



WP477 (v1.0) 2016 年 6 月 14 日

UltraRAM：在 UltraScale+ 器件上集成 嵌入式存储器取得突破性成功

UltraRAM 是 UltraScale+™ 系列中最新的存储器模块，能够实现高达 500Mb 的总片上存储容量，相当于赛灵思 28nm FPGA 片上存储器容量的 6 倍。

摘要

传统的 FPGA 和 SoC 包含的片上存储器以 block RAM 和分布式 RAM 的形式存在。由于器件能以更高数据速率处理更多数据，因此越发需要将数据缓冲或存储在靠近处理器的位置。

UltraScale+ 系列中的新产品是一款容量更大的称为 UltraRAM 的灵活存储器模块。UltraRAM 模块可级联在一起构成更大的片上存储器。

在设计中使用 UltraRAM 很简单，因为赛灵思提供了所有必要工具，利用这些工具可以在设计中集成功能强大的最新模块。

缓冲与存储的需求增加

UltraScale+™ FPGA 和 MPSoC 能够传输和处理的数据比前一代器件多得多。最大的 Virtex® UltraScale+ FPGA (即 VU13P) 包含 128 个数据速率高达 32.75Gb/s 的 GTY 收发器, 以及超过 11,000 个工作频率接近 900MHz 的 DSP Slice。这样可得到 8.4Tb/s 串行带宽和 21 TMAC/s 的信号处理速度, 因此, 片上存储器也需要实现同样的增长, 以使系统能够有效地缓冲处理前进入的数据和处理后流出的数据。

到达 (进入) UltraScale™ FPGA 器件, block RAM 和分布式 RAM 可作为快速和高效的片上存储器。为了成功处理数百兆位的数据, 通常需要使用外部存储器, 例如 DDR4、DDR3 或 RDRAM3。

UltraScale+ 系列产品则采用一种新的存储方法。UltraRAM 是一种大容量、重量轻的存储器模块, 使 UltraScale+ 器件具备 500Mb 以上的高能效、低成本片上数据存储器, 相当于赛灵思 28nm FPGA 片上存储器的 6 倍。

灵活的级联功能使得 UltraRAM 模块能够采用多种配置——既可单独使用, 也可在器件中将所有模块连接在一起使用。这使得用户能够构建大小适合应用需求的存储器阵列, 并省去一些 PCB 上的外部存储器组件, 例如 QDR SRAM。

UltraRAM 细节与特性

赛灵思很多代的 FPGA 和 SoC 都采用分列架构, 将不同资源列并排布置在一起。列的数量和高度直接决定 FPGA 或 SoC 的尺寸和容量。

大多数 Kintex® UltraScale+ 和 Zynq® UltraScale+ 器件都包含一列 UltraRAM; 高端 Virtex UltraScale+ FPGA 则包含多达 5 列 UltraRAM。UltraRAM 可在设计中作为单个 288Kb 存储器, 或者连接在一起构成更大的 RAM 阵列。相同列中的所有 UltraRAM 都可连接在一起, 通过专用的级联路由传送地址、数据和控制信号。在 Kintex UltraScale+ 和 Zynq UltraScale+ 器件中, 级联得到的 RAM 阵列可高达 36Mb, 在 Virtex UltraScale+ 器件中可达到 22.5Mb。Virtex UltraScale+ 系列中, 所有 UltraRAM 列都可通过光纤路由连接在一起, 在最大器件中可构成容量达 360Mb 的存储器阵列。

每个 UltraRAM 模块都是一个双端口同步 288Kb RAM, 采用深度为 4,096、宽度为 72 位的固定配置。端口 A 和端口 B 共享相同时钟信号。在外部时钟的单个周期内, 端口 A 操作总是在端口 B 操作之前完成。每个端口在每个时钟周期都可独立执行一个读操作或一个写操作。当两个端口在相同时钟周期内以相同地址 (即地址冲突) 执行写操作时, 端口 B 的写操作有效, 因为端口 A 上的写操作被覆盖。当端口 A 执行读操作, 而端口 B 以相同地址执行写操作时, 端口 A 获得存储器阵列中的旧数据, 然后端口 B 上的新数据被写到存储器阵列。当端口 A 执行写操作, 端口 B 执行读操作时, 端口 A 上的新数据被写入存储器阵列, 新数据也被读到端口 B。对于每个端口, 当执行写操作时, 该端口的读输出不变, 保持之前的值。

每个 UltraRAM 存储器阵列的内容都被每个端口上的汉明码单误差校正双误差检测 (SECDED) ECC 码保护，以确保数据完整性。启动 ECC 后，UltraRAM 提供 64 位宽的被保护数据字。UltraRAM 的 SECDED ECC 代码与 block RAM 的兼容。这样可实现端到端数据保护方案，即在一个存储器输入端进行编码的数据，在下游存储器输出端对其进行解码之前，可在多个流水线级、存储器类型和 / 或逻辑上保持受保护状态。

UltraRAM 每两个端口包含多达四个流水线级，并可针对 1-4 个时钟周期的时延进行配置。当通过级联 UltraRAM 搭建较大的 RAM 阵列时，时延取决于所用 UltraRAM 的数量（即阵列的大小以及目标工作频率）。

功耗降低

UltraRAM 提供各种内置功能以最大化功率效率，通常无需用户干预。这些功能包括：

- 将设计中不用的 UltraRAM 断电
- 自动对不使用的流水线寄存器进行时钟门控
- 当 UltraRAM 长时间不使用时，使其进入休眠模式

UltraRAM 可通过手动或自动进入休眠模式。用户可以访问 UltraRAM 的 SLEEP 端口。激活 SLEEP 时，UltraRAM 从下个时钟周期开始进入休眠模式。休眠模式下，UltraRAM SRAM 存储器的外设逻辑断电，但 SRAM 保持加电并保留其中的内容。

每个 UltraRAM 都包含相应的电路用来预测和识别接下来 N 个周期内模块是否会被使用，如果不使用，那么在未使用时间足够长之后，模块会被送入休眠模式。这种自动休眠模式使用休眠端口，利用内置的控制逻辑来检查将即将输入的地址和使能信号的状态。AUTO_SLEEP_LATENCY 属性决定地址和使能信号需要比其他输入信号提前多久到达。该信息用来确定是否有足够的时间使单个 UltraRAM 进入休眠模式并在下个活动之前将其唤醒。

在设计中使用 UltraRAM

UltraScale+ 产品组合使用赛灵思的 Vivado® Design Suite。设计工具提供一系列代码模板，可帮助用户成功锁定器件内的可用资源。有三种方法可以在 RTL 设计中使用 UltraRAM：编写代码以调用存储器；实例化器件原语；使用最近添加的赛灵思参数化宏 (XPM)。大多数用户都熟悉调用和实例化。Vivado Design Suite 包含调用模板，所在位置：

模板 → VHDL/Verilog → 综合构建 → 编码实例 → RAM

在 Vivado Design Suite v2016.1 中，用户必须指定 `ramstyle= "ultra"` 才能明确指示 Vivado 综合使用 UltraRAM。

如果用户需要对 UltraRAM 模块的连接具有最终控制权，应使用以下位置的器件原语：

模板 > VHDL/Verilog > 器件原语实例化 > Kintex/Virtex UltraScale+ > BLOCKRAM > URAM

尽管能让用户对单个组件的连接实现最严格的控制，但如果采用很多组件并有很多端口和信号，那么这种方法很快就会变得很费劲。

要在 RTL 设计中采用 UltraRAM，第三种方法是 XPM：

模板 → VHDL/Verilog → 赛灵思参数化宏 (XPM) → 存储器 (XPM_MEMORY) → RAM

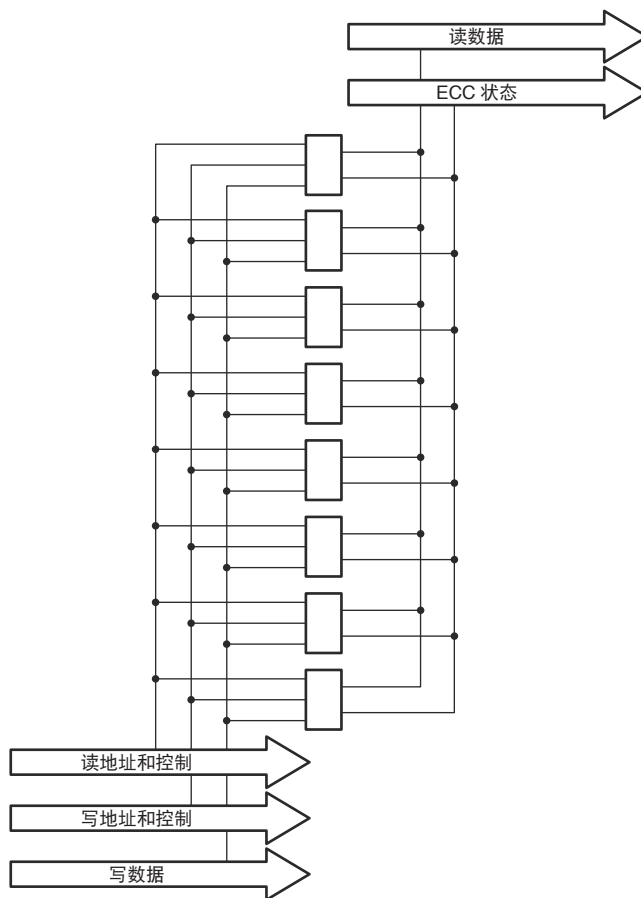
XPM 是一种用来根据用户指定要求创建 RAM 和 ROM 结构的最新工具。用户可在 XPM 代码中指定多个类属，包括存储器大小、时钟模式、ECC 模式等。这些要求随后被 Vivado 综合转化为适当的存储器阵列尺寸和风格。

在 Vivado Design Suite v2016.1 中，用户必须在 MEMORY_PRIMITIVE 类属上指定值 UltraRAM，以明确指示 Vivado 综合使用 UltraRAM。

创建存储器阵列

UltraRAM 架构扩展性非常强，允许很多个 UltraRAM 模块高效地连接在一起，以构成深逻辑存储器，仅使用很少甚至无需架构资源，而且访问时延相对较短。每个 UltraRAM 包含所有必要的逻辑，用以将相同列中相邻 UltraRAM 模块之间的全部输入以及输出接口进行完全级联，并有选择性地进行了流水线化。此外，每个 UltraRAM 都包含每端口上的附加逻辑和控制输入，以独立确定 UltraRAM 模块阵列中经过的读或写事务处理是否针对本地阵列，还是应该被忽略掉。这种分布式的决策逻辑能实现灵活而且低开销的扩展性。

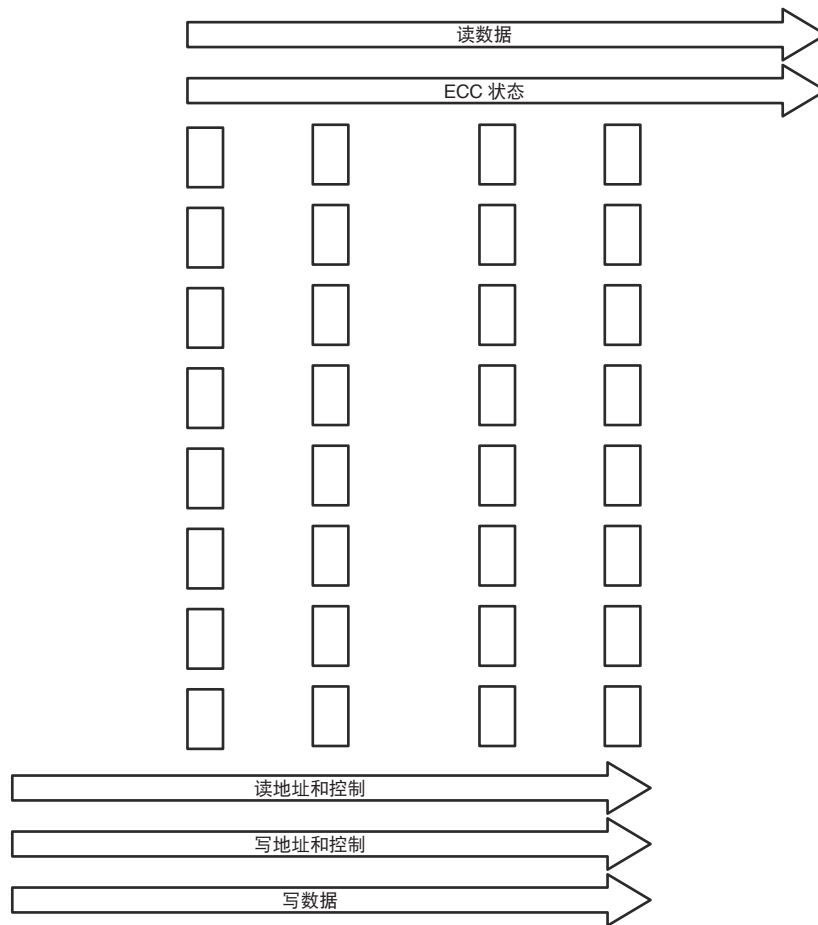
当设计中的指定存储器阵列需要一个以上 UltraRAM 模块时，必须决定如何将多个 UltraRAM 连接在一起。有两个选择分别是：使用一个列中的所有 UltraRAM，或者使用多列阵列中的 UltraRAM。如果在一个列中构建 UltraRAM 阵列，那么用来级联 UltraRAM 的所有必要路由都包含在该列中，使工具能够充分利用所提供的专用资源。见图 1。



WP477_01_032616

图 1：在一个列中级联的 UltraRAM

如果设计要求决定了多列阵列更加适合，例如，如果所需的阵列大于单个列的容量，那么需要创建多列阵列（见图 2）。这种情况下，需要用一些架构资源将多个列连接在一起。最大可实现频率以及多列阵列中的访问时延取决于阵列尺寸、所用列的数量以及其他设计因素（例如 UltraRAM 列之间有多少架构资源被设计的其他部分使用）。



WP477_02_032616

图 2 : 多列 UltraRAM 阵列

当创建 UltraRAM 阵列时，地址、控制和写数据信号被提供至阵列的左下角，数据从阵列的右上角读取。这样能确保，通过对固定大小的阵列进行恰当流水线化后，从任何地址读取任何数据所产生的时延总相同。

与每个 UltraRAM 自身可提供的 64 位 ECC 保护或 72 位非 ECC 保护字宽相比，宽数据或控制路径设计需要更高的总线带宽。这种情况下，可以并列使用多个相同的 UltraRAM 模块实例或以上描述的级联结构来构建非常高效的宽字数据存储结构。

寄存器和流水线化

每个 UltraRAM 包含多个输入和输出寄存器，如图 3 所示，其中有一些是可选的，取决于配置情况和所需的功能。

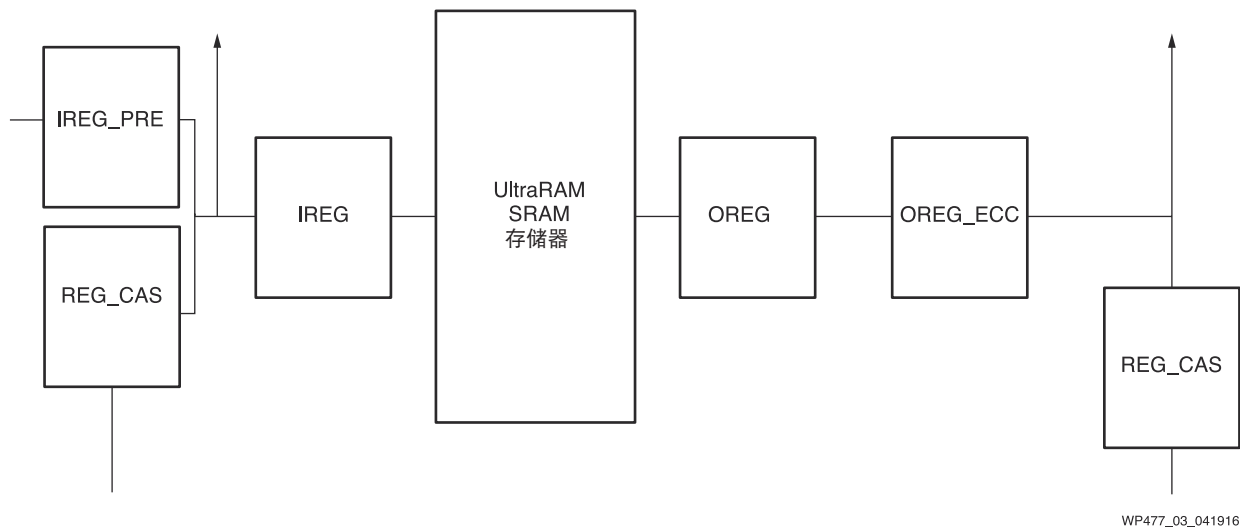


图 3：UltraRAM 寄存器

IREG、OREG 和 OREG_ECC 寄存器根据用户所需的存储器功能进行激活。此外，当把多个 UltraRAM 级联在一起时使用的寄存器为：IREG_PRE 和 REG_CAS。

当把多个 UltraRAM 连接在一起时需要遵守一些规则，以保持最大工作频率：

- 单列阵列中的第一个实例或多列阵列每个列中的第一个实例都必须使用 IREG_PRE 寄存器。
- 如果设计中需要，OREG 和 / 或 OREG_ECC 寄存器应针对阵列中的所有 UltraRAM 一直启用。
- 必须每经过四个 UltraRAM 模块激活 REG_CAS 寄存器。
- 单列阵列中的最后一个实例或者多列阵列每列中的最后一个实例必须使用 REG_CAS 寄存器。
- 如果阵列垂直跨越多个时钟区域，那么需要在时钟区域边界两侧提供附加流水线级。每个时钟区域的每列上有 16 个 UltraRAM。

如需了解 UltraRAM 模块的更多信息和详细框图，请参考 [UG573](#) 《UltraScale 架构存储器资源用户指南》的第二章。

迁移到 UltraRAM

用户一定要知道所选目标器件中不同存储器资源的数量。有些情况下，用户最开始可以针对只有 block RAM 的器件，但设计最终将会针对包含 UltraRAM 的器件。这种情况下，用户需要了解 UltraRAM 的特性，并编写代码以利用 block RAM 和 UltraRAM 中的特性。这样，工具就可使用目标器件中可使用的 UltraRAM，而且，当改用不同器件时用户不必重新设计。

表 1 给出了不同 UltraScale+ 系列中 UltraRAM 的数量。在 Virtex UltraScale+ 器件中，可将所有 UltraRAM 连在一起，以构成容量高达 360Mb 的多列阵列；不过，也可实现更为常用的 10Mb-144Mb 范围的多实例使用模型。

表 1 : UltraScale+ 产品组合中的 UltraRAM 资源

系列	Kintex UltraScale+	Virtex UltraScale+	Zynq UltraScale+
UltraRAM	0-128	320-1,280	0-120
UltraRAM (Mb)	0-36	90-360	0-36
列高 (Mb)	0-36	18.0 或 22.5	0-36

所有 UltraScale+ 器件还包含可实现较小存储器的 block RAM 和分布式 RAM，从而在 VU13P 中实现总容量超过 500Mb 的片上存储器。

[DS890](#)，《UltraScale 架构和产品简介》给出了器件资源方面的更多详情。

如果所选的器件中有 UltraRAM，应该将其用作 144Kb 或更大的存储器。

替代外部存储器组件

高达 360Mb 的 UltraRAM 几乎可配置成无数种方式，因此会有很多种使用模式。常见的 UltraRAM 使用模式是如果不实现在外部存储器（例如 QDR SRAM）中，可以将存储器整合在 FPGA 或 MPSoC 中。片上存储器的优势在于可采用所需要的容量大小，而不是存储器厂商提供的大小；而且存储器在逻辑上非常接近设计中用来缓冲或存储数据的位置。此外，为了与外部存储器进行连接，数据必须通过存储器接口或控制器，并且通过 FPGA/MPSoC 上的 I/O 引脚，而且外部存储器组件会给系统增加很多功耗。通过以 UltraRAM 的方式在片上实现大容量存储，这样可消除 FPGA/MPSoC 的 I/O 以及外部存储器的 I/O 产生的功耗。

输入多播

每个 UltraRAM 端口都可确定级联模式中的事务处理是否属于局部存储器阵列。该功能可用来实现动态的事务处理级写单播、多播和广播存储器结构。这种结构可以构成高效输出缓冲存储器开关的基础。

有一个 UltraRAM 存储器端口专门用来写，另一个专门用来读。所有 UltraRAM 模块的写端口都采用每模块唯一的独热编码 SELF_ADDR 和相配的独冷编码 SELF_MASK 值配置为连续级联形式。每个写事务处理都可以被送到单个、多个或所有 UltraRAM 模块，方法是设置一个或多个相应的高地址位，以匹配所有目标模块的 SELF_ADDR 属性的 logical-OR。所有 UltraRAM 模块的读端口都是非级联的，能够让单个写事务处理传播到多个消费模块。

每个 UltraRAM 模块代表一个消费模块的缓冲要求。要获得更大的存储容量，可将每个 UltraRAM 模块改成 N 个 UltraRAM 模块级联成的消费模块。

假设 C 个消费模块从 C 个逻辑存储器（每个深度为 N 个 UltraRAM 模块）并行读取，那么写端口需要一个包含所有 C * N 个 UltraRAM 模块的连续级联，而读端口需要 C 个独立的级联，每个深度为 N 个 UltraRAM。

图 4 给出了这样一种配置，其中每个消费模块有四个输出，每个输出需要一个 UltraRAM 模块作为缓冲，并有两个样本多播事务处理分别针对 UltraRAM 模块 0 和 1 以及 1 和 3。

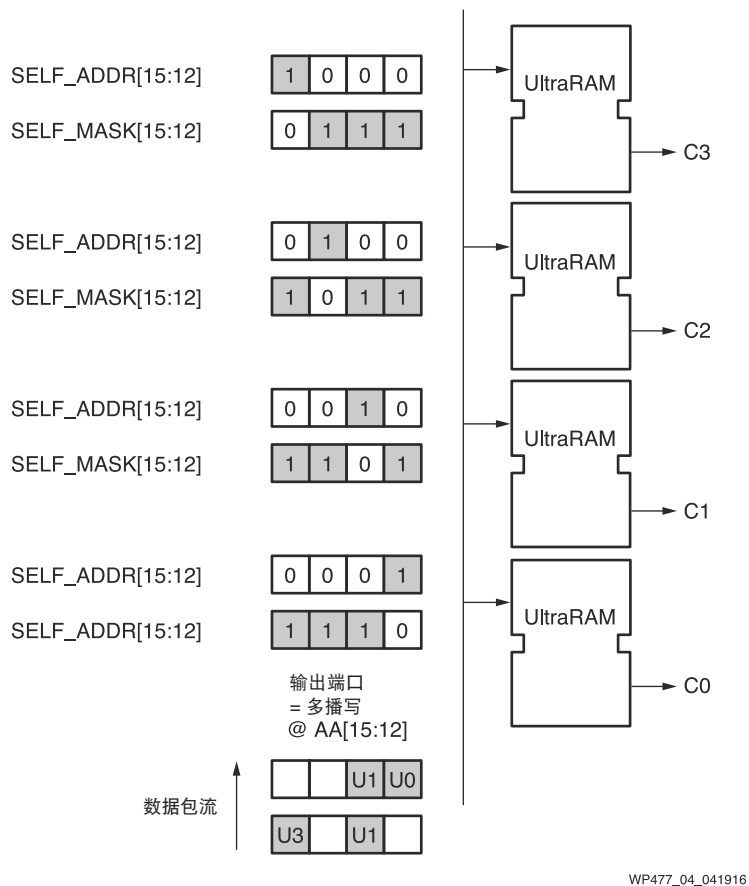


图 4：输入多播

结论

每个设计都需要和使用这样或那样的存储器。将大容量、灵活的 UltraRAM 模块添加到 UltraScale+ 产品系列后，为用户提供的片上存储器容量比以往任何器件或竞争器件都要大。由于模块可连接在一起构成不同大小的存储器阵列，这样用户就能在设计中在需要的位置构建理想的存储器解决方案，消除了使用外部 SRAM 组件所带来的功耗和电路板复杂性。

修订历史

下表列出了本文档的修订历史：

日期	版本	修订描述
2016 年 06 月 14 日	v1.0	赛灵思初始版本

免责声明

本文向贵司 / 您所提供的信息（下称“资料”）仅在选择和使用赛灵思产品时供参考。在适用法律允许的最大范围内：(1) 资料均按“现状”提供，且不保证不存在任何瑕疵，赛灵思在此声明对资料及其状况不作任何保证或担保，无论是明示、暗示还是法定的保证，包括但不限于对适销性、非侵权性或任何特定用途的适用性的保证；且 (2) 赛灵思对任何因资料发生的或与资料有关的（含对资料的使用）任何损失或赔偿（包括任何直接、间接、特殊、附带或连带损失或赔偿，如数据、利润、商誉的损失或任何因第三方行为造成的任何类型的损失或赔偿），均不承担责任，不论该等损失或者赔偿是何种类或性质，也不论是基于合同、侵权、过失或是其他责任认定原理，即便该损失或赔偿可以合理预见或赛灵思事前被告知有发生该损失或赔偿的可能。赛灵思无义务纠正资料中包含的任何错误，也无义务对资料或产品说明书发生的更新进行通知。未经赛灵思公司的事先书面许可，贵司 / 您不得复制、修改、分发或公开展示本资料。部分产品受赛灵思有限保证条款的约束，请参阅赛灵思销售条款：<http://china.xilinx.com/legal.htm#tos>；IP 核可能受赛灵思向贵司 / 您签发的许可证中所包含的保证与支持条款的约束。安全保护功能，不能用于任何需要专门故障安全保护性能的用途。如果把赛灵思产品应用于此类特殊用途，贵司 / 您将自行承担风险和责任。请参阅赛灵思销售条款：<http://china.xilinx.com/legal.htm#tos>。

汽车应用免责声明

赛灵思产品并非为故障安全保护目的而设计，也不具备此故障安全保护功能，不能用于任何需要专门故障安全保护性能的用途，比如与下列有关的用途：(1) 安全气囊设置；(2) 车辆控制，除非在该赛灵思产品中具备故障安全保护或者额外功能（但不包括对安装在赛灵思设备中用于执行该等额外功能的软件的使用）且会对操作人员操作失误发出警告信号；或者 (3) 可能会导致死亡或者人身损害的用途。客户应当自行承担因赛灵思产品被用于该等用途而产生的全部风险和责任。