。

示例3：GB/T 34590.2-XXXX,6.4.11 要求对相关项进行功能安全审核。 由于 PLD 制造商无法在相关项层面进行安全审核，因此只在 PLD 器件层面上进行处理。

示例4：GB/T 34590.2-XXXX,7.4.2.1 要求（第三方）组织指定责任人并给与相应授权，如 GB/T 34590.2-XXXX,5.4.2.7 中所述，以维持相关项在投入生产后的功能安全。对于 PLD 制造商而言，这意味着指定人员在 PLD 投入生产后将维持 PLD 的功能安全，而不是负责保证整个相关项的功能安全。

5.3.1.3.3 GB/T 34590.3 （概念阶段）

根据GB/T 34590.3，PLD制造商在概念阶段通常不必承担责任，除非PLD制造商也是相关项集成商。如果PLD用户负责相关项层面，则此部分适用。

5.3.1.3.4 GB/T 34590.4 （产品开发：系统层面）

PLD可以按SEooC概念开发。 对于SEooC的开发，GB/T 34590.4-XXXX，第6章和GB/T 34590.4-XXXX，第7章，在部分或全部范畴内。SEooC的开发指南可以在GB/T 34590.10[61]中找到。

示例：为支持技术安全概念，PLD 制造商可在 PLD 器件上实现专门的硬件安全措施。 其他措施（的实现）取决于用户所实现的电路，而且需要特定措施（例如，逻辑冗余，外部看门狗），并且由用户自己责任。 PLD 制造商对系统层面措施所做的假设（条件），由 PLD 用户文档化并完成验证。

如果PLD用户也是相关项集成商，则GB/T 34590.4 （的要求）完全在该范畴内。

5.3.1.3.5 GB/T 34590.5 （产品开发：硬件层面）

GB/T 34590.5 所有章节，包括GB/T 34590.5-XXXX，第8章和GB/T 34590.5-XXXX，第9章，都适用于PLD制造商和PLD用户，按照他们对整体安全概念的贡献程度。

示例：如果 PLD 不包含任何硬件安全机制，PLD 制造商的主要作用是给出基本失效率、失效模式和失效模式分布（使用，例如本部分 4.6 中描述的方法）。可以提供硬件架构度量的参考或计算示例，而在 PLD 中实现的特定设计，度量由 PLD 用户计算。

关于GB/T 34590.5-XXXX，第8章和GB/T 34590.5-XXXX，第9章，通常PLD制造商有责任提供PLD用户所需的信息、方法和/或工具， 使其能够计算和验证度量， 这些内容包含：

——失效模式的分布；及

——PLD 中嵌入的安全机制的诊断覆盖率的值（见 5.3）。

关于GB/T 34590.5-XXXX，第10章，对于半导体组件，设想这部分不仅与集成测试有关，而且它也适用于PLD制造商和PLD用户的测试活动有关，但要按照他们各自在整体安全概念中的贡献程度。5.3.4给出了关于诊断覆盖率进一步的信息。

5.3.1.3.6 GB/T 34590.6 （产品开发：软件层面）

基于GB/T 34590.4-XXXX, 6.4.6.5，GB/T 34590.5 和GB/T 34590.6的要求可以合并起来，用于可编程逻辑，如PLD。

在高级综合流情况下，使用如OpenCL，C-to-HDL流或基于模型等方法开发，与GB/T 34590.6-XXXX的要求的对照，都属于高级语言代码相关的开发。 GB/T 34590.5-XXXX 与用于传统PLD开发的后续步骤有关。

如果PLD用户和PLD制造商的开发流程都是基于HDL语言，这点与微控制器的开发流程类似，因此采用 GB/T 34590.5-XXXX，这种情况下GB/T 34590.6-XXXX则无关。

注：用户PLD电路开发所用的特定技术和措施在5.3.5.3中讨论。 许多方法，与GB/T 34590.6-XXXX中所规定的有相似之处，例如， 遵守编码指南。

5.3.1.3.7 GB/T 34590.7 （生产和运行）

一般来说，GB/T 34590.7 调整到适合的层级，对于PLD制造商是可行的。 当涉及到相关项硬件要素或相关项本身的生产时，对于PLD用户同样可行。

示例1：在 GB/T 34590.7-XXXX，5.4.1.1 中，要求通过对相关项进行评估，来制定生产流程计划。 对应 PLD 制造商的情况，完成该计划，只需要评估 PLD 而不是相关项。

示例2：GB/T 34590.7-XXXX,5.4.1.4 要求识别合理可预见的流程失效，及其对功能安全的影响，并采取适当措施解决这些问题。这点可以不加修改的应用于 PLD 生产。

示例3：GB/T 34590.7-XXXX,5.4.3.5 报废说明的要求通常不适用于 PLD

示例4：为符合 GB/T 34590.7-XXXX,7.4.1.1（的要求），PLD 制造商应该实现 PLD 的现场监控流程。

5.3.2 PLD 失效模式

根据5.3.1.3中所示的生命周期，表43总结了PLD用户可能（需要）关注的失效模式。 PLD的失效模式可以使用4.3.2中提到的关键词得出。

注：该列表未声明详尽（所有失效模式），也可根据其他已知失效模式进行调整。

表43 PLD 失效模式示例

要素（见图25）	描述	已分析的失效模式
固定功能 IP[a]		见表30。
PLD 数字I/O		见GB/T 34590.5-XXXX，表D.1，“数字I/O”要素和表30。
逻辑块[d]		逻辑块所实现功能的永久性损坏。 逻辑块所实现功能的瞬态损坏[b]。

配置技术	见 5.3.1.1	逻辑块配置的非预期永久性改变。 逻辑块配置的非预期瞬时改变[c]。
PLD 模拟I/O		见GB/T 34590.5-XXXX，表D.1，“模拟I/O”要素和表36。
用户存储器		见5.1.3。
信号路由能力[e]		由一组逻辑块所实现的功能的永久性损坏，包括功能的时间延迟。 由一组逻辑块所实现的功能的瞬态损坏。

[a] 如5.3.1中所述，固定功能IP是指与微控制器内相类似的要素的组合。它们通常按照非固定功能在分离区域中实现，因此，它们可以在各个方面被考虑，类似于GB/T 34590.5-XXXX，表D.1和5.1.2和5.1.3中讨论的数字组件的要素。

[b] 该失效模式的相关性取决于PLD的技术类型和逻辑块的类型，见5.3.1.2。

[c] 该失效模式的相关性取决于PLD的技术类型，见5.3.1.2。

[d] I/O 配置逻辑可以在固定功能IP内部，或在I/O本身内。

[e] “信号路由能力”需要考虑（信号）走线和路由配置的技术。

5.3.3 PLD 安全分析说明

5.3.3.1 PLD 的定量分析

5.1中讨论的类似方法也可用于PLD。 根据PLD用户可用的信息，PLD的定量分析包括用户设计，可在不同的抽象层级上执行。

关于PLD用法和用户设计的信息，在设计开发阶段被细化，并基于最新信息的相关分析（工作）也会重复进行。 通过5.3.3.2中描述的相关失效分析，使PLD设计的定量分析得到补充（增强）。

以下两条描述了PLD裸片失效率计算的示例，以及失效率分布到确定的失效模式示例。

硬件架构度量，通过本部分附录C中给出的例子类似的方法来确定。 分析的详细程度取决于目标ASIL等级和应用的需要。

5.3.3.1.1 使用 4.6.2.1.1 中的模型来计算 PLD 裸片失效率的示例

失效率可按4.6所述进行估算。

为了估算PLD裸片的失效率，以下因素需要考虑：

——与配置技术相关的失效率。根据行业来源，配置技术对应的晶体管的处理是不同的，比如，配置技术被看做计算的独立实体，或在逻辑块中，在用户存储器实体中以及在其他相关要素中的配置技术。

——未用资源的失效率。 两种可能都适用。 一种方法是将未用资源视为与安全无关。这取决于PLD 结构，相关失效分析可以分析未用逻辑对用户设计的影响。另一种方法是将未用的逻辑视为安全相关的，并估计可能导致安全失效（根据 GB/T 34590.10 [61]，Fsafe）的相应故障部分。通过 PLD 制造商提供的信息支持的定量分析，可以（帮助）完成这种估算。

注1：如果使用PLD制造商提供的失效率，则用于所提供数据的修正因子都应是可用的。

注2：本条补充了4.6.2.1.1.1中的例子。 由于假设（条件）类似，所以不再重复每条注释。 该例子采用具有表44中所述特性的PLD。

表44 PLD 资源概述

要素	资源	IEC 62380 假设分类
逻辑块	1000	CLDD（EPLD、MAX、FLEX、FPGA等）
用户存储器	16kb	低功耗静态随机存储器
固定功能IP	20 k gates	数字电路、微控制器、DSP
配置技术	10 kb	低功耗静态随机存储器
注：对于逻辑块，图10采用的CPLD实体作为例子。 对于现代易失性FPGA器件，逻辑单元阵列LCA（基于RAM）实体可能更为合适。		

如表45所示可以计算出完整的PLD失效率。表45中的失效率可用于实际用户设计的失效率计算。 表46给出了用户设计采用的假设（条件）。

表45 PLD 失效率计算示例

要素	λ_1	N	α	λ_2		基础FIT	温度修正	有效FIT
逻辑块	2.0×10^{-5}	100 000 （每个宏胞（单元）100个晶体管）	10	34		34.0604	0.17	5.7903
用户存储器	1.7×10^{-7}	98 304 （6个晶体管/位用于低功耗SRAM）	10	8.8		8.8005	0.17	1.4961
固定功能IP	3.4×10^{-6}	80 000 （4个晶体管/栅极）	10	1.7		1.7082	0.17	0.2904
配置技术（基于静态随机存储器）	1.7×10^{-7}	61 440 （6个晶体管/位用于低功耗SRAM）	10	8.8		8.8003	0.17	1.4961
总和						53.3694		9.072 9
注1：假设每个宏胞的晶体管数量（100，如图10所导出）不包括与配置技术相关的晶体管。 因此，配置技术被看作								

计算（元器件）单独实体。 另一种方法可以是，采用晶体管的数量将配置技术包含在逻辑块内，用户存储器实体和其他相关元件中，一起用到（计算）晶体管数量。

注2：该表也可用于，将产生的有效FIT除以要素数，得出单一的FIT。

示例：FIT/逻辑块，可以这样计算 5.7903/1 000 = 0.005 7。

注3：如4.6所示，温度修正系数可能有其他方法。这些备选方法也用于PLD。

表46　用户设计资源使用和失效率计算示例

要素	资源使用	有效FIT
逻辑块	23 %	1.3318
用户存储	10 %	0.1496
固定功能IP	100 %	0.2904
配置技术（基于SRAM）	15 %	0.224 4
总计		1.9962

如果有更多用户设计的细节，则数据可以进一步细化。例如，逻辑块有不同的配置选项，而用户设计可能只用某一种配置。 这样可以进一步降低算出的失效率。

注3：注3：相关失效分析可用于分析不同配置选项对用户设计的影响。

注4：注4：通过合适的设计工具可以帮助修正系数的推导。

5.3.3.1.2　PLD 瞬态失效率计算示例

PLD瞬态失效率的计算可遵循4.6。

注：如果PLD制造商提供的瞬态失效率包括修正系数（例如，基于PLD平均利用率或基于运行工况），则该系数需要向PLD用户给予解释。

表46 用前一章同样的计算瞬态故障失效率的方法，也可用于特定用户设计的失效率计算。

5.3.3.1.3　PLD 失效率对应失效模式的分布的示例

一旦PLD失效率被估算出来，它将被分配到确定的失效模式，即算出失效模式分布。

对于PLD制造商，可以按照5.1中的描述计算失效模式分布。

对PLD用户来说，以下是识别失效模式和确定失效模式分布的可能方法的示例：

a) 在 PLD 用户端设计时，在功能块级别上识别失效模式；假设对应已确定的失效模式，PLD 失效率均等分布；

b) 在 PLD 用户端设计时，在功能块级别上识别失效模式；根据专家判断，对应已确定的失效模式，估计 PLD 失效率的分布，与估计所耗资源（如固定功能 IP、逻辑块数量、用户存储等）一并考虑，文档化证据支持；及

c) 通过基础子元器件上所实现的 PLD 用户设计进行分区的方法，可以确定失效模式；通过 PLD 制造商提供的详细资源利用信息，利用这些信息可以确定基于所实现的 PLD 用户设计的失效模式，进而估算出失效率的分布。这些工作可以通过适当的设计工具来支持。

注1：在PLD制造商应用场景下，基础子元器件可被视为一组触发器和相关扇入门。同样，在PLD用户应用场景下，基础子元器件可以作为逻辑单元组，由逻辑块中的触发器和逻辑块表示的组合逻辑构成。详细程度，即考虑的基础子元器件的数量取决于所用安全机制的类型和应用。

注2：得出的定量数据的准确度水平因所用方法而异。

示例1：如果有关于用户 PLD 设计的信息，那么方法 c）可以提供最高的准确度。如果此信息不可用，并且没有论据说明为什么某一种故障模式比另一种更可能出现，那么可以用方法 a）。

注3：所需的失效模式分布精度水平还取决于所用安全机制的类型和应用。

示例2：在 PLD 的用户端锁步设计案例中，方法 a）可能是足够的，因为失效模式分布的非均匀分布值不会影响声称的诊断覆盖率。对于依赖软件测试库周期性测试 PLD 硬件的用户 PLD 设计，如果存在论据，证明其中一种失效模式比其他概率更高，则按照所需的准确度级别使用方法 b)或 c)。

注4：详细的失效模式定义（如方法c）提供的定义）有助于提供诊断覆盖率的理由。

注5：对于瞬态故障，资源利用率可以考虑逻辑块中包含的触发器数量、PLD用户端设计的用户存储器位数量和PLD用户端设计使用的配置位数量。

表47显示了基于附件E的上述三种方法的示例。它考虑在PLD中实现的SPI模块。

表47　PLD 用户级的 PLD 失效模式分布计算方法示例

失效模式	所包括的子元器件	a)	b) 见注1	c) 见注2
错误或没有时钟	时钟生成	25 %	10/110 = 9.09 %	10/90 = 11.11 %
错误或无数据接收	外设总线接口 输入移位寄存器 数据接收寄存器 I/O　端口	25 %	40/110 = 36.36 %	30/90 = 33.33 %
错误或无数据发送	外设总线接口 输出移位寄存器 数据发送寄存器 I/O　端口	25 %	40/110 = 36.36 %	30/90 = 33.33 %
SPI错误配置	配置寄存器 外设总线接口	25 %	20/110 = 18.18 %	20/90 = 22.22 %

注1：对于本示例，假设每个子元器件用到10个逻辑块，因此，子元器件失效模式的失效模式的分布，就是子元器件所用逻辑块之和按比例估算划分。

注2：b）和c）之间的区别在于，特定失效模式的资源使用不做估算，而是计算导致失效模式的实际资源数量。如果导致失效模式的逻辑块跨越不同的子元器件，则可以在子元器件层面上进行，也向下到基础子元器件层面上进行。在本示例中，对输入移位寄存器、输出移位寄存器、数据接收寄存器和数据发送寄存器这样计算：对各自的失效模式贡献率为100%，

对其他失效模式贡献率为0%；外设总线接口对数据相关失效模式贡献率50%，对配置（参数）失效模式贡献率为100%；I/O引脚对与数据相关的失效模式贡献率按50%计算。

5.3.3.1.4 验证硬件实现的安全机制的完整性和正确性

如4.8所述，开发阶段的故障注入仿真是一种有效的方法，用于验证对应硬件安全要求的安全机制实现的完整性和正确性，以及辅助验证安全故障，以及故障数量和失效模式覆盖率的计算，如5.1.10所述。这些同样适用于PLD制造商。

对于PLD用户来说，如果需要进行故障注入，而关于PLD用户设计与PLD逻辑块映射关系的详细信息还没有，则可以在映射之前，对逻辑设计执行故障注入。

示例：通过软件测试库，对用户 PLD 设计进行周期测试，对于这些测试是否可以达到所宣称的诊断覆盖率，如果需要进行故障注入作为依据，那么故障注入也可以在不同的层级执行。例如，从 RTL 设计开始，来描述用户 PLD 设计，然后对其进行整合，得到参考网表，并执行故障注入。如果参考网表与 PLD 设计不符合，则提供一定理由解释，为什么注入的故障对于 PLD 设计的假定实现是有必要的。

5.3.3.2 PLD 相关失效分析

对于集成电路来说，考虑相关失效是很重要的，尤其是在硬件安全机制或冗余要求在同一组件中实现的情况下。

注：本条所考虑的相关失效分析流程与4.7的特性要求是等同的。对于PLD制造商和PLD用户，如果有，表48 描述的特性可以和 4.7中定义的步骤一起考虑。

表48 4.7 PLD 制造商和 PLD 用户的相关失效分析特性

步骤（见图 23）	PLD 制造商	PLD 用户
B1：确认硬件和软件要素。	定义见4.7。	定义见4.7。
B2：确认相关失效引发源。	分析还要考虑可配置逻辑和固定逻辑之间的相互作用，包括复位或配置技术相关的作用。[a]	分析还要考虑失效对配置技术的影响，因此可能同时影响多个逻辑块。
B6：确定必要的安全措施，以控制或减缓相关失效引发源。	分析还要考虑在可配置逻辑和固定逻辑之间提供隔离的可能性。	分析还要考虑在逻辑块之间提供隔离的可能性
B10：评估控制或避免相关失效的（措施的）有效性。	定义见4.7。	定义见4.7。
[a] 例如，固定逻辑中的故障导致可配置逻辑的配置丢失。		

表49和表50中列出的相关失效引发源，与4.7中的说明等效。其他考虑相关失效引发源或其应对措施，都适用于PLD制造商和PLD用户。

表49 对照 4.7、PLD 制造商和 PLD 用户的相关失效引发源特性

相关失效引发源	PLD制造商的相关失效引发源	PLD用户的相关失效引发源
共享资源失效[a]	定义见4.7	可用时钟网络的潜在依赖性 配置技术失效（例如，共享短距或长距公共互连） 共享可编程I/O失效 由于外部配置存储器或相关互连失效，导致PLD配置的错误
单个物理性根本原因	定义见4.7	导致PLD配置完全或部分丢失的故障（例如，在复位逻辑中）
开发故障	固定逻辑和可配置逻辑之间的距离或隔离不充分	错误使用PLD制造商提供的工具[b]。同时见4.7
生产故障	定义见4.7	错误使用配置编程工具[b]
安装故障	定义见4.7	定义见4.7
服务故障	定义见4.7	错误使用在线重新配置功能
[a] 在 PLD 上下文中，“共同”不仅仅是可配置逻辑、固定逻辑内部共享资源，而且包括在可配置逻辑固定逻辑间共享资源。 [b] 例如， 用户错误运用 隔离/分离限制条件。		

表50　PLD 制造商和 PLD 用户的相关失效引发源对策

相关失效引发源	PLD制造商的对策	PLD用户的对策
共享资源失效[a]	定义见4.7	时钟网络的相关性分析和专门时钟监控 配置技术失效分析和后续采用分离/隔离技术 共享可编程I/O失效分析和后续采用I/O安全协议 运行时PLD配置完整性检查（例如，通过CRC检查）
单个物理性根本原因	定义见4.7	复位网络相关性分析和专门看门狗
开发故障	固定逻辑和可配置逻辑之间正确的隔离或分离	定义见4.7
生产故障	定义见4.7	（使用）PLD 工具手册防止相关性失效的正确指导
安装故障	定义见4.7	定义见4.7

服务故障	定义见4.7	限制使用在线重新配置功能
[a] 在 PLD 上下文中，“共同”不仅仅是可配置逻辑、固定逻辑内部共享资源，而且包括在可配置逻辑固定逻辑间共享资源。		

5.3.4 PLD 安全机制示例

表51 列出的安全机制示例，可用于解决表 43中所述PLD失效模式。

注：本表并非详尽，可使用其他技术，前提是有证据支持所声明的诊断覆盖率。

表51 PLD 安全机制和 GB/T 34590.5-XXXX，附录 D 对照

要素	安全机制示例
固定功能IP	表34
时钟	GB/T 34590.5-XXXX，表D.8 片内时钟状态指示[a]
供电电源	GB/T 34590.5-XXXX，表D.7 分离供电（电压）层[b]
数字输入输出 I/O	GB/T 34590.5-XXXX，表D.5
模拟输入输出 I/O	GB/T 34590.5-XXXX，表D,5
逻辑块	GB/T 34590.5-XXXX，表D.4 表34 通过重新配置，实现时间、空间冗余的融合
片外通讯	GB/T 34590.5-XXXX，表D.6
配置技术	表32，表33 回读通过下载设备[c]下载的内容[c]
用户存储器	表32，表33
信号路由能力	表35

[a] 许多 PLD 提供时钟生成和管理的资源，还提供时钟功能的监控，并通过相关状态管脚/寄存器，以指示某个时钟还在正常工作（例如，通过主时钟输入，判断时钟输出是否处于正确的状态）。

[b] 电压层指电隔离的电压供应平面，可连接到外部电源电压。

[c] 是指许多可编程器件是否能够检查其配置寄存器内容，并与预期的（特定设计相关）配置内容进行比较。如果检测到不匹配，这些功能可以改变输出管脚的状态，或产生中断，以便系统能够做出适当响应。为了提高在线监控安全机制的可用性，有效的回读测试可对器件内安全相关和非安全相关的元器件划分优先顺序。安全相关元器件可以更高频次进行检查，从而大大缩短失效探测时间。

5.3.5 PLD 系统故障避免

5.3.5.1 避免 PLD 实现中的系统性故障

由于与其他数字组件制造商使用的流程相比，PLD制造商采用的规范、设计和验证流程没有显著差异，因此可以采用5.1.9（及表31有关）中给出的相同建议。

5.3.5.2 关于 PLD 支持工具

PLD 相关工具可分为两类：

——生产前使用的工具（如 PLD 制造商使用的工具）；及

——PLD 用户使用的工具。

按照GB/T 34590.8-XXXX第11章的要求，分析这两类工具使用的置信度。

示例1：根据 GB/T 34590.8-XXXX 11 章，因为工具故障可导致被开发的安全相关元素出现错误，所以 PLD 制造商用于布置和布线的工具可看作为 TI2；如果证据显示，采用了设计规则检查（DRC）和使用了合适规则集进行布局与原理图比对（LVS）检查，如最新的集成电路设计流程中所预见的，能够以高度的置信度检测由工具引入的可能错误，可声称 TD1。在这种情况下，根据 GB/T 34590.8-XXXX，表 3，可以将其视为 TCL1。

示例2：根据 GB/T 34590.8-XXXX 11 章，因为工具故障可导致被开发的安全相关要素出现错误，所以 PLD 制造商用于布置和布线的工具可看作为 TI2；。如果后续的硬件和集成测试，错误检测可以达到中等置信度，考虑电路的复杂性，则可以将其视为 TD2。因此，根据 GB/T 34590.8-XXXX 表 3，可以将其看作 TCL2。如果相应的相关项 ASIL 等级是，例如，ASIL B 那么工具供应商可以通过“结合（实际）使用提高置信度”和“工具开发流程评估”两种方法的适当组合对软件工具合格评判。

5.3.5.3 PLD 用户系统故障避免

就像微控制器一样，对于PLD制造商，PLD是基于标准化开发过程开发的，如采用5.1.9中的例子。

以下两种方法是通过示例，说明如何提供证据的证明，PLD用户在开发过程中，使用合适的流程，避免系统性失效，所采取的措施是充分的：

——使用检查表（见表 52）；及

——使用与目标器件采用相同工艺开发的类似产品的现场数据（例如使用 GB/T 34590.8-XXXX 第 14 章）。

表52 PLD 用户避免系统性失效的措施示例

GB/T 34590.5-XXXX 要求	设计阶段	技术/措施	目的
7.4.1.6 模块化设计特性	设计入口	结构化描述与模块化	PLD 的功能描述构成，按易于阅读的方

			式组织，如电路功能可以在描述的基础上直观地理解，无需仿真工作。
7.4.1.6 模块化设计特性		HDL 中的设计描述	以硬件描述语言，如 VHDL 或 Verilog，在上层对功能进行描述
7.4.2.4 鲁棒性设计原则		遵守编码准则	严格遵守编码风格，带来的好处是句法和语义正确的电路代码
7.4.2.4 鲁棒性设计原则	设计入口	限制使用异步构造	避免综合过程中的典型时间异常，避免仿真和综合过程中因建模或设计的可测试性不足而产生的模糊性。 这并不排除对于某些类型的 PLD 实现，异步逻辑还是可用的；在这种情况下，目的是建议对这些电路的处理和验证（需要）格外注意。 异步复位的时间存在风险，因为大量可能的附加要素，（造成）传导时间不同。由于异步复位信号与附加同步要素的时钟不相关，因此在复位释放掉时，（复位信号）亚稳态可能是个问题。是否造成问题，可预料取决于设计和环境因素，如温度和重置网络的输出。
7.4.2.4 鲁棒性设计原则		初级输入的同步和亚稳态控制	避免因（信号）建立和保持时间违反，而导致电路动作不确定
7.4.4 硬件设计验证		HDL　仿真	通过仿真方法，对 VHDL 或 Verilog 中描述的电路进行流片前验证
7.4.4 硬件设计验证		模块级功能测试（例如，使用 HDL 测试台）	“自下而上”流片前验证
7.4.4 硬件设计验证		顶层功能测试	PLD 验证（全功能）
7.4.4 硬件设计验证		功能和结构覆盖率驱动的验证（验证目标的覆盖率以百分比表示）	功能测试期间，针对所用的验证场景进行定量评估。覆盖率的目标等级被定义和证明。
7.4.4 硬件设计验证		使用代码检查程序	使用代码检查工具自动检验编码规则（“编码样式”）。

7.4.4 硬件设计验证		仿真结果归档	为验证特定电路功能，成功仿真所用的每个数据都要归档。
7.4.4 硬件设计验证		软 IP 集成与验证	见 4.5。
7.4.4 硬件设计验证	综合，映射，层规划，布置和布线	检查 PLD 供应商的要求和约束	在 PLD 设计期阶段，PLD 供应商定义的要求和约束就被考虑了。
7.4.4 硬件设计验证		PLD 支持工具输出（物）分析	对 PLD 支持工具的输出物进行分析。为消除（输出物）警告和错误，提供理由（解释）。
7.4.1.6 模块化设计特性		约束、结果和工具的归档	PLD 设计的最佳综合、映射、放置和布线，所需的定义的每个约束条件需要归档
7.4.1.6 模块化设计特性		基于脚本的过程	结果的可重现和综合、映射、放置和布线的自动化
7.4.4 硬件设计验证		最终网表的仿真和时间验证	在综合、映射、放置和布线之后对网表进行独立验证-包括时间验证
7.4.4 硬件设计验证		最终网表与参考模型的比对（形式等价性检查）	使用 RTL 对最终网表进行功能等效性检查。
7.4.2.4 鲁棒性设计原则		使用不足三年的工艺技术（需要保证）充足时间余量	即使在强（压力）过程和参数波动下，也能保证所实现电路功能的鲁棒性。在库文件或由 PLD 用户（实现的），考虑在时间分析中保证时间余量。
7.4.4 硬件设计验证		设计规则检查（DRC）	按平面规划逻辑执行设计规则检查
9.4.1.2、9.4.13 专用措施 10 硬件集成与验证	PLD 集成和测试	PLD 验证	验证 PLD 原型，包括 PLD 正确配置验证（例如：使用校验和）。
7.4.5 生产、运行、服务和报废 9.4.1.2、9.4.1.3 专用措施 10 硬件集成与验证		PLD 集成	PLD 在系统中的验证和集成

5.3.6 PLD 安全文档示例

5.1.11给出了有关SEooC数字组件安全文档的建议，这可以合并到“安全手册”或“安全应用说明”（这类文档）中。这些建议也可供PLD制造商和PLD用户使用，备注如下：

——PLD 制造商和 PLD 用户之间明确哪些文档可获得，及向 PLD 用户提供到（怎样的）详细程度；

——PLD 制造商提供的安全档案的主要要点是：

- 根据GB/T 34590 的适用要求，对PLD制造商的开发流程的分析结果的说明；
- 根据GB/T 34590 的适用要求，对PLD的支持工具的分析结果的说明；
- 在安全分析期间， PLD用户使用的信息提供（例如，PLD失效率、PLD失效模式及其相关失效模式分布、PLD中已实施的安全机制的声称的诊断覆盖率等）；
- 安全机制的建议或示例，例如，有关相关失效等的示例；及
- 随PLD提供的（并用于）指导PLD用户正确使用安全相关信息的（前提）假设列表；

——安全生命周期的工作成果由 PLD 用户提供。工作成果的完整性取决于 PLD 用户是否也是相关项集成者的角色。

5.3.7 PLD 安全分析示例

本文件附录E 给出了PLD定量安全分析的详细示例。

5.4 多核组件

5.4.1 多核组件的类型

有两种类型的多核组件：

——仅包含相同（内核）物理实体 PE 的同构多核组件；及

——具有不相同（内核）物理实体 PE 异构多核组件，典型的（例子），具有不同的指令集架构（ISA）。

示例：图 27 给出了一个通用的同构双核系统的框图，具有 CPU 本地一级缓存，和一个裸片内共享的二级缓存。

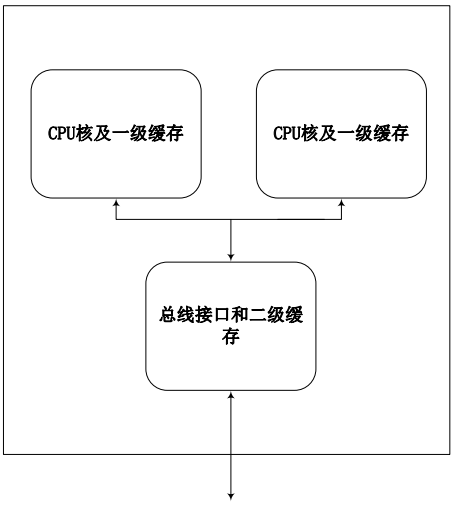


图27 双核系统通用框图

5.4.2 多核组件在 GB/T 34590 中的含义

5.4.2.1 简介

本条提供了多核安全要求分配的指导，之前这些安全要求是分配给多个组件的。

5.4.2.2 多核组件免于干扰（FFI）的说明

如果是在多核的情况下，共存有多个不同ASIL等级的软件要素，按照GB/T 34590.9-XXXX第6章，需要进行免于干扰分析。

GB/T 34590.6-XXXX附录D中列出的典型故障，可作为该分析的起点。

注1：本条仅关注在物理实体PE中被执行的软件要素之间的级联故障。干扰也可能由硬件相关失效引起，这种情况适用GB/T 34590.9-XXXX第7章。

关于对GB/T 34590.6-XXXX，D.2.3“存储”实体的干扰（分析），需要考虑对私有资源的干涉情况。这种类型的干涉可能影响到PE的数据或程序区域。

示例1：私有数据可能是属于某个 PE 的安全相关软件要素的变量:来自其他 PE 的这些变量的损坏导致软件的故障。在这种情况下，读写监控的安全机制和确保访问独占的安全机制，有助于避免干扰。。该例子与软件干扰有关（如，由软件错误引起变量损坏）。干涉也可能由硬件相关失效引起，这种情况适用 GB/T 34590.9-XXXX 第 7 章。

示例2：私有程序区域可能与非易失性存储器内的程序损坏有关。在这种情况下，限定只可以从较高 ASIL 等级要素进行编程的机理，有助于避免干扰。该例子可用于与软件相关的干扰（当程序损坏是由软件错误引起的，例如，错误的权限导致软件改写了程序存储）。这种情况适用 GB/T 34590.9-XXXX 第 7 章。

这种类型的干涉也会影响不同PE之间共享的资源。

示例3：多个内核使用 CAN 外设与其他 ECU 交换信息。干扰会导致错误的信息传输。在这种情况下，使用可靠的端到端保护机理（例如，GB/T 34590.5-XXXX 表 D.6 中列出的机理）可以帮助检测干扰。

示例4：将读取和监控外部传感器的任务分配给软件。初始要求划分为 ASIL X。在后续开发步骤中，该要求分配给软件要素 software_mon.1，ASIL Y（X）和软件要素 software_mon.2，ASIL Z（X）。相关失效分析分析说明，共享资源（内核、RAM 和软件驱动程序“软件外设”将传感器值传到 software_mon.1 和 software_mon.2）的问题，可能严重影响到独立性要求，比如，在 software_mon.1 和 software_mon.2 之间造成存储、时间、执行或信息交换的干扰。在此示例中，共享内核问题的解决，可通过将 software_Mon.1 和 softwae_Mon.2 映射到两个不同的物理实体来解决，如非共享内核。存储干扰方面通过操作系统配置的存储保护单元 MPU，对存储进行封装来解决。在这种情况下，操作系统也是一种安全机制，保证 software_mon.1 和 software_mon.2 之间的独立性，所以它需要按照 ASIL X 进行开发。共享软件资源“软件外设”的问题，按照初始 ASIL 等级（即 ASIL X）进行开发，可以得到解决。

根据GB/T 34590.6-XXXX，D.2.2中对“时间和执行”实体的干涉，需要考虑的主要情况，是影响内核的程序执行延迟或正确编程顺序的干涉。

示例5：多个内核使用 CAN 外设与其他 ECU 交换信息。如果处理较低 ASIL 等级任务的 PE 连续请求来自 CAN 外设的传输，则在另一个核中运行的较高 ASIL 等级任务不能接收和/或传输所需信息。时间监控机理（例如，使用 GB/T 34590.5-XXXX 表 D.8 中列出的安全机制的原理描述）可以帮助识别这种情况。

注2：5.4.2.3中描述了与时序相关的其他要求。

根据GB/T 34590.6-XXXX，D.2.4中对“信息交换”实体的干涉，“存储”或“时间和执行”的失效带来的干涉，有可能是不同PE之间的信息交换失效导致的。

示例6：非安全相关内核发出的消息，被解读为安全相关（伪装故障）。

注3：使用鲁棒的端到端（通讯）保护机理（例如，GB/T 34590.5-XXXX表D.6中列出的机理）可以帮助检测干扰。

当对软件分区（例如，功能或要素分离，以避免级联失效）来实现软件组件之间的免于干扰时，可采用GB/T 34590.6-XXXX，7.4.9。

Hypervisor管理程序等技术，可以帮助实现软件分区（例如，参考文献[26]和[5]）。

注4：也可采用其他技术，如微内核（例如，参考文献[12]）。

在涉及管理程序技术的多核安全分析中，以下几点值得考虑：

——虚拟化技术可以支持保证多核运行的软件要素之间免于干扰的观点。需要对软件层面的进行相关失效分析，通过考虑 GB/T 34590.6-XXXX 附录 D 中列出的失效模式得到支持；及

注5：虚拟化技术在免于干扰方面的好的效果，可能会因为监管软件中的系统故障，而达不到预期的效果。同样，支持虚拟化技术的硬件资源（如存储管理单元）或相关共享资源的硬件故障，也会对虚拟化技术造成影响。按照GB/T 34590.9-XXXX第8章中描述的方法，对这些故障进行分析，而5.1 给出数字组件专门指导意见。硬件相关失效也会影响虚拟化技术，这种情况适用GB/T 34590.9-XXXX第7章。

注6：如果将任何管理功能分派给软件分区中的任务，那么注5中提到的分析也（相应）扩展到分区。

——虚拟化技术通常无法充分防止或探测影响多核的永久性故障或瞬态故障。

注7：如果随机故障破坏虚拟化过程中的强制软件分区，则虚拟化技术可能探测这些故障。根据GB/T 34590.9-XXXX第8章中描述的方法，可以通过逐案详细分析来证明特定硬件失效模式的探测。数字组件专用指南见5.1。

5.4.2.3 多核组件的时间要求

GB/T 34590.6包括与执行时间要求有关的章节，例如：

——GB/T 34590.6-XXXX，6.4.2，e）要求软件安全要求规范考虑时间约束；

——GB/T 34590.6-XXXX，7.4.13 要求对嵌入式软件所需资源进行使用上限的估算（包括执行的时间）；

——GB/T 34590.6-XXXX，表 10 注 c）表明硬件和软件之间关系可能影响，例如，处理器平均和最快性能、最小或最大执行时间；及

——GB/T 34590.6-XXXX，附录 D 将时间和执行失效模式（包括错误分配执行时间）描述为软件要素之间干扰的潜在起因；

多核可能会受到时序故障的影响（见参考文献[26]），因此，通过专门的分析和适当应对措施的实现，可以考虑前面所列章节。

示例1：在识别可能违反安全目标的时序故障的专门分析中，其典型方法是基于执行时间的上限估计（例如:参考文献[6]）。

示例2：探测违反时间要求的典型硬件应对措施是看门狗、定时监控单元和特定硬件电路（例如，参考文献[26]）。基于软件的应对措施也是可能的（例如:参考文献[3]）。

5.5 传感器和转换器

5.5.1 传感器和转换器术语

如GB/T 34590.1-XXXX， 3.172的定义，转换器是将能量从一种形式转换为另一种形式的硬件元器件，因此，它是汽车功能安全里需要考虑的关键要素。输出能量形式相对于输入能量形式的量化取决于转换器的灵敏度。输入能量包括存储在化学键内的能量。

传感器要素至少包含一个转换器和一个硬件要素，其中硬件要素用于支持、调制或进一步处理转换器的输出，以用于电气/电子系统中。

示例1：直流偏置、放大、滤波。

转换器和传感器之间的关系如图28所示。

注1：图28中的转换器可以是一个单独的组件，支持电路可以是一个单独的或多个组件。转换器和支持电路的功能共同构成传感器的功能。

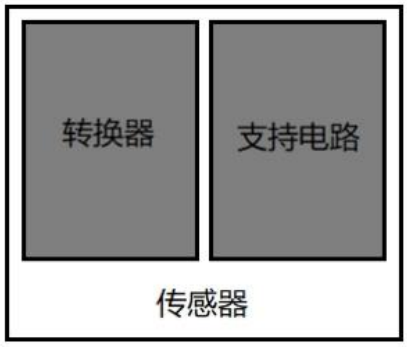


图28　传感器和转换器之间的一般关系

与其他电气/电子要素一样，传感器可以由元器件和子元器件组成，并且具有不同的复杂性。

示例2：由转换器和放大器构成的具有模拟输出的半导体组件。

示例3：由外壳，带数字信号处理和数字输出的传感器 IC，必要的外部组件（例如电阻器，电容器）和与线束接口的连接器所组成的要素（见图 29）。在此示例中，传感器 IC 和其他要素都可以被归类为传感器，但位于不同的层级。

注2：本条中的术语“转换器”特指那些使用半导体工艺技术制造的转换器，包括微机电系统（MEMS）。本条中的术语“传感器”特指那些包含上述的转换器的传感器，并具有电气输出。

传感器可以通过各种方式进行分类，如参考文献[43]所示。

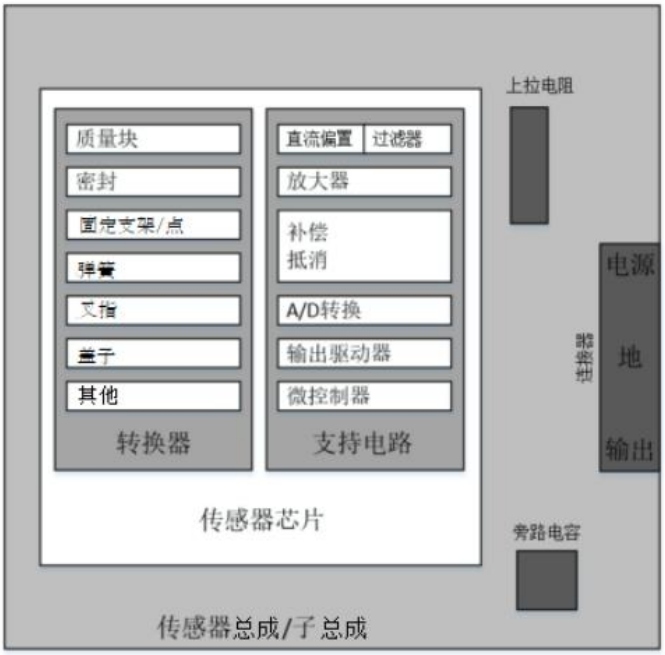


图29　复杂的分层传感器示例

5.5.2　传感器和转换器的失效模式

在本条范围内，每个转换器的输出都在电气域内。任何原因造成的转换器失效模式都将是电气失效模式。始于转换器的信号路径中任何要素的 失效都可能对传感器输出产生影响。

转换器的失效模式可以通过4.3.2中提到的方法导出。

表53包括各种不同类型和复杂度的转换器通用的失效模式（与测量、检测手段、转换手段等无关）[42]。 该表只是转换器的电气失效模式的示例，并不详尽，因为转换器的电气失效模式取决于特定转

换器的类型和功能。传感器信号路径中包含的数字或模拟支持电路的失效模式在5.1和5.2中有相关介绍。

转换器的失效模式表现为与传感器标称输出的偏差。 传感器的失效模式也源于转换器输出和传感器输出之间的信号路径中的支持电路的故障。 转换器的失效模式与传感器输出的失效模式之间的相关性将取决于传感器中转换器的具体实现方式。 按照GB/T 34590.5-XXXX，表D.1，需要对实际传感器类型进行详细分析，以识别每种失效模式。

表53包括转换器失效模式对系统输出的可能的影响。这些影响是否被认为是传感器的相关失效模式取决于分配给传感器的安全要求。通常，只要偏差是可预测的，就可以在系统或要素层面评估指定范围内传感器标称性能的偏差。任何超出预期范围或行为模型的性能偏差都可能导致违反传感器的安全要求。

表53　转换器失效模式示例（电气）

技术规范	失效模式	描述
偏移	偏移超出指定范围	在没有激励（输入能量）的情况下，转换器输出偏离理想值
	温度偏移误差	温度偏移误差超出规定限值
	偏移漂移	偏移值随时间而变化
动态范围	超出范围	转换器输出超出规定的工作范围
灵敏度（增益）	灵敏度过高/过低	灵敏度偏离超出规定的限值
	卡滞	由于机械和电气失效（例如，粒子短路，粘滞），灵敏度为零
	非参数灵敏度	灵敏度偏离其指定范围内的数学关系，包括不连续性或输出响应的削波
	噪音，重复性差	克服动态本底噪声所需的可变阈值
	温度敏感度误差	因温度引起灵敏度偏离超出规定的限值
注：系统层面可能产生的影响包括：切换阈值不准确，切换阈值随温度变化，切换阈值随时间变化，功能丧失，切换阈值不准确，相移（超前，滞后），占空比变化，输出切换阈值的变化，切换阈值随温度变化，相位随温度偏移，占空比随温度的变化。		

示例：典型的基于摄像机的图像传感器可以由以下元器件和子元器件组成：像素阵列；模拟链，时钟和电源；配置和标定电路；存储器包括 RAM，OTP；特殊电路；数字控制；接口。数字控制，存储器和相关接口的失效模式按照 5.1 的描述进行分析，而模拟链，时钟和电源的失效模式按照 5.2 的描述进行分析。以下是根据表 53 中列出的类别， 给出的可能影响像素阵列及其余元器件和子元器件的失效模式示例：

——特定失效模式：摄像机故障（作为阵列的主要故障导致全图像故障）；单个图像单行丢失或水平线失效；单个图像单列丢失或垂直线失效；图像帧丢失；

——与灵敏度（增益）有关：图像中像素数据丢失或比特位损坏；图像中的噪声；

——与偏移有关：水平或垂直的图像移位；及

——与动态范围有关：曝光不足或过度曝光的图像/像素，包括与动态范围相关的问题。

5.5.2.1 生产过程和失效模式

基于半导体的传感器和转换器的制造是多步骤过程，包括许多机械工序，例如晶片研磨/减薄，锯，拾取和放置，裸片粘接，引线键合，裸片堆叠和封装。由这些过程引起的机械应力可以影响材料特性，例如会导致设备参数的波动的材料电子迁移率。 转换器/传感器的技术规格，例如偏移，直接受到组装过程的应力的影响。 在机械生产过程之前没有表现出特定失效模式的传感器或转换器在该过程之后不能保证没有该失效模式。

在供应商发货之前，通过多种方法对传感器标定，使得它们的技术规范（例如，偏移，灵敏度）集中在各自的范围内。 然而，供应商的生产过程并不是由组装引起的机械应力的唯一来源。 直接客户的生产过程以及在供应链下游的那些生产过程，可能引入机械应力或其他可能导致传感器的某种失效模式的环境因素。 这些工艺包括但不限于表面安装、夹紧、拾取和放置、回流焊和敷形涂层工艺。 如果可能，在每个连续的供应商的生产流程的最后阶段之后，需要验证转换器/传感器运行是否符合规范。

表54列出了可能由装配过程造成的传感器/转换器所发生的失效模式。本表并非详尽。在设计阶段需考虑检测任何由这些过程引起的传感器性能偏差的能力，以及偏差减轻措施，以确保足够的鲁棒性。（例如，偏移消除，灵敏度调整和测试模式）。有关在开发阶段避免系统性故障的更多信息，请参阅5.5.5。

表54 在生产过程中可能会引入的传感器异常

生产相关的失效模式	可能影响	可能原因
灵敏度偏移	不准确的切换阈值，相移占空比偏移	机械应力（压电电阻），温度引起的机械应力，机械短路或开路（例如破碎的金属，异物，ILD空隙），陷阱电荷，跌落，冲击，压缩/减压，振动，水分侵入，温度循环引起的塑性变形，材料固化
灵敏度丢失	系统丢失	
偏移	切换阈值不准确	

5.5.2.2 微机电失效原因

MEMS传感器用于各种应用中，并采用机械探测的方法，通过典型的弹性电（基于运动）转换方式来感知环境。由于转换方法是机械的，因此转换器的性能直接受其物理结构和结构中与标称规格偏差的影响。

图30和图31是通用MEMS转换器的图示。图30显示了通用MEMS转换器的各个元器件，包括电极，质量块，固定支架，弹簧和电容器极板。 图31显示了侧视图的其他细节，包括空腔，密封盖和防粘连涂层。这些元器件的任何非理想的物理/机械特性都会对转换器输出产生（电）影响。

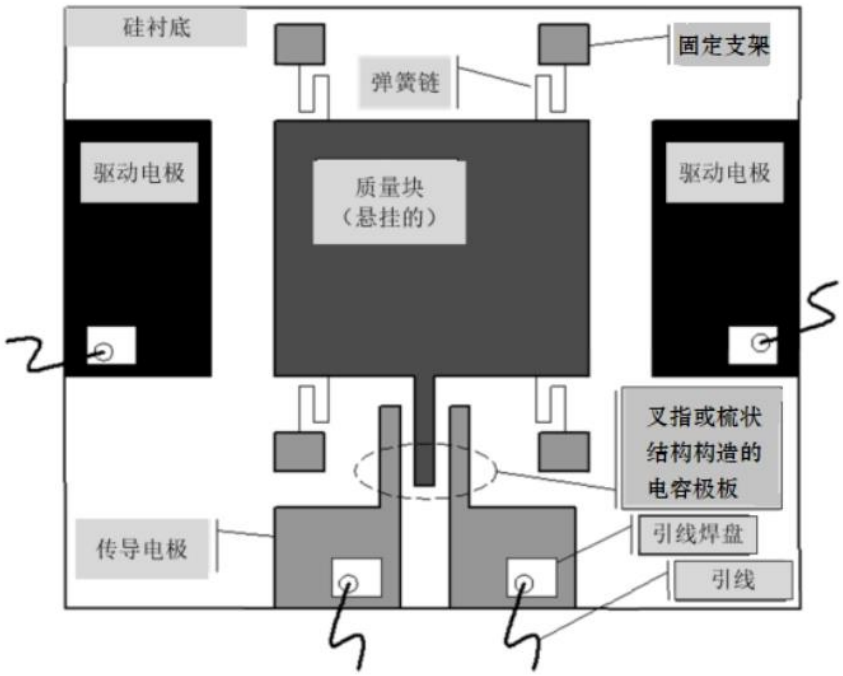
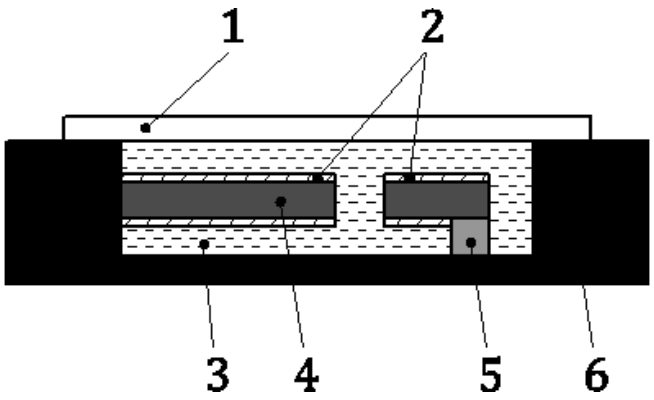


图30　MEMS 转换器示例 （顶视图）



说明：

1 ——密封盖

2 ——防粘连涂层

3 ——密封腔-加压（正压或负压）

4 ——质量块：多晶硅薄膜（如悬臂）

5 ——固定支架

6 ——硅衬底

图31　MEMS 转换器示例 （侧视图）

表55中列出了MEMS转换器的一些常见机械失效根本原因以及相关失效模式，但这并非详尽。

表55　MEMS 转换器的失效根本原因以及相关失效模式示例[45]

机械失效根本原因	转换器失效模式	描述
弹簧断裂	非参数灵敏度	MEMS运动转换器通常设计有一组弹簧，以提供机械定位，建立线性灵敏度并限制行程。 如果该组中的一个弹簧破裂，则质量块变得不平衡，使得部分行程看起来正常，但是最接近断裂弹簧的部分将放松甚至无约束，导致非线性灵敏度。
叉指断裂	灵敏度漂移，偏移漂移，传感器动态变化	MEMS运动转换器通常设计有多组电容叉指,用于感测质量块运动。 灵敏度与器件总电容成正比，后者是每个单个叉指电容的总和。 如果叉指断裂，则总电容减小，导致灵敏度的降低和偏移的漂移。
密封腔破裂		由于MEMS腔结构内部存在密封气体分子,叉指之间的间隙提供空气动力学阻尼。 灵敏度与密封气体的压力成比例。 如果密封性被破坏，则压力降低，导致灵敏度增加，最终改变传感器动态性能（例如截止频率的变化）。
薄膜破裂	偏移漂移，卡滞	MEMS压力转换器通常设计为薄膜型,用于在压阻元件上施加应力或改变电容间隙。 如果隔膜破裂，可能会发生灵敏度的偏移或完全失去灵敏度，从而导致卡滞到地的故障。
固定支架破裂		MEMS运动转换器的设计中通常包含弹簧的固定支架,或者包含用于限制行进距离的类似结构。 如果固定支架或行程限制器破裂,则质量块会偏离方向或行进到与腔的内表面接触的允许边界之外，从而导致卡滞故障。
颗粒物	灵敏度漂移 非参数灵敏度 偏移漂移 卡滞	颗粒物能够引入多种失效模式,这取决于颗粒的导电性和它所接触的转换器的各个部分。 如果颗粒是导电的，它可以将元器件短接在一起，如果它是电阻性的，它会阻碍元器件的运动。 如果颗粒异物在腔内自由移动，颗粒也会导致瞬态失效和一般的不可预测性。 在生产过程中或是由于运行期间的破损/磨损，可能会产生颗粒异物。
抗粘连涂层异常	灵敏度漂移 非参数灵敏度 卡滞	表面张力或静电力会导致悬浮/悬臂表面粘附到其他移动表面或固定表面,这是由于用于防止这种效应的涂层异常造成的。
一般机械过应力	灵敏度漂移 非参数灵敏度	机械过应力的来源可能包括冲击，疲劳，振动，腐蚀或电过应力（EOS）或静电放电（ESD）的影响，导致MEMS传感器元

	偏移漂移 卡滞	器件或子元器件的结构损坏。

5.5.3 传感器和转换器的安全分析

5.5.3.1 确定和分配基础失效率的注意事项

在确定集成转换器的失效率并将基础失效率分配给转换器和支持电路时存在一些特定的挑战。进行定量分析时，需要考虑以下几点：

——无源转换器占据了相当大比例的裸片面积，其中还包括有源电路；

示例1：基于霍尔单元的传感器。

注1：有源要素和无源要素之间的失效率可能存在差异， 同样的，几何形状较大的器件与几何形状较小的器件失效率也存在差异。

——在有源电路上方制造的转换器不占用有源裸片的区域；

示例2：GMR（巨磁电阻）。

——由于 MEMS 技术的发展迅速，因此手册通常不覆盖 MEMS 要素；

——转换器失效率分布类型取决于结构；

示例3：用于压力传感器的 MEMS，具有腔体，相对较大的隔膜和小型压电转换要素。

——转换器可以在没有支持电路的情况下组装，因此不能使用常用的可靠性标准来确定基础失效率；

——对于新技术，缺乏现场数据并且可靠性数据有限；及

——转换器与支持电路的失效率可以从不同的源得出。

注2：如果失效率不是来自相同的源和条件，则需要适当的换算。

在各种情况下，确定传感器的基础失效率以及如何将失效率分配给转换器要素的方法需要基于合理和文档化的理由。

示例4：以下是确定新 MEMS 转换器失效率的方法示例（无现场/可靠性数据）：

1) 从已确定的 MEMS 器件的失效模式开始，其包括总体失效率，失效机理（例如，颗粒物，粘连，空腔破裂）和基于已确定的数据分布（例如，现场返回或其他类似的可靠性源）；
2) 确定每种失效机理的基础失效率；
3) 对于每种失效机理，分配一个敏感因子，将设计/评估下的转换器与用于导出上述步骤 1 和 2 中的数据的转换器进行比较。该敏感因子对参考转换器与评估中的转换器之间的相对风险进行评估，例如：更高、更低或相同；
4) 将来自步骤 2 和 3 的数据组合起来，为评估中的转换器的每个失效机理产生加权失效率；及
5) 应用来自步骤 1 的失效模式分布以产生新的 MEMS 转换器的单个预测失效率。

注3：这只是一个示例方法。 所定义的程序既不详尽也不局限，也不限于MEMS，并且假定基于已经文档化并通过适当证据证实的基本原理。

5.5.3.2 传感器和转换器的相关失效分析

如果要求独立性或免于干扰，则按照4.7中描述的流程执行相关失效分析。 表56给出了各种类型传感器的相关失效引发源示例。

表56 传感器和转换器的相关失效引发源

4.7.5中定义的相关失效引发源类型	示例
由于共享资源的随机硬件故障产生的相关失效引发源	共用的标定和/或配置资源（例如，用于控制基于CMOS的图像传感器的eFUSE）
由于随机的物理性根本原因产生的相关失效引发源	时变噪声或固定模式噪声
由于环境条件产生的系统相关失效引发源	长时间暴露在过热，潮湿或强烈阳光下 静电放电
由于开发类错误产生的系统相关失效引发源	图像传感器设计错误
由于生产类错误产生的系统相关失效引发源	传感器制造缺陷
由于安装类错误产生的系统相关失效引发源	磁传感器的目标轮离轴（偏心）安装 图像传感器的镜子安装位置错误

用以评估控制或避免传感器和转换器相关失效的有效性的方法，可以从4.7.5.2中描述的典型方法推导得到。

5.5.3.3 定量分析

对于传感器，关于硬件架构度量的评估和由于随机硬件失效导致违背安全目标的评估的定量分析，对比其他硬件要素，没有程序上的差异。

显著差异是在分析中要纳入转换器要素，因为传感器安全要求的违背与转换器要素的失效模式十分相关。对于定量分析将考虑以下几点：

——颗粒度级别（如何将其分类为元器件或子元器件）；

——量化的失效率及其来源；

注：可靠性和HTOL测试可用于推导出除手册数据之外的，关于转换器的新技术、新应用和实现技术的失效率。 另见4.6.1.6。

——失效模式分布；及

——包含传感器特定的安全机制（见 5.5.4）。

按照GB/T 34590.5-XXXX，第8章和GB/T 34590.5-XXXX，第9章，对半导体元器件和机械元器件的电气失效模式进行定量分析。

按照5.1章节中针对数字电路和5.2章节中针对模拟电路的指导，进行支持电路的定量分析。

5.5.4 传感器和转换器安全机制示例

表57列举了传感器/转换器一些通用的安全机制的示例，它对在评估环境中的转换器要素起独特作用。

由于传感器包括多种数量和类型的支持电路，因此这些安全机制是分别对于5.1（关于数字），5.2（关于模拟），5.3（关于PLD）章节中所含任何模拟或数字安全机制的补充。

表57中包括的示例并不详尽，也可以使用其他技术。 提供支持声称的诊断覆盖率的理由。

注：不可能为传感器/转换器的诊断覆盖率提供通用指南，因为诊断覆盖率很大程度上取决于具体的技术，电路类型以及用例。

表57 传感器/转换器的安全机制示例

安全机制/措施	参阅技术概述	备注
高压密封质量滤波器	5.5.4.1	MEMS特定实现。
冗余隔膜	5.5.4.2	MEMS特定的片上校准参考。
偏移抵消	5.5.4.3	允许偏移优化。
转换器特定的自检	5.5.4.4	各种测试信号路径完整性的方法。
自动增益控制	5.5.4.5	允许低水平的环境激励并增加动态范围。
灵敏度调整	5.5.4.6	使灵敏度居中。
MEMS特定的非电气/电子安全措施	5.5.4.7	评估MEMS转换器物理特性的措施。

5.5.4.1 密封质量滤波器

目的：提供一种低通滤波器机理，可以抑制噪声，否则噪声可能会混入有用的频段。 常用于MEMS加速度计转换器。

描述：用大于大气压力密封的防护质量室可以抑制环境引起的MEMS转换器元器件的移动。

示例：MEMS 转换器可以由多组“梳”齿组成，其间隙由精密公差限定。 由于防护质量室在压力下密封，周围气体呈现挤压膜阻尼效果，类似于减震器，过滤高频振动。 较高的压力会捕获更多的气体分子，会降低截止频率。 较低的压力捕获较少的气体分子，允许更高的截止频率。

5.5.4.2 冗余隔膜

目的：提供永久性参考，以便与系统的主要转换要素进行比较。

描述：包含一个参考转换器，以便将由于环境因素而允许移位的主传感隔膜与相同但不可移动的隔膜进行比较。 常用于MEMS压力转换器。

示例：MEMS 转换器可以被制造成具备一个不（随压力）位移的“孪生转换器”，其在相同的工艺步骤和临界尺寸下同时形成，并且受制于相同的工艺公差。 因此，诸如灵敏度之类的由温度或施加的电压引起的共因变量将被共享并且在数学上相互抵消，使得对比位移与非位移反应成为采样到的唯一剩余差异。

5.5.4.3 偏移抵消

目的：最小化转换器输出的偏移。

描述：有各种硬件和软件方法可用于消除由转换器的非理想特性引起的内置偏移。 所选择的方法取决于所用转换器的类型。

示例：线性磁传感器在没有磁场的情况下提供特定的 VCC / 2 静态电压。 在每个上电周期运行标定程序，以在没有磁激励的情况下量化偏移电压。 存储该值并用于调整在操作模式下获取的读数。

5.5.4.4 转换器特定的自检

目的：提供评估特定类型转换器的方法。

描述：由于转换器对环境有反应，因此在没有环境条件的情况下评估传感器/转换器的完整性是具有挑战性的。有多种方法通过自检来激励转换器，这些测试的准确性和可用性取决于所用转换器的具体类型和所评估的技术规格。 通常，通过测试对整个信号路径或隔离信号路径的一部分评估完整性，例如靠近转换器的模拟前端或经过数字处理的后端。

示例：MEMS 转换器可以包含两组感测电极，以相反的极性电连接。 两个绝对值的求和设置为零（在指定的公差范围内），且与 MEMS 机械运动无关。 超出允许零范围的值将表示质量块或传感电极完整性的不平衡或破裂。

5.5.4.5 自动增益控制

目的：在低等级的环境激励下支持传感器功能。

描述：通常，转换器的电输出会被放大，以便进一步用于传感系统。 自动增益控制（AGC）允许根据转换器输出信号的幅度调节转换器的放大增益。当转换器输出电平较低时，增益增加，当转换器输出电平较高时，增益降低，以允许更大的动态范围。

5.5.4.6 灵敏度调节

目的：将灵敏度保持在指定范围内

描述：在传感器工作温度范围内，传感器/转换器的灵敏度是在指定范围内的，以确保准确的输出。有多种方法来调整转换器的灵敏度以应对环境波动。

示例1：使用由电流激活的微加热器来保持 MEMS 元器件在温度范围内的灵敏度[46]。

示例2：通过改变流经霍尔单元的偏置电流以保持对温度的灵敏度。

示例3：向 MEMS 叉指施加静电电势，以造成电阻尼运动并降低灵敏度。

示例4：连接到 MEMS 的组件具有内置温度传感器。基于温度信息，对 MEMS 的灵敏度变化进行正确补偿。

5.5.4.7 MEMS 特定的非电气/电子安全机制

目的：提供MEMS转换器元器件专用的机械安全机制

描述：在大多数情况下，通过电子装置（在信号链的转换器接口之后）探测转换器中的非电气失效是基于估计失效对信号本身的影响来完成的。在这些情况下，通常不可能直接观察失效，因此只能使用推断来确定转换器是否发生了失效[见图32 b)] 。这种推断方法的性质可能会导致误检或漏检。

示例：转换器的量程内故障。

除了转换后级的电气或电子技术之外，还有其他的方法和技术可以永久地用于MEMS转换器内，以直接检测或控制转换器本身的失效模式[见图32a)]。 例如，可以在转换器内使用附加的机械或光学机理（例如参考文献[44]和[45]）作为安全机制，例如简单的止动或悬浮的悬臂叉指。

这些简单的机械机理可以包括单独的信号输出，以允许转换器在探测到失效模式时进入安全状态，从而将转换器从系统中特定安全目标或硬件要求的相关失效引发源中去除。 这可以是任何专用措施或传统电气/电子安全机制的补充，并且可以潜在地提供针对转换器内的随机和系统故障的覆盖率。

可以在诊断覆盖率的应用中定义这种非电气/电子安全机制。 非电气/电子安全机制为特定用例提供的诊断覆盖率水平需要领域专家进行合理的工程评估，以获得适当值，并将每个基本依据和验证活动完整文档化并包含在安全档案中。 一旦经过验证和确认，组件中的这种非电气/电子安全机制可以有助于系统或要素达到给定安全要求或安全目标的ASIL等级。

a)机械的（非-电气/电子）安全机制　　b)电气的安全机制

图32　机械探测和电气探测推断转换器失效之间的区别

5.5.4.8　传感器的专用措施

如GB/T 34590.5-XXXX,9.4.1.2和9.4.1.3中所述，可以考虑采用专用措施来确保在评估违反安全目标或要求的概率时声称的失效率。

传感器和转换器的专用措施示例包括：

——过设计传感器或转换器的元器件或子元器件以获得鲁棒性（例如电气或热应力等级）；

——对关键传感器或转换器规范进行特殊样品测试或 100％生产测试，以降低发生失效模式的风险；

——布局相关措施；

示例1：配置四个霍尔单元，以最大限度地减少与压力相关的偏移。

——调整键合焊盘顺序，以最大限度地减少交互机会；

示例2：共模杂散电容或漏电流,以影响开关电容质量块移动。

——工艺措施。

示例3：使用湿蚀刻代替干蚀刻技术来去除掩埋氧化层，从而产生更光滑的表面并增加 MEMS 元器件的强度[46]。

5.5.5　关于避免传感器和转换器的系统性故障

除了5.1.9和5.2.5中针对数字和模拟组件所描述的内容外，表58中描述的措施可用于传感器和转换器。

表58　在传感器或转换器的开发期间实现符合 GB/T 34590.5-XXXX 要求的技术或措施的示例

GB/T 34590.5-XXXX 要求	设计阶段	技术/措施	目的

7.4.4 硬件设计验证	验证	内部接口验证	通过专用测试验证传感器或转换器的机械、机电、光电、磁性元器件与相关模拟和/或数字元器件之间的正确集成。
10.5.1 硬件集成和验证	硬件集成和验证	测试封装的影响	测试封装（例如像镜子这样的支持元器件）对传感器/转换器特性的影响。
7.4.4 硬件设计验证	设计	有限元分析（FEA）	减轻诱导应力的影响。 为了确保分析的有效性，要显示 FEA 结果与产品开发后期或早期样品或早期产品中可获得的测量值之间的相关性。
7.4.3 安全分析	设计	FMEA	考虑转换器失效模式的完整性和正确性，包括失效模式，分布及其对传感器输出的影响
7.4.2.4 鲁棒性设计原则	设计	可制造性设计	考虑传感器/转换器电气特性的制造工艺变化，以提高鲁棒性。
7.4.4 硬件设计验证	设计	可测试性设计	设计必要的硬件，以便全面评估转换器性能和传感器/转换器安全机制。
7.4.5 生产、运行、服务和报废 9.4.1.2,9.4.1.3 专用措施	芯片生产中安全相关特殊特性	光学图案检测，以检测和剔除早期失效	将半导体工艺的特定层与参考几何形状进行光学比较，以便检测图案异常现象。
10.5.1 硬件集成和验证活动	评估硬件要素	环境测试以模拟实际操作条件	执行扩展可靠性测试，模拟使用环境条件。如，振动试验。
10.5.1 硬件集成和验证活动	硬件集成验证	针对具有环境激励的传感器的独特测试	能够将传感器/转换器暴露于其感测的环境激励中，例如 加速度，磁场，压力

5.5.6 传感器和转换器的安全文档示例

传感器和转换器的安全文档是根据数字组件（见5.1.11）和模拟组件（见5.2.6）所述的文档制作的。 包括：

——基础失效率，包括评估时的假设和理由；

注：如果基础失效率显示失效率如何分布在可能影响传感器和转换器的不同故障模型上，则非常有用。

示例：对于基于图像传感器的相机，提供像素阵列故障可影响单个像素，整列，整行，多个像素或完整阵列的百分比。

——转换器失效模式列表，具有终端影响和失效模式分布；及

——用户信息，如安全手册或安全应用笔记，特别强调以下内容：

- 集成在设备中的安全机制及其可用性；
- 可能影响设备安全特性的配置或标定参数（和相关程序）；及
- 影响功能安全的生产相关说明。

附 录 A
（资料性）
有关如何使用数字失效模式进行诊断覆盖率评估的示例

A.1 DMA安全机制评估示例

A.1.1 用例描述

以下是此示例中考虑的DMA用例：
——通信外设每隔X毫秒接收一条消息；
——一旦消息被通信外设接收，它就触发DMA请求；
——DMA将消息从外设接收缓冲区传输到RAM区域；
——总是传输到相同的RAM区域，独立于消息内容；
——DMA完成传输后，会触发一个CPU中断；及
——CPU根据消息ID将消息复制到RAM中的另一个缓冲区。

A.1.2 安全机制的描述

在此示例中，下列安全机制可用于监控DMA活动是否正确：
——SafMech_01_DMA_MPU:定义可通过DMA访问的存储器区域的专用存储器保护单元：
- 写访问仅限于目标地址；及
- 读取访问仅限于源地址；

——SafMech_02_E2E_Protection:
- DMA传输的消息采用以下机制进行端到端保护：
  - 覆盖数据内容、消息ID和消息计数器的8位CRC；
  - 消息ID(4位)；及
  - 消息计数器(4位)；
- 在2^4=16个消息ID中，只有12个ID是有效的；
- 计数器达到最大值0xF后重置为零；及
- 接收到数据传输完成信号后,CPU将消息复制到另一个RAM区域。DMA无法访问此存储区域。在CPU执行复制操作之后，检查E2E保护机理。应用程序只使用此副本，不使用DMA的目标地址中的数据；

——SafMech_03_Timeout_Mon:数据传输应该是周期性的，系统知悉数据传输的频率，从而监测在规定时间范围内是否有数据传输发生；及
——SafMech_04_IR_Source_Mon：在中断请求的情况下，这个安全机制将检查触发是否来自合法源。

A.1.3 失效模式的定义和诊断覆盖率的估算

根据描述的用例和安全机制，定义了下列失效模式，并可以估算以下诊断覆盖率的值。

A.1.3.1 DMA_FM1: 请求的数据传送未发生

由于在指定的时间范围内没有数据传输完成信号，此失效模式可以被SafMech_03_Timeout_Mon探测到。 FMC_{DMA_FM1}估算为100％。

A.1.3.2 DMA_FM2：没有请求，但发生了数据传输

DMA将数据从源地址传送到目标地址。它指示了数据传输的完成。取决于源地址的内容，传输的数据可能是以前的消息(DMA_FM2.1)或随机值(DMA_FM2.2;“白噪声”模型，即每个可能的错误状态都是等概率的)。

更多详情：

——DMA_FM2.1：通过 E2E 保护机制（SafMech_02_E2E_Protection）中的消息计数器或消息 ID 可以检测是否是之前的消息。 $FMC_{DMA_FM2.1}$估算为 100％；

——DMA_FM2.2:在随机值的情况下：

- 随机匹配合法CRC值的概率pCRC,legal为$1/2^8$；
- 随机匹配合法ID的概率pID,legal为12/16；
- 随机匹配正确计数器值的概率pCounter,legal为$1/2^4$（因为2^4个值中只有一个值是正确的）；
- 未触发错误的总体概率pRF为pRF = pCRC,legal×pID,legal×pCounter,legal = 0,000183；及
- $FMC_{DMA_FM2.2}$估算为1 - pRF，因此等于99.98%。

为了准确估计总体失效模式覆盖率，本文对两种失效模式DMA_FM2.1和DMA_FM2.2之间的失效模式分布进行了估算。

由于这两个值都很高并且非常接近，因此忽略了对这两种失效模式分布的估算，并且仅使用较低的值：

FMC_{DMA_FM2}和$FMC_{DMA_FM2.2}$估算值为99.98％。

A.1.3.3 DMA_FM3：数据传输过早/过晚

为了评估，更详细的失效模式如下：

——DMA_FM3.1：在正确请求之前触发数据传输。此失效模式等同于 DMA_FM2，此处不再进一步评估。 $FMC_{DMA_FM3.1}$估算为 100％；

——DMA_FM3.2 在正确的请求之后过晚触发数据传输。根据延迟的不同，其影响可能是下列情况之一:

- DMA_FM3.2a：根据通信外设不同，消息可能在被DMA提取之前被后续消息覆盖，或者可能无法接收到后续消息。这两种情况都会导致消息丢失。这种情况可以被SafMech_03_Timeout_Mon或SafMech_02_E2E_Protection（通过消息计数器）探测到，且FMC = 100％。取决于通信外设，其本身可以产生附加的错误信号；
- DMA_FM3.2b：在DMA获取消息期间，通讯外设接收到下一个消息并部分覆盖前一个消息。这将造成一个损坏的消息，其内容由前后两个消息的部分组成：
  - ID 是合法的(pID,legal = 1)；
  - 当前消息的计数器值在很高的概率下可能与前一个消息的计数器值相同（如果两个消息具有相同的传输频率）。这里假设最坏情况的概率为 pCounter, legal = 1；
  - 将数据损坏模型化为“白噪声”，生成随机匹配合法 CRC 值的概率 pCRC,legal 为 $1/2^8$；及

- FMC = 1 – pCRC,legal × pID,legal × pCounter,legal = 99.6 %.

- 取决于通信外设:

  • 可以生成附加的错误信号，增加有效的 FMC，或

  • 此失效模式不可能发生，失效模式只有 DMA_FM3.2a。

为了准确估算$FMC_{DMA_FM3.2}$，需要推导DMA_FM3.2a和DMA_FM3.2b的失效模式分布。对于保守的初次估算，可以使用两者中较低的FMC: $FMC_{DMA\_FM3.2}$ = 99.6%。

——DMA_FM3.3：在传输完成之前发出传输完成信号。这将导致一个消息的部分损坏，其目标缓冲区中的消息由两个消息混合组成。就 SafMech_02_E2E_Protection 的探测而言，可以证明$FMC_{DMA_FM3.3}$的估算值类似于 DMA_FM3.2b，为 99.6%；

——DMA_FM3.4：传输完成后，数据传输完成的信号提供过晚。这种失效模式会导致：

- DMA_FM3.4a：在被CPU提取之前，该消息被后续的消息覆盖。这会导致消息丢失，该失效可以被SafMech_03_Timeout_Mon或SafMech_02_E2E_Protection探测到，且FMC = 100%。这类似于DMA_FM3.2a；及
- DMA_FM3.4b：在CPU提取期间，该消息被DMA覆盖。这会导致消息部分损坏。 FMC = 99.6%（类似于DMA_FM3.2b）。

按照与之前相同的证明，$FMC_{DMA_FM3.4}$总体值可估算为99.6%。

A.1.3.4 DMA_FM4：错误输出

和先前与时序相关的失效模式相比,该失效模式属于时序正确但是输出不正确的问题。在此示例中，DMA具有以下输出：

——控制信号：读取或写入；

——控制信号：访问宽度（8 位，16 位，32 位）；

——控制信号:要访问的地址；

——数据(在写入的情况下)；及

——四种不同的中断请求信号。

可区分出以下子类别的失效模式：

——DMA_F4.1a：写入操作被执行为读取操作；

——DMA 对 RAM 目标地址的写入操作被执行为对该地址的读取访问。该消息将不会被更新。完成“传输”之后，DMA 仍然会触发 CPU 中断请求。SafMech_02_E2E_Protection 通过检查 ID 或计数器值的方式探测到这是一条旧消息。此外，SafMech_01_DMA_MPU 会探测到非法访问（写入操作被误执行为读取操作）。 $FMC_{DMA_FM4.1A}$估算为 100%。

——DMA F4.1b: 读取操作被执行为写入操作；

- 读取操作被执行为写入操作：DMA对通信外设的读取访问被执行为写入操作。根据通信外设的不同，这可能会导致通信外设的错误响应。此外，SafMech_01_DMA_MPU将探测到非法的写访问。$FMC_{DMA_FM4.1b}$估算为100%。

——DMA_F4.2: 错误的访问宽度；

- 错误的访问宽度：此失效模式将导致消息损坏，可通过SafMech_02_E2E_Protection的CRC探测到。通过ID检查和非法的消息计数器值也可以探测到该错误（参见SafMech_01_DMA_MPU）。$FMC_{DMA_FM4.2}$估算为99.6%。

——DMA_F4.3:错误的访问地址；

- 错误的访问地址：此失效模式将导致DMA访问非法地址，并将被SafMech_01_DMA_MPU探测到。$FMC_{DMA_FM4.1b}$估算为100%。

——DMA_F4.4： 错误的数据输出；及

- 错误的数据输出：该失效模式将导致消息的随机损坏，类似于DMA_FM2.2。 $FMC_{DMA_FM4.4}$估算为99.98％。

——DMA_F4.5：错误的中断请求

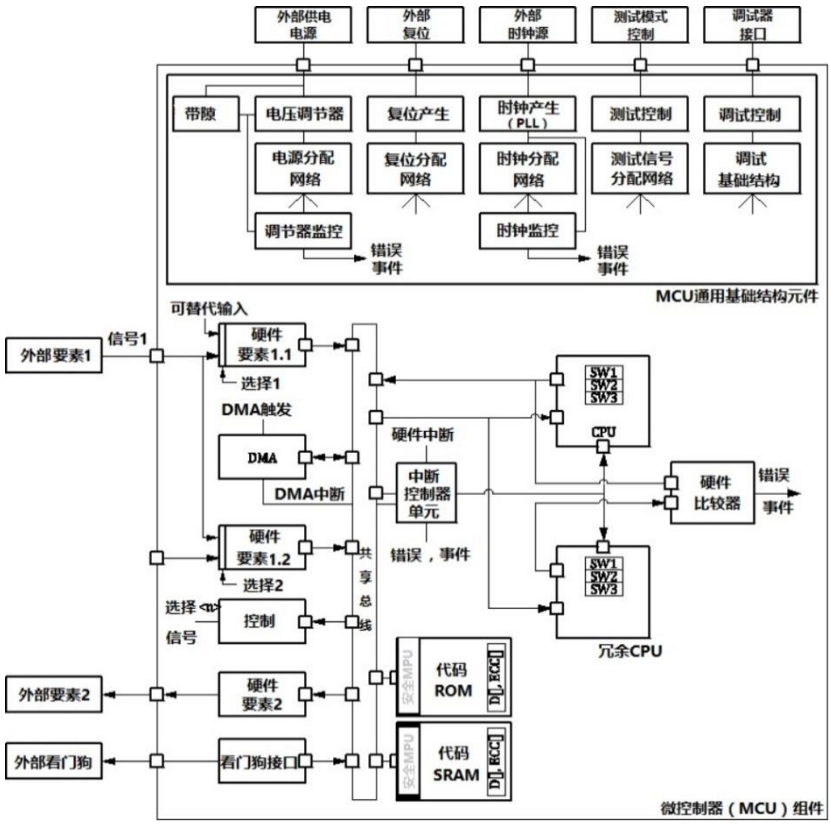
- 错误的中断请求：在此示例中，DMA只触发一个CPU中断请求。因此SafMech_04_IR_Source_Mon将探测到此故障。 $FMC_{DMA_FM4.5}$估算为100％。

附 录 B
（资料性）
相关失效分析示例

B.1 微控制器示例

B.1.1 描述

采用图B.1中描述的微控制器组件，来对数字组件的相关失效分析方法进行说明。



图B.1 微控制器组件示例

首先介绍硬件和软件要素，以此来重点介绍相关失效分析用到的硬件安全机制。本示例的讨论范围不涵盖硬件安全要求和安全机制的全面定义。

——硬件要素 1.1：接口处理要素，从连接到微控制器的硬件要素（例如，从外部要素 1 收到的信号 1）接收信息。

——硬件要素 1.2： 从功能角度看，等同于硬件要素 1.1 的接口处理要素

——硬件要素 2：该要素用于控制外部要素 2。

——控制：该要素提供选择信号，能够控制硬件要素 1.1 和 1.2 与微控制器的不同输入接口的连接。

——CPU：执行软件要素的中央处理单元。

——数据静态随机存储器（Data SRAM）：软件要素保存它们的私有变量的存储器。还包含软件和 DMA 之间、以及软件要素之间的通信缓冲区。

——代码只读存储器：只读存储器，存放软件要素执行的代码，及可能存有被软件要素使用的常数。

——软件要素：在此示例中，列出了三个软件要素：软件 1、软件 2 和软件 3。

——看门狗接口：通过它与外部看门狗硬件要素进行通信。

——共享资源：识别以下共享资源：

- DMA（直接内存访问）硬件要素：DMA可由每个软件要素使用，并对可寻址资源（存储器、配置寄存器）具有读写访问权。
- EVR（嵌入式稳压器）：除由“外部供电电源”供电的输入/输出焊盘外，EVR可以为微控制器内的每个硬件要素提供电源。
- 复位生成和分发：根据外部复位源产生的复位命令，或由硬件、软件要素控制的内部复位动作，来控制微控制器的复位状态。
- 时钟生成与分配：通过使用外部时钟源的锁相环PLL，为每个硬件要素提供需要的时钟。
- 测试逻辑：微控制器生产测试所需的测试结构。

功能安全概念和要求概念定义如下。信号S1是指示执行器状态的模拟信号。要求是“应识别出非预期状态，并使执行机构停止工作”。这被认为是安全状态。为此，信号S1转换为数字信息，然后由软件要素软件1处理，以识别执行器可能的危险状态。软件要素软件2负责冗余地从硬件要素1.1和1.2获取信息。软件2的主要任务是控制DMA，从硬件要素1.1和1.2中获取转换结果，并将它们存储在数据SRAM中的共享缓冲区中的单独数据集中。DMA通过向ICU发送中断通知软件2传输完成。接收到该事件后，软件2会比较数据集的合理性，如果不匹配，它会向软件1提供预定义的错误信息。软件要素软件3负责定期刷新外部看门狗。刷新要求发送具有给定序列的动态代码。要发送的代码仅由软件要素软件1提供。如果软件3未能刷新看门狗或发送错误代码，则外部看门狗进入超时状态，使执行机构停止。

本附录提供了示例性安全要求。该组安全要求的规范简化为适用于 DFA 的最小集：

——MCU-REQ-1：“硬件要素 1.1 处理信号 1 时的故障，应在 20 毫秒内被检测到[ASIL X]”：

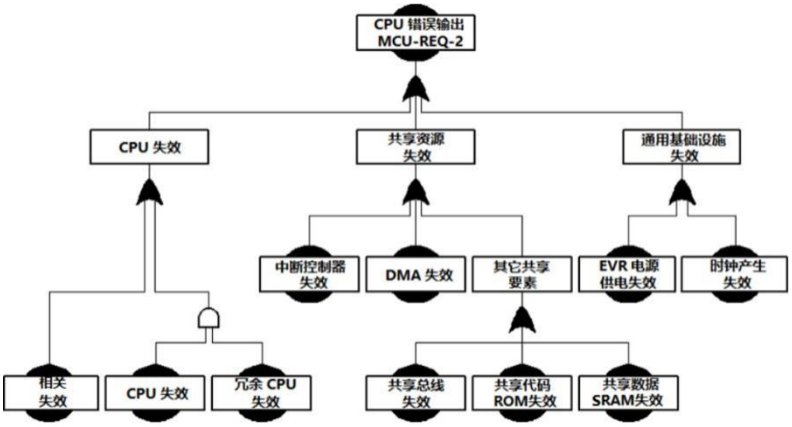
- MCU-REQ-1.1：“信号1应由硬件要素1.2进行冗余处理”；及
- MCU-REQ-1.2：“硬件要素1.1和1.2的结果应由软件进行监控。在结果出现不匹配时，软件通过看门狗接口，向外部看门狗发送报错消息”。

——MCU-REQ-2：导致 CPU 输出错误的随机硬件故障，应在 20 毫秒内被检测到[ASIL X]：

- MCU-REQ-2.1：“中央处理单元CPU应由冗余CPU监控。CPU和冗余CPU的输出通过硬件比较器在每个时钟周期都进行比较；及
- MCU-REQ-2.2：“当CPU和冗余CPU的输出出现不匹配时，应生成错误事件。

B.1.2 相关失效分析

相关失效分析将只针对可能导致违反安全要求MCU-REQ -2的相关失效引发源进行分析。分析将依照所建议的工作流程。为了简化分析，将不会考虑每个步骤。根据MCU-REQ -2的要求，本步骤侧重于相关失效分析工作流程中的B1和B2步骤，重点关注架构的分析。通过定性故障树（参见图B.2）方法支持的分析，识别出共享资源和冗余要素。



图B.2 共享要素概览

对于共享资源，每个失效基础事件或“与门”都是被独立分析的。对于CPU和冗余CPU，已经引入了基础事件相关失效，因为在所建议的架构层面已经可以看到安全机制。建议对具有全局影响的通用基础设施要素分别单独进行分析，以避免为每个共享要素单独考虑它们。对于电源和时钟产生电路，这样分析是可能的，因为它们都有自己的安全机制。但是，对于复位产生电路、测试信号和调试的基础设施，有必要在较低的层次进行分析，以便分析它们对共享要素安全机制的影响。对于通用基础设施要素，分析将集中在电源和时钟产生电路。

表B.1 给出了微控制器相关失效分析的示例。

表B.1 微控制器相关失效分析示例

序号	要素	冗余要素	相关失效引发源		相关失效分析	
	简称和描述	简称和描述	共享资源	单个物理性根本原因	避免(A)或控制(C)故障的措施	验证方法
通用基础设施要素						
PS1	电源	电源监控： 在电源运行条件下，测量电压电平	共用的带隙有可能导致无法探测到过压		（C）增加一个带隙监控	芯片级鲁棒性测试
PLL1	时钟	时钟监控 频率测量	共享输入频率有可能阻碍准确的频率测量		（C）增加一个独立的时钟源（振荡器）来测量 PLL 频率 （A）差异化设计：由于采用不同的实现，锁相环与时钟监视器所用的参考振荡器的漂移行为之间会有差异。	设计检查 芯片级鲁棒性测试

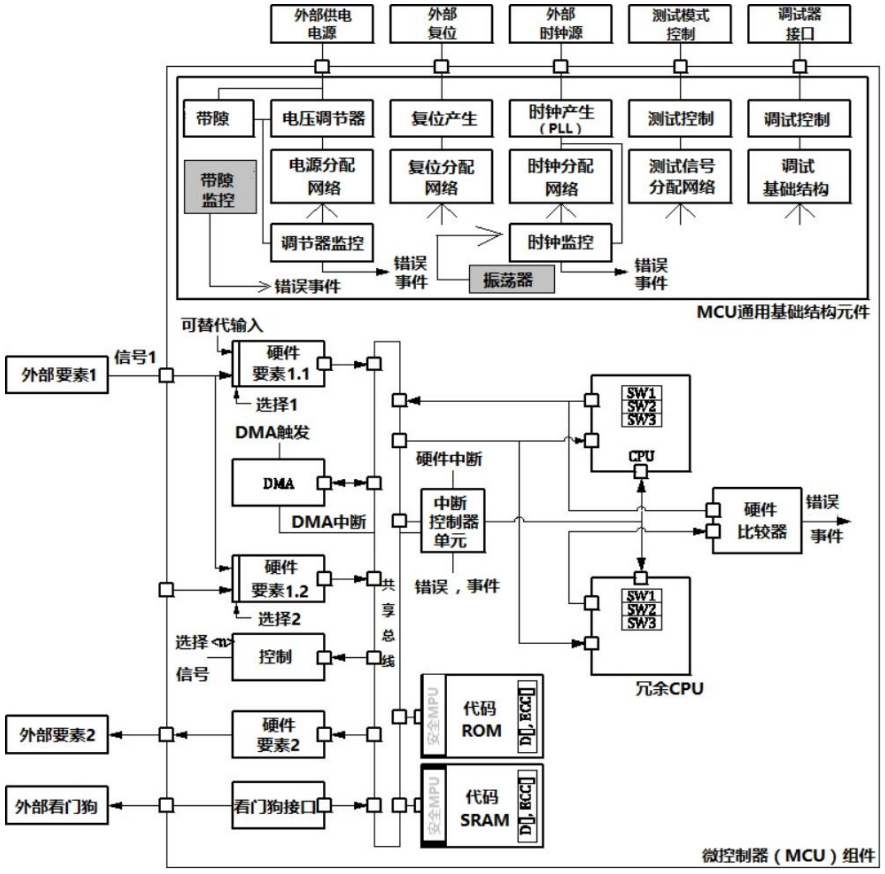
PLL2	时钟	时钟监控 频率测量	时钟丢失，导致监控电路无法报告失效情况		（C）通过外部看门狗功能对半导体进行监控	
PLL3	时钟	时钟监控 频率测量		它是基于一个详细的时钟生成和时钟监控的框图进行分析的，其中相关的接口、边带信号和配置寄存器是可见的。		
处理要素						
CPU1	CPU 计算	冗余 CPU+硬件比较器	供电		通过电源分析覆盖	
CPU2	CPU 计算	冗余 CPU+硬件比较器	时钟： 错误频率		通过 PLL 分析覆盖	
CPU3	CPU 计算	冗余 CPU+硬件比较器	时钟：时钟小误差			
CPU4	CPU 计算	冗余 CPU+硬件比较器	共享的总线			
CPU5	CPU 计算	冗余 CPU+硬件比较器	数据 SRAM		通过安全分析覆盖数据 SRAM 的安全机制（如，ECC） ECC 是通过冗余 CPU 来评估的，冗余 CPU 能够控制与数据 SRAM 接口相关的相关失效。	
CPU6	CPU 计算	冗余 CPU+硬件比较器	代码 SRAM			

CPU7	CPU 计算	冗余 CPU+硬件比较器	ICU			
CPU8	CPU 计算	冗余 CPU+硬件比较器		属于 CPU 的信号与属于冗余 CPU 的信号之间发生短路	（A）按照工艺设计规则做物理分隔	设计规则分析 物理布局检查
CPU9	CPU 计算	冗余 CPU+硬件比较器		影响属于 CPU 的逻辑和属于冗余 CPU 的逻辑的闩锁	（A）按照工艺设计规则做物理分隔，以隔离标准单元避免闩锁 （A）与由软错误导致的闩锁相关的物理分隔	设计规则分析 物理布局检查

在DFA带来了架构上的增强之后，微控制器组件框图将更新为：

——新的带隙监控元件，使与带隙漂移失效模式相关的相关失效被减轻；及

——新的振荡器要素，使与时钟漂移失效模式相关的相关失效被减轻。



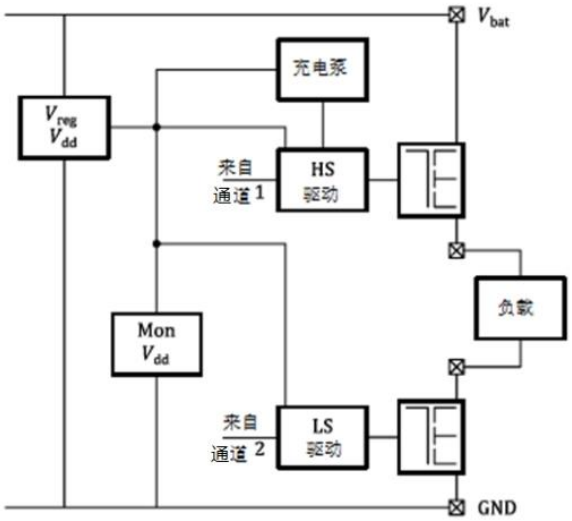
图B.3 增强的微控制器组件

B.2 模拟示例

B.2.1 描述

模拟示例旨在为模拟组件，元器件或子元器件的相关失效分析应用提供指导。详细的失效模式，相关的相关失效引发源，安全要求以及考虑到的安全和减轻措施的选择都是典型的示例，但这些是不详尽的，并且可以根据应用，系统架构，电路设计和IC技术的细节而调整。

在以下章节中，基于假定的开关输出级架构，解释模拟元器件的相关失效分析。该输出级的架构如图B.4所示，它使用高压N-DMOS开关晶体管来开启负载的电流路径，该负载可以是安全应用中的执行器的一部分。为了避免开关晶体管或其栅极驱动器的故障可能无意中激活执行器，会在负载的高边和低边电流路径中，冗余地放置开关。高边和低边驱动器由稳压器V_{reg} V_{dd}供电，该V_{reg} V_{dd}远低于连接到车辆12 V电池板的外部电源V_{bat}。电源稳压器的输出已经由电压监控器监控，该电压监控器用于非安全目的，例如提供上电复位电源。打开高边N-DMOS开关晶体管所需的栅极电压由电荷泵提供，以使驱动器对电源网络上的EMC不敏感。



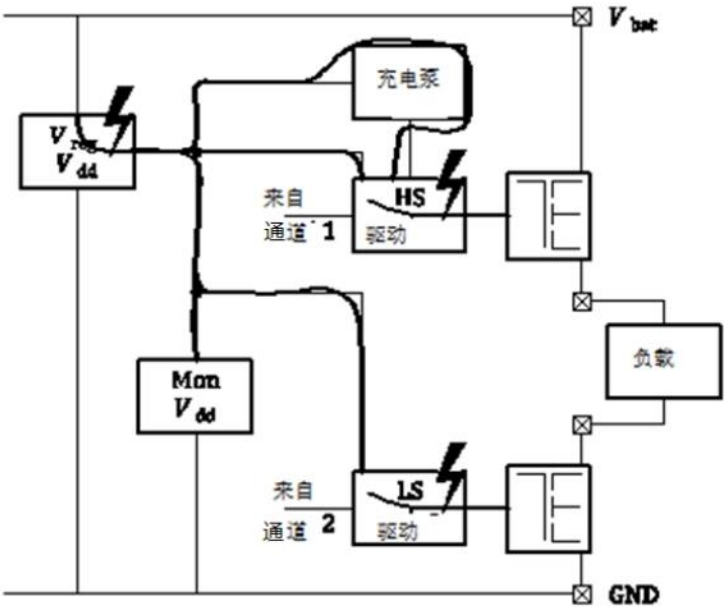
图B.4 模拟输出驱动器示例

为了能够识别相关失效机理，假设以下安全要求：“在非激活状态下，高边开关晶体管输出和低边开关晶体管输出之间连接的负载上不应提供超过 1 mA，持续时间超过1 ms的电流”。

注：假设1 mA的电流远低于开关打开时负载汲取的电流（例如1 A）。

B.2.2 共用电源电压调节器的相关失效

导致示例中相关失效的主要故障如图B.5所示。为控制开关晶体管栅电压提供内部驱动电路的电源电压调节器发生失效，导致通断装置(通断装置是位于电源电流路径上的晶体管)永久打开。故障机理可能是通断晶体管本身的缺陷或控制环路的故障，导致不稳定性，例如， 失去补偿电容器。 结果是内部电源电平V_{dd}上升到外部电源电平V_{bat}。



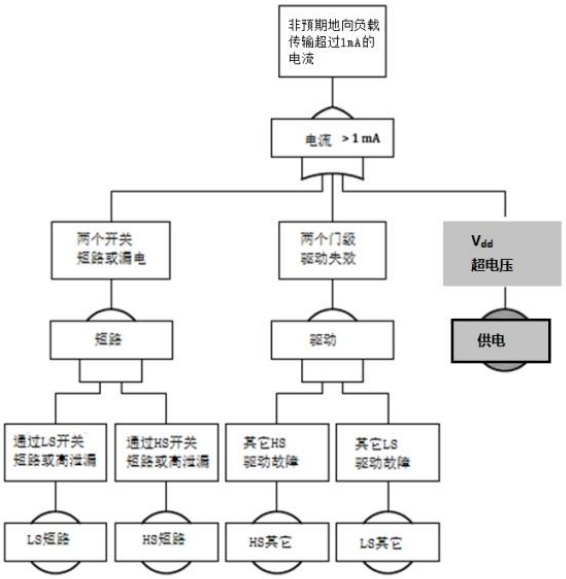
图B.5 共用电源电压调节器的相关失效

假设该示例中的复杂驱动器电路无法在其与外部电源短路时正常工作，如果该故障出现，就会违反安全要求。

因此，假定驱动器严重损坏并且驱动器输出不能将开关晶体管的栅极电压保持在使开关晶体管保持在高阻抗状态的电平上。 因此，假定由施加到驱动级的供电的“过电压”引起的相关失效对于驱动级具有最坏后果。因此，它会传播到图B.6所示的故障树中的顶层失效。

在定量安全分析中，电源电压调节器的“过压”失效模式的SPFM（无需是电源电压调节器的每个失效模式，例如欠压）将直接添加到违反定义的安全目标的SPFM，如图所示，从灰色的基础事件，即V_{dd}电源电压调节器的过压，连接到FTA的顶层“或”门。

注：在电源电压调节器提供过压的情况下，可能会出现其他相关失效。 第一个是在电荷泵中引起的故障，其在框图中以虚线示出。 在最坏的情况下，该故障可能具有与V_{dd}电源输入处的过压导致的高边驱动器损坏相同的影响，因此已经以V_{dd}电源过压故障的方式包含在FTA中。 可能由过压引起的另一个相关失效是电压监控器的损坏，这可能导致过压未被检测到；稍后将讨论减轻栅极驱动器相关失效的措施。



图B.6 包括共用电源的FTA

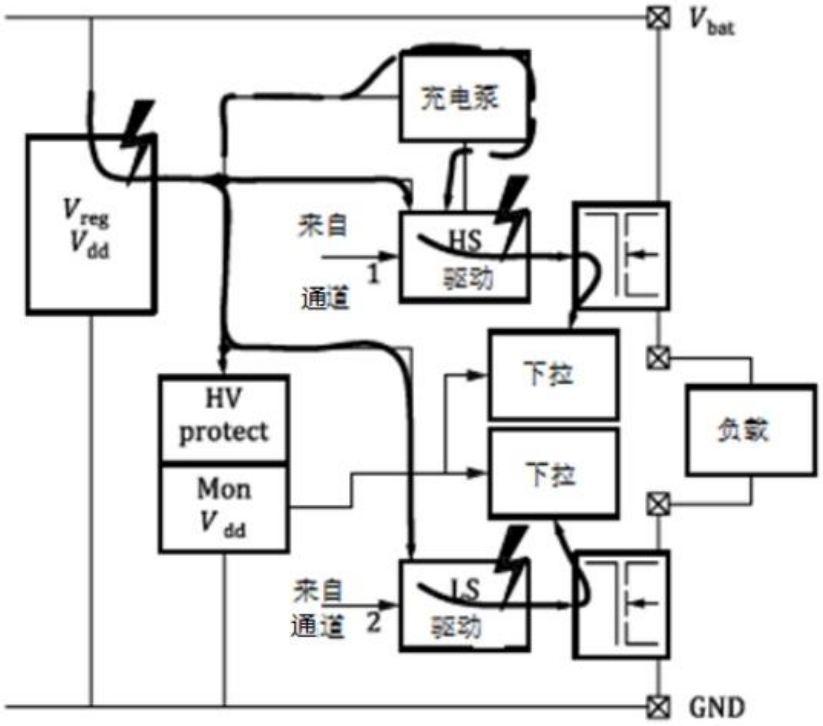
为确保在电源电压调节器出现该故障的情况下达到安全要求，可以导出以下免于干扰要求：“电源电压调节器模块中的失效不应导致激活高边或低边开关晶体管，使其输出可以向负载提供超过1 mA的电流超过1 ms。”

为了实现免于干扰，定义了安全措施，以避免在驱动级的内部电源和外部电源电压V_{bat}之间连接的情况下违反安全目标。 采取措施的例子如图B.7所示：

——引入子元器件以将开关晶体管栅-源极电压下拉至低于其阈值电压。 下拉电路由电源监控模块激活；及

——可以通过驱动器输出和开关晶体管栅极之间的连接上电流的限制，以确保在栅极驱动器输出处发生电源短路的情况下，下拉能够使栅-源极电压保持足够低。

由于引入了上述安全机制，系统的结构发生了变化，只要下拉子元器件被激活，由内部电源上升到电源网络的电压水平的初始相关失效不再导致违反安全要求。 如果没有其他可能影响该安全机制功能的级联效应，那么相关失效就得到了足够的减轻。 按照由相关失效分析结果得到的减轻措施对故障树进行调整，如图B.8所示。



图B.7 公用电源故障减轻

如果架构变化引入的其他附加相关失效机制，可能影响为减轻初始相关失效的新安全机制（a）和（b）有效性，则需要额外的免于干扰要求。 对于这种情况，新的免于干扰要求可以表述如下：“电源电压调节器将内部电源Vdd短路至外部电源电压Vbat的失效不应导致电压监控器失效或下拉电路失效，该失效阻断下拉电流路径，使得开关晶体管阈值超出时间长于1 ms。”

为了实现这种新的免于干扰要求，为开关晶体管安装了附加的安全措施。 这些下拉模块不受初始故障（内部电源Vdd短接到外部网络电源Vbat）的影响，保持输出开关晶体管的栅极被拉低。

采取措施的示例：

——为电源监控器引入高压保护块（a）；及

——其规格适合于在外部电源电压下工作的栅极下拉的设计（b）。

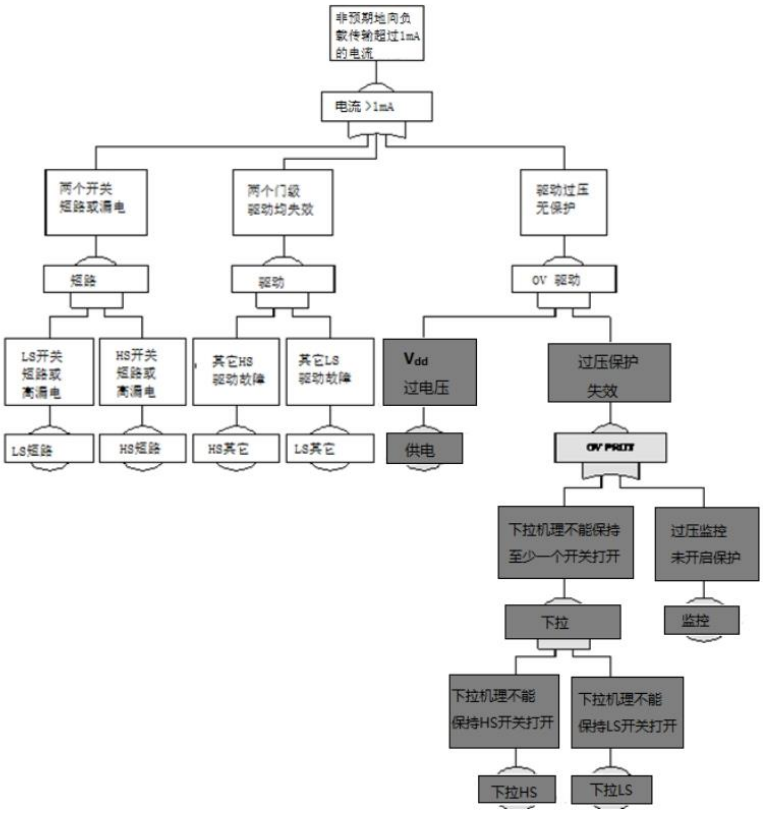
对于该示例，假设IC技术允许以提供足够安全裕度的方式实施这些措施。 这种假设在定性评估中是合理的，因为电源监控器和下拉模块很小，可以通过某种方式（例如，增加通道长度，级联高压晶体管，串联电阻器）实现比电源稳压器更高的安全裕度（对电源稳压器而言，需要更高的绝对最大额定值）。当然，安全要求，故障机理和建议的减轻方法仅仅是示例性的，并且基于以下边界条件的假设：

——电路架构；

——应用要求；及

——用于制造电路的 IC 工艺的能力。

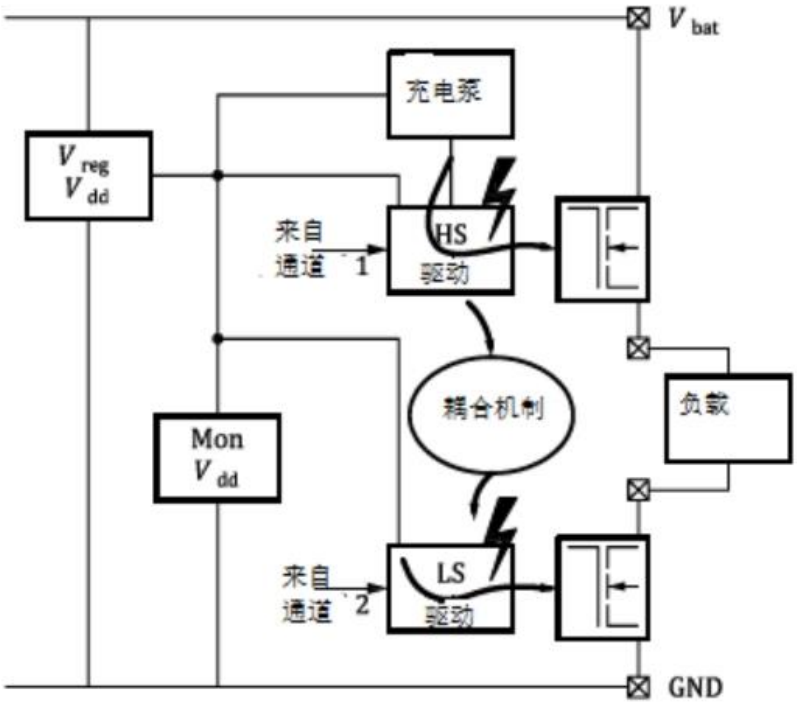
该示例的目的是解释如何执行模拟元器件的相关失效分析，而不是作为在实际的开关输出级中，减轻由电源电压调节器的过压失效引起的相关失效的参考。 可以使用其他方法来减轻相同的故障，这取决于真实边界条件的最终信息（例如，技术选项，外部安全机制）。 最后，对为减轻由电源过压引起的相关失效而引入的新要素执行潜伏故障分析。 该分析可以确定是否需要在重复的时间间隔内（例如在每次系统启动时）测试它们。



图B.8 公用电源故障减轻的 FTA

B.2.3 耦合机制导致的相关失效

导致第二示例性相关失效的主要故障如图B.9中所示。 这是高边驱动器中出现的随机硬件故障。它导致高边路径的失效，从而导致高边开关晶体管的导通。 它进一步激活耦合效应，该效应可以引发低边路径中的相关失效。



图B.9 耦合机制导致的相关失效

独立性要求可以表述为：“高边路径的失效不会导致低边路径的失效，从而导致低边开关晶体管的激活，使其能够提供超过1 mA的电流。” 作为对相关失效引发源清单的评估结果，确定了以下相关的引发源（见表B.2）及其相应的耦合机制，需要为这些机制定义特殊的减轻措施。

注：这只是一个例子，并不意味着这3个相关失效引发源是唯一与栅极驱动器相关的。

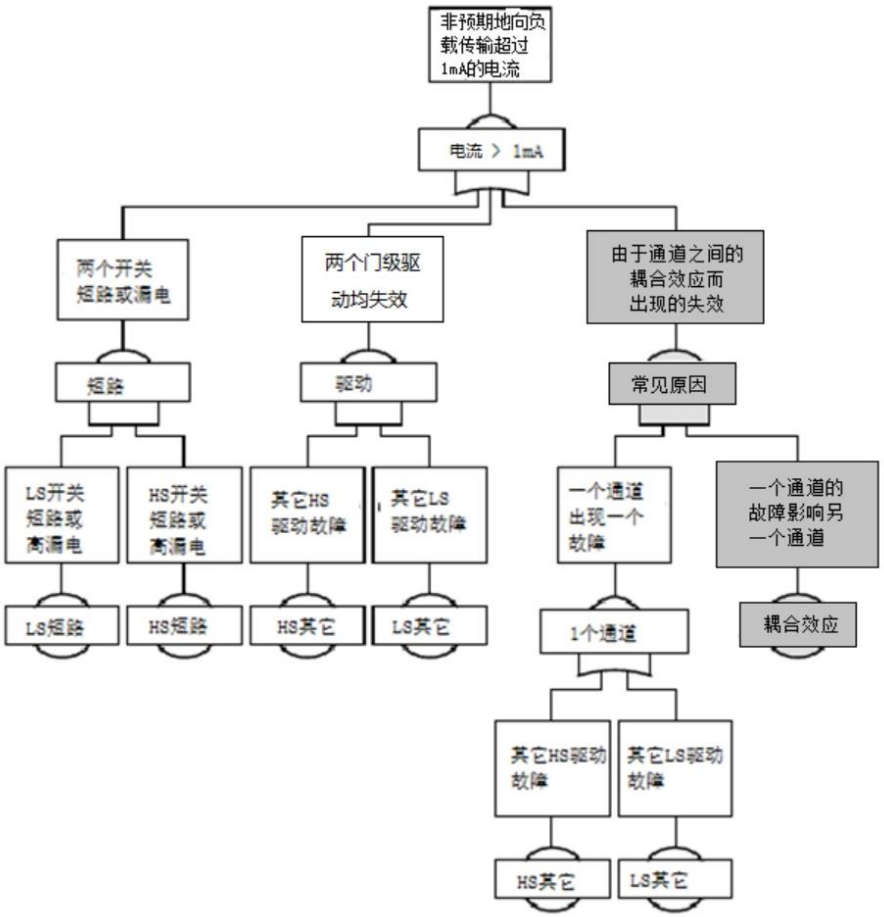
表B.2中列出的每个相关失效，皆使用图B.10中的故障树。 它表明，除了每个通道中的独立随机故障之外，通道之间的耦合可能导致不会受到初始故障的直接影响的第二通道中的故障。

对于温度升高（表B.2中的参考编号1）或电源中断（表B.2中的参考编号2）的情况，可以通过实施检测耦合效应的安全机制来避免相关失效。 使系统或元件进入安全状态。 对于衬底电流注入的情况（表B.2中的参考编号3），可以通过破坏耦合机制的技术和/或布局措施来实现减轻。

表B.2 确定相关耦合机制的示例

参考编号	相关失效引发源	耦合机制
1	其中一个栅极驱动器电路中的局部热点（例如，由于有缺陷的器件的功耗增加而导致栅极驱动器内的器件的升温）。	经由衬底的热传导导致超过另一栅极驱动器的温度范围的最大额定值。
2	其中一个栅极驱动器发生短路，导致电流消耗高于电源电压调节器的规格。	另一个栅极驱动器的电源跌落会导致其处于未定义状态（既不在操作范围内也不在导致上电复位的范围内）。

3	在一个栅极驱动器内将电流注入衬底，例如，由衬底pn结的缺陷导致，或通过激活功率器件的寄生双极晶体管引起的。	由于沿着电流路径，从衬底到GND上的电压降增加，导致包括其他栅极驱动器电路的元件引起的闩锁。



图B.10 包含耦合效应的故障树

为了减轻已识别的相关失效，表B.3中定义了补充的安全机制。

注：减轻相关失效可能需要一个或组合的减轻措施，所选措施的证据的最终证明可用于实际设计、布局、技术、包装和应用。

表B.3 减轻耦合效应的示例

参考编号	相关失效减轻
1	栅极驱动器附近的温度检测（可接受的距离取决于散热路径的热阻，可通过热仿真得出，检测元件可以是电阻器或双极型晶体管），并且在过温情况下关闭栅极驱动器电源。 电源电压调节器中的电流限制，以限制可使芯片升温的功率，并使调节器进入指定的欠压复位状态。

	独立路径（高边和低边路径，每个由开关晶体管及其相关的栅极驱动器组成）的热隔离（例如，暴露的管芯焊盘与背面散热器组合的足够距离），需足以防止非故障路径（不受初始故障影响的路径）过热。 可以通过例如热仿真评估所需分离的尺寸。
2	区域供电的电流测量，并在过流情况下关闭栅极驱动器电源。 具有欠压复位的电压监控器通过将复位阈值设置在电路的安全工作范围内来避免未定义状态。 栅极的被动下拉，例如，如果电源电压低，则用电阻将开关晶体管保持在关断状态。
3	物理分离（例如间隔，保护环，分离阱，沟槽，埋层，沉降片 - 取决于IC技术），目的在于阻隔声称独立的元器件之间的闩锁机理。

附 录 C
（资料性）
数字组件定量分析示例

C.1 描述

以下是使用5.1.7中所述方法进行定量分析的例子。

注1：此示例使用的数字（例如失效率、安全故障数量和失效模式覆盖率）只是示例。它们随架构不同而不同。

注2：以下示例将数字组件的一部分划分到子元器件级别。如5.1.7所讨论的，必要的详细程度取决于分析阶段和所用的安全机制。

注3：以下示例使用定量的方法计算瞬态故障的特定目标“单点故障度量”值。如5.1.7.2所讨论的，瞬态故障也可以通过定性原理来解决。该原理包括定性方法是充分的原因。

该例子仅考虑数字组件的一小部分，即只有两部分：

——一个小型 CPU，分为五个子元器件：寄存器组、算数逻辑单元 ALU、载入-存储单元、控制逻辑和调试。每个子元器件再进一步划分为若干子元器件；及

——一个 16 KB 的随机存储器 RAM，分为三个子元器件：存储单元阵列、地址解码器和下线测试逻辑，以及随机存储 RAM 的备用行（冗余）管理。

注4：本示例所示的FIT数值不包括外设或其他功能，如封装、处理或过载。只是用它们作为一个例子，说明计算FIT值的可能的方法。因此，这些值与含完整封装的数字组件的FIT值(例如SN29500中所示)没有可比性。

注5：下面这个例子目的是避免在系统层面分析中要求处理每个最小数字组件子元器件。在系统级分析中，组件或元器件层面的详细度可能是足够的。本示例的目的是提供证据，证明对于数字组件的自身层面可能需要进行更深入的分析（例如，在子元器件层面），以便以要求的精度计算元器件和子元器件的失效率和失效模式覆盖率，供系统工程师随后使用。换句话说，如果没有准确而详细的数字组件的自身级别分析，就很难获得用于系统级别分析的良好数据。

考虑以下四种安全机制：

——硬件安全机制（SM1），对 CPU 的程序序列进行逻辑监控。这个安全机制能够按一定覆盖率检测到控制逻辑中的故障，这些故障可能导致软件运行顺序错误。然而，该安全机制在检测导致数据错误的故障方面（如，错误的算术运算）能力较弱；

注6：在此示例中，假设每个被检测到的影响CPU的永久性一位故障都被发送给系统（例如，通过将数字组件的输出信号激活）。基于这一假设可以得出，根据GB/T 34590.5 所述，潜伏故障的失效模式覆盖率可以设为100%。在系统层面或要素层面提出要求以合理使用该信号（例如，进入安全状态并通知驾驶员）。对于可疑的瞬态故障，CPU可以尝试通过复位来消除这些故障。如果故障依然存在，则意味着它是永久性的，从而可以按照之前所述通知系统。如果故障消失（即确实是瞬态的），CPU可以继续工作。

——处理随机硬件失效的基于软件的安全机制（SM2），在系统启动时执行以验证对 CPU 的程序序列进行逻辑监控的安全机制（SM1）不存在潜伏故障

——错误探测-校正逻辑 ECC（SM3），能够校正随机存储 RAM 的所有单个位故障（单位错误校正，SEC）和探测所有双位故障（双位错误探测，DED）；及

注7：在此示例中，假设每个被探测到的永久性单个位故障 – 即使被ECC纠正 – 都会通知到软件（例如，通过中断），软件作出相应的反应。基于这一假设可以得出，根据GB/T XXXXX-5 所述，潜伏故障的失效模式覆盖率可以假设为100%。在系统层面或要素层面提出要求以合理使用该事件（例如，进入安全状态并通知驾驶员）。对于被ECC修复但可疑的瞬态故障，CPU可以尝试向存储器中写回正确的值，来消除这些故障。如果故障依然存在，则意味着它是永久性的，从而可以按照之前所述通知系统。如果故障消失（即确实是瞬态的），CPU可以继续工作。为了区分间歇性和瞬态故障，对修正次数计数可能是一种可行的方法。

——处理随机硬件失效的基于软件的安全机制（SM4），在系统启动时执行以验证ECC（SM3）不存在潜伏故障

为了更清晰，图C.1分为三个分开的计算表格。

图C.1 从子元器件层面给出了失效模式。图C.2揭示了如何确定低层级失效模式，以及如何按照4.4中描述的方法计算总的失效分布。

示例1：该表给出了触发器X1及其相关扇入永久性故障的失效率为0.01 FIT。将这些低层级失效模式相加，就可以计算出整个ALU逻辑的永久性故障的失效率（0.0348 FIT）。使用相同的步骤，通过将子元器件相关的每个失效率相加，可以计算出ALU中永久性故障的FIT值。

注8：从失效模式抽象树往上（即从低层级失效模式到高层失效模式）中，不同子元器件失效模式的失效率组合起来，可以计算出高层失效模式的失效率，特别是如果这些高层失效模式是以更通用的方式来定义。

示例2：如果较高层的失效模式（例如，在元器件级别）像 “CPU 执行了错误指令”这样定义，则此失效模式的失效率可以由子元器件级别失效模式的失效率组合得到，例如，流水线中的永久性故障、寄存器组中的永久性故障等。因此，如果低层失效率可用，更高层的失效率可以用自下而上的方法计算（假设各故障独立）。

注9：表中各列与GB/T 34590.10 [61]中所述的故障分类和故障等级占比计算的流程图相关：

——失效率（FIT）等于λ；

——安全故障量等于F_{safe}；

——违反安全目标的失效模式覆盖率等于$K_{FMC,RF}$；

——残余或单点故障率等于λ_{SPF}或λ_{RF}，取决于故障是单点故障还是残余故障。本示例不考虑单点故障，所以该失效率始终等于λ_{RF}；

——潜伏失效的失效模式覆盖率等于$K_{FMC,MPF}$；及

——潜伏多点故障失效率等于λ_{MPF}。

注10：安全故障数量是失效模式的一部分，安全故障即不会在缺乏安全机制的情况下违反安全目标，也不会与其他子元器件的独立失效一起违反安全目标。

注11：失效模式的覆盖率，是通过详细分析安全机制SM1覆盖每个子元器件的能力而计算得到的。在此示例中，R0和R1是编译器选择用来传递函数参数的寄存器，所以它们有略高的可能性导致程序序列错误，而这可以被SM1检测到。本示例的目的是提供证据，以证明通过详细分析可确定不同子元器件的覆盖率的不同。

注12：ECC（SM3）的失效模式覆盖率可通过例如对ECC探测单个位和双位错误的高概率以及对多位错误探测的低概率（可能低于90%）的组合进行详细分析来计算。如图C.2所示。

注13：特定子元器件可由多个安全机制覆盖：在这种情况下，最终的失效模式覆盖率是将通过详细分析确定的每个失效模式的覆盖率组合而得到。

注14：该示例表明，如果ECC（SM3）没有对多位错误的合适的覆盖率，并且没有对RAM地址解码器的覆盖率，则很难实现高单点故障度量。

注15：此示例表明，某些安全机制可能直接违反安全目标，因此在计算残余故障时会考虑这些安全机制。在此示例中，ECC（SM3）中的故障可能会损坏任务数据，而存储器中没有相应的故障。

注16：示例表明，在数字组件中，与安全无关的子元器件可以共存，但它们不可能与安全相关的子元器件（调试内部逻辑）建立明确的分离或区分。相反，其他元器件（调试接口）可以很容易地被隔离和禁用，这样它们就可以被认为是与安全无关的，没有风险。

注17：安全故障量按照GB/T 34590.10 [61]中描述的分类方法确定。这些计算可以通过例如设计分析或故障注入仿真来完成。这表示某些低层级失效模式（例如触发器X2及其扇入中的单粒子翻转和单粒子瞬态故障）是安全的（例如，因为该位很少被ALU体系结构使用）。

注18：对于导致n>2比特位错误的单一永久性故障，存储器的故障率会被计算，例如考虑存储器布局信息、地址解码器的结构等。

注19：计算大于双位错误的ECC（SM3）覆盖率时，应伴随详细的分析，并考虑每个编码字中的比特位数量（在本示例中为32）和编码比特位数量（在本示例中为7）。根据这些参数，覆盖率可能会高得多。

元器件	子元器件	基础子元器件	非安全相关组件？	失效模式	永久性失效 失效率(FIT)	安全故障数量（见注1）	防止违反安全目标的安全机制	关于违反安全目标的失效模式覆盖率	残余或单点故障失效率/FIT	预防潜伏故障的安全机制	对于潜在失效的失效模式覆盖率	潜在多点故障失效率/FIT	瞬态故障 失效率（FIT）	安全故障数量（见注1）	防止违反安全目标的安全机制	关于违反安全目标的失效模式覆盖率	残余或单点故障失效率/FIT
CPU	寄存器组	寄存器 R0	SR	永久性故障	0.0029	0%	SM1	40%	0.00174	SM1	100%	0.00000	0.032095	0%	SM1	40%	0.01920
				瞬态故障													
		寄存器 R1	SR	永久性故障	0.0029	0%	SM1	40%	0.00174	SM1	100%	0.00000	0.032095	0%	SM1	40%	0.01920
				瞬态故障													
		寄存器 R2	SR	永久性故障	0.0029	0%	SM1	20%	0.00232	SM1	100%	0.00000	0.032095	0%	SM1	10%	0.02880
				瞬态故障													
		寄存器 R3	SR	永久性故障	0.0029	0%	SM1	20%	0.00232	SM1	100%	0.00000	0.032095	0%	SM1	10%	0.02880
				瞬态故障													
	ALU	ALU	SR	永久性故障	0.0348	0%	SM1	20%	0.02784	SM1	100%	0.00000	0.00038	20%	SM1	10%	0.00027
				瞬态故障													
		MUL	SR	永久性故障	0.0290	0%	SM1	20%	0.02320	SM1	100%	0.00000	0.00037	70%	SM1	10%	0.00010
				瞬态故障													
		DIV	SR	永久性故障	0.0232	0%	SM1	20%	0.01856	SM1	100%	0.00000	0.00036	70%	SM1	10%	0.00010
				瞬态故障													
	控制逻辑	流水线	SR	永久性故障	0.0174	0%	SM1	90%	0.00174	SM1	100%	0.00000	0.00103	20%	SM1	90%	0.00008
				瞬态故障													
		序列	SR	永久性故障	0.0406	0%	SM1	90%	0.00406	SM1	100%	0.00000	0.00307	50%	SM1	90%	0.00015
				瞬态故障													
		栈控制	SR	永久性故障	0.0029	0%	SM1	70%	0.00087	SM1	100%	0.00000	0.000325	50%	SM1	40%	0.00010
				瞬态故障													
	载入存储单元	地址生成	SR	永久性故障	0.0174	0%	SM1	60%	0.00696	SM1	100%	0.00000	0.00103	10%	SM1	60%	0.00037
				瞬态故障													
		载入单元	SR	永久性故障	0.0145	0%	SM1	50%	0.00725	SM1	100%	0.00000	0.000345	10%	SM1	50%	0.00016
				瞬态故障													
		存储单元	SR	永久性故障	0.0145	0%	SM1	50%	0.00725	SM1	100%	0.00000	0.000345	10%	SM1	50%	0.00016
				瞬态故障													
	调试	调试内部逻辑	SR	永久性故障	0.0058	20%	none	0%	0.00464	none			0.00017	20%	none	0%	0.00014
				瞬态故障													
		调试接口	NSR	永久性故障	0.0783								0.001635				
				瞬态故障													
				Σ					0.11049			0.00000					0.09764
				总失效率	0.29000					总失效率			0.13708				
				总安全相关失效率	0.21170					总安全相关失效率			0.13545				
				总非安全相关失效率	0.07830					总非安全相关失效率			0.00164				
				单点故障度量	47.8%								单点故障度量				27.91%
				潜伏故障度量	100.0%												
易失性存储器	RAM (16KB)	随机存储数据位	SR	永久性故障	1.5000	0%	SM3	96.9%	0.04688	SM3	100%	0.00000	131.072	0%	SM3	99.69%	0.40894
				瞬态故障													
		地址解码器	SR	永久性故障	0.0087	0%	none	0%	0.00870				0.000335	0%	none	0%	0.00034
				瞬态故障													
		测试/冗余	SR	永久性故障	0.0058	50%	none	0%	0.00290				0.00033	90%	none	0%	0.00003
				瞬态故障													
				Σ					0.05848			0.00000					0.40931
				总失效率	1.51450					总失效率			131.07				
				总安全相关失效率	1.51450					总安全相关失效率			131.07				
				总非安全相关失效率	0.00000					总非安全相关失效率			0.00				
				单点故障度量	96.1%								单点故障度量				99.69%
				潜伏故障度量	100.0%												
安全机制	SM1	探测逻辑	SR	永久性故障	0.0029	0%				SM2	90%	0.00029					
				瞬态故障									0.000195				
		预警生成	SR	永久性故障	0.0029	50%				SM2	90%	0.00015	0.000055				
				瞬态故障													
	SM3	EDC 编码器	SR	永久性故障	0.0029	0%	SM3	90%	0.00029	SM4	90%	0.00026	0.000325	0%	none	0%	0.00033
				瞬态故障													
		EDC 解码器	SR	永久性故障	0.0029	0%	SM3	90%	0.00029	SM4	90%	0.00026	0.000325	0%	none	0%	0.00033
				瞬态故障													
		预警生成	SR	永久性故障	0.0029	50%				SM4	90%	0.00015					
				瞬态故障													
		随机存储EDC位	SR	永久性故障	0.328125	0%	SM3	96.9%	0.01025	SM4	90%	0.03179	28.6720	0%	SM3	99.69%	0.08946
				瞬态故障													
				Σ					0.01683			0.03289					0.09011
				总失效率	0.34263					总失效率			28.67281				
				总安全相关失效率	0.34263					总安全相关失效率			28.67281				
				总非安全相关失效率	0.00000					总非安全相关失效率			0.00000				
				单点故障度量	96.8%								单点故障度量				99.69%
				潜伏故障度量	90.1%												

图C.1 量化分析示例（子元器件等级）

					永久性失效								瞬态失效				
元器件	子元器件	基础子元器件	非安全相关组件/安全相关组件	失效模式	失效率(FIT)	安全故障数量	防止违反安全目标的安全机制	关于违反安全目标的失效模式覆盖率	残余或单点故障失效率	预防潜伏故障的安全机制	对于潜在失效的失效模式覆盖率	潜在多点故障失效率/FIT	失效率(FIT)	安全故障数量	防止违反安全目标的安全机制	关于违反安全目标的失效模式覆盖率	残余单点故障失效率
CPU	ALU	ALU	SR	触发器X1及其相关输入永久性故障	0.0100	0%	SM1	20%	0.00800	SM1	100%	0.00000					
				触发器X1及其相关输入中的SEU和SET									0.0001	0%	SM1	10%	0.00009
				触发器X2及其相关输入永久性故障	0.0150	0%	SM1	20%	0.01200	SM1	100%	0.00000					
				触发器X2及其相关输入中的SEU和SET									0.0001	70%	none	0%	0.00003
				等等。。。。。。	...	..	..	..	..	..	..	..	..	..	..	..	..
				Σ	0.0348				0.02784			0.00000	0.00038				0.00027
易失性存储器	RAM (16KB)	随机存储数据位	SR	在同一编码字中引起≤2位错误的永久性故障	1.3500	0%	SM3	100.0%	0.00000	SM3	100%	0.00000					
				在同一编码字中SEU≤2									129.76128	0%	SM3	100.00%	0.00000
				在同一编码字中引起≥2位错误的永久性故障	0.1500	0%	SM3	68.8%	0.04688	SM3	100%	0.00000					
				在同一编码字中SEU≥2									1.31072	0%	SM3	68.8%	0.40894
				Σ	1.5000				0.04688			0.00000	131.0720				0.40894

图C.2 量化分析示例（低层级失效等级）

附 录 D
（资料性）
模拟组件的定量分析示例

D.1 描述

以下是使用5.2.3中描述的方法进行定量分析的示例，以便计算分配给图D.1中所示的混合信号硬件要素的给定安全要求的单点故障度量和潜伏故障度量。

该示例包含混合信号硬件要素，包括：

——低压差稳压器（图 D.2 中的低压差稳压器），提供规定范围内的输出电压；

——电压监控器（图 D.3 中的电压监控器）能够通过监控调节电压 VA 并将其与两个预定阈值进行比较，检测 LDO 输出上的过压（VA〉OVth）和欠压（VA〈UVth）；预定阈值由独立带隙（图 D.3 中的电压带隙 2）提供的参考电压产生，以确保相对于电压调节器的独立性；

——通过数字系统控制的模拟 BIST（数字控制器未在图 D.1 的框图中描述）；及

——一个 ADC 通道。

ASIL B安全要求是：“调节电压输出不会超出调节范围，即调节电压VA不超过UV_{th}-OV_{th}范围大于1 ms。”

当检测到超出调节范围并且向系统/相关项的外部要素发信号通知时，可以将该组件视为处于安全状态。 外部系统负责故障响应，包括将系统或要素转换到安全状态。

如图D.3所示，电压监控器由两个电压比较器、一个无源网络和一个带隙组成; 如图D.2所示，低压差稳压器包括一个带隙，一个限流器，一个偏置发生器和一个稳压器内核。

ADC包含在混合信号硬件要素中，但它不用于与安全要求相关的任何功能，因此其潜在的失效不会导致违反此类要求; 因此假设ADC非安全相关。

考虑以下安全机制：

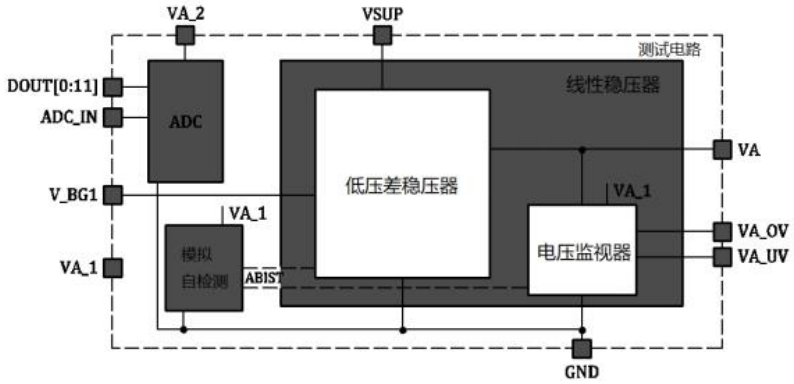
——电压监控器探测过压（安全机制 SM2）和欠压（安全机制 SM1）失效，诊断覆盖率为 99.9%。安全机制在 5.2.4.2 中描述。

——模拟 BIST 探测影响电压监控器的失效，诊断覆盖率为 60%（安全机制 SM6）。 安全机制在 5.2.4.10 中描述。

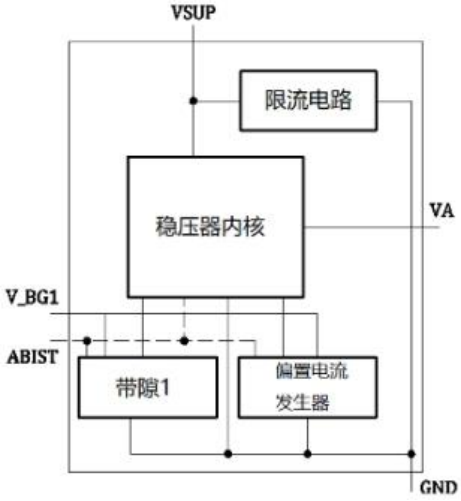
安全机制所声称的覆盖率见表D.1。 假设他们已通过仿真、表征测试和硅的特性确认测试被证实，并在产品安全档案中文档化相关证据。本示例不提供这些证据。

每个安全机制向系统/相关项的要素发出故障探测信号，然后该外部要素负责将系统或要素转换到安全状态。

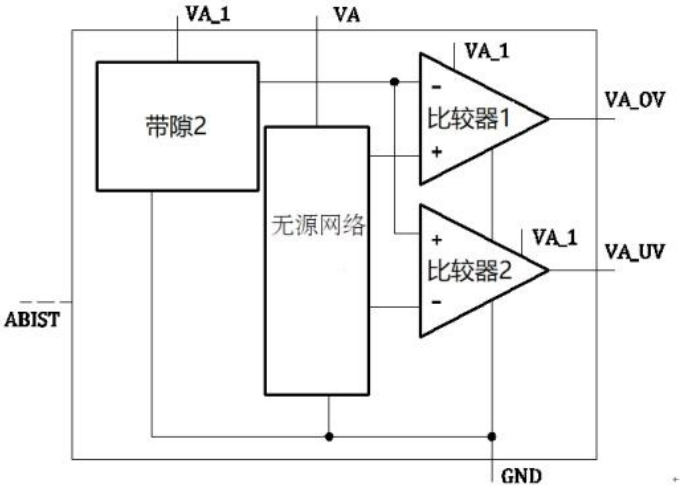
在此假设下，根据GB/T 34590.5-XXXX，附录E中的示例，声称与低压差稳压器相关的潜伏失效的失效模式覆盖率为100%。



图D.1 模拟和混合信号硬件要素示例（分析电路）



图D.2 低压差稳压器元器件的详细框图



图D.3 电压监控器元器件的详细框图

表D.1 示例中考虑的安全机制和对硬件要素的相关覆盖率

编号	安全机制	声称的失效模式覆盖率

SM1	欠压（UV）监控	99.9 %
SM2	过压（OV）监控	99.9 %
SM6	模拟BIST诊断	60 %

注1：该示例表明，在某种方式上可以容易地隔离和禁用的元器件，它们可被没有风险的认为是与安全无关的，可与安全相关的元器件共存。

注2：安全机制的有效性可能受到相关失效的影响。 应如5.2.3.6所述考虑采取适当措施。

根据5.2.3中提供的指南，可以通过以下方式计算模拟和混合信号硬件要素的失效率和度量：

——首先，硬件要素分为元器件或子元器件;

注3：在相关失效分析期间，确定了特定元器件独立性假设的有效性。

注4：必要的详细程度（例如，在元器件层级或子元器件层级进行分析）可取决于分析阶段和安全机制。

——第二，每个元器件或子元器件的失效率可以使用 4.6.2.4 和 5.2.3.3 中描述的方法之一计算;

注5：在此示例中，假设使用表D.6中报告的值，永久性故障和瞬态故障的失效率分布与面积成正比。

——对于每个元器件/子元器件，列出相关的失效模式，并为每个元器件/子元器件分配失效模式分布；

注6：表D.2和表D.3的示例中的失效模式分布，是在每个元器件/子元器件的失效模式中均等分布的。 该假设仅供参考，对于特定示例有效。

——通过将故障分为安全故障，残余故障，探测到的双点故障和潜伏的双点故障，完成评估；

——最后，确定关于该元器件或子元器件的残余和潜伏故障的失效模式覆盖率。

注7：此示例中使用的数字（例如失效率，安全故障数量和失效模式覆盖率）可能因架构而异。

在表D.2和表D.3中给出了定量分析的示例，使用了与图C.1相同的格式，仅限于永久性故障。定量分析给出了失效模式在子元器件层面的概览。

注8：在本示例中，不包含对瞬态故障的单独分析，但可以在与瞬态故障相关时进行添加。

根据系统功能和安全要求，不同的操作阶段可能是安全相关的，因此可以考虑更多的失效模式。

示例：对于需要符合启停要求的系统，稳压器启动阶段可能是安全相关的，可以添加失效模式“启动时间不正确（即超出预期范围） - 电压上升太快”。

表D.2 定量分析示例 -- 任务元器件

元器件	子元器件	安全相关组件或非安全相关组件	失效模式	在缺少安全机制时，失效模式对IC级别的潜在影响[a]	故障模型[b]	失效分布	失效率（FIT）	安全故障数量	预防违反安全要求的安全机制	对于违反安全要求的失效模式覆盖率	残余或单点故障失效率/FIT	预防潜伏故障的安全机制	对于潜伏失效的失效模式覆盖率	潜伏多点故障失效率/FIT
线性稳压器	低压差稳压器	SR	输出电压高于规定范围的高阈值（即过压 - OV）	调节电压高于VA_OV	P	14 %	2.16E-03	0 %	SM2	99.9 %	2.16E-06	SM2	100 %	0.0E+00
		SR	输出电压低于规定范围的低阈值（即欠压 - UV）	调节电压低于VA_UV	P	14 %	2.16E-03	0 %	SM1	99.9 %	2.16E-06	SM1	100 %	0.0E+00
		SR	输出电压受尖峰影响	调节电压超出预期范围（VA_UV-VA_OV）	P	14 %	2.16E-03	0 %	SM1、SM2	99.9 %	2.16E-06	SM1、SM2	100 %	0.0E+00
		SR	输出电压在规定范围内振荡	无影响 - 调节电压在预期范围内但精度较低	P	14 %	2.16E-03	100 %			0.0E+00			0.0E+00
		SR	输出电压快速振荡超出规定范围但平均值在规定范围内	调节电压超出预期范围（VA_UV-VA_OV）	P	14 %	2.16E-03	0 %	SM1、SM2	99.9 %	2.16E-06	SM1、SM2	100 %	0.0E+00
		SR	输出电压在规定范围内漂移	无影响 - 调节电压在预期范围内但精度较低	P	14 %	2.16E-03	100 %			0.0E+00			0.0E+00

		SR	启动时间不正确（即超出预期范围）－电压上升过快	无影响－假设在电压调节器启动期间相关项处于安全状态	P	0 %	0.00E+00	100 %			0.00E+00			0.0E+00
		SR	启动时间不正确（即超出预期范围）－电压上升太慢	无影响－假设在电压调节器启动期间相关项处于安全状态	P	0 %	0.00E+00	100 %			0.00E+00			0.0E+00
		SR	静态电流（即由调节器消耗的电流以控制其内部电路正常工作）超过最大值	调节电压可能具有低精度或超出调节范围,具体取决于实际的静态电流	P	14 %	2.16E-03	50 %			1.08E-03	SM1、SM2	100 %	0.0E+00
ADC	ADC	NSR			P	100 %	7.00E-03				0.0E+00			0.0E+00
							总计				0.00109			0.0E+00
							总失效率 0.0221							
							总安全相关 0.0151							
							总非安全相关0.0070							
											单点故障度量 92.8%			
											潜伏故障度量 100%			

[a] 根据复杂性，最好在 FMEA 中有一个专项提供每种失效模式的潜在根本原因和最终影响的更多详细信息。

[b] 故障模型可以是永久性故障（P）或瞬态故障（T）；本示例仅限于永久性故障。

表D.3 定量分析示例－安全机制

元器件	子元器件	安全相关组件或非安全相关组件	失效模式	在缺少安全机制时，失效模式对IC级别的潜在影响[a]	故障模型[b]	失效分布	失效率（FIT）	安全故障数量	预防违反安全要求的安全机制	对于违反安全要求的失效模式覆盖率	残余或单点故障失效率/FIT	预防潜伏故障的安全机制	对于潜伏失效的失效模式覆盖率	潜伏多点故障失效率/FIT
线性稳压器	电压监控器（SM1，SM2）	SR	UV 监控器（SM1）错误地触发 UV 事件	在标称稳压器负载情况下错误的关闭。	P	25 %	1.45E-03	100 %						0.0E+00
		SR	UV 监控器（SM1）不会触发有效的 UV 事件	调节电压低于 VA_UV	P	25 %	1.45E-03	0 %				SM6	60 %	5.80E-04
		SR	OV 监控器（SM2）错误地触发 OV 事件	在标称稳压器负载情况下错误的关闭。	P	25 %	1.45E-03	100 %						0.0E+00
		SR	OV 监控器（SM2）不触发有效的 OV 事件	调节电压高于 VA_OV	P	25 %	1.45E-03	0 %				SM6	60 %	5.80E-04
模拟 BIST	模拟 BIST	SR	模拟 BIST（SM6）错误地探测线性稳压器的异常行为	无影响[c]	P	50 %	3.50E-03	100 %						0.0E+00

	(SM6)	SR	模拟 BIST（SM6）不检测线性稳压器的不良行为	无影响[c]	P	50 %	3.50E-03	100 %						0.0E+00
总计														1.16E-03
总失效率 0.01280 总安全相关 0.01280 总非安全相关0.00000 单点故障度量 100% 潜伏故障度量 90.0%														

[a] 根据复杂性，最好在 FMEA 中有一个专项提供每种失效模式的潜在根本原因和最终影响的更多详细信息。

[b] 故障模型可以是永久性故障（P）或瞬态故障（T）；本示例仅限于永久性故障。

[c] 在成为安全相关之前需要两个以上的故障：＃1故障：主安全机制（SM1，SM2或SM6）失效，＃2故障：LDO超出规定，＃3故障：BIST诊断失效。

将表D.2和表D.3的结果结合在一起，总体值为：

——单点故障度量 = 96.1 %；及

——潜伏故障度量 = 95.7 %。

对具有更严格安全要求的相同硬件要素，以下示例考虑了更精细的子元器件颗粒度的好处：“调节电压的精度和稳定性是VA <VAO +Δ和VA> VAO-Δ，其中VAO在Vmin至Vmax内，并且Δ= 5mV。”

当检测到低精度/低稳定性状态，并将信号发送到系统/相关项的外部要素时，可以将该组件视为安全状态。外部系统负责故障响应，包括将系统或要素转换到安全状态。

表D.5使用与图C.1相同的格式报告了永久性故障的定量分析示例。 分析中考虑的安全机制是：

——电压监控器探测过压（安全机制 SM2）和欠压（安全机制 SM1）失效；

——独立 ADC 通道检测调节电压的变化高于 Δ= 5mV（安全机制 SM3）。 安全机制在 5.2.4.11 中描述；

——限流器探测由低压差稳压器供电的电路失效影响 （安全机构 SM5）。 安全机制在 5.2.4.5 中描述；及

——模拟 BIST 探测影响电压监控器的失效。

注9：需要提供证据表明限流器相对于调节器内核的独立性。

注10：假定用作安全机制SM3的ADC在所分析的硬件要素外部，因此在FMEA中不予考虑。 硬件要素中包含一个不是SM3的ADC：因此它在FMEA中报告为不与安全相关。

安全机制所声称的覆盖率见表D.4。

表D.4 新安全要求的示例中考虑的安全机制

编号	安全机制	生成的失效模式覆盖率
SM1	欠压（UV）监控器	99.9 %
SM2	过压（OV）监控器	99.9 %
SM3	独立的ADC监控	97 %
SM5	限流器	98 %
SM6	模拟BIST诊断	90 %

注11：安全机制的有效性可能受到相关失效的影响。 如5.2.3.6所述，考虑采取适当措施。

此外，每个安全机制向系统/相关项的外部要素发出故障探测信号，然后该外部要素负责将系统或要素转换到安全状态。

根据这一假设，按照GB/T 34590.5-XXXX，附件E，声称与任务电路相关的潜伏故障的失效模式覆盖率为100％。

表D.5显示了以比表D.2和表D.3中更精细的颗粒度进行的任务元器件的定量分析。 这些示例表明，不同的安全要求会影响一个或多个安全机制的层级划分和诊断覆盖率要求。

注12：在此示例中，不包含瞬态故障的分析，但可以在相关时添加。

表D.5 定量分析示例（按精细的颗粒度进行）- 任务元器件

元器件	子元器件	安全相关组件或非安全相关组件	失效模式	在缺少安全机制时，失效模式对IC级别的潜在影响[a]	故障模型[b]	失效分布	失效率（FIT）	安全故障数量	预防违反安全要求的安全机制	对于违反安全要求的失效模式覆盖率	残余或单点故障失效率/FIT	预防潜伏故障的安全机制	对于潜伏失效的失效模式覆盖率	潜伏多点故障失效率/FIT
低压差稳压器	稳压器内核	SR	输出电压高于规定范围的高阈值(即过压 - OV)	调节电压高于VA_OV	P	14%	1.49E-03	0%	SM2	99.9%	1.49E-06	SM2	100%	0.00E+00
		SR	输出电压低于规定范围的低阈值（即欠压 - UV）	调节电压低于VA_UV	P	14%	1.49E-03	0%	SM1	99.9%	1.49E-06	SM1	100%	0.00E+00
		SR	输出电压受尖峰影响	调节电压超出预期范围（VA_UV-VA_OV）	P	14%	1.49E-03	0%	SM1 SM2	99.9%	1.49E-06	SM1、SM2	100%	0.00E+00
		SR	输出电压在规定范围内振荡	调节电压在预期范围内但精度较低	P	14%	1.49E-03	0%	SM3	97.0%	4.46E-05	SM3	100%	0.00E+00
		SR	输出电压快速振荡超出规定范围但平均值在规定范围内	调节电压在预期范围内但精度较低	P	14%	1.49E-03	0%	SM1、SM2	99.9%	1.49E-06	SM1、SM2	10 %	0.00E+00

		SR	输出电压在规定范围内漂移	调节电压在预期范围内但精度较低	P	14%	1.49E-03	0%	SM3	97.0%	4.46E-05	SM3	100%	0.00E+00
		SR	静态电流(即由调节器吸取的电流以控制其内部电路以便正常工作）超过最大值	调节电压可能精度低,具体取决于实际的静态电流	P	14%	1.49E-03	50%	SM3	97.0%	2.23E-05	SM3	100%	0.00E+00
	带隙1	SR	输出卡滞（高或低）	调节电压超出预期范围（VA_UV-VA_OV）	P	20%	6.00E-04	0 %	SM1、SM2	99.9%	6.00E-07	SM1、SM2	100%	0.00E+00
		SR	输出是浮动的(例如开路）	调节电压超出预期范围（VA_UV-VA_OV）	P	20%	6.00E-04	0%	SM1、SM2	99.9%	6.00E-07	SM1、SM2	100%	0.00E+00
		SR	输出电压在预期范围内振荡	调节电压在预期范围内但精度较低	P	20%	6.00E-04	0%	SM3	97.0%	1.80E-05	SM3	100 %	0.00E+00
		SR	输出电压值不正确（即超出预期范围）	调节电压超出预期范围（VA_UV-VA_OV）	P	20%	6.00E-04	0%	SM1、SM2	99.9%	6.00E-07	SM1、SM2	100%	0.00E+00
		SR	输出电压精度太低，包括漂移	调节电压在预期范围内但精度较低	P	20%	6.00E-04	50%	SM3	97.0%	9.00E-06	SM3	100%	0.00E+00
		SR	输出电压受尖峰影响	由于电路实现方式而不适用	P	0%	0.00E+00	0%			0.00E+00	SM1、SM2	100%	0.00E+00

	偏置电流发生器	SR	一个或多个输出卡滞（高或低）	调节电压超出预期范围（VA_UV-VA_OV）	P	10%	2.00E-05	0%	SM1、SM2	99.9%	2.00E-08	SM1、SM2	100%	0.00E+00
		SR	一个或多个输出是浮动的（即开路）	调节电压超出预期范围（VA_UV-VA_OV）	P	10%	2.00E-05	0%	SM1、SM2	99.9%	2.00E-08	SM1、SM2	100%	0.00E+00
		SR	参考电流不正确（即在预期范围之外）	调节电压超出预期范围（VA_UV-VA_OV）	P	10%	2.00E-05	0%	SM1、SM2	99.9%	2.00E-08	SM1、SM2	100%	0.00E+00
		SR	参考电流精度太低，包括漂移	调节电压在预期范围内但精度较低	P	10%	2.00E-05	0%	SM3	97.0%	6.00E-07	SM3	100%	0.00E+00
		SR	参考电流受尖峰影响	如果没有滤除尖峰，则在有限的时间内调节电压精度低； 否则没有影响	P	10%	2.00E-05	50%			1.00E-05	SM1、SM2	100%	0.00E+00
		SR	参考电流在预期范围内振荡	调节电压在预期范围内但精度较低	P	10%	2.00E-05	0%	SM3	97.0%	6.00E-07	SM3	100%	0.00E+00
		SR	一个或多个偏置电流超出预期范围，而参考电流是正确的	调节电压精度低或失调	P	10%	2.00E-05	0 %	SM3	97.0%	6.00E-07	SM3	100%	0.00E+00

		SR	一个或多个偏置电流精度太低，包括漂移	调节电压在预期范围内但精度较低	P	10%	2.00E-05	0%	SM3	97.0%	6.00E-07	SM3	100%	0.00E+00
		SR	一个或多个偏置电流受尖峰影响	如果没有滤除尖峰,则在有限的时间内调节电压精度低； 否则没有影响	P	10%	2.00E-05	50%			1.00E-05	SM1、SM2	100%	0.00E+00
		SR	一个或多个偏置电流在预期范围内振荡	调节电压在预期范围内但精度较低	P	10%	2.00E-05	0%	SM3	97.0%	6.00E-07	SM3	100%	0.00E+00
ADC	ADC	NSR			P	100%	7.00E-03				0.0E+00			0.0E+00
总计											0.00017			0.0E+00
总失效率 0.0206 总安全相关 0.0136 总非安全相关0.0070 单点故障度量 98.8% 潜伏故障度量 100%														

[a] 根据复杂性，最好在 FMEA 中有一个专项提供每种失效模式的潜在根本原因和最终影响的更多详细信息。

[b] 故障模型可以是永久性故障（P）或瞬态故障（T）；本示例仅限于永久性故障。

D.1.1 模拟组件失效率的计算示例

模拟和混合信号组件的基础失效率的计算方法在4.6中描述。

基础失效率分配给组成硬件组件的不同要素，可以根据要素的类型使用不同的分配方法。

可以认为基础失效率与电路面积成比例。

示例1：将基础失效率除以组件的总面积，以获得每个相关故障模型的 FIT / mm2。

表D.6 基于面积的基础失效率分配

故障模型	失效率	单位
永久性故障	2.00E-02	FIT/mm^2
瞬态故障	2.00E-05	FIT/mm^2

通过使用表D.6中报告的FIT / mm2计算前一示例中所示的模拟和混合信号组件的每个子元器件的失效率。

考虑到前一示例的框图 ，计算结果列于表D.7中。

表D.7 每个元器件/子元器件的失效率

元器件	子元器件	区域面积(mm^2)	永久性故障失效率（FIT）	瞬态故障失效率（FIT）
低压差稳压器	稳压器内核	0.52	0.0104	0.0000104
	带隙 1	0.15	0.0030	0.0000030
	偏置电流发生器	0.01	0.0002	0.0000002
	限流器	0.075	0.0015	0.0000015
	总和	0.755	0.0151	0.0000151
电压监控器	比较器1	0.03	0.0006	0.0000006
	比较器2	0,03	0,000 6	0,000 000 6
	无源网络	0.08	0.0016	0.0000016

	带隙2	0.15	0.0030	0.0000030
	总和	0,29	0,0058	0.0000058
ADC	ADC	0,85	0.0170	0.0000170
模拟BIST	模拟BIST	0.35	0.0070	0.0000070
总和		2.535	0.0507	0.0000507

注1：此处报告的数字仅为示例。

注2：此处报告的块区域包括内部布线。 如果相关，顶层布线包含在单独的块中。

作为基于区域研究方法的替代方案，如5.1.7.1所示，可以基于每个子元器件或基础子元器件的等效晶体管的数量来估计失效率和失效模式分布。就混合信号或模拟组件而言，在估计等效晶体管的数量时可以考虑有源器件，无源器件和布线之间的区别。 所使用的方法可以基于所分析的电路的布局（或计划的布局），或者基于硬件要素之间是如何共享失效模式的分析来选择。

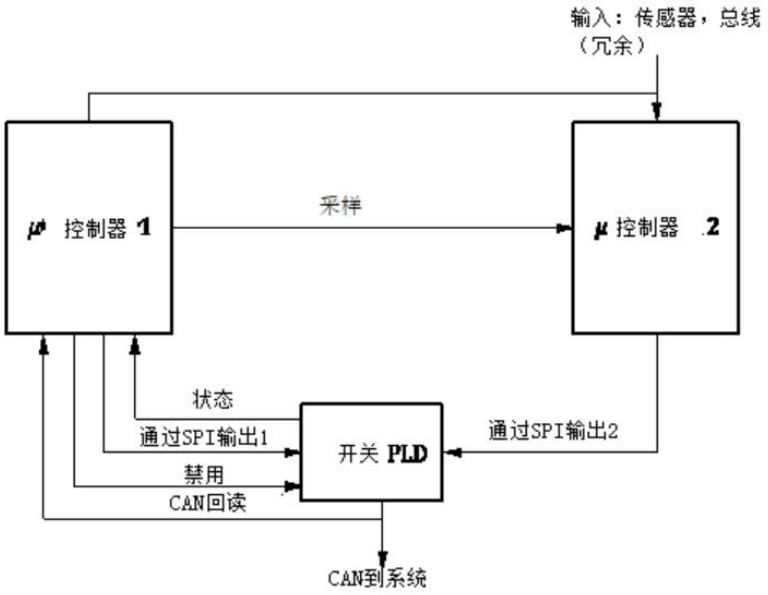
注3：对于瞬态故障模型，与面积成比例的基础失效率分配是一个简化的例子，因为实际上，并非混合信号电路中的每个要素具有相同的失效概率。

示例2：在开关电容架构中，与电路的其他部分相比，保持信号的电容对瞬态故障更敏感，因为它们被用作存储要素。

附 录 E
（资料性）
PLD 组件定量分析示例

E.1 架构示例

图E.1给出了以下示例中采用的系统。该系统用于安全相关应用，使用两个微控制器用于冗余，最终控制输出用可编程逻辑器件PLD实现。两个微控制器将其值通过SPI（串行外设接口）发送到PLD，PLD通过CAN（控制器局域网）总线进行通信。在本示例中，假设计算所得的输出值过高（即超过了正常系统给出的值再加上阈值的总和），是有潜在危害的，但输出值过低，从功能安全角度来看是可以接受的。同时，假设接收CAN消息的组件可以检测到CAN消息的丢失，并采取适当的措施，例如将接收信号默认设为最小值，并且接收模块可以容忍X条CAN消息损坏（如高于预期值）。



图E.1 PLD 使用实例 – 输出开关

注： “控制器”硬件组件用两个微控制器和一个PLD实现。

硬件“控制器”所得的安全要求：

——SafReq_hardware_Comp_Controller_001：“应避免输出连续 x 条包含大于正确值加上阈值的错误值消息”；及

——SafReq_hardware_Comp_Controller_002：“应避免超过 y 毫秒，未检测 CAN 消息输出丢失”。

硬件组件“控制器” 由两个微控制器（微控制器1和微控制器2）加一个PLD实现。微控制器1和微控制器2 在相同时间，有相同的输入/输出，并将其结果传输到PLD。当没有故障发生时，两个输出在阈值范围内一致。PLD从两个信号取更小值，通过CAN通讯，将该输出发送给系统的其余部分。SafReq_hardware_Comp_Controller_002可由控制器外部实体完成（例如超时监控）。

PLD 所得的安全要求可能是：

——SafReq_PLD_001: “应避免输出大于由微控制器 1 和微控制器 2 输入的较小值的值”（由安全要求 SafReq_hardware_Comp_Controller_001 得到）；及

——SafReq_PLD_002: “应避免 PLD 发送的 CAN 输出值损坏未能检测，而导致过高的值被输出（由安全要求 SafReq_hardware_Comp_Controller_001 得到）。”

以下章节用两种不同方法为例，说明PLD的安全（分析）和相关失效分析（的处理方法）。有关微控制器1和微控制器2的安全分析和相关失效分析，不在本部分（讨论）范围内。

PLD失效通过以下两种方法处理：

——使用 PLD 外部安全措施，或

——使用 PLD 内部安全措施。PLD 含有检测 PLD 故障的诊断措施。这些故障通过状态信号传递给微控制器 1，后者可以根据故障的严重程度，（决定是否）停止 PLD 运行。

E.2 PLD 外部措施

以下安全机制，由PLD以外的要素实施：

——SafMech_PLD_001: CAN 消息读回和比对。 PLD 的 CAN 消息输出，由微控制器 1 读回。微控制器 1 检查 PLD 的输出值，是否等于或小于它的输出。如果检查结果不正确，则微控制器 1 通过禁用信号，停止 PLD 运行；及

——SafMech_Network_001：接收器（端）实现超时监控。

在安全分析的第一步，可以确定相关失效模式。由于PLD内未实现任何安全机制，因此对PLD输出端可观测的失效模式，进行描述就足够了：

——FM_PLD_OP_01：无输出；

——FM_PLD_OP_02：输出旧的消息；

——FM_PLD_OP_03：输出损坏；

——FM_PLD_OP_04：未输出最小值；

——FM_PLD_OP_05：一直输出微控制器 1 的值；

——FM_PLD_OP_06：一直输出微控制器 2 的值；及

——FM_PLD_OP_07：主动“禁用”离散信号并不阻止 CAN 消息传输。

如5.3.3.1.3所述，为了得出上述失效模式的概率分布，通常需要详细了解PLD的内部结构。如果该信息无法提供，也无法给出依据，说明其中某失效模式比其他失效模式更可能发生，则可采用5.3.3.1.3 a） 所述的方法，如下表E.1所示。

表E.1 采用 PLD 外部措施时，PLD 安全分析实例

失效模式	永久分布	瞬态分布	PVSG	MPF	安全机制
FM_PLD_OP_01:无输出	14.3 %	14.3 %	1	0	SafMech_Network_001
FM_PLD_OP_02:输出旧的消息	14.3 %	14.3 %	1	0	SafMech_PLD_001
FM_PLD_OP_03:输出损坏	14.3 %	14.3 %	1	0	SafMech_PLD_001
FM_PLD_OP_04:未输出最小值	14.3 %	14.3 %	0	1	SafMech_PLD_001

FM_PLD_OP_05:一直输出微控制器1的值	14.3 %	14.3 %	0	1	
FM_PLD_OP_06:一直输出微控制器2的值	14.3 %	14.3 %	0	1	SafMech_PLD_001
FM_PLD_OP_07:主动“禁用”离散信号并不阻止 CAN消息传输	14.3 %	14.3 %	0	1	
注：PVSG=直接违反安全目标的可能性；MPF=多点失效					

就相关失效分析 （不在本部分范围内）而言，需要考虑以下要素间的相关性：

——PLD 和微控制器 1；

——PLD 和微控制器 2；

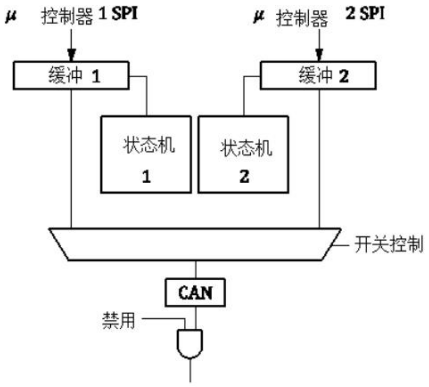
——微控制器 1 和微控制器 2。

E.3 PLD 内部措施

示例其余部分考虑使用PLD内部的安全措施。PLD的内部架构如图E.2所示。从微控制器发送的数据，在通过CAN总线发出之前先被缓冲。缓冲区被实现为用户存储器，从而状态机可以控制缓冲区的运行，多路选择器用逻辑块来实现，CAN模块则是一个固定功能的IP。逻辑块的功能及块和存储器之间的（信号）路由通过配置技术控制。为简单起见，（通过）开关控制逻辑决定是发送缓冲区1的数据，还是缓冲区2的数据，此示例就不做讨论了。

该设计也容易受到间歇性和永久性硬件失效的影响。所有芯片基础设施，如时钟或电源，都可能是共因失效的根源。这些失效可以通过探测和报告单模失效的冗余（机理）得到处理。其他示例，还包括初始化时不正确的代码加载和存储器中的位翻转。这些可以通过校验和，和奇偶校验来探测；但是，其中一些失效（仍）可能导致违反安全目标，并带来不可接受的危险。错误-探测-纠正码（ECC）是一种很好技术，因为可以纠正错误，并在纠错后报告芯片中存在的潜在问题。芯片I/O中的单个失效只影响某一路输出，存在的风险较小。

注1：根据所实现电路的功能，除了修复故障以恢复所设计的功能（例如，配置技术中的故障，即使配置技术中的故障被修复，仍导致状态机进入不可恢复状态），还必须有进一步的措施（活动）。



图E.2 PLD 架构

如果在内部安全机制未检测到故障，导致可能违反安全目标，通过微控制器1检测CAN信号丢失，或SPI输出与CAN读回之间不匹配。假如微控制器1可以通过“禁用”信号关闭PLD功能，仍然是可以接受的。进行相关失效分析，以确保PLD违反安全目标且通过禁用信号关闭功能的失效同时发生的风险足够低。

示例：如果开关无法响应来自微控制器 1 的禁用命令，则可能发生潜在危害。就像微控制器 1 和微控制器 2 都还好；而 PLD 输出仍然代表安全值，是一种多点故障的情况。只有当其中一个微控制器故障，且 PLD 响应错误时，才有潜在风险。为了检测多点故障，可以对禁用逻辑进行定期测试。由于该操作在系统或要素层面执行，而具体细节不在本部分范畴内，所以不作进一步描述。

注2：在这个简单的例子中，外部措施可以取代内部安全机制。通常，存在需要采取内部措施以达到目标诊断覆盖率的情况，因此，可以采用本条所述的内部安全机制的详细分析。

随机硬件故障可在设计中，采用归纳故障分析法（如 FMEA）进行分析。同时考虑用户设计的缺陷，及PLD技术缺陷，并考虑永久性故障和瞬态故障。在设计的定性分析之后进行定量分析，就像本部分附录C中所述的分析。

如5.3.3.1所述，可由PLD制造商提供的 PLD基础子元器件的故障率，和失效模式分布，作为定量分析输入（条件）。

注3：在考虑PLD内部措施的情况下，如何确定失效模式分布，最好采用5.3.3.1.3 b）或c）中所述的方法。

表 E.2 为以上设计的定量分析提供了一个框架，可以用类似于图C.1的信息作补充。

注4：如5.1.7所讨论，根据设计所处的分析阶段和所用的安全机制，再决定分析所要的详细程度。

表E.2 场景 2 定量分析的模板框架示例

元器件	子元器件	安全有关（SR）或安全不相关（NSR）要素	失效模式
I/O 接口	IO 缓冲器	SR	永久
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
缓冲器1	随机存储数据位	SR	永久
			瞬态
	地址解码器	SR	永久
			瞬态
	测试/冗余	SR	永久
			瞬态
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
缓冲器2	随机存储数据位	SR	永久

			瞬态
	地址解码器	SR	永久
			瞬态
	测试/冗余	SR	永久
			瞬态
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
状态机 1	逻辑块	SR	永久
			瞬态
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
状态机 2	逻辑块	SR	永久
			瞬态
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
多路转换开关	逻辑块	SR	永久
			瞬态
	配置技术	SR	永久
			瞬态
	（信号）路由资源	SR	永久
			瞬态
控制器局域网络	逻辑	SR	永久
			瞬态
	随机存储数据位	SR	永久
			瞬态
	地址解码器	SR	永久
			瞬态
注1：根据系统中每个PLD元器件的作用，可以进行更详细的分析。 **注2：**为了简单起见，本示例没有列出量化数字。			

分析还包括与PLD相关的外部组件，如电源、时钟和复位电路。此外，如果PLD的配置由外部器件加载，还要分析配置加载到PLD是否看作安全相关，或者配置加载过程是否会导致相关项失效。

特别是，如果PLD由微控制器1加载，则需要考虑微控制器1的加载机理，和影响微控制器1功能的共因失效。如果在PLD内实现了独立通道或诊断措施，则需要做相关失效分析。本部分的附录B可找到该分析的例子。在此示例中，由于已通过微控制器将CAN模块的输出读回，来探测PLD的故障，因此没有考虑PLD各子元器件的独立性。

参 考 文 献

[1] Askari S., Nourani M. Design methodology for mitigating transient errors in analogue and mixed-signal circuits. Circuits, Devices & Systems [online]. IET. November 2012, 6(6), 447-456 [viewed 2017-10-10]. Available at: 10.1049/iet-cds.2012.0053.

[2] Baumann R.C. Radiation-Induced Soft Errors in Advanced Semiconductor Technologies. IEEE Transactions on device and materials reliability [online]. IEEE. December 2005, 5(3), 305-316 [viewed 2017-10-10]. Available at: 10.1109/TDMR.2005.853449.

[3] BARUAH S.K., GOOSSENS J. Rate-monotonic scheduling on uniform multiprocessors. Proceedings of the 23rd International Conference on Distributed Computing Systems [online]. IEEE. May 2003, 360-366 [viewed 2017-10-10]. Available at: 10.1109/ICDCS.2003.1203485.

[4] BÖRCSÖK J., SCHAEFER S., UGLJESA E. Estimation and Evaluation of Common Cause Failures. Second International Conference on Systems [online]. IEEE. April 2007, 41 [viewed 2017-10-10]. Available at: 10.1109/ICONS.2007.25.

[5] BRESSOUD T.C., SCHNEIDER F.B. Hypervisor-based fault tolerance. Proceedings of the fifteenth ACM symposium on Operating systems principles [online]. ACM. December 1995, 1-11 [viewed 2017-10-10]. Available at: 10.1145/224057.224058.

[6] CHATTOPADHYAY S., KEE C.L., ROYCHOUDHURY A., KELTER T., MARWEDEL P., FALK H. A Unified WCET Analysis Framework for Multi-core Platforms. IEEE 18th Real-Time and Embedded Technology and Applications Symposium [online]. IEEE. April 2012, 99-108 [viewed 2017-10-10]. Available at: 10.1109/RTAS.2012.26.

[7] CLEGG J.R. Arguing the safety of FPGAs within safety critical systems. Incorporating the SaRS Annual Conference, 4th IET International Conference on Systems Safety [online]. IET. October 2009, 1-6 [viewed 2017-10-10]. Available at: 10.1049/cp.2009.1569.

[8] CONMY P.M., PYGOTT C., BATE I. VHDL guidance for safe and certifiable FPGA design. 5th IET International Conference on System Safety [online]. IET. October 2010, 1-6 [viewed 2017-10-10]. Available at: 10.1049/cp.2010.0832.

[9] FIDES Guide 2009 Edition A September 2010, Reliability Methodology for Electronic Systems.

[10] Fleming, P.R., Olson, B.D., Holman, W.T., Bhuva, B.L., Massengill, L.W. Design Technique for Mitigation of Soft Errors in Differential Switched-Capacitor Circuits. IEEE Transactions on Circuits and Systems II: Express Briefs [online]. IEEE. May 2008, 55(9), 838-842 [viewed 2017-10-10]. Available at: 10.1109/TCSII.2008.923437.

[11] FRANKLIN M. Incorporating Fault Tolerance in Superscalar Processors. Proceedings of International Conference on High Performance Computing [online]. IEEE. December 1996 [viewed 2017-10-10]. Available at: 10.1109/HIPC.1996.565839.

[12] HAYEK A., BORCSOK J. SRAM-based FPGA design techniques for safety-related systems conforming to IEC 61508 a survey and analysis. 2nd International Conference on Advances in

Computational Tools for Engineering Applications (ACTEA) [online]. IEEE. December 2012, 319-324 [viewed 2017-10-10]. Available at: 10.1109/ICTEA.2012.6462892.

[13] HEISER G. The role of virtualization in embedded systems. Proceedings of the 1st workshop on Isolation and integration in embedded systems [online]. ACM. April 2008, 11-16 [viewed 2017-10-10]. Available at: 10.1145/1435458.1435461.

[14] IEC 61508 2:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems.

[15] IEC 61709:2017, Electrical components — Reliability — Reference conditions for failure rates and stress models for conversion.

[16] JEDEC JEP122H, Failure Mechanisms and Models for Semiconductor Devices.

[17] JEDEC JESD89A, Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices.

[18] Keckler, S.W., Olukotun, K., Hofstee, H.P. Multicore Processors and Systems. 2009. Springer.

[19] Kervarreca, G., et al. A universal field failure based reliability prediction model for SMD Integrated Circuits. Microelectronics Reliability [online]. Elsevier. June-July 1999, 765-771 [viewed 2017-10-10]. Available at: https://doi.org/10.1016/S0026-2714(99)00099-2.

[20] LAZZARI C. ET AL. Phase-Locked Loop Automatic Layout Generation and Transient Fault Injection Analysis: A Case Study. 12th IEEE International On-Line Testing workshop [online]. IEEE. July 2006, 117-127 [viewed 2017-10-10]. Available at: 10.1109/IOLTS.2006.48.

[21] BENSO A., BOSIO A., DI CARLO S., MARIANI R. A Functional Verification based Fault Injection Environment. 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems [online]. IEEE. September 2007 [viewed 2017-10-10]. Available at: 10.1109/DFT.2007.31.

[22] Mariani R. Soft Errors on Digital Components. Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation [online]. Springer. 2003 [viewed 2017-10-10]. Available at: https://doi.org/10.1007/0-306-48711-X_3.

[23] MIL-HDBK-217, Military Handbook — Reliability Prediction of Electronic Equipment.

[24] Mitra, S., Saxena, N.R., Mccluskey, E.J. Common-mode failures in redundant VLSI systems: a survey. IEEE Transactions on Reliability [online]. IEEE. September 2000, 49(3), 285-295 [viewed 2017-10-10]. Available at: 10.1109/24.914545.

[25] MUKHERJEE S.S. ET AL. A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor in microarchitecture. Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture [online]. IEEE. December 2003, 29-40 [viewed 2017-10-10]. Available at: 10.1109/MICRO.2003.1253181.

[26] NIIMI Y. ET AL. Virtualization Technology and Using Virtual CPU in the Context of GB/T 34590: The E-Gas Case Study. SAE Technical Paper [online]. SAE. April 2013 [viewed 2017-10-10]. Available at: https://doi.org/10.4271/2013-01-0196.

[27] PAOLIERI M., MARIANI R. Towards functional-safe timing-dependable real-time architectures. IEEE 17th International On-Line Testing Symposium (IOLTS) [online]. IEEE. July 2011, 31-36 [viewed 2017-10-10]. Available at: 10.1109/IOLTS.2011.5993807.

[28] SINGH M. ET AL. Transient Fault Sensitivity Analysis of Analog-to-Digital Converters (ADCs). Proceedings of the IEEE Workshop on VLSI (WVLSI '01) [online]. IEEE. April 2001 [viewed 2017-10-10]. Available at: 10.1109/IWV.2001.923153.

[29] WHITE M., BERNSTEIN J.B. Microelectronics Reliability: Physics-of-Failure Based Modeling and Lifetime Evaluation. JPL Publication [online]. February 2008 [viewed 2017-10-10]. Available at: http://www.acceleratedreliabilitysolutions.com/images/_NASA_Physics_of_Failure_for_Microelectronics.pdf.

[30] Arlat J., et al. Fault Injection and Dependability Evaluation of Fault-Tolerant Systems. IEEE Transactions on Computers [online]. IEEE. August 1993, 42(8), 913 [viewed 2017-10-10]. Available at: 10.1109/12.238482.

[31] Benso A. and Prinetto P. Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation. Springer. 2003 [viewed 2017-10-10]. Available at: https://doi.org/10.1007/0-306-48711-X_3.

[32] Wei Jiesheng, et al. Quantifying the accuracy of high-level fault injection techniques for hardware faults. Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference [online]. IEEE. June 2014 [viewed 2017-10-10]. Available at: 10.1109/DSN.2014.2.

[33] Kejun Wu, Pahlevanzadeh H., Peng Liu, Qiaoyan Yu. A new fault injection method for evaluation of combining SEU and SET effects on circuit reliability. Circuits and Systems (ISCAS), 2014 IEEE International Symposium on [online]. IEEE. June 2014, 602,605 [viewed 2017-10-10]. Available at: 10.1109/ISCAS.2014.6865207.

[34] Van De Goor A.J. Testing Semiconductor Memories, Theory and Practice, 2nd. ComTex Publishing.

[35] ENAMUL AMYEEN M., et al. Evaluation of the Quality of N-Detect Scan ATPG Patterns on a Processor. Proceedings of the International Test Conference 2004, ITC'04 [online]. IEEE. October 2004, 669-678 [viewed 2017-10-10]. Available at: 10.1109/TEST.2004.1387328.

[36] BENWARE B., et al. Impact of Multiple-Detect Test Patterns on Product Quality, Proc. of the International Test Conference 2003, ITC'03 [online]. IEEE. October 2003, 1031-1040 [viewed 2017-10-10]. Available at: 10.1109/TEST.2003.1271091.

[37] PATEL J.H. Stuck-At Fault: A Fault Model for the Next Millennium? Proceedings of the International Test Conference 1998, ITC'98 [online]. IEEE. August 1988, 1166 [viewed 2017-10-10]. Available at: 10.1109/TEST.1998.743358.

[38] SN 29500:2004, Siemens AG, "Failure Rates of Components — Expected Values, General".

[39] Paschalis A., and Gizopoulos D. Effective Software-Based Self-Test Strategies for On-Line Periodic Testing of Embedded Processors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [online]. IEEE. December 2004, 88-99[viewed 2017-10-10]. Available at: 10.1109/TCAD.2004.839486.

[40] IEC/TR 62380:2004, Reliability data handbook — Universal model for reliability prediction of electronics components, PCBs and equipment.

[41] ITRS 2009, The International Technology Roadmap For Semiconductors (ITRS), 2009 Edition.

[42] IEEE STD 2700-2014, IEEE Standard for Sensor Performance Parameter Definitions.

[43] White Richard M. A Sensor Classification Scheme. IEEE Transactions On Ultrasonics, Ferroelectrics, And Frequency Control [online]. IEEE. March 1987, 34(2), 124-126 [viewed 2017-10-10]. Available at: 10.1109/T-UFFC.1987.26922.

[44] Gupta Vijay, R. Snow, M.C. Wu, A. Jain, J. Tsai. Recovery of Stiction-Failed MEMS Structures Using Laser-Induced Stress Waves. Journal of Microelectromechanical Systems [online]. IEEE. August 2004, 13(4), 696-700 [viewed 2017-10-10]. Available at: 10.1109/JMEMS.2004.832185.

[45] Walraven Jeremy A. Failure Mechanisms in MEMS. IEEE ITC International Test Conference [online]. IEEE. October 2003, 828-833 [viewed 2017-10-10]. Available at: 10.1109/TEST.2003.1270915.

[46] J. Iannacci. Reliability of MEMS: A perspective on failure mechanisms, improvement solutions and best practices at development level. Elsevier Displays [online]. Elsevier. April 2015, 37, 62-71 [viewed 2017-10-10]. Available at: https://doi.org/10.1016/j.displa.2014.08.003.

[47] Vonkyoung Kim, Chen T. Assessing SRAM test coverage for sub-micron CMOS technologies. VLSI Test Symposium, 1997, 15th IEEE [online]. IEEE. May 1997, 24-30 [viewed 2017-10-10]. Available at: 10.1109/VTEST.1997.599437

[48] GINEZ O. ET AL. An overview of failure mechanisms in embedded flash memories. VLSI Test Symposium, 2006. Proceedings. 24th [online]. IEEE. April 2006 [viewed 2017-10-10]. Available at: 10.1109/VTS.2006.19.

[49] DI CARLO S., FABIANO M. PIAZZA, ROBERTO; PRINETTO, P. Exploring modeling and testing of NAND flash memories. Design & Test Symposium (EWDTS), 2010 East-West [online]. IEEE. September 2010, 47-50 [viewed 2017-10-10]. Available at: 10.1109/EWDTS.2010.5742059.

[50] Al-Ars, Z.; Hamdioui, S.; Van De Goor, A.J., Space of DRAM Fault Models and Corresponding Testing. Design, Automation and Test in Europe, 2006. DATE '06. IEEE. March 2006, 1, 1-6 [viewed 2017-10-10]. Available at: 10.1109/DATE.2006.244080.

[51] IATF 16949:2016, Quality management system requirements for automotive production and relevant service parts organizations.

[52] Daniel J. Sorin, Mark D. Hill, David A. Wood. A Primer on Memory Consistency and Cache Coherence (1st ed.). Morgan & Claypool Publishers.

[53] JEDEC JESD94, Application Specific Qualification Using Knowledge Based Test Methodolog.

[54] G. Kervarrec, et al. A universal reliability prediction model for SMD integrated circuits based on field failures. European Symposium on Reliability of Electron Devices, Failure Physics and Analysis [online]. Microelectronics Reliability Elsevier. July 1999, 39(6), 765-771 [viewed 2017-10-10]. Available at: https://doi.org/10.1016/S0026-2714(99)00099-2.

[55] E-GAS. Standardized E-GAS Monitoring Concept for Gasoline and Diesel Engine Control Units. [viewed 2017-10-10]. Available at: https://www.iav.com/sites/default/files/attachments/seite//ak-egas-v6-0-en-150922.pdf.

[56] IEEE P1804, IEEE Draft Standard for Fault Accounting and Coverage Reporting to Digital Modules [viewed 2017-10-10].

[57] R. Leveugle, A. Calvez, P. Maistri and P. Vanhauwaert, Statistical fault injection: Quantified error and confidence. 2009 Design, Automation & Test in Europe Conference & Exhibition [online]. IEEE. April 2009, 502-506 [viewed 2017-10-10]. Available at: 10.1109/DATE.2009.5090716.

[58] Philip Mayfield. Understanding Binomial Confidence Intervals [viewed 2017-10-10]. Available at: http://www.sigmazone.com/binomial_confidence_interval.htm.

[59] SAE J1211:201211, Handbook for Robustness Validation of Automotive Electrical/Electronic Modules, SAE.

[60] JEDEC JESD88E, Dictionary of Terms for Solid-State Technology — 6th Edition.

[61] ISO 26262-10:2018, Road vehicles — Functional safety — Part 10: Guideline on ISO 26262.

[62] AEC, AEC-Q100: Failure Mechanism Based Stress Test Qualification For Integrated Circuits.

[63] ISO 26262-2:2018, Road Vehicles — Functional Safety — Part 2: Management of functional safety

[64] ISO 26262-3:2018, Road vehicles — Functional safety — Part 3: Concept phase

[65] ISO 26262-4:2018, Road vehicles — Functional safety — Part 4: Product development at the system level

[66] ISO 26262-5:2018, Road vehicles — Functional safety — Part 5: Product development at the hardware level

[67] ISO 26262-6:2018, Road vehicles — Functional safety — Part 6: Product development at the software level

[68] ISO 26262-7:2018, Road vehicles — Functional safety — Part 7: Production, operation, service and decommissioning ISO 26262-8:2018,

[69] ISO 26262-9:2018, Road vehicles — Functional safety — Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses