Application Note

适用于 MCU+ SDK 上 Sitara™ 处理器的增强型 OSPI PHY 调优算法

Nikhil Jain, Aryamaan Chaurasia, Vaibhav Kumar, Soumya Tripathy, Brian Elies, Nuruddin Mahmood

摘要

本应用手册概述了 OSPI DQS PHY 调优算法，这是 MCU+ SDK 11.02 中引入的一项新功能，旨在优化德州仪器 (TI) Sitara™ 处理器系列（包括 AM243x、AM62x、AM62Ax、AM62Dx、AM62Px、AM64x 和 AM275x 器件）上的高速串行 NOR 和 NAND 闪存访问。PHY 调优过程会自动校准三个时序参数，包括 Tx DLL、Rx DLL 和读取延迟。这些参数补偿板级传播延迟、过程、电压和温度变化。目标受众包括嵌入式软件工程师、系统架构师，以及在 Sitara 处理器上使用 OSPI 闪存存储器的硬件设计人员。

内容

商标

1 简介

现代嵌入式系统需要快速可靠地访问外部闪存以进行代码执行和数据存储。OSPI 支持与串行 NOR 和 NAND 闪存器件进行高带宽通信，采用单倍数据速率 (SDR) 和双倍数据速率 (DDR) 模式，支持在八路配置中实现高数据速率。

在这些高速下，信号时序至关重要。如果 OSPI 控制器在错误的时间采样数据，电路板布线长度、温度变化、电压波动和元件公差的微小变化可能会损坏数据。OSPI PHY（物理层）模块包括可编程延迟线，可对信号时序进行细粒度控制，但手动查找最佳延迟设置不切实际。

为了确定延迟设置，PHY 调优算法会搜索参数空间来识别稳定的工作点。本应用手册介绍了算法的工作原理、影响调优成功的因素，以及 DQS 调优算法中执行的步骤。

2 术语

PHY：

指 OSPI 驱动器的 PHY（物理层）模式。PHY 模式使用专用计时电路来管理存储器的数据传输。在该模式下，每个参考时钟周期在标准传输中产生一个完整的存储器时钟周期，在双倍速传输中产生半个周期。系统提供四种不同的时序配置，可使用内部信号或来自存储器芯片的外部反馈信号。

启用 PHY 后会绕过输入时钟分频器。因此，有效频率是指输入时钟频率。PHY 调优算法通过改变 rxDLL、txDLL 和读取延迟来计算最佳调优点。请参阅此处了解更多信息。

QSPI：

四线串行外设接口是一种使用 4 条数据线 (DQ0-DQ3) 进行串行数据传输的增强型 SPI 型号。支持单通道/双通道/四通道模式以适应不同传输阶段，在保持向后兼容性的同时，相较于标准 SPI 可实现高达 4 倍的带宽提升。

OSPI：

八路串行外设接口 - 一种使用 8 条数据线 (DQ0-DQ7) 进行串行数据传输的高级 SPI 型号。支持所有 QSPI 模式和八路模式，可获得更高带宽。在进行源同步数据采集时，它可使用或不使用 DQS（数据选通）信号运行。

SDR：

单倍数据速率模式在时钟信号的单一沿传输数据，每根数据线每时钟周期发送一位。这是一种更简单，更传统的时钟方案，可在中等速度下提供良好的可靠性。在 8 条数据线的八通道 SDR 模式下，理论最大数据速率为每个时钟周期 8 位。

DDR：

双倍数据速率模式在时钟信号的上升沿和下降沿均传输数据，相较于 SDR 模式有效提升两倍数据吞吐量。在 8 条数据线的八通道 DDR 模式下，每个时钟周期传输 16 位数据（每边 8 位 × 2 边）。

协议（命令-地址-数据）：

协议模式格式为 WR-WR-WR，其中第一个 WR 表示命令位宽和速率，第二个 WR 表示命令修饰符位宽和速率，第三个 WR 表示数据位宽和速率。位宽 (W) 可以是 1 位或 8 位。速率 (R) 可以是 SDR 中的 S，也可以是 DDR 的 D。SDR 在上升时钟沿和下降时钟沿传输相同的值，而 DDR 可能在每个边沿传输不同的值。

例如，1S-1S-1S 表示所有相位都使用 1 位宽 SDR。符号 8D-8D-8D 表示所有相位都使用 8 位宽 DDR。

DQS：

数据选通是闪存器件在 DDR 模式下的闪存读取操作期间提供的源同步信号。DQS 边与有效数据窗口的中心对齐，从而使控制器能够在最佳时间对数据进行采样。DQS 用于 DDR 模式，不适用于 SDR 模式。

基准时钟：

基准时钟是 OSPI 控制器的输入时钟信号。它通常由系统时钟提供。该时钟经过分频，生成发送到闪存器件的串行时钟。

DLL：

延迟锁定环是一种为信号时序控制生成精确、可编程延迟的数字电路。DLL 由延迟元件链组成，通常是逆变器或缓冲器，其传播延迟可以进行调整。环路通过连续调整延迟链直至输出与反馈信号对齐，从而锁定至基准。在 OSPI PHY 中，独立的发送和接收 DLL 以一个延迟元件步长的分辨率控制输出和输入信号的时序，从而对建立时间和保持时间进行精细控制。

Tx DLL：

传输 DLL 是 PHY 中的可编程延迟线，用于调整控制器将输出数据和命令驱动到闪存的时间。它还可确保满足建立和保持时间要求。取值范围通常为 0 至 127。

Rx DLL：

接收 DLL 是 PHY 中的可编程延迟线，用于调整控制器对来自闪存的输入数据进行采样的时间。DLL 延迟采样时钟或 DQS 信号，使其与数据有效窗口的中心对齐。取值范围通常为 0 至 127。

读取延迟：

读取延迟是应用于数据采集时序的额外可编程延迟，以基准时钟周期进行测量。它提供了对细粒度 DLL 设置的补充粗调整。取值范围通常为 0 至 4。

OTP：

最佳调优点是 Tx DLL、Rx DLL 和读取延迟值的特定组合，由调优算法选择。它代表了稳定工作区域的中心，在所有方向上都具有最大裕度。

亚稳态间隙：

亚稳态间隙是一个参数空间，在该空间内并非均匀地通过或失败。相反、它包括时序亚稳态并读取失败的区域之间的对角线间隙，或者随温度、电压和其他环境因素偏移的边界。该间隙表示采样时钟沿在数据切换沿上移动的切换区域。这从根本上来说是不稳定的，必须加以避免。

半径验证：

半径验证是一种验证技术，用于测试候选调优点周围圆形区域内的所有参数组合。它可确保所选点在所有方向上都有足够的边距。

对角线搜索：

对角线搜索是最新的调优算法使用的核心搜索策略。算法不是在参数空间中进行水平和垂直搜索，而是沿着有效穿越通过区域和亚稳态间隙的 45 度对角线进行搜索。

3 PHY 调优算法

PHY 调优是自动校准过程，用于确定启用 PHY 时的最佳时序参数。

该过程涉及：

- 将测试模式（称为攻击向量）写入闪存。
- 系统地测试时序参数的不同组合。
- 验证每个参数组合下测试模式的闪存读取。
- 识别闪存读取始终成功的区域。
- 在具有最大裕度的稳定区域内选择最佳调优点 (OTP)。

在系统能够可靠地高速访问闪存之前，必须成功完成调优算法。如果不进行调优，系统需要采用保守的时序设置，这会限制性能，或者在极端温度和电压条件下面临数据损坏的风险。

备注

算法、步骤、参数和相应的值已根据 MCU+ SDK 11.02 进行配置。

4 关键调优参数

4.1 非 DQS PHY 调优算法的参数配置

参数	用途	典型值	影响
最小读取延迟	读取延迟搜索的下界	0	由硬件定义
最大读取延迟	读取延迟搜索的上界	3	由硬件定义
RxDLL 搜索 - TxDLL 高结束	在 RxDLL 窗口中搜索的 Tx DLL 的最高值	127	在 RxDLL 窗口搜索期间固定为 TxDLL 值
RxDLL 低搜索启动	启动 Rx DLL 以进行下界搜索	0	定义开始窗口搜索的最小 Rx DLL
RxDLL 高搜索结束	结束 Rx DLL 以进行上界搜索	127	定义用于搜索窗口结束的最大 Rx DLL
RxDLL 和 TxDLL 搜索步长	搜索期间 Rx 和 Tx DLL 递增/递减的步长	8	较小的值可提供更精细的分辨率，但会增加调优时间

4.2 DQS PHY 调优算法的参数配置

参数	用途	典型值	影响
搜索半径	在候选调优点周围验证的圆形区域大小	10	较大的值可提供更大的裕度，但会增加调优时间
最小通过大小	必须将通过区域的最小平方长度视为有效	100	较大的值会拒绝缺少足够裕度的狭窄区域
连续通过点	确认亚稳态间隙所需的相邻通过点数量	10	较高的值可改善稳定性检测，但可能更难验证
连续失效点	确认亚稳态间隙所需的相邻失效点数量	5	帮助区分真正的间隙和隔离失效
对角线偏移	如果主对角线失败，搜索对角线的偏移量	10	较大的值会加快搜索速度，但覆盖范围较小
最大对角线偏移	声明失败前的对角线偏移上限	70	确定搜索参数空间的详尽程度
读取延迟搜索步长	最初定位有效读取延迟值时的步长	16	较大的值完成得更快，但可能会错过较窄的通过区域
最小 DLL 值	DLL 参数搜索的下界	0	由硬件定义
最大 DLL 值	DLL 参数搜索的上界	127	由硬件定义
最小读取延迟	读取延迟搜索的下界	0	由硬件定义
最大读取延迟	读取延迟搜索的上界	4	由硬件定义

5 PHY 调优算法的先决条件

5.1 硬件要求

5.1.1 闪存器件准备

- 验证闪存器件是否已正确初始化且可访问。
- 器件必须支持预期的协议模式。
- 从要写入攻击向量的区域擦除闪存。

5.1.2 PHY 配置

- 基准时钟必须稳定并以正确的频率运行。
- 根据工作频率设置 PHY 模式（主模式或旁路模式）。
- 适当配置 DLL 锁定模式（全周期或半周期）。

5.2 攻击向量

攻击向量是一种特定的 128 字节模式，旨在可靠地检测时序违规。

该模式包括：

- 产生频繁数据切换的交替位模式。
- 长时间运行零和一来测试偏移问题。
- 用于测试位对位隔离的单个位切换。
- 偶数字节和奇数字节对齐，以涵盖 DQS 模式下的两个采样相位。

在调优开始之前，将攻击向量写入已知的闪存地址。保留此地址：

- 在闪存的可访问、不受保护区域中。
- 在不会干扰引导加载程序或应用代码的位置。
- 在写入模式之前正确擦除。

在调优期间，算法会重复读取该地址，并将结果与预期模式进行比较。任何不匹配都表示该参数组合存在时序违规。

5.3 通过区域与失效区域

具有半周期锁定的 DQS PHY 调优算法的延迟值存储在三维数组中，其维度为：

- 第一个维度：读取延迟值 [0 至 4] - 5 个总值。
- 第二个维度：传输 DLL 值 [0 至 127] - 128 个总值。
- 第三个维度：接收 DLL 值 [0 至 127] - 128 个总值。

这使得大小为 [5][128][128] 的数组包含总共 81,920 个元素。数组索引为：数组 [readDelay][TxDLL][RxDLL]。

每个数组元素存储一个简单的二进制结果：

- 通过区域标有值：1（表示攻击向量读取成功）。
- 失效区域标有值：0（表示读取错误或数据不匹配）。

例如：

- 数组 [2][64][64] = 1：在读取延迟 = 2、Tx DLL = 64、Rx DLL = 64 时，测试通过。
- 数组 [2][65][65] = 0：在读取延迟 = 2、Tx DLL = 65、Rx DLL = 65 时，测试失败。

5.4 主模式与旁路模式

OSPI PHY 可以在两种基本不同的模式下工作，这两种模式改变了 DLL 延迟值的解释和应用方式。

MCU+ SDK 11.02 根据配置的基准时钟频率自动选择适当的模式：

- 基准时钟 < 166MHz：已选择旁路模式。
- 基准时钟 ≥ 166MHz：已选择主模式。

5.4.1 旁路模式

在旁路模式下，DLL 值直接控制使用的物理延迟元件的数量。每个延迟元件都是一个固定缓冲器或门，会增加较小的绝对传播延迟。旁路模式遵循 Tx DLL 与用于延迟传输时钟信号的硬件元件数量之间的直接映射。总延迟即为各元件延迟之和。在此模式下，延迟是固定的时间值，与工作时钟频率无关。

5.4.2 主模式

在主模式下，主 DLL 电路通过锁定延迟链来连续测量时钟周期，从而产生全时钟周期或半时钟周期的延迟。然后，将发送和接收 DLL 值解释为该基准延迟的一部分，而不是绝对延迟元件计数。MCU+ SDK 11.02 支持两种类型的时钟机制：

- 全周期锁定：如果主 DLL 锁定到全时钟周期，则 DLL 值 64 表示 64/128 = 一个完整时钟周期的 50%。
- 半周期锁定：如果主 DLL 锁定到半时钟周期，则 DLL 值 64 表示 64/128 = 半个周期的 50%，等同于一个完整时钟周期的 25%。

在主模式下，延迟与时钟周期成正比，并随着频率自动调整。半周期锁定可在时钟周期的较小部分内提供更精细的分辨率，从而进行精确的时序控制。在此模式下，延迟被指定为时钟周期的一部分，无论绝对时钟频率如何，都能保持正确的时序关系。这对于在高速下的可靠操作至关重要，因为此时必须相对于时钟边沿精确控制建立时间和保持时间。

6 需要更新的调优算法

新的 DQS 算法解决了之前算法中的关键可靠性限制。旧方法主要根据区域大小和温度相关的位置来选择调优点，而不验证所选点周围的裕度，从而使调优点更接近极端值，使系统更容易受到噪声相关变化的影响。之前的方法使用具有固定 Tx/Rx 搜索和边界二进制搜索的角查找，这可能会遗漏狭窄区域或错误描述不规则的通过区域几何形状。此外，当主搜索失败时，旧方法缺少回退策略。新的 PHY 调优算法引入了圆形半径验证，以在所有方向上提供足够的裕度，采用显式亚稳态间隙检测，通过连续通过点和失效点计数来滤除孤立噪声，采用对角线搜索，遵循 DQS 通过区域的自然几何形状，采用几何中点计算以实现最优放置，执行最小区域尺寸，以及采用对角线偏移作为后备策略。这些增强功能有助于在各种电路板设计、闪存器件和 PVT 条件下实现可靠操作。

6.1 温度变化

温度对半导体时序有重大影响：

- 温度越高，操作越慢。随着温度升高，由于硅晶体晶格中的原子振动增加，电子通过晶体管的速度会更慢。
- 较低的温度可加快操作速度。在低温下，原子晶格的活性较低，允许电子更自由、更快速地通过晶体管移动。
- 在典型的工作温度范围（-40°C 至 +125°C）内，延迟元件会改变传播延迟特性，从而导致参数空间中 DLL 位置发生偏移。

因为信号之间的相对时序发生偏移，亚稳态间隙会随着温度的变化而移动。较新的 DQS 调优算法通过选择具有足够裕度的点进行补偿，以便通过区域在整个温度范围内保持有效。

旧算法和新算法计算的 OTP 值的比较如下图所示。

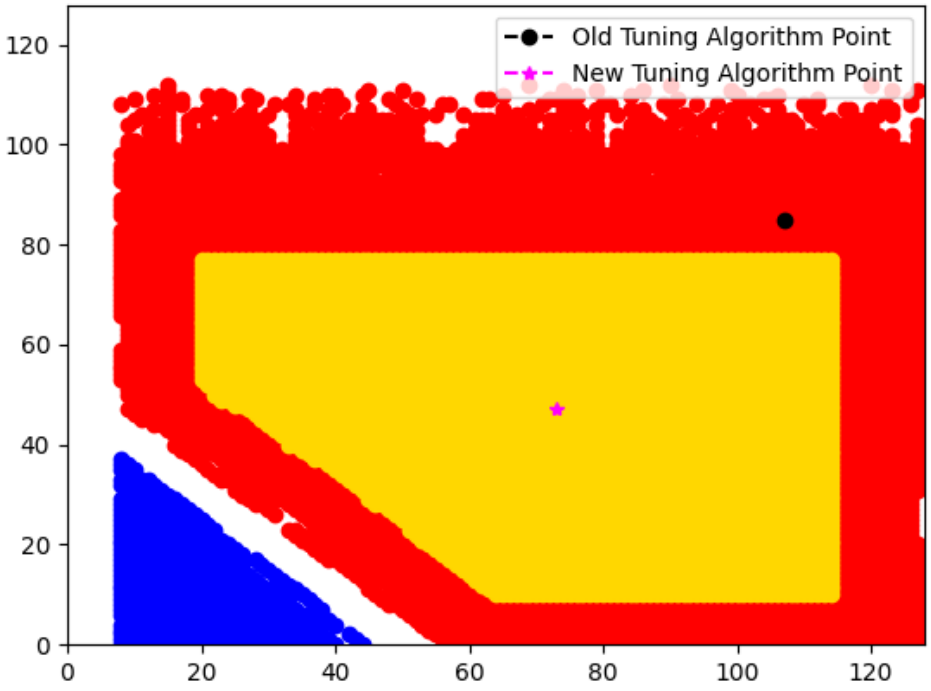

图 6-1. 新旧 OTP 的比较

7 算法实现方案

7.1 DQS PHY 调优算法

DQS 调优算法使用对角线搜索策略来确定具有 DQS 信令的高速运算的最佳时序参数。

7.1.1 对角线选择

该算法开始沿 (0,0) 至 (127,127) 的对角线搜索，表示 TX 和 RX DLL 值的相等增量。如果找不到有效的调优点，对角线会以交替模式向上移动 10 点（增加 Y 偏移）或向右移动 10 点（增加 X 偏移）。偏移可沿原始对角线方向持续增加，最大可达 70 个单位，从而在保持 45 度斜率的同时，系统性地覆盖参数空间。

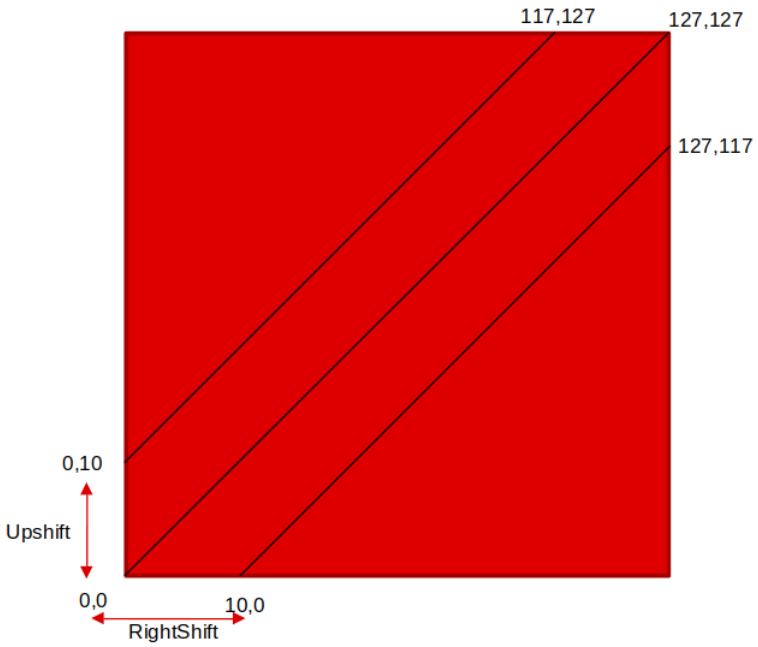

图 7-1. 对角线选择

7.1.2 有效读取延迟选择

从最小读取延迟开始，算法沿对角线以 16 为步长执行粗搜索，以识别读取成功的区域。通过点确认该读取延迟设置处存在有效的工作区域。一旦找到，则以一为步长执行细粒度搜索，映射出完整的有效区域。该过程会对每个读取延迟值重复进行。如果在所有读取延迟中未找到通过点，则会移动对角线以寻找新的潜在工作点。

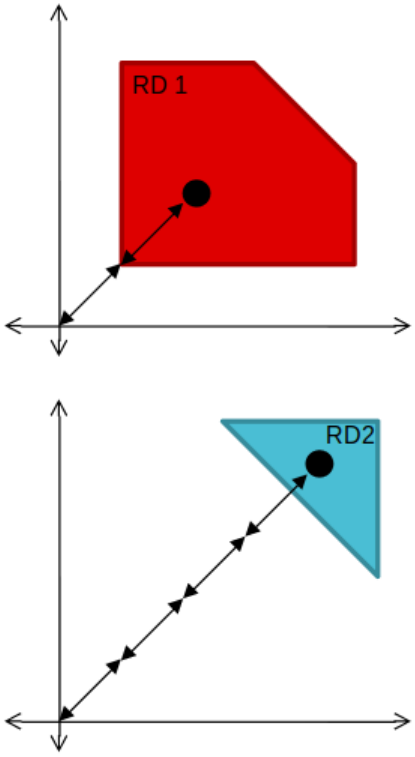

图 7-2. 有效读取延迟选择

7.1.3 角点识别

该算法沿着对角线定位通过区域的精确边界。对于最小有效读取延迟，算法从对角线起点开始搜索，以 1 为步长递增 Tx DLL 和 Rx DLL，直到找到第一个通过点（下界）。对于最大有效读取延迟，算法从对角线终点开始搜索，递减两个 DLL，直到找到第一个通过点（上界）。这些角点定义了接口可靠工作的完整范围。

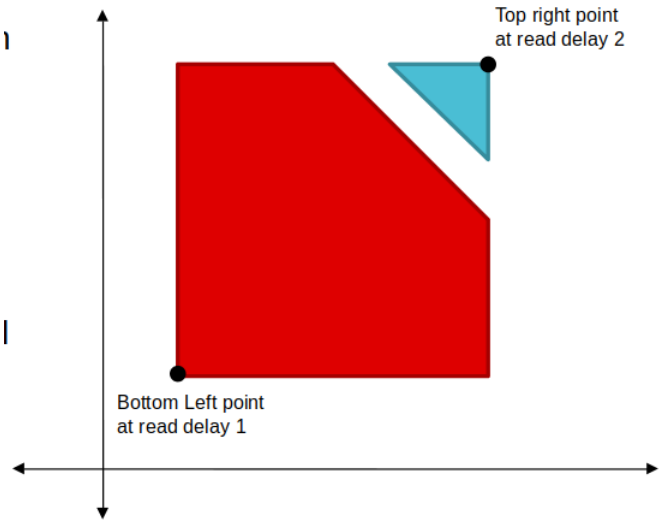

图 7-3. 角点识别

7.1.3.1 仅用于一个读取延迟值的角点选择

当两个端点存在相同的读取延迟时，存在一个单一通过区域。该算法会查找连续通过点最多的线的角点，来表示稳定的采样窗口。如果线长度（即角点之间的距离）超过最小通过长度阈值，则继续进行调优点选择。否则，由于窗口过窄无法可靠操作，将报告失败。

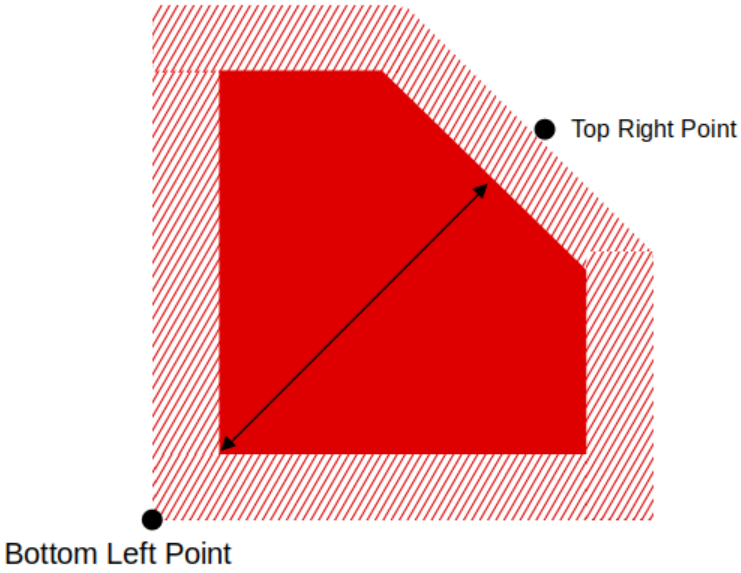

图 7-4. 用于一个读取延迟值的角点选择

7.1.3.2 用于两个不同读取延迟值的角点选择

当对角线端点出现不同的读取延迟时，两个通过区域由亚稳态间隙隔开。算法通过寻找具有最少连续通过点的点，并测试多个相邻点以提高稳定性，从而对角点进行优化。对于每个有效点，算法识别第一个失效点并确认最小连续失效点，以建立清晰的边界。没有足够连续通过点的区域被消除为不稳定。算法沿对角线向上搜索最小读取延迟，向下搜索最大读取延迟。

优化后，算法识别通过点数量最多的线。如果任何一条线长度超过最小通过长度，则会继续进行调优点选择；否则，将报告校准失败。

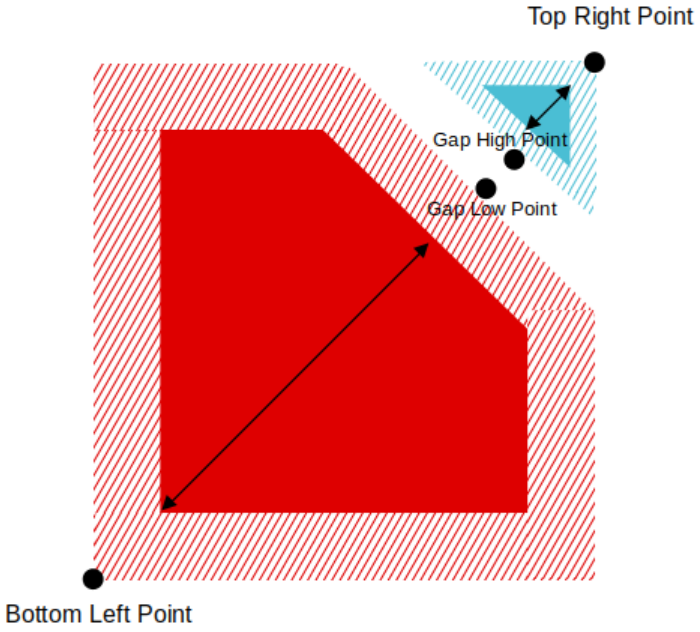

图 7-5. 用于两个不同读取延迟值的角点选择

7.1.4 调优点选择

该算法选择长度较长的区域并计算中点（中点 1）。识别出中点 1 处垂直于对角线的线角点，这些角点代表连续的通过点。计算垂直线的中点（中点 2），代表稳定区域的中心宽度。半径验证测试中点 2 周围指定半径的圆内的所有点是否均通过。如果成功，中点 2 将成为最佳调优点。

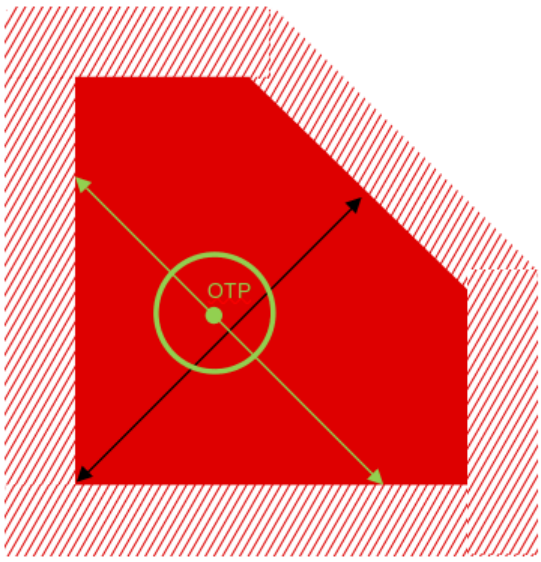

图 7-6. 中点处的调优点选择

如果半径验证失败，垂直线将在中间点 1 分为两个线段。对较长线段的中点（中点 3）进行半径验证。如果成功，中点 3 将成为调优点。

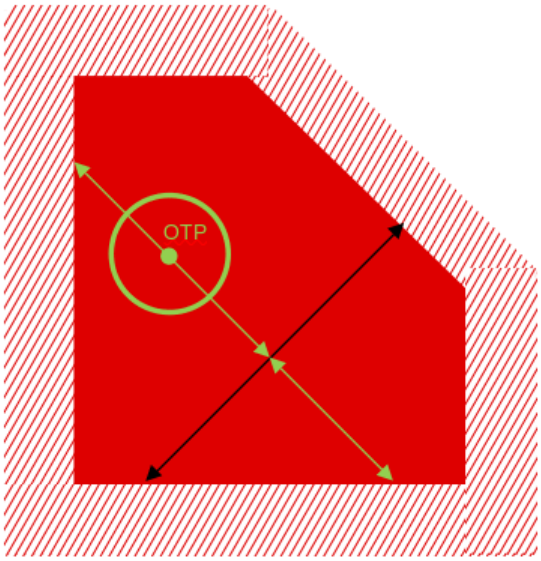

图 7-7. 较长线段中点处的调优点选择

如果中点 2 和中点 3 均失败，则当第二读延迟区域的长度超过最小通过长度时，对该区域进行评估，重复选择过程。

如果所有区域和段均失败，算法报告调优失败，并按照预定义模式选择其他对角线，以实现全面覆盖。每个对角线代表不同的延迟参数

组合，为寻找稳定工作区域提供多次机会。

对于 PHY 失败，请参阅*常见启动问题和调试*应用手册中的 PHY 失败一节。

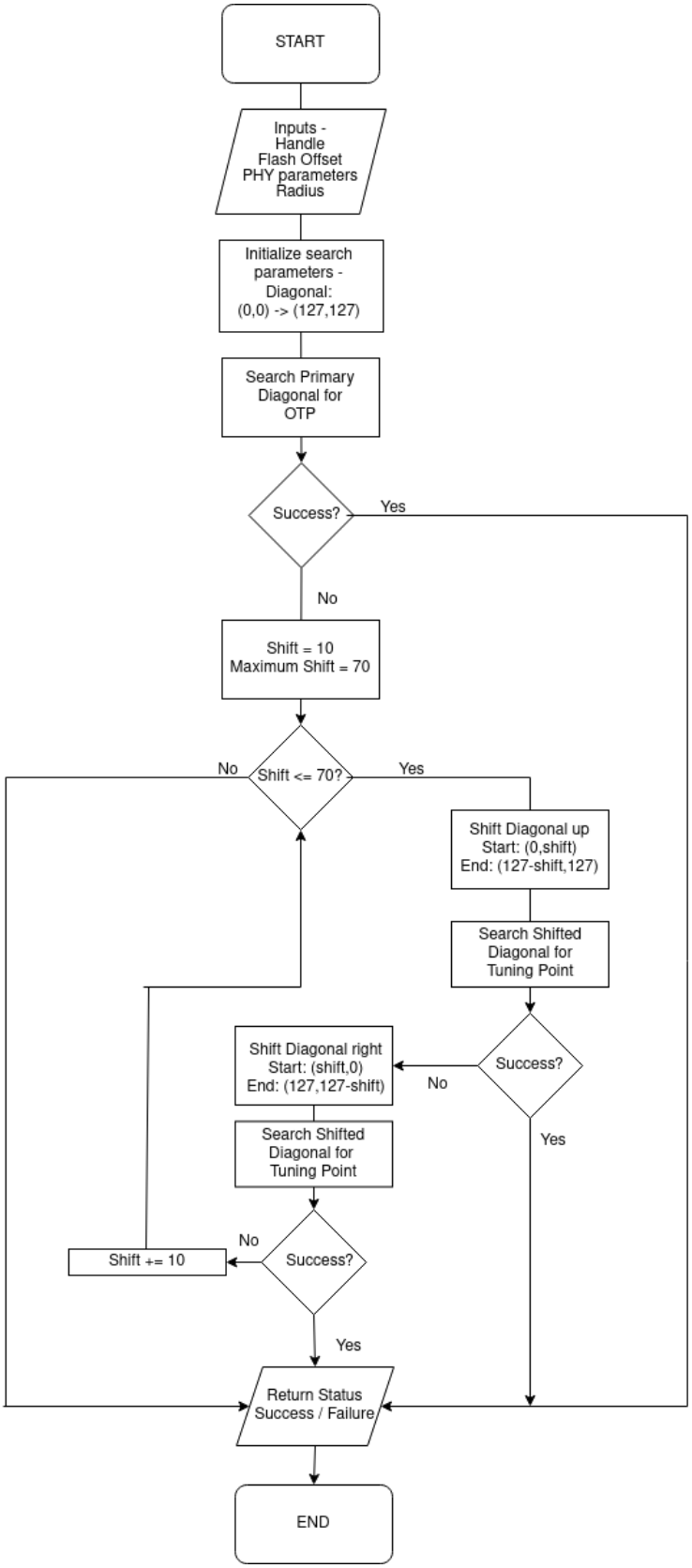

图 7-8. DQS 调优算法流程图

7.2 非 DQS PHY 调优算法

非 DQS 调优算法使用基于窗口的简化方法，该方法专为无 DQS 信号支持的低速运算而设计。与 DQS 算法中的二维对角线搜索不同，非 DQS 算法沿着 Rx DLL 轴执行一维搜索，同时保持固定的 Tx DLL 值。

7.2.1 固定 Tx DLL 值

该算法首先将 Tx DLL 设置为预先确定的高值。这种固定配置为大多数支持非 DQS 模式的闪存器件提供了足够的建立时间，并将搜索问题从两个维度减少到一个维度。通过消除优化 Tx DLL 的需要，该算法现在可以专注于寻找最合适的 Rx DLL 时序。

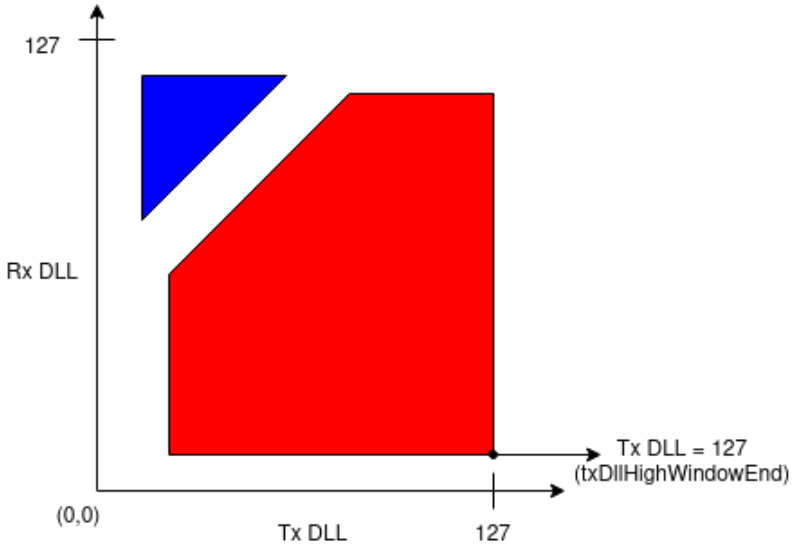

图 7-9. 固定 Tx DLL 值

7.2.2 查找 Rx 窗口 1

从最小读取延迟值开始，算法会搜索第一个有效的 Rx DLL 窗口。在每一步测试攻击向量读取时，Rx DLL 从最小值递增。读取成功的第一个 DLL 值成为窗口开始 (rxStart1)，失败恢复前的最后一个值成为窗口结束 (rxEnd1)。窗口大小的计算公式为 rxWindow1 = rxEnd1 - rxStart1。

如果在当前读取延迟找不到通过窗口，算法会增加读取延迟并重复搜索。此过程会一直持续，直到找到窗口或超过最大读取延迟。通过窗口表示接收 DLL 值的连续范围，其中接口时序满足所有建立和保持要求。

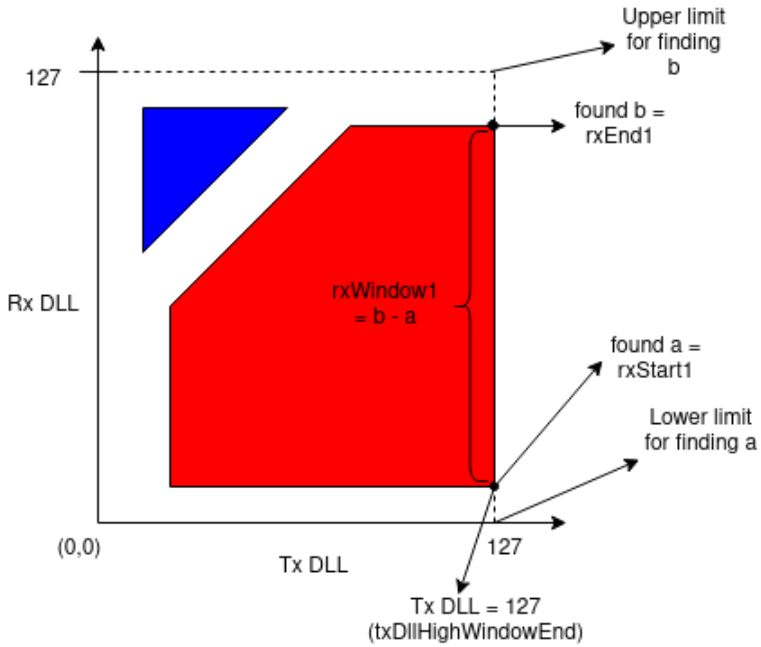

图 7-10. 选择 Rx 窗口 1

7.2.3 查找 Rx 窗口 2

识别第一个窗口后，算法尝试在下一个较高的读取延迟值（窗口 1 + 1 的读取延迟）处定位第二个窗口。重复相同的搜索过程以查找 rxStart2 和 rxEnd2，第二个窗口大小的计算公式为 rxWindow2 = rxEnd2 - rxStart2。

如果没有找到第二个窗口，rxWindow2 将设置为 0，这不会被视为失败。不同的读取延迟值会使基准时钟和数据采样点之间的相对时序发生偏移。测试多个读取延迟值，以便算法找到最大的可用窗口并提供针对环境变化的最大裕度。

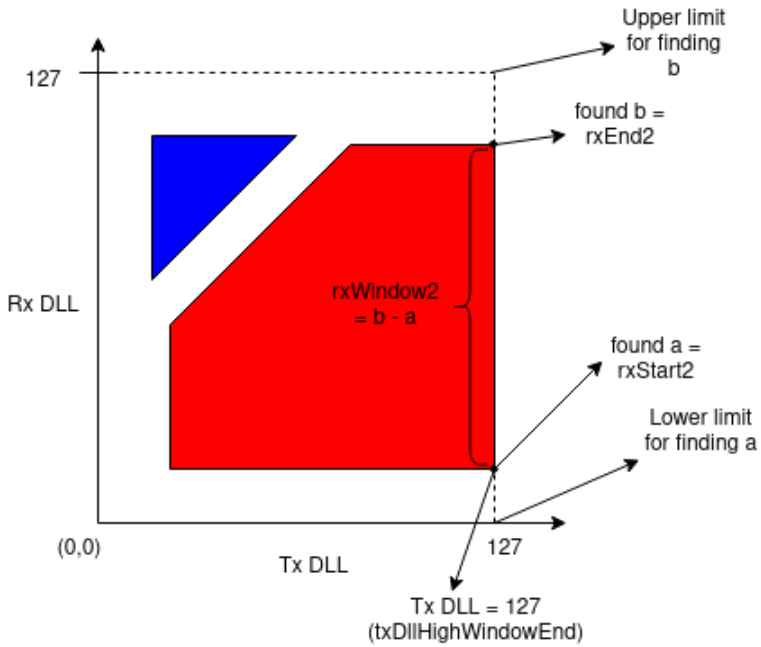

图 7-11. 选择 Rx 窗口 2

7.2.4 选择更大的 Rx 窗口

该算法会比较两个窗口大小并选择较大的窗口。如果 rxWindow2 大于 rxWindow1，则选择窗口 2 及其相关参数（rxStart2、rxEnd2 和读取延迟）。否则使用窗口 1。窗口越大表示时序裕度越大，对温度变化、电压波动和制造变化的容忍度越高。

7.2.5 计算 OTP

最终调优点由三个参数决定：

1. 传输 DLL：第 1 步中的固定值。
2. 读取延迟：与所选窗口关联的读取延迟。
3. 接收 DLL：所选窗口的中点 = rxStart + (rxEnd - rxStart)/2。

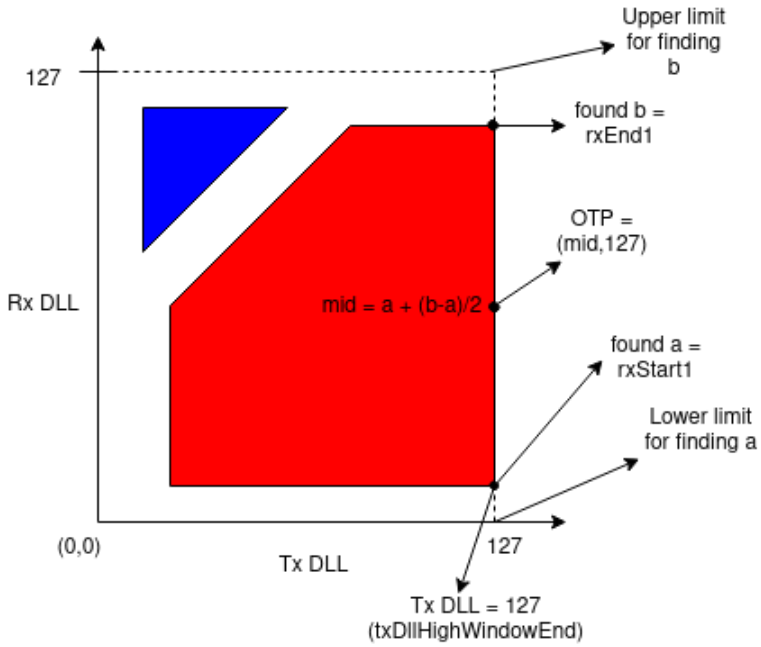

图 7-12. OTP 计算

7.2.6 温度注意事项

SoC 的内部温度传感器（即 VTM 模块）测量当前的芯片温度。算法使用 42.5°C 的参考温度来计算与温度相关的偏移：

温度因数 = (当前温度 - 42.5°C)/165°C × 窗口大小 × 0.75

然后，调整后的接收 DLL 等于中点和温度因数之间的差值。在较高的温度下，由于延迟会随着温度的升高而增加，调优点会向较低的接收 DLL 值偏移。在较低的温度下，由于延迟减少，调优点会向较高的值偏移。

计算调优点后，参数将应用于 PHY 硬件寄存器。在算法返回成功之前，攻击向量的验证读取会确认调优成功。

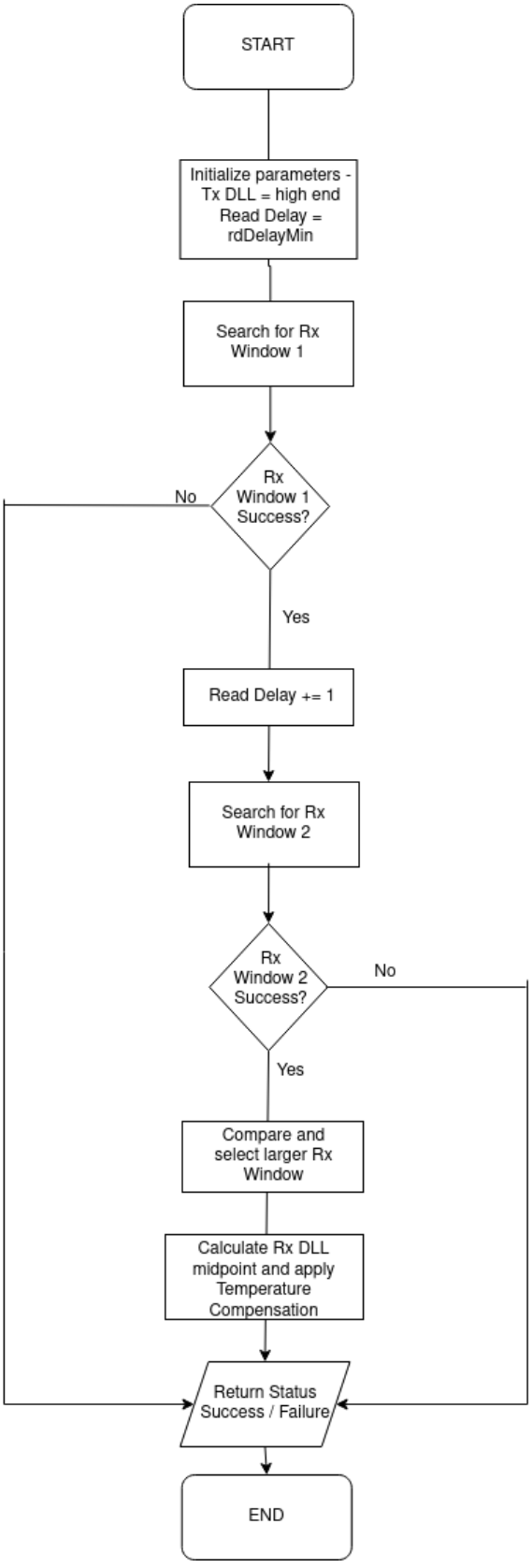

图 7-13. 非 DQS 调优算法流程图

8 调优增强功能

8.1 调优时间优化 – 跳过调优特性

MCU+ SDK 11.02 具有跳过 PHY 调优的功能。此特性可减少后续级的引导时间。对于具有多级引导过程的系统，可以在第一级（例如在主引导加载程序中）执行一次调优，并通过执行以下步骤在后续级重复使用：

1. 第一引导级执行完整的调优，并将参数值存储在硬件寄存器中。
2. 当在 SysConfig 中启用跳过调优时，后续级会检索驱动器直接应用已保存的参数。

8.2 运行时验证 – 验证 OTP

MCU+ SDK 11.02 具有验证 OTP 的功能。在 SysConfig 中启用后，应用会在启用 PHY 的闪存读取操作期间重新验证调优点。验证过程通过执行以下步骤来完成：

1. 在闪存读取操作期间，如果在 SysConfig 中启用 OTP 验证，则会自动进行调用。
2. 该特性在配置半径的圆形区域内围绕当前调优点执行对角线检查。
3. 如果验证失败，PHY 调优算法会自动重新运行。

9 总结

OSPI PHY 调优算法自动校准时序参数，以实现可靠的高速闪存访问。DQS 模式使用对角线搜索策略，该策略可以高效地表征通过区域、计算最佳中点，并通过圆形区域测试验证裕度。非 DQS 模式使用更简单的、带有温度补偿的一维窗口搜索。最新的 DQS 调优算法可对影响半导体时序特性的制造变化和环境变化进行补偿。工程师可以使用跳过调优来进行优化，以加快引导速度，或对生产进行预表征。

10 参考资料

1. 德州仪器 (TI)，*AM62x MCU+ SDK* 。
2. 德州仪器 (TI)，*OSPI 控制器 PHY 调优算法*，应用手册
3. 德州仪器 (TI)，*OSPI PHY 调优算法*，Github 存储库
4. 德州仪器 (TI)，*MCU+ SDK 上 OSPI 和 QSPI 串行 NOR/NAND 闪存操作的吞吐量特性* ，应用手册
5. 德州仪器 (TI)，*MCU+SDK 的 xSPI 定制闪存调试指南* ，应用手册

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 TI 销售条款)、TI 通用质量指南 或 ti.com 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。 除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

最后更新日期：2025 年 10 月