

1 Gbit (128 MByte)/512 Mbit (64 MByte), GL-T MirrorBit® Eclipse™ 闪存非易失性存储器系列

概述

赛普拉斯® S29GL01GT/512T 是基于 45 nm 工艺技术的 MirrorBit Eclipse 闪存产品。这些设备提供 15 ns 的快速页面读取时间，相应的随机访问时间可达 100 ns。它们配备写入缓冲器，在一个操作中最多可编程 256 字 /512 字节，与标准编程算法相比，有效编程时间更短。因此，对于当今需要更高容量、更好性能和更低功耗的嵌入式应用程序来说，这些设备是理想的选择。

特殊特性

- 45 nm MirrorBit Eclipse 技术
- 单电源 (V_{CC}) 为读取 / 编程 / 擦除操作供电 (2.7 V ~ 3.6 V)
- 多用途 I/O 功能
 - 宽 I/O 电压范围 (V_{IO}): 1.65 V ~ V_{CC}
- x8/x16 数据总线
- 异步 32 字节页面读取
- 512 字节编程缓冲器
 - 多页面编程，最多 512 字节
- 支持单字编程和对同一字多次编程 (比特编程操作)
- 扇区擦除
 - 统一的 128 kb 扇区
- 编程和擦除操作的挂起和恢复命令
- 可通过状态寄存器、数据轮询和就绪 / 忙碌引脚等方法确定设备状态
- 高级扇区保护 (ASP)
 - 为每个扇区提供易失性和非易失性的保护方法
- 独立的 2048 字节一次性编程 (OTP) 区域
 - 四个可锁定区域 (SSR0 - SSR3)
 - SSR0 为出厂锁定
 - SSR3 为密码读取保护
- 通用闪存接口 (CFI) 参数表
- 温度范围:
 - 工业级 (-40°C ~ +85°C)
 - 扩展的工业级 (-40°C ~ +105°C)
 - 扩展 (-40°C ~ +125°C)
- 耐久性
 - 工业级和扩展的工业级: 通常情况下, 可对任意扇区进行 100,000 次擦除操作
 - 扩展: 通常可对任意扇区进行 10,000 次擦除操作
- 通常情况下, 数据保持时间为 20 年
- 封装选项
 - 56 引脚 TSOP
 - 64 球形焊盘 LAA 加固的 BGA, 13 mm x 11 mm
 - 64 球形焊盘 LAE 加固的 BGA, 9 mm x 9 mm
 - 56 球形焊盘 VBU 加固的 BGA, 9 mm x 7 mm

性能总结

工业级温度范围 (-40°C ~ +85°C) 的性能总结

最大读取访问时间					
容量	电压范围	随机访问时间 (t _{ACC})	页面访问时间 (t _{PACC})	CE# 访问时间 (t _{CE})	OE# 访问时间 (t _{OE})
512 (512 Mb)	V _{CC} 的最大值 = V _{IO}	100	15	100	25
	多用途 I/O 的 V _{IO}	110	25	110	35
1 Gb	V _{CC} 的最大值 = V _{IO}	100	15	100	25
	多用途 I/O 的 V _{IO}	110	25	110	35

扩展的工业级温度范围 (-40°C ~ +105°C) 的性能总结

最大读取访问时间					
容量	电压范围	随机访问时间 (t _{ACC})	页面访问时间 (t _{PACC})	CE# 访问时间 (t _{CE})	OE# 访问时间 (t _{OE})
512 (512 Mb)	V _{CC} 的最大值 = V _{IO}	110	15	110	25
	多用途 I/O 的 V _{IO}	120	25	120	35
1 Gb	V _{CC} 的最大值 = V _{IO}	110	15	110	25
	多用途 I/O 的 V _{IO}	120	25	120	35

扩展温度范围 (-40°C ~ +125°C) 的性能总结

最大读取访问时间					
容量	电压范围	随机访问时间 (t _{ACC})	页面访问时间 (t _{PACC})	CE# 访问时间 (t _{CE})	OE# 访问时间 (t _{OE})
512 (512 Mb)	V _{CC} 的最大值 = V _{IO}	120	15	120	25
	多用途 I/O 的 V _{IO}	130	25	130	35
1 Gb	V _{CC} 的最大值 = V _{IO}	120	15	120	25
	多用途 I/O 的 V _{IO}	130	25	130	35

典型编程和擦除速率

操作	-40°C ~ +85°C	-40°C ~ +105°C	-40°C ~ +125°C
缓冲区编程 (512 B)	1.14 MB/s	1.14 MB/s	1.14 MB/s
扇区擦除 (128 KB)	245 KB/s	245 KB/s	245 KB/s

最大电流消耗

操作	-40°C ~ +85°C	-40°C ~ +105°C	-40°C ~ +125°C
读取 (5 MHz, 30 pF)	60 mA	60 mA	60 mA
编程	100 mA	100 mA	100 mA
擦除	100 mA	100 mA	100 mA
待机	100 μA	200 μA	200 μA

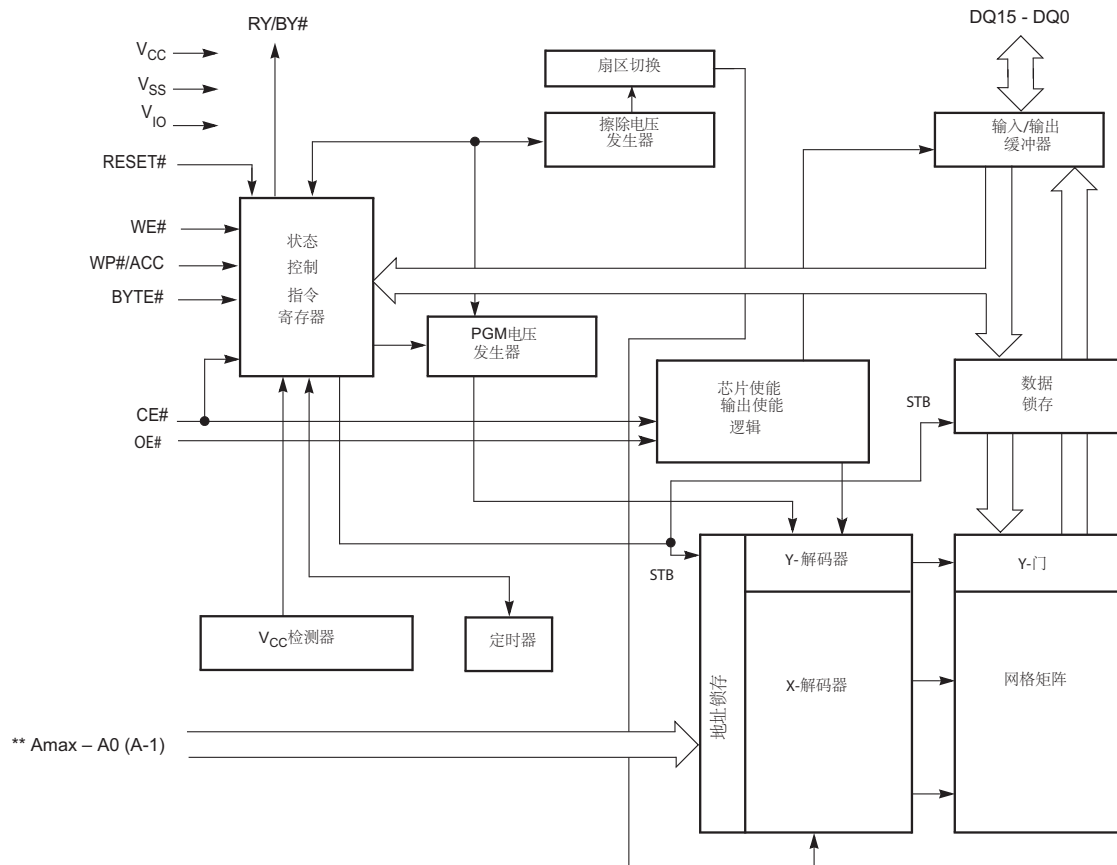
目录

概述	1	9.5 电容特性	67
特殊特性	1	10. 时序规范	68
性能总结	2	10.1 波形切换	68
1. 产品概述	4	10.2 交流测试条件	68
软件接口		10.3 上电复位 (POR) 和热复位	69
2. 地址空间重叠模式	6	10.4 交流特性	71
2.1 闪存存储器阵列	7	11. 物理接口	84
2.2 器件 ID 和 CFI (ID-CFI) ASO	8	11.1 56-TSOP	84
2.3 状态寄存器 ASO	9	11.2 64 球 FBGA	86
2.4 数据轮询状态 ASO	9	11.3 56 球 FBGA	89
2.5 安全硅区域 ASO	9	12. 关于 FBGA 封装的特殊处置说明	91
2.6 扇区保护控制	10	13. 订购信息	91
3. 数据保护	11	14. 其他资源	93
3.1 器件保护方法	11	14.1 软件链接	93
3.2 指令保护	11	14.2 应用笔记链接	93
3.3 安全硅区域 (OTP)	11	14.3 规范公告	93
3.4 扇区保护方法	11	15. 文档修订记录	94
4. 读取操作	16		
4.1 异步读取	16		
4.2 页模式读取	16		
5. 嵌入式操作	17		
5.1 嵌入式算法控制器 (EAC)	17		
5.2 编程和擦除简介	17		
5.3 指令集	19		
5.4 状态监控	31		
5.5 错误类型和清除步骤	37		
5.6 嵌入式算法性能表	40		
6. 软件接口参考	43		
6.1 命令汇总	43		
6.2 器件 ID 和通用闪存接口 (ID-CFI) ASO 映射	51		
硬件接口			
7. 信号描述	56		
7.1 地址和数据配置	56		
7.2 输入 / 输出简介	56		
7.3 字 / 字节配置	57		
7.4 多用途 I/O 功能	57		
7.5 就绪 / 忙碌 # (RY/BY#)	57		
7.6 硬件复位	57		
8. 信号协议	58		
8.1 接口状态	58		
8.2 关闭电源 (硬件数据保护)	59		
8.3 节能模式	59		
8.4 读取	59		
8.5 写入	60		
9. 电气规范	61		
9.1 最大绝对额定值	61		
9.2 闩锁特性	61		
9.3 工作范围	61		
9.4 直流特性	64		

1. 产品概述

GL-T 系列包括 512 Mb 到 1 Gb、3.0 V 内核、多用途 I/O、非易失性、闪存存储器器件。这些器件具有一个 8 位（字节）/16 位（字）带宽的数据总线，并且只使用了一个字节 / 字宽的地址边界。所有读取访问在每个总线传输周期提供 8/16 位数据。所有写入在每个总线传输周期提取 8/16 位数据。

图 1.1 框图



注意:

** Amax GL01GT = A25, Amax GL512T = A24。

GL-T 系列结合了 eXecute In Place（芯片内执行，XIP）和 Data Storage（数据存储）闪存的最佳特性。在拥有大容量、快速编程的数据存储闪存的基础上，该系列产品还能对 XIP 闪存进行快速的随机访问。

对任意随机位置的读取访问需要 100 ns 到 120 ns，具体按器件容量和 I/O 电源电压而定。每次随机（初始）访问时读取整个以 32 字节排列的组数据（称为页面）。读取同一页面内的其他字时，更改字地址的低 4 位即可。同一页面内的每次读取需要 15 ns 到 25 ns。这种操作被称为页面模式读取。更改字地址的高位时，将选择不同的页面并开始新的初始读取。所有读取访问均为异步。

表 1.1 S29GL-T 地址映射

类型	x16		x8	
	数量	地址	数量	地址
页面内的地址	16	A3 - A0	32	A3 - A-1
写入缓冲区内的地址	256	A7 - A0	256	A6 - A-1
页	4096	A15 - A4	4096	A15 - A4
写入 - 缓冲区 - 行	256	A15 - A8	512	A15 - A7
扇区	1024 (1 Gb) 512 (512 Mb)	Amax - A16	1024 (1 Gb) 512 (512 Mb)	Amax - A16

器件控制逻辑分成两个并行的操作区，**Host Interface Controller**（主机接口控制器，**HIC**）和 **Embedded Algorithm Controller**（嵌入式算法控制器，**EAC**）。**HIC** 监控器件输入的信号电平，并根据需要驱动输出，以完成主机系统内的读取和写入数据传输。**HIC** 在读取传输时将传递当前所在的地址空间中的数据；将写入传输地址和数据信息放入 **EAC** 指令存储器内；告知 **EAC** 电源转换、硬件复位以及写入传输的情况。**EAC** 在写入传输后查看指令存储器中的合法指令序列，并执行相关的嵌入式算法。

更改存储器阵列中的非易失性数据时，需要执行复杂的操作序列，这些操作被称为嵌入式算法（**EA**）。这些算法完全由器件内部的 **EAC** 来管理。主算法执行主阵列数据的编程和擦除。主机系统将指令代码写入到闪存器件地址空间内。**EAC** 接收指令用于执行所有必要的步骤以完成指令，并在 **EA** 执行期间提供状态信息。

每个存储器位的擦除状态为一个逻辑 **1**。编程操作会将逻辑 **1**（高电平）修改为逻辑 **0**（低电平）。只有通过擦除操作才可以将逻辑 **0** 修改为逻辑 **1**。擦除操作必须在整个扇区（即 **128 kB** 对齐的数据组）内执行。从赛普拉斯出厂时，所有扇区均为擦除状态。

编程通过一个 **512** 字节的写入缓冲区完成。开始编程操作前，在 **x16** 模式下，可以向写入缓冲区内的任意位置上写入 **1** 到 **256** 个字。在闪存阵列内，每个以 **512** 字节排列的 **512** 字节组称为行。开始编程操作之前，在 **x8** 模式下，可以向写入缓冲区内（**A7 = 0** 或 **A7 = 1**）任意位置写入 **1** 到 **256** 个字节。编程操作将易失性数据从写入缓冲区传输到非易失性存储器阵列行中。该操作称为写入缓冲区编程。

复位后或使用写入缓冲区完成任何操作后，写入缓冲区的内容被置 **1**。对于“写入到缓冲区”指令未写入 **0** 的位置，在默认情况下仍为 **1**。在编程操作期间，写入缓冲区中的任何 **1** 都不会影响存储器阵列中的数据。

加载到写入缓冲区中的每个数据页面都被传输到存储器阵列行中。

Advanced Sector Protection（高级扇区保护，**ASP**）功能集可以分别实现对扇区的擦和写保护。**ASP** 提供多种硬件和软件控制的、易失性和非易失性的方法，来选择需要擦写保护的扇区。

软件接口

2. 地址空间重叠模式

在闪存存储器器件的地址范围内，可能出现多个单独的地址空间。在任意给定时间，只有一个地址空间可见（进入）。

- 闪存存储器阵列：主非易失性存储器阵列，用于存储可通过异步读取操作进行随机访问的数据。
- ID/CFI 区域：用于存储赛普拉斯工厂预编程的器件特性信息。该区域包含器件标识（ID）和通用闪存接口（CFI）信息表。
- 安全硅区域（SSR）：一次性可编程（OTP）非易失性存储器阵列用于存储赛普拉斯工厂预编程的永久性数据以及客户可编程的永久性数据。
- 锁定寄存器：一个 OTP 非易失性字，用于配置 ASP 功能和锁定 SSR。
- 持久保护位（PPB）：非易失性闪存存储器阵列（每扇区一位）。编程后，每一位都会保护相关的扇区，以防止擦除和编程。
- PPB 锁定位：它是一个易失性寄存器位，用于使能或禁用对 PPB 位进行的编程和擦除。
- 阵列密码：它是一个 OTP 非易失性阵列，可存储 64 位密码，允许在使用密码模式扇区保护时更改 PPB 锁定位的状态。
- SSR3 密码：它是一个 OTP 非易失性阵列，可存储 64 位密码，并允许读取 SSR3。
- 动态保护位（DYB）：它是易失性阵列（每扇区一位）。设置该位后，每一位都会保护相关的扇区，以防止进行擦除和编程。
- 状态寄存器：它是一个易失性寄存器，用于显示嵌入式算法的状态。
- 数据轮询状态：它是一个易失性寄存器，用作显示嵌入式算法状态的备用方式，与旧版软件兼容。

主闪存存储器阵列是主要同时也是默认的地址空间，但随时会被另一个地址空间覆盖掉。每个备用地址空间都被称为地址空间重叠（ASO）。

每个 ASO 取代（重叠）闪存器件的整个地址空间。没有被特定 ASO 地址映射定义的地址范围将留作将来使用。ASO 地址映射之外的所有读取访问均返回无效（未定义）的数据。读取地址映射之外的位置取得的数据，无论是 1 还是 0，都未定义其含义。

器件工作模式有四种，可以确定闪存器件地址空间在任何特定时刻的内容：

- 读取模式
- 数据轮询模式
- 状态寄存器（SR）模式
- 地址空间重叠（ASO）模式

在读取模式下，主机系统存储控制器可能直接读取整个闪存阵列。存储器件的嵌入式算法控制器（EAC）在加电期间、硬件复位后、指令复位后以及嵌入式算法（EA）挂起后，都会将器件置于读取模式。在读取模式下，器件可接受读取访问和指令写入。当 EA 挂起时，在读取模式下器件可接受部分指令。

在任一模式下，都可以发出状态寄存器读取指令，从而在器件地址空间中的每个字地址都出现状态寄存器 ASO。在该状态寄存器 ASO 模式下，器件接口将等待读取访问，写入访问被忽略。随后的一次器件读取访问，读取状态寄存器的内容并退出状态寄存器 ASO，然后返回之前收到状态寄存器读取指令时的（调用）模式。

在 EA 模式下，EAC 执行嵌入式算法，如编程或擦除非易失性存储器阵列。在 EA 模式下，闪存器件的整个地址空间被数据轮询状态 ASO 取代，主闪存阵列不可读。器件地址空间中的每个字位置均出现数据轮询状态。

在 EA 模式下，只接受编程挂起 / 擦除挂起指令或状态寄存器读取指令。所有其他指令均被忽略。因此，无法从 EA 模式进入其他 ASO。

当嵌入式算法挂起时，在器件挂起 EA 前，数据轮询 ASO 一直可见。当 EA 挂起时，数据轮询 ASO 退出，闪存阵列数据可用。当挂起的 EA 得到恢复时，将重新进入数据轮询 ASO，并一直持续到 EA 重新挂起或完成为止。当嵌入式算法完成时，数据轮询 ASO 退出，器件进入先前（调用）模式（嵌入式算法在该模式下启动）。

在 ASO 模式下，进入其中一个剩余的重叠地址空间（覆盖主闪存阵列地址映射）。在任意特定时间内，只能进入一个 ASO。对器件执行的指令对当前进入的 ASO 产生影响。每个 ASO 都有自己特定的有效指令。它们被罗列在第 43 页上的表 6.1 中每个 ASO 的相关章节内。

下列 ASO 包含非易失性数据，可通过编程将 1 改为 0：

- 安全硅区域
- 锁定寄存器
- 持久保护位（PPB）
- 密码
- 只有 PPB ASO 中包含非易失性数据，并可通过擦除该数据将 0 改为 1。

在进入其中一个非易失性 ASO 后发出编程或擦除指令时，EA 将对该 ASO 执行操作。当 EA 处于活动状态时，该 ASO 不可读。EA 完成后，ASO 仍留在进入状态，但重新变成可读。在任意 ASO 的 EA 过程中，挂起和恢复指令无效。

2.1 闪存存储器阵列

S29GL-T 系列具有统一的扇区架构，扇区大小为 128 kB。下表显示的是不同器件的扇区架构。

表 2.1 S29GL01GT 扇区和存储地址映射

扇区容量（千字节）	扇区数量	扇区范围	地址范围（16 位）	地址范围（8 位）	注释
128	1024	SA0	0000000h-000FFFFh	0000000h-001FFFFh	扇区起始地址
		:	:	:	-
		SA1023	3FF0000h-3FFFFFFh	7FF0000h-7FFFFFFh	扇区结束地址

表 2.2 S29GL512T 扇区和存储地址映射

扇区容量（千字节）	扇区数量	扇区范围	地址范围（16 位）	地址范围（8 位）	注释
128	512	SA0	0000000h-000FFFFh	0000000h-001FFFFh	扇区起始地址
		:	:	:	-
		SA511	1FF0000h-1FFFFFFh	3FF0000h-3FFFFFFh	扇区结束地址

注意： 这些表只是在一个页面上简单列出了整个器件的扇区相关信息。对于未明确列出的扇区以及它们的地址范围（如 GL512T 上的 SA1-SA510），其扇区起始地址和结束地址与该尺寸的所有其他扇区具有相同的组合。例如，所有 128 kb 扇区的组合均为 XXX0000h-XXXFFFFh（在 x16 模式下）和 XXX0000h-XXX1FFFF（在 x8 模式下）。

2.2 器件 ID 和 CFI (ID-CFI) ASO

系统可以通过两种传统方法识别系统中已经安装的闪存类型。一种被传统定义为“自动选择”，现在指的是器件标识 (ID)。另一种方法称为通用闪存接口 (CFI)。

对于 ID，使用一条指令来启用一个地址空间重叠，最多可从这个空间内读取 16 字位置，用于从闪存中获取 JEDEC 制造商标识 (ID)、器件 ID 和一些配置及保护状态信息。系统可以使用制造商和器件 ID 为闪存器件选择相应的驱动程序软件。

CFI 也使用一个指令来使能一个地址空间重叠，从其中读取关于闪存组织和操作的标准信息的详细表。通过该方法，在编写驱动程序软件时，便不用再熟知每种可能的存储器件细节。在编写驱动程序软件时，只需根据 CFI 表中的信息来调整驱动程序的运行方式即可，无需按照常规方式来处理多种不同的器件。

传统上，这两个地址空间为不同的重叠，分别使用单独的指令。不过，这两个地址空间的映射是非重叠的，因此可以组合成一个地址空间并一同出现在一个重叠中。进入“自动选择” (ID) 或 CFI 映射的任意一条传统指令都可以进入现在组合的 ID-CFI 地址映射。

ID-CFI 地址映射可能会覆盖掉整个闪存阵列。

ID-CFI 地址映射起始于所选扇区的 0 位置。对于从所定义的 ID-CFI ASO 最大地址到所选扇区最大地址之间的位置，其数据未被定义。ID-CFI 进入指令使用与前一代存储器相同的地址和数据值分别用来读取 JEDEC 制造商 ID (自动选择) 和通用闪存接口 (CFI) 信息。

表 2.3 ID-CFI 地址映射概览

字地址	字节地址	说明	读取 / 写入
(SA) + 0000h 到 000Fh	(SA) + 0000h 到 001Fh	器件 ID (传统的“自动选择”值)	只读
(SA) + 0010h 到 0079h	(SA) + 0020h 到 00F2h	CFI 数据结构	只读
(SA) + 0080h 到 FFFFh	(SA) + 00F3h 到 1FFFFh	未定义	只读

有关完整的地址映射信息，请参见第 51 页上的表 6.3。

2.2.1 器件 ID

美国电子工程设计发展联合会 (JEDEC) 标准 JEP106T 定义了兼容存储器的制造商 ID。通用行业用法定义了从存储器件读取制造商 ID 和器件特定 ID 的方法和格式。制造商和器件 ID 信息的主要目的是为了使编程装置自动匹配器件和相应的编程算法。赛普拉斯在此 32 字节地址空间中增加了更多字段。

初始行业格式的结构适合任一内存数据总线宽度，如 x8、x16、x32。传统上 ID 代码值为字节宽度，但位于总线带宽地址边界。因此，器件地址输入递增时，将读取连续的字节、字或双字位置，并且 ID 代码始终位于数据总线上最低有效字节的位置。由于器件数据总线为字宽度，因此每个代码字节分别位于每个字位置的下半部。基础行业格式要求高字节始终为 0。赛普拉斯已经修改了该格式，从而能在地址控制的某些字中使用两种字节。有关器件 ID 地址映射的详细说明，请参见第 51 页上的表 6.3。

2.2.2 通用闪存接口 (CFI)

JEDEC 通用闪存结构 (CFI) 规范 (JESD68.01) 定义了可从闪存器件读取的标准化数据结构，它允许厂商在整个器件系列中使用其指定的软件算法。数据结构包含系统配置信息，如各种电气和时序参数以及器件支持的特殊功能。这样，软件支持就变得与器件和器件 ID 无关，并且对整个闪存器件系列前后兼容。

系统可以从选定扇区中各个地址读取 CFI 信息，如第 51 页上的器件 ID 和通用闪存接口 (ID-CFI) ASO 映射所示。

与器件 ID 信息相似，CFI 信息的结构也适合任一内存数据总线宽度，如 x8、x16、x32。代码值总是字节宽度，但位于数据总线带宽地址边界。因此，器件地址递增时，将读取连续的字节、字或双字位置，并且代码始终位于数据总线上最低有效字节的位置。由于数据总线为字宽度，因此每个代码字节分别位于每个字位置的下半部，并且高字节始终为 0。

更多有关信息，请参考 *Spansion CFI 规范，版本 1.4*（或更高版本）以及 *JEDEC 出版物 JEP137-A 和 JESD68.01*。欲了解 JEDEC 标准，请访问该组织的网站 <http://www.jedec.org>。如要了解 Spansion CFI 规范，请访问赛普拉斯公司网站：<http://www.spansion.com/Support/TechnicalDocuments/Pages/ApplicationNotes.aspx>（在发布本文档时提供），或者联系本公司网站上所提供的本地赛普拉斯经销商。

2.3 状态寄存器 ASO

状态寄存器 ASO 具有一个包含了嵌入式算法的易失性状态的单字寄存器值。发出状态寄存器读取指令时，寄存器将（在 WE# 的上升沿上）捕获当前的状态，并进入 ASO。状态寄存器的内容将会出现在所有字位置上。第一次读取访问使系统退出状态寄存器 ASO（在 CE# 或 OE# 的上升沿上），并将其返回到发出状态寄存器读取指令时所使用的地址空间映射。发出写入指令不会使器件退出状态寄存器 ASO 状态。

2.4 数据轮询状态 ASO

数据轮询状态 ASO 包含一个易失性存储器的单字，用于指明 EA 的进度。在任一启动 EA 的指令序列的最后一个写入周期完成时，立即进入数据轮询状态 ASO。启动 EA 的指令包括：

- 字编程
- 写入缓冲区到闪存
- 芯片擦除
- 扇区擦除
- 擦除恢复 / 编程恢复
- 编程恢复增强方法
- 空白检查
- 锁定寄存器编程
- 密码编程
- PPB 编程
- 所有 PPB 擦除
- 评估擦除状态

数据轮询状态字出现在器件地址空间中的所有字位置。当 EA 完成时，器件将退出数据轮询状态 ASO，器件地址空间则返回 EA 启动时的地址映射模式。

2.5 安全硅区域 ASO

安全硅区域（SSR）提供一个额外的存储器区域，该区域可编程一次并受到永久保护，以后不能更改。也就是说，它是一次性编程（OTP）区域。SSR 的长度是 2048 个字节。它包括大小为 512 个字节的工厂锁定的安全硅区域（SSR0）、大小为 1024 个字节的用户锁定的安全硅区域（SSR1 和 SSR2）以及大小为 512 个字节且带有读取密码的用户锁定的安全硅区域（SSR3）。

出厂时，SSR0 已经被锁定，用于防止发生意外的编程。SSR1 和 SSR2 都是 OTP，它们都带有单独的锁定位。一旦被锁定，则不能对这些区域进行任何更改。SSR3 也是一个 OTP，对该区域进行读取或编程时，需要一个 SSR3 密码。如果 SSR3 被锁定，便不能对该区域进行任何更改。

安全硅进入指令期间所提供的扇区地址将选择闪存存储器阵列扇区，该扇区将被安全硅区域地址映射覆盖掉。SSR 从所选扇区 0 位置开始覆盖。考虑到未来的兼容性，建议使用扇区 0 地址。进入 SSR ASO 后，所有其他扇区的内容变成用于读取的存储器内核数据。在 ASO 外，不能进行编程。

表 2.4 安全硅区域

字地址范围	字节地址范围	内容	区域	大小
(SA) + 0000h 到 00FFh	(SA) + 0000h 到 01FFh	工厂锁定的安全硅区域	SSR0	512 个字节
(SA) + 0100h 到 01FFh	(SA) + 0200h 到 03FFh	用户锁定的安全硅区域	SSR1	512 个字节
(SA) + 0200h 到 02FFh	(SA) + 0400h 到 05FFh	用户锁定的安全硅区域	SSR2	512 个字节
(SA) + 0300h 到 03FFh	(SA) + 0600h 到 07FFh	用户锁定的安全硅区域，带有读取密码	SSR3	512 个字节
(SA) + 0400h 到 FFFFh	(SA) + 0800h 到 1FFFFh	未定义	n/a	126 kB

2.6 扇区保护控制

2.6.1 锁定寄存器 ASO

锁定寄存器 ASO 包含 OTP 存储器的一个单字。进入 ASO 后，锁定寄存器出现在器件地址空间中的所有字位置。不过，考虑到未来的兼容性，建议只在器件地址空间 0 位置对锁定寄存器 ASO 执行读取或编程操作。

2.6.2 持久保护位 (PPB) ASO

PPB ASO 包含了器件中每个扇区的闪存阵列的一位。进入 PPB ASO 后，扇区的 PPB 位出现在扇区中每个地址的最低有效位 (LSB)。读取扇区中的任一地址时会显示数据，其中 LSB 指明该扇区的非易失性保护状态。不过，考虑到未来的兼容性，建议只在扇区的地址 0 上对 PPB ASO 执行读取或编程操作。该位是 0 时，扇区受到保护，不能对其执行编程和擦除操作。该位是 1 时，PPB 不保护扇区。扇区可能受到 ASP 其他功能的保护。

2.6.3 PPB 锁定 ASO

PPB 锁定 ASO 包含易失性存储器的一位。该位会决定 PPB ASO 中的各个位能否被编程或擦除。该位是 0 时，PPB ASO 受到保护，不能对其进行编程和擦除。该位是 1 时，PPB ASO 不受保护。进入 PPB 锁定 ASO 后，PPB 锁定位出现在器件地址空间中每个地址的最低有效位 (LSB)。不过，考虑到未来的兼容性，建议只在器件的地址 0 对 PPB 锁定 ASO 执行读取或编程操作。

2.6.4 密码 ASO

密码 ASO 包含 OTP 存储器的四个字。进入 ASO 后，密码出现在器件地址空间中从 0 地址开始的位置。第四个字之上的所有位置均为未定义。

2.6.5 动态保护位 (DYB) ASO

DYB ASO 包含器件中每个扇区的易失性存储器阵列的一位。进入 DYB ASO 后，扇区的 DYB 位出现在扇区中每个地址的最低有效位 (LSB)。读取扇区中的任一地址时会显示数据，其中 LSB 指明该扇区的易失性保护状态。不过，考虑到未来的兼容性，建议只在扇区的位置 0 上对 DYB 执行读取、设置或清除操作。该位是 0 时，扇区受到保护，不能对其进行编程和擦除。该位是 1 时，DYB 不保护扇区。扇区可能受到 ASP 其他功能的保护。

3. 数据保护

为防止通过硬件方式恶意或意外地修改任何扇区，器件提供了多项保护功能。

3.1 器件保护方法

3.1.1 加电写入禁止

在加电复位（POR）期间，RESET#、CE#、WE# 和 OE# 均被忽略。在 POR 期间，不能选择器件，在 WE# 的上升沿上器件不会接受任何指令，也不驱动输出。主机接口控制器（HIC）和嵌入式算法控制器（EAC）在 POR 期间复位至待机状态，准备读取阵列数据。在 POR 结束（ t_{VCS} ）前，CE# 或 OE# 的电压必须转为 V_{IH} 。

POR 结束时，器件状况如下：

- 所有内部配置信息已加载，
- 器件处于读取模式，
- 状态寄存器处于默认值，
- DYB ASO 中的所有位被设置为取消保护所有扇区，
- 写入缓冲区全部加载 1，
- EAC 处于待机状态。

3.1.2 低电平 V_{CC} 写入禁止

当 V_{CC} 小于 V_{LKO} 时，HIC 不会接受任何写入周期和 EAC 复位。这可在 V_{CC} 加电和断电期间保护数据。系统必须向控制引脚提供正确的信号，以防止当 V_{CC} 大于 V_{LKO} 时意外写入。

3.2 指令保护

将指令序列写入到 EAC 指令存储器中，从而启动嵌入式算法。指令存储器阵列没有 ASO，也不能被主机系统读取。每个主机接口写入是发给器件的一个指令或指令序列的一部分。EAC 检查每个写入传输中的地址和数据，以确定写入是否是合法指令序列的一部分。如果合法指令序列是完整的，EAC 将启动相应的 EA。

如果写入的地址或数据值不正确或者写入了错误的序列，一般会导致 EAC 返回其待机状态。不过，此类错误指令序列可能使器件进入未知状态。在这种情况下，系统必须写入复位指令，或者可能需要通过将 RESET# 信号置为低电平来提供一个硬件复位，才能使 EAC 返回其待机状态，并准备执行随机读取。

每个写入中提供的地址可能包含一个位组合，有助于识别该写入是否是发给器件的指令。地址的高位也选择了执行指令操作的扇区地址。扇区地址（S）包括 Amax 到 A16 闪存地址位（系统字节地址信号 Amax 到 A16）。指令位组合位于 A10 到 A0 闪存地址位（系统字节地址信号 A11 到 A1）。

每个写入中提供的数据可能是：位组合，有助于识别该写入是否是指令；代码，用于识别要执行的指令操作，或提供为执行操作而必需的信息。有关器件接受的所有指令的列表，请参见第 43 页上的表 6.1。

3.3 安全硅区域（OTP）

请参见第 9 页上的 2.5 节，安全硅区域 ASO，以便了解安全硅区域的说明。请参考第 29 页上的 5.3.9.3 节，安全硅区域 ASO，以便了解所允许的指令。

3.4 扇区保护方法

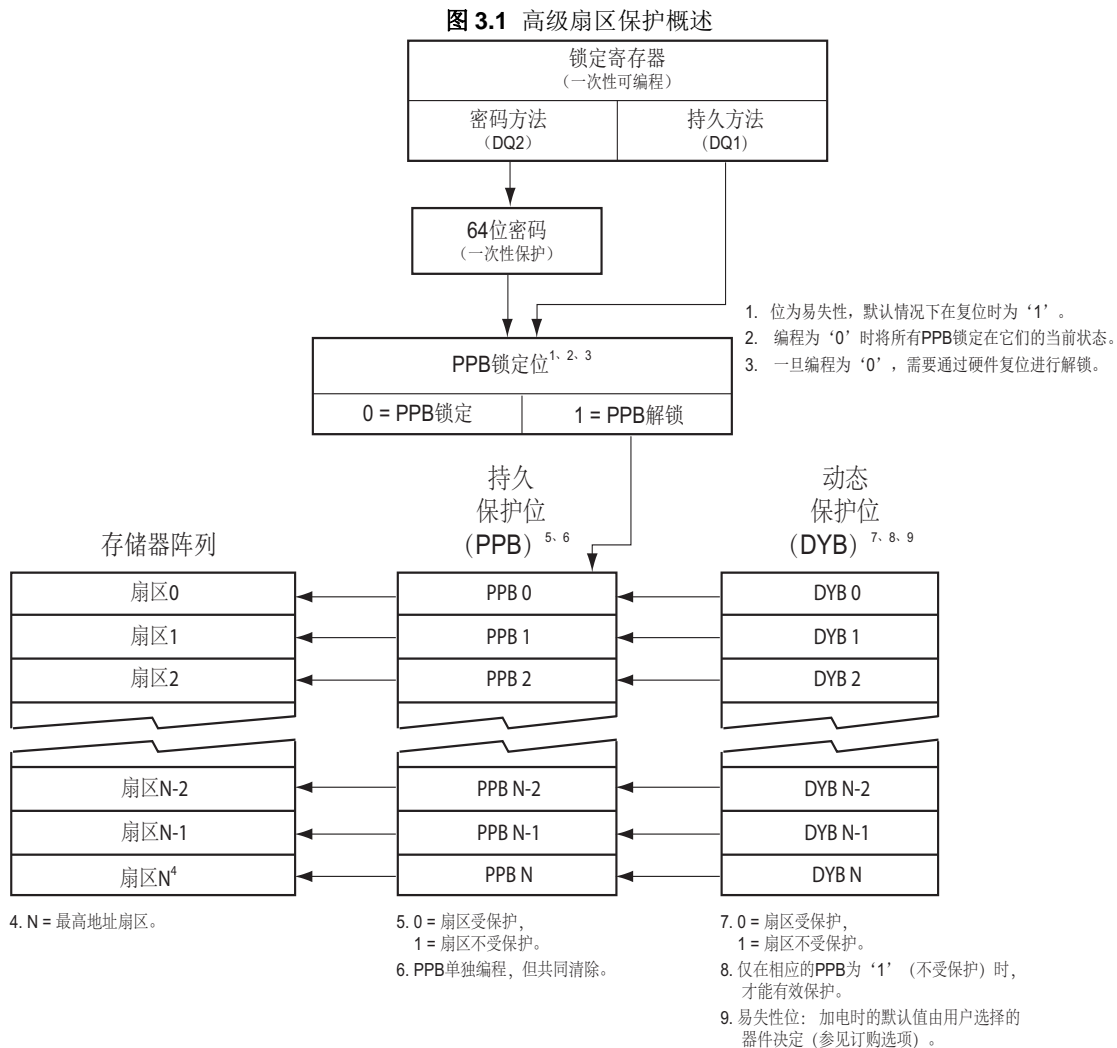
3.4.1 写入保护信号

$WP\# = V_{IL}$ 时，最低或最高地址扇区受到保护（与任何其他 ASP 配置无关），不能对这些扇区执行编程或擦除操作。受保护的扇区是最低扇区还是最高扇区则取决于所选择的器件订购选项（型号）。 $WP\# = V_{IH}$ 时，最低或最高地址扇区不受 $WP\#$ 信号的保护，

但可能受到其他 ASP 配置的保护。WP# 有一个内部上拉电阻；当断开连接时，WP# 将处于 V_{IH}。在执行任意嵌入式操作期间，WP# 不会在 V_{IL} 和 V_{IH} 间切换。

3.4.2 ASP

高级扇区保护（ASP）是一组独立的硬件和软件方法，分别用于禁止或允许对任何或所有扇区执行编程或擦除操作。此部分介绍存储器阵列中所存储数据的各种保护方法。图 3.1 是这些方法的概述。



每个主闪存阵列扇区都有与其有关的一个非易失性 (PPB) 和一个易失性 (DYB) 保护位。如果某位是 '0'，则全部扇区受到保护，从而不能对其执行编程和擦除操作。

PPB 锁定位为 '0' 时，不能执行编程和擦除操作。PPB 锁定位的状态可通过两种方法进行管理：持久保护和密码保护。

持久保护方法在 POR 或硬件复位期间将 PPB 锁定位设为 '1'，因此 PPB 位不会受到器件复位的保护。有一个指令可将 PPB 锁定位清除为 '0'，以保护 PPB。持久保护方法没有指令可设置 PPB 锁定位，因此 PPB 锁定位将一直为 '0'，直到下一次关闭电源或硬件复位为止。持久保护方法允许引导代码通过编程或擦除 PPB 来更改扇区保护，然后通过清除 PPB 锁定位在正常系统操作的剩余时间内保护 PPB，不让更多更改。这有时称为引导代码控制的扇区保护。

密码方法在 POR 或硬件复位期间将 PPB 锁定位清除为 ‘0’，以保护 PPB。对于密码方法，可以永久地编程并隐藏一个 64 位密码。通过一个指令来提供一个密码，将其与隐藏密码进行比较。如果密码匹配，则 PPB 锁定位设为 1，以取消 PPB 保护。PPB 锁定位可以通过一个指令来清除为 0。

通过编程锁定寄存器中的 OTP 位，可以永久性选择要使用的 PPB 锁定管理方法。

锁定寄存器还包含 OTP 位，用于保护 SSR。

从赛普拉斯出厂时，PPB 位被清除，因此所有主闪存阵列扇区不受保护。安全硅区域在工厂可能设成保护或不保护，具体视订购选项（型号）而定。

3.4.3 PPB 锁定

持久保护位锁定是一个易失性位，用于保护所有 PPB 位。清除为 0 时，它锁定所有 PPB；设成 1 时，允许更改 PPB。每个器件只有一个 PPB 锁定位。

PPB 锁定指令用于将该位清零。只有当所有 PPB 位均配置为所需的设置后，才将 PPB 锁定位清零。

在持久保护模式下，PPB 锁定在 POR 或硬件复位期间被设为 1。清除后，所有软件指令序列都不能设置 PPB 锁定，只有通过硬件复位或上电才能设置 PPB 锁定位。

在密码保护模式下，PPB 锁定在 POR 或硬件复位期间被清除为 0。PPB 锁定只能通过密码解锁指令序列设为 1。可以使用 PPB 锁定位清除指令将 PPB 锁定位清零。

3.4.4 持久保护位（PPB）

持久保护位（PPB）位于一个单独的非易失性闪存阵列中。为每个扇区分配一个 PPB 位。当一个 PPB 位为 0 时，相应的扇区受到保护，不能对它执行编程和擦除操作。PPB 位可单独编程，但必须按组进行擦除。这与字相似，各个字可以在主阵列中单独编程，但整个扇区必须同时擦除。擦除前的预编程和验证由 EAC 执行。

编程一个 PPB 位需要典型的字编程时间。在编程或擦除一个 PPB 位期间，数据轮询状态 DQ6 跳变位将持续跳变，直至操作完成。擦除所有 PPB 需要典型的扇区擦除时间。

如果 PPB 锁定位为 0，PPB 编程或擦除指令不被执行，并且被超时。

与特定扇区相关的 PPB 的保护状态，可通过在进入 PPB ASO 后执行一个 PPB 状态读取指令来检查。

3.4.5 动态保护位（DYB）

动态保护位是易失性位，可以单独更改该位。每个扇区只有一个唯一的动态保护位。DYB 只控制那些已清除 PPB 的扇区的保护。如果某个扇区的 PPB 位为 1，则通过发送 DYB 设置或清除指令序列，可分别将 DYB 设置为 0 或清除为 1，因此该扇区将相应进入保护或无保护状态。使用该功能，可以轻易保护扇区，避免意外改变相应扇区。另外需要更改时也可以轻易取消对其保护。DYB 可以在需要时随时设为 0 或清除为 1。

3.4.6 扇区保护状态汇总

每个扇区均可以处于下面一种保护状态：

- 解锁 – 扇区不受保护，保护状态可通过一个简单的指令进行更改。在关闭电源或硬件复位后，保护状态默认为不保护。
- 动态锁定 – 扇区受保护，保护状态可通过一个简单的指令进行更改。在关闭电源或硬件复位后，保护状态不被保存。
- 持久锁定 – 扇区受保护，只有将 PPB 锁定位设置为 1 时才能更改保护状态。保护状态是非易失性的，在关闭电源或硬件复位后仍被保存。更改保护状态需要编程或擦除 PPB 位。

表 3.1 扇区保护状态

保护位值			扇区状态
PPB 锁定	PPB	DYB	
1	1	1	无保护 — PPB 和 DYB 可更改
1	1	0	保护 — PPB 和 DYB 可更改
1	0	1	保护 — PPB 和 DYB 可更改
1	0	0	保护 — PPB 和 DYB 可更改

表 3.1 扇区保护状态

保护位值			扇区状态
PPB 锁定	PPB	DYB	
0	1	1	无保护 — PPB 不可更改, DYB 可更改
0	1	0	保护 — PPB 不可更改, DYB 可更改
0	0	1	保护 — PPB 不可更改, DYB 可更改
0	0	0	保护 — PPB 不可更改, DYB 可更改

3.4.7 锁定寄存器

锁定寄存器保持非易失性 OTP 位, 用于控制 SSR 保护和确定 PPB 锁定位的管理方法 (保护模式)。

表 3.2 锁定寄存器

位	默认值	名称
15-12	1	保留
11	1	SSR 区域 3 密码保护模式锁定位
10	1	SSR 区域 3 (用户) 锁定位
9	1	SSR 区域 2 (用户) 锁定位
8	0	保留
7	1	保留
6	1	SSR 区域 1 (用户) 锁定位
5	1	保留
4	1	保留
3	1	保留
2	1	密码保护模式锁定位
1	1	持久保护模式锁定位
0	0	SSR 区域 0 (工厂) 锁定位

安全硅区域 (SSR) 保护位必须谨慎使用。一旦锁定, 没有办法可解锁安全硅区域中受保护的部分, 也没有办法可修改受保护的安全硅区域存储空间中的所有位。一旦安全硅区域受到保护, 任何在该区域中继续编程的尝试都会失败, 其状态表现为要编程的区域受到保护。区域 0 指示位位于锁定寄存器中的位 0, 区域 1 的位于位 6, 区域 2 的位于位 9, 区域 3 的位于位 10。

从工厂出厂时, 所有器件在上电时均默认使用持久保护方法, 所有扇区均不受保护。器件编程器或主机系统随后可以选择使用哪种扇区保护方法。通过对下面两个一次性可编程、非易失性位中的一位进行编程, 可将器件永久性锁定于相应模式:

- 持久保护模式锁定位 (DQ1)
- 密码保护模式锁定位 (DQ2)

如果同时编程这两个锁定位, 操作将中止。一旦编程了密码模式锁定位, 持久模式锁定位永远被禁用, 保护方案无法更改。类似的, 如果编程了持久模式锁定位, 将永久性禁用密码模式。

如果选择密码模式, 必须在设置相应的锁定寄存器位之前编写密码。设置密码保护模式锁定位 (DQ2) 可阻止对密码进行编写或读取操作。

锁定寄存器的编程时间与典型的字编程时间相同。在锁定寄存器编程 EA 期间, 数据轮询状态 DQ6 跳变位将持续跳变, 直至编程完毕。系统还可通过读取状态寄存器来确定锁定寄存器的编程状态。欲了解这些状态位的信息, 请参见第 31 页上的 [状态寄存器](#)。

用户不能同时编程 DQ2 或 DQ1 与 DQ6 或 DQ0 位。这使得用户可以在选择器件保护方案之前或之后锁定 SSR。编程锁定位时, 必须将保留位设置为 1 (被掩码)。

3.4.8 持久保护模式

持久保护方法在 POR 或硬件复位期间将 PPB 锁定设为 1, 因此进行器件复位后 PPB 位为不保护状态。可以通过一个指令来将 PPB 锁定位清除为 0, 以保护 PPB。持久保护方法没有任何指令可将 PPB 锁定位设为 1, 因此 PPB 锁定位一直为 0, 直到下一次关闭电源或硬件复位为止。

3.4.9 密码保护模式

3.4.9.1 PPB 密码保护模式

PPB 密码保护模式使用 64 位密码来设置 PPB 锁定，因此能够提供比持久扇区保护模式级别更高的安全性。除了密码要求外，在上电和复位后，PPB 锁定还清除为 0 以确保在上电时提供保护。通过输入整个密码并成功执行密码解锁指令后，PPB 锁定将设为 1，从而允许修改扇区 PPB。

密码保护注意事项：

- 密码编程指令只能编程“0”。
 - 从赛普拉斯出厂时，密码 64 位全部是 1。它位于自己的存储空间中，可通过使用密码编程和密码读取指令进行访问。
 - 所有 64 位密码组合均为有效密码。
 - 编程和验证密码后，必须设置密码模式锁定位，以防止读取或修改密码。
 - 一旦编程了密码模式锁定位，即可防止在数据总线上读取 64 位密码和进一步编程密码。所有对密码区域进行读取的指令均被禁用（读取数据返回的结果全是 1）。编程了密码保护模式锁定位后，无法验证密码内容。只能在选择密码保护模式前进行密码验证。对锁定扇区进行任何编程操作都会失败，并且报告为一个一般的编程故障。
 - 不能擦除密码模式锁定位。
 - 只有输入准确的密码，才能解锁。
- 地址可按任何序列进行加载，但需要全部 4 个字以实现成功匹配。
- 密码地址 / 数据被加载时，会将扇区地址（Amax - A16）和字线地址（A15 - A8）与“零”进行比较。如果扇区地址或字线地址不匹配，那么在写周期结束时将报告错误。状态寄存器返回就绪状态、编程状态位被设为 1，并且写入缓冲区中止状态位被设为 1，用于表示编程操作失败。数据轮询状态将保持活动状态，DQ7 设成密码解锁指令的最后一个字中 DQ7 位的补码，并且 DQ6 跳变。RY/BY# 将保持低电平状态。
- 发送“从缓冲区写入闪存”指令后，将比较特定地址和数据。如果它们与内部设置值不匹配，则状态寄存器将返回就绪状态，并且编程状态位设为 1，用于表示编程操作失败。数据轮询状态将保持活动，DQ7 设成密码解锁指令的最后一个字中 DQ7 位的补码，并且 DQ6 跳变。RY/BY# 将保持低电平状态。
- 为器件设置有效的 64 位密码后，需要进过 t_{PPB} 时间来设置 PPB 锁定。这样，如果黑客试图通过实现所有 64 位组合来找到正确的匹配密码，需要经过第一段不可思议的超长时间（5800 万年）。EA 状态检查方法可用于确定 EAC 何时准备好接受新的密码指令。
 - 如果在设置密码模式锁定位后密码丢失，没有办法可清除 PPB 锁定。

4. 读取操作

4.1 异步读取

可以对存储器中的任意位置进行读取访问（随机读取）。每个随机读取访问均为自定时，与从 $CE\#$ 或地址到数据有效的延迟（ t_{ACC} 或 t_{CE} ）相同。

4.2 页模式读取

每个随机读取并行访问整个 32 字节的页面。后续读取同一页面时，读取访问速度会加快。页面由高位地址（ $A_{max}-A_4$ ）选择，页面中的特定字由最低有效地址位 A_3-A_0 （在 x8 模式中为 A_3-A-1 ）选择。高位地址保持不变，只有 A_3-A_0 （在 x8 模式中为 A_3-A-1 ）发生改变，以选择同一页面中不同的字。这是异步读取，当 $CE\#$ 为低电平、 $OE\#$ 为低电平，并且满足异步页面访问时间（ t_{PACC} ）时，数据出现在 $DQ_{15}-DQ_0$ （在 x8 模式中为 DQ_7-DQ_0 ）上。对于后续访问，如果 $CE\#$ 拉至高电平再返回到低电平，则执行随机读取访问，并且需要（ t_{ACC} 或 t_{CE} ）时间。

5. 嵌入式操作

5.1 嵌入式算法控制器 (EAC)

EAC 从主机系统接收关于编程和擦除闪存阵列的指令，并执行改变非易失性存储器状态时所需的所有复杂操作。这样可以减轻主机系统的负担，并且不需要管理编程和擦除过程。

EAC 操作有四种类别：

- 待机（读取模式）
- 地址空间切换
- 嵌入式算法（EA）
- 高级扇区保护（ASP）管理

5.1.1 EAC 待机

在待机模式下，电流消耗明显降低。当没有执行任何指令和嵌入式算法时，EAC 进入待机模式。如果在嵌入式算法期间取消了选择器件（CE# = 高电平），器件仍会消耗工作电流，直至操作完成为止（I_{CC3}）。第 64 页上的直流特性中的 I_{CC4} 表明主机接口和 EAC 处于其待机状态时的待机电流规范。

5.1.2 地址空间切换

写入特定地址和数据序列（指令序列）可将存储器件地址空间从主闪存阵列切换到其中一个地址空间重叠（ASO）。

嵌入式算法对当前活动（进入的）ASO 中的可见信息进行操作。在系统发出 ASO 退出指令、执行硬件复位之前，或者断开器件电源之前，系统可以一直访问 ASO。ASO 退出指令可将器件从 ASO 切换回主闪存阵列地址空间。进入特定 ASO 后所接受的指令被罗列在指令定义表中，该表位于 ASO 进入和退出指令之间。有关所有指令序列的地址和数据要求，请参见第 43 页上的命令汇总。

5.1.3 嵌入式算法 (EA)

更改存储阵列中的非易失性数据时，需要执行复杂的操作序列，这些操作被称为嵌入式算法（EA）。这些算法完全由器件内部的嵌入式算法控制器（EAC）来管理。主算法执行对主阵列数据和 ASO 的编程和擦除操作。主机系统将指令代码写入到闪存器件地址空间内。EAC 接收指令用于执行所有必要的步骤以完成指令，并在 EA 执行期间提供状态信息。

5.2 编程和擦除简介

闪存数据位以大组（称为扇区）的形式被并行擦除。擦除操作将扇区中的每个数据位置于逻辑 1 状态（高）。每个闪存数据位可以分别进行编程，使其从擦除 1 状态变为编程逻辑 0（低）状态。数据位 0 不能编程为 1。后续的读取结果显示，数据位仍然为 0。只有通过执行擦除操作才能将 0 变换为 1。多次对同一个字位置进行编程不同的 0 位，会引起原数据和正在编程的新数据间进行逻辑 AND 运算。

第 40 页上的嵌入式算法性能表介绍了编程和擦除时长。

编程和擦除操作可能被挂起。

- 擦除操作可能被挂起，允许编程或读取另一个扇区（不是在擦除扇区中）。在擦除挂起期间，不能启动其它擦除操作。
- 编程操作可能被挂起，允许读取另一个位置（并非位于正在编程的线中）。
- 在编程操作挂起期间，不能启动其它编程或擦除操作；在该时间内，编程或擦除指令将被忽略。
- 在完成嵌入的编程操作或读取访问后，可以恢复被挂起的擦除或编程操作。如果器件当前没有执行其它指令，则可以随时恢复被挂起的操作。
- 编程和擦除操作可以根据需要被中断，但为了正常完成编程或擦除操作，恢复和下一次挂起指令之间的时长必须大于或等于第 40 页上的嵌入式算法性能表中所介绍的 t_{PRS} 或 t_{ERS}。
- 完成嵌入式算法（EA）时，EAC 返回至启动 EA 时所在的操作状态和地址空间（擦除挂起、EAC 待机、...）。

系统可以通过读取状态寄存器或使用数据轮询状态来确定编程或擦除操作的状态。欲了解这些状态位的信息，请参见第 31 页上的状态寄存器。有关更多信息，请参考第 33 页上的数据轮询状态。

除了编程挂起（x51h）、状态读取指令（x70h）和擦除挂起 / 编程挂起指令（xB0h）外，在嵌入式编程算法期间写入器件的所有指令都被忽略。

除了状态读取（x70h）和擦除挂起 / 编程挂起指令（xB0h）外，在嵌入式擦除算法期间写入器件的所有指令都被忽略。

硬件复位会立即终止正在执行的所有编程 / 擦除操作，并在经过 t_{RPH} 时间后返回到读取模式。一旦器件返回待机状态，应重新启动被终止的操作，以确保数据完整性。

出于性能和可靠性原因，读取和编程操作在全部 32 字节页面上内部完成。

第 64 页上的 [直流特性](#) 中的 I_{CC3} 显示了写入（嵌入式算法）操作的工作电流规范。

5.2.1 编程粒度

S29GL-T 支持两种编程方法：字或写入缓冲区编程。每个页面可以使用任一方法进行编程。对于工业温度版本（-40°C 至 +85°C），一线中可以包含使用不同方法进行编程的页面。对于扩展的工业级温度版本（-40°C ~ +105°C）和扩展温度版本（-40°C ~ +125°C），在两次擦除操作之间，器件仅支持每个页面一次编程操作，并且不支持单字编程指令。

字编程检查指令中提供的数据字，在寻址的存储阵列字中编程 0，以匹配指令数据字中的 0。

写入缓冲区编程检查写入缓冲区，在寻址的存储阵列行中编程 0，以匹配写入缓冲区中的 0。写入缓冲区不需要全部填入数据。在一个编程操作中，可以尽可能少地编程，如单个位、多个位、单个字、多个字、一个页面、多个页面、或者整个缓冲区。使用写入缓冲区方法可以减轻主机系统在写入编程指令方面的负担，并能够减轻存储器件在编程操作方面的内部负担。与使用字编程指令对各个字进行编程相比，写入缓冲区编程更快、更有效率。

5.2.2 增量编程

同一字位置可以通过字或写入缓冲区编程方法进行多次编程，从而增量方式将 1 改为 0。

5.3 指令集

5.3.1 编程方法

5.3.1.1 字编程

字编程用于编程主闪存阵列中任意位置的单个字。

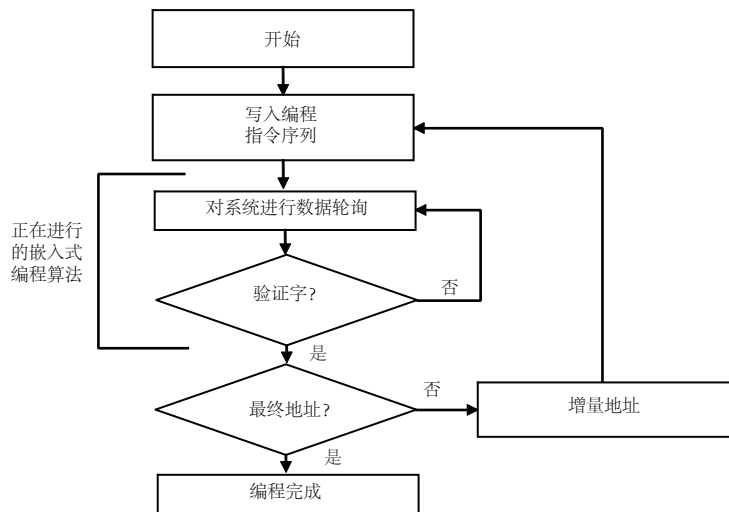
字编程指令是四个写入循环序列。编程指令序列开始于两个解锁写入循环，然后是编程设置指令。接下来写入编程地址和数据，以启动嵌入式字编程算法。系统不需要提供其它控制或时序。器件自动生成编程脉冲，并内部检验编程的网格边距。当嵌入式字编程算法完成时，EAC 随后返回其待机模式。

系统可以通过使用数据轮询状态、读取状态寄存器、或监控 RY/BY# 输出等方式来确定编程操作的状态。欲了解这些状态位的信息，请参见第 31 页上的 [状态寄存器](#)。欲了解这些状态位的信息，请参见第 33 页上的 [数据轮询状态](#)。欲了解字编程操作的流程图，请参见第 19 页上的图 5.1。

除了编程挂起指令外，在嵌入式编程算法期间写入器件的所有指令都被忽略。请注意，硬件复位 (RESET# = V_{IL}) 可立即终止编程操作，并经过 t_{RPH} 时间后可使器件返回到读取模式。为确保数据完整性，一旦器件完成硬件复位操作，应重新启动编程指令序列。

修改版的字编程指令没有解锁写入周期，从而能够在进入锁定寄存器、密码和 PPB ASO 或解锁旁通模式后进行编程。进入 PPB 锁定和 DYB ASO 后，可以使用相同的指令更改易失性位。欲了解编程指令序列，请参见第 43 页上的表 6.1。

图 5.1 字编程操作



5.3.1.2 写入缓冲区编程

写入缓冲区用于在 512 字节边界 (线) 上对齐的 512 字节地址范围内编程数据。因此，整个写入缓冲区编程操作必须与线边界对齐。不足 512 字节的编程操作可以在任意字边界上开始，但不能跨越线边界。在写入缓冲区编程操作开始时，缓冲区中所有位的位置全部为 1 (FFFFh 字)，因此未加载的位置 仍保持现有数据。有关地址映射的信息，请参见第 4 页上的 [产品概述](#)。

写入缓冲区编程在一个操作中最多可编程 512 字节。在每个写入缓冲区编程操作中，可以编程 1 位到 512 字节。建议写入多个页面，每个页面只写入一次。为达到最佳性能，编程应在 512 字节边界对齐的全部 512 字节线上完成。

仅在主闪存阵列或 SSR ASO 中支持写入缓冲区编程。

写入缓冲区编程操作开始于两个写入解锁周期。随之是写入到缓冲区指令的第三个写入周期，其中包含要编程的扇区地址 (SA)。接下来，系统会写入字位置数量减去 1。这会告知器件有多少写入缓冲区地址加载了数据以及何时会发出“编程缓冲区到闪存”的确认指令。写入到缓冲区指令和写入字计数指令中的扇区地址必须匹配。要编程的扇区必须解锁 (不受保护)。

系统随后写入起始地址 / 数据组合。这个起始地址是要编程的第一个地址 / 数据对，选择写入缓冲区线地址。扇区地址必须与通过写入到缓冲区指令进行编程的扇区地址匹配，否则操作将中止并返回中止状态。所有后续地址 / 数据对必须是连续顺序。所有写入缓冲区地址必须在同一线内。如果系统尝试加载此范围之外的数据，操作将中止并返回中止状态。

每个数据加载操作后，计数器递减。请注意，随着数据写入倒计时，每个写入操作都被认为数据正在加载到写入缓冲区。在写入缓冲区加载期间，不能执行任何指令。唯一可停止写入缓冲区加载的方式是写入编程操作线之外的地址。这个无效地址将立即中止写入到缓冲区指令。

一旦已加载指定的写入缓冲区地址数量，系统必须在扇区地址内写入编程缓冲区到闪存指令。器件随之开始忙碌。嵌入式编程算法自动编程数据，并验证数据是否为正确的数据组合。在这些操作期间，系统不需要提供任何控制或时序。如果加载的写入缓冲区位置的数量不正确，操作将中止并返回中止状态。由于在字计数结束时需要写入编程缓冲区到闪存指令，所以如果此时写入其它任何指令，编程操作中中止。

写入缓冲区嵌入式编程操作可以通过编程挂起指令来挂起。完成嵌入式编程算法时，EAC 返回到编程操作启动时所在的 EAC 待机或擦除挂起待机状态。

系统可以通过使用数据轮询状态、读取状态寄存器、或监控 RY/BY# 输出等方式来确定编程操作的状态。欲了解这些状态位的信息，请参见第 31 页上的 [状态寄存器](#)。欲了解这些状态位的信息，请参见第 33 页上的 [数据轮询状态](#)。欲了解编程操作的流程图，请参见第 21 页上的图 5.2。

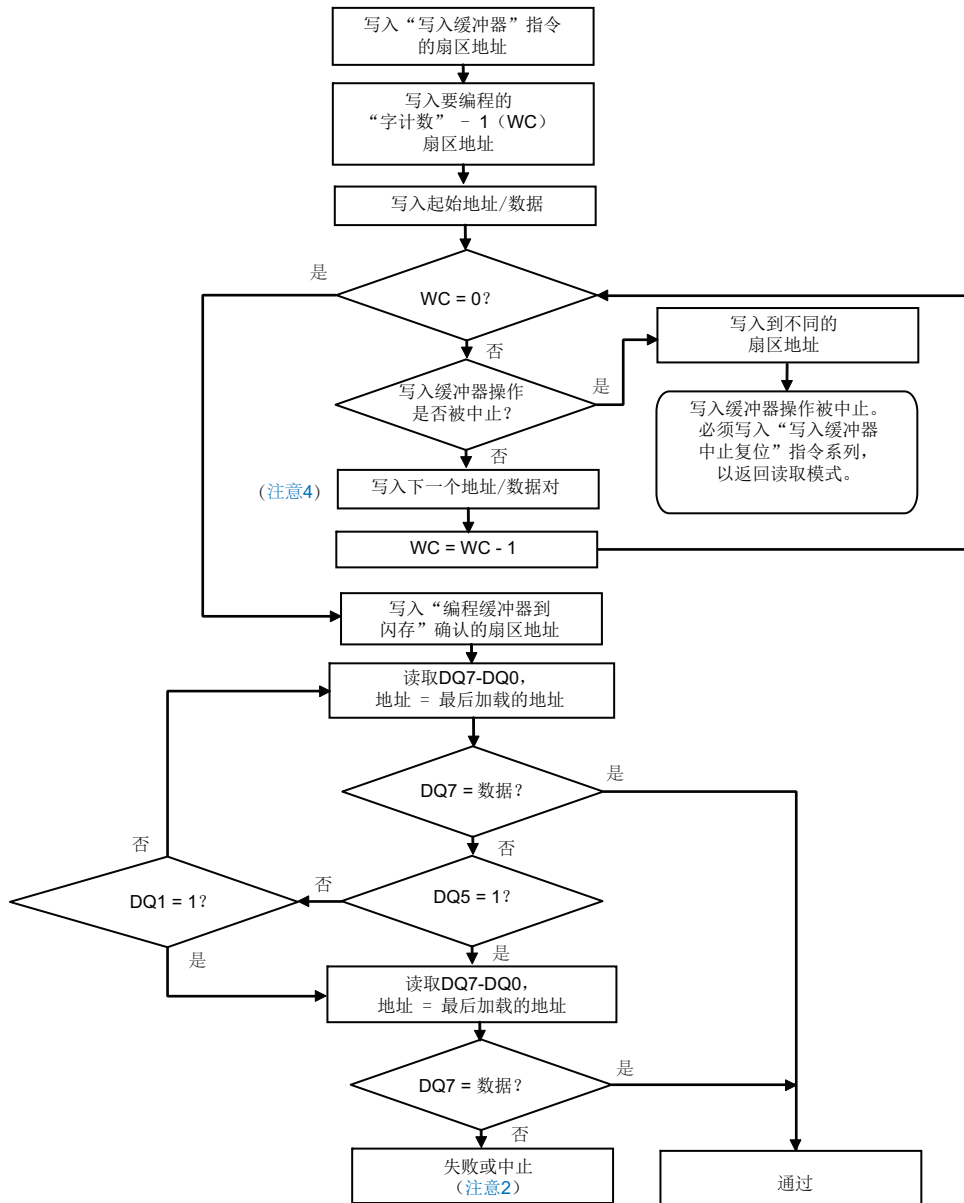
在下列情况下，写入缓冲区编程序列将中止：

- 加载的字计数值超过缓冲区容量（255）。
- 写入的地址超出在写入到缓冲区指令中提供的线。
- 加载完写入字计数的数据字数量后不发出写入缓冲区到闪存指令。

当导致写入缓冲区指令中止的任何条件发生时，指令操作将立即中止，并在状态寄存器的位置位 4 指明编程失败（PSB = 1），因为写入缓冲区中止位置位 3 被设置（WBASB = 1）。下一次成功的编程操作将清除失败状态，或者可以使用清除状态寄存器指令清除 PSB 状态位。

有两种方法可终止写入缓冲区编程序列：硬件复位或关开电源。不过，使用任一方法都可能造成正在编程的区域处于中间状态，即包含无效或不稳定的数据值。在此情况下，需要使用相同的数据对该区域进行重新编程或者擦除该区域，以确保正确编程数据值或适当擦除它们。

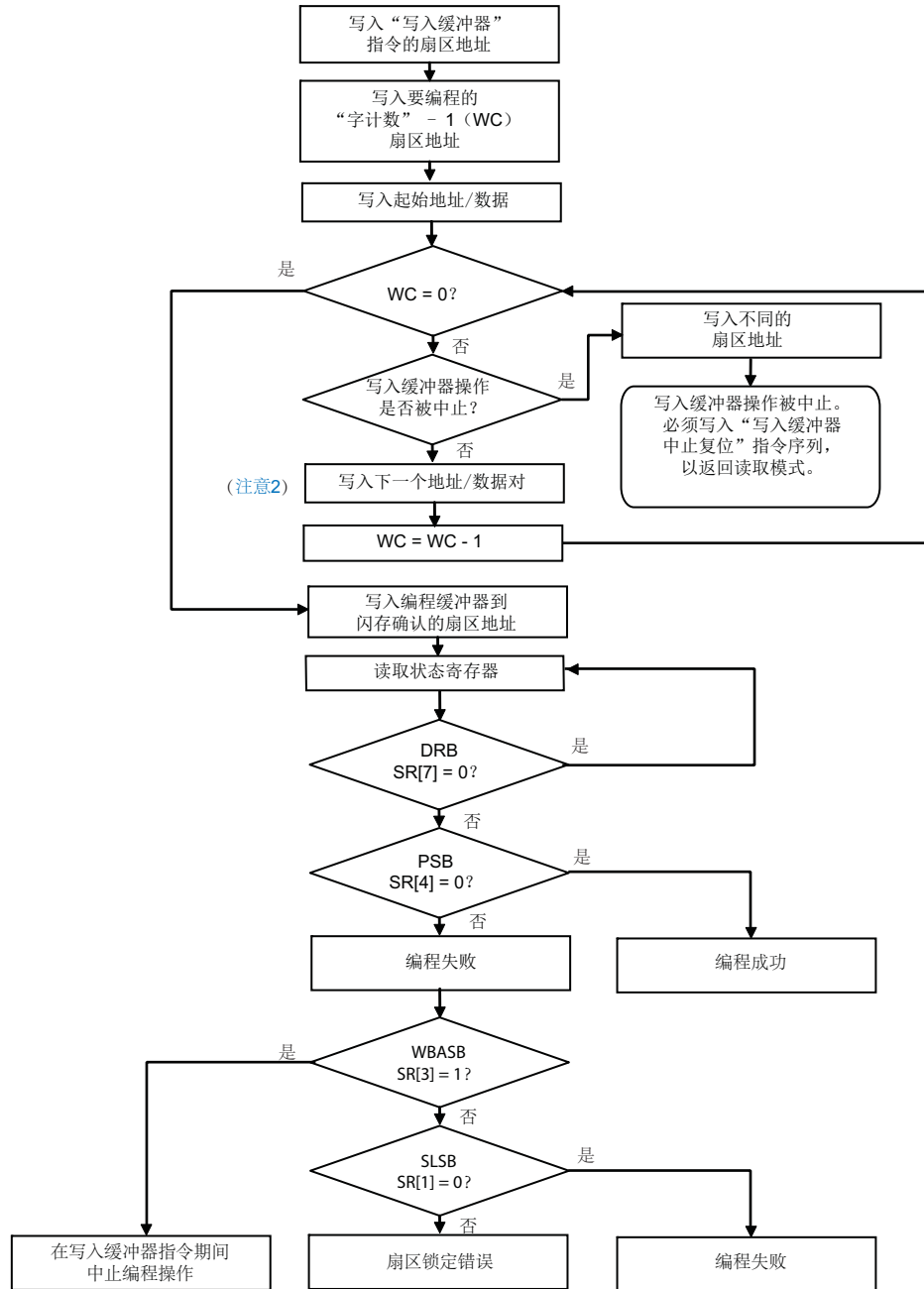
图 5.2 使用数据轮询状态验证写入缓冲区编程操作



注意:

1. 由于 DQ7 可能与 DQ5 同时改变，因此即使 DQ5 = 1，也应重新检查 DQ7。
2. 如果由于 DQ5 = 1 而到达此流程图位置，则器件发生故障。如果由于 DQ1 = 1 而到达此流程图位置，则写入缓冲区操作已中止。无论在哪种情况下，都必须向器件写入正确的复位指令，以使器件返回读取模式。如果 DQ1 = 1，则写入缓冲区编程中止复位；如果 DQ5 = 1，则软件复位或写入缓冲区编程中止复位。
3. 欲了解写入缓冲区编程所需的指令序列，请参见第 43 页上的表 6.1，指令定义 x16。
4. 当指定了扇区地址时，所选扇区中的任何地址均可接受。不过，为写入缓冲区地址位置加载数据时，所有地址必须在所选写入缓冲区页面范围内。

图 5.3 使用状态寄存器验证写入缓冲区编程操作



注意:

- 欲了解写入缓冲区编程所需的指令序列，请参见第 43 页上的表 6.1，指令定义 x16。
- 当指定了扇区地址时，所选扇区中的任何地址均可接受。不过，为写入缓冲区地址位置加载数据时，所有地址必须在所选写入缓冲区页面范围内。

表 5.1 写缓冲区编程指令序列

序列	x16		x8		注释
	地址	数据	地址	数据	
发出解锁指令 1	555	AA	AAA	AA	
发出解锁指令 2	2AA	55	555	55	
在扇区地址发出写入到缓冲区指令	SA	0025h	SA	25h	
在扇区地址发出位置数量 例如: WC = 0, 需要编程一个字 WC = 1, 则需要编程两个字	SA	WC	SA	WC	WC = 要编程的字的数量 - 1 (在 x8 模式下, WC = 要编程的字节数 - 1)
加载起始地址 / 数据对	起始地址	PD	起始地址	PD	选择写入缓冲区页面, 然后加载第一个地址 / 数据对。
加载下一个地址 / 数据对	WBL	PD	WBL	PD	所有地址必须位于所选写入缓冲区页面边界内, 并且必须按连续顺序加载。
加载最后一个地址 / 数据对	WBL	PD	WBL	PD	所有地址必须位于所选写入缓冲区页面边界内, 并且必须按连续顺序加载。
在扇区地址发出写缓冲区编程确认	SA	0029h	SA	29h	该指令必须在最后一个写入缓冲区位置加载后执行, 否则操作会中止。
器件开始忙碌					

图标:

SA = 扇区地址 (非扇区地址位不用考虑。扇区内的任何地址足够)

WBL = 写入缓冲区位置 (必须位于起始地址指定的写入缓冲区行的边界内)

WC = 字计数

PD = 编程数据

5.3.2 编程挂起 / 编程恢复指令

编程挂起指令允许系统中断嵌入式编程操作, 以便能够从非挂起的行中读取数据。当在某个编程过程中写入编程挂起指令时, 器件在 t_{PSL} (编程挂起延迟) 时间内暂停编程操作并更新状态位。在写入编程挂起指令时, 不用考虑地址。

编程挂起可通过两个指令来执行。传统组合擦除 / 编程挂起指令 (B0h 指令代码) 和单独的编程挂起指令 (51h 指令代码)。编程恢复也可通过两个指令来执行。传统组合擦除 / 编程恢复指令 (30h 指令代码) 和单独的编程恢复指令 (50h 指令代码)。建议在编程时使用单独的编程挂起和恢复指令, 只在擦除挂起和恢复时使用传统组合指令。

编程操作挂起后, 系统可以读取任何非挂起的行中的阵列数据。当擦除操作被挂起时, 仍可以在编程期间发出编程挂起指令。在此情况下, 可以从擦除挂起或编程挂起之外的地址读取数据。

写入编程恢复指令后, 器件返回编程状态, 并且状态位被更新。系统可以通过读取状态寄存器或使用数据轮询来确定编程操作的状态。欲了解这些状态位的信息, 请参见第 31 页上的 [状态寄存器](#)。更多有关信息, 请参考第 33 页上的 [数据轮询状态](#) 中介绍的内容。

在编程挂起期间有效的访问和指令包括:

- 读取任何其他非擦除挂起的扇区
- 读取任何其他非编程挂起的行
- 状态读取指令
- 状态寄存器清除
- 退出 ASO 或指令集退出
- 编程恢复指令

系统必须写入编程恢复指令, 才能退出编程挂起模式并继续编程操作。进一步写入的编程恢复指令被忽略。在器件恢复编程后, 便可写入另一个编程挂起指令。

编程操作可以根据需要经常中断, 但为了使编程操作正常完成, 恢复和下一次挂起指令之间的时长必须大于或等于第 17 页上的 [嵌入式算法控制器 \(EAC\)](#) 中所述的 t_{PRS} 。

进入 ASO 后, 不支持编程挂起和恢复。

5.3.3 加速编程

当系统激活 WP#/ACC 或 ACC 引脚上的 V_{HH} 时，器件支持加速编程操作。当 WP#/ACC 或 ACC 引脚上的电压下降到 V_{IH} 或 V_{IL} 时，器件将退出加速编程模式，并返回执行正常操作。WP#/ACC 可以承受 V_{HH} 电压，但不能加速编程功能。如果系统激活输入端的 V_{HH} ，器件将自动进入解锁省略模式。系统随后可以使用解锁省略模式提供的写入缓冲区加载指令序列。请注意，如果在解锁省略模式下需要“写入到缓冲区中止复位”，则必须使用完整的 3 周期 RESET 指令序列来复位器件。如果在嵌入式编程操作完成时从 ACC 输入去除 V_{HH} ，器件将返回正常操作。注意：除了加速编程操作外，WP#/ACC 引脚不能处于 V_{HH} ，否则器件可能损坏。WP# 有一个内部上拉电阻；当断开连接时，WP# 将处于 V_{IH} 。仅在室温条件下才支持加速编程。

- 在 WP#/ACC 提升至 V_{HH} 前，必须解锁扇区。
- 建议在上电序列完成后才在 WP#/ACC 上应用 V_{HH} 。另外，在 V_{CC}/V_{IO} 断电前，建议将 WP#/ACC 上的 V_{HH} 变回到 V_{IH}/V_{IL} 范围内的电压。

5.3.4 解锁省略

此器件提供解锁省略模式，可缩短编程指令。一旦器件进入解锁省略模式，只需两个写周期（而不是正常的四个周期）即可编程数据。该器件也支持写入缓冲区指令而仅需要四个写周期以上。

该模式不使用标准编程指令序列所需的前两个解锁周期，从而缩短总计编程时间。第 43 页上的命令汇总介绍了解锁省略指令序列的要求。

在解锁省略模式下，只有读取、编程、写入缓冲区编程、写入到缓冲区中止复位、状态寄存器读取、状态寄存器清除、软复位、解锁省略扇区擦除、解锁省略芯片擦除、解锁省略擦除暂停 / 恢复、解锁省略暂停 / 恢复以及解锁省略复位等指令均有效。为了退出解锁省略模式，系统必须发出两周期解锁省略复位指令序列。第一个周期地址为“无需关注”和数据 90h。第二个周期只需包含数据 00h。扇区随后返回读取模式。

软件函数和代码示例

下面是使用解锁省略进入、编程、以及退出功能的 C 源代码示例。有关赛普拉斯闪存存储器软件开发指南的通用信息，请参见赛普拉斯底层驱动器用户指南。

表 5.2 解锁省略进入 (LLD 函数 = lld_UnlockBypassEntryCmd)

周期	说明	操作	字节地址	字地址	数据
1	解锁	写入	基址 + AAAh	基址 + 555h	00AAh
2	解锁	写入	基址 + 555h	基址 + 2AAh	0055h
3	进入指令	写入	基址 + AAAh	基址 + 555h	0020h

```

/* Example: Unlock Bypass Entry Command */
*( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (UINT16 *)base_addr + 0x555 ) = 0x0020; /* write unlock bypass command */
/* At this point, programming only takes two write cycles. */
/* Once you enter Unlock Bypass Mode, do a series of like */
/* operations (programming or sector erase) and then exit */
/* Unlock Bypass Mode before beginning a different type of */
/* operations. */

```


表 5.3 解锁省略编程 (LLD 函数 = lld_UnlockBypassProgramCmd)

周期	说明	操作	字节地址	字地址	数据
1	编程设置	写入	基址 + XXXh	基址 + XXXh	00A0h
2	编程指令	写入	编程地址	编程地址	编程数据

```

/* Example: Unlock Bypass Program Command */
/* Do while in Unlock Bypass Entry Mode! */
*( (UINT16 *)base_addr ) = 0x00A0; /* write program setup command */
*( (UINT16 *)pa ) = data; /* write data to be programmed */
/* Poll until done or error. */
/* If done and more to program, */
/* do above two cycles again. */

```

表 5.4 解锁省略复位 (LLD 函数 = lld_UnlockBypassResetCmd)

周期	说明	操作	字节地址	字地址	数据
1	复位周期 1	写入	基址 + XXXh	基址 + XXXh	0090h
2	复位周期 2	写入	编程地址	编程地址	0000h

```

/* Example: Unlock Bypass Exit Command */
*( (UINT16 *)base_addr ) = 0x0090;
*( (UINT16 *)base_addr ) = 0x0000;

```

5.3.5 评估擦除状态

评估擦除状态 (EES) 指令用于验证对已指定地址的扇区进行的擦除操作是否完全成功 (即 Trust Worthy — 值得信任)。使用 EES 指令可以检测擦除操作失败的原因, 原因可能是: 断电、复位或擦除操作过程失败。

要想对一个扇区启动评估擦除状态 (EES), 可以在 EAC 处于待机状态时向该扇区中的地址 555h 写入 35h。

当器件正在编程、擦除或挂起时, 不能写入 EES 指令。

EES 指令不允许在评估擦除状态期间读取阵列。

通过使用状态寄存器或轮询方法 (仅 DQ6 切换) 可以确定该器件当前处于繁忙还是完成状态。完成后, 使用状态寄存器读取指令来确认该扇区是否可靠。如果该扇区可靠, 则状态寄存器 (SR[5]) 的位 5 将被清除为 0。如果该扇区不可靠, 则 SR[5] 将被置 1, RD/BY# 为低电平, 并且需要执行软件复位 /ASO 退出指令或状态寄存器清除指令, 从而使器件返回到待机状态。

EES 完成后, EAC 返回待机状态。

EES 指令需要经过 t_{EES} 时间来完成并更新状态寄存器中的擦除状态。可以读取 DRB 位 (SR[7]), 以确定 EES 指令完成的时间。如果 SR[5]=1, 则表示扇区未被擦除, 那么必须重新执行擦除操作, 以确保该扇区中数据存储的可靠性。

5.3.6 空白检查

空白检查指令将确认所选的主闪存阵列扇区是否已擦除 (即 Trust Worthy (值得信任) 和 Blank (空白))。空白检查指令不允许在空白检查期间读取阵列。若在执行此指令时读取阵列, 会返回轮询数据。

要想对一个扇区启动空白检查, 可以在 EAC 处于待机状态时向该扇区中的地址 555h 写入 33h。

当器件正在编程、擦除或挂起时, 不能写入空白检查指令。

通过使用状态寄存器或轮询方法 (等于嵌入式擦除操作) 可以确定该器件处于繁忙还是完成状态。完成后, 状态寄存器和轮询方法将显示该扇区为空 (等于成功擦除操作) 还是未清除该扇区。如果该扇区为空, 状态寄存器 (SR[5]) 的位 5 将清除为 0。如果该扇区为非空, 则 SR[5] 将被置 1, RD/BY# 为低电平, 并且需要执行软件复位 /ASO 退出指令或状态寄存器清除指令以使器件返回待机状态。

一旦发现任何位没有擦除, 器件将中止操作并报告结果。

空白检查完成后, EAC 返回待机状态。

5.3.7 擦除方法

5.3.7.1 芯片擦除

芯片擦除功能擦除整个主闪存阵列。器件不要求系统在擦除之前进行预编程。嵌入式擦除算法在执行电擦除前，自动编程和验证整个存储器是否为全 0 数据组合。芯片擦除成功后，器件内的所有位置均包含 FFFFh。在这些操作期间，系统不需要提供任何控制或时序。芯片擦除指令序列通过写入两个解锁周期来启动，随后是设置指令。另外两个解锁写周期后面是芯片擦除指令，用于激活嵌入式擦除算法。当 WE# 在第 6 个命令周期结束后被拉高时，RY/BY# 将变为低电平。

嵌入式擦除算法完成后，EAC 返回待机状态。请注意，正在执行嵌入式擦除操作时，系统不能从器件读取数据。系统可以通过读取 RY/BY#、状态寄存器或使用数据轮询来确定擦除操作的状态。有关 RY/BY# 的信息，请参考第 57 页上的就绪/忙碌# (RY/BY#)。有关这些状态位的信息，请参见第 31 页上的状态寄存器。更多有关信息，请参考第 33 页上的数据轮询状态中介绍的内容。

芯片擦除操作开始后，只有状态读取、硬件复位或开关电源有效。所有其他指令均被忽略。不过，硬件复位或开关电源会立即终止擦除操作，并在经过 t_{RPH} 时间后返回读取模式。如果芯片擦除操作被终止，一旦器件返回闲置状态，必须重新启动芯片擦除指令序列，以确保数据完整性。

有关参数和时序框图的信息，请参见第 40 页上的表 5.7、第 76 页上的异步写入操作和第 82 页上的备用 CE# 控制的写入操作。

受 ASP DYB 和 PPB 位保护的扇区不会被擦除。请参见第 12 页上的 ASP。如果一个扇区在芯片擦除期间受到保护，芯片擦除将跳过受保护的扇区，并继续对下一个扇区执行擦除。如果对受保护扇区的擦除失败，状态寄存器擦除状态位和扇区锁定位将不设置为 1。

5.3.7.2 扇区擦除

扇区擦除功能擦除存储阵列中的一个扇区。器件不要求系统在擦除前进行预编程。嵌入式擦除算法在执行电擦除前，自动编程和验证整个扇区是否为全 0 数据组合。扇区擦除成功后，被擦除扇区内的所有位置均包含 FFFFh。在这些操作期间，系统不需要提供任何控制或时序。扇区擦除指令序列通过写入两个解锁周期来启动，随后是设置指令。另外两个解锁写周期后面是要擦除的扇区地址以及扇区擦除指令。当 WE# 在第 6 个命令周期结束后被拉高时，RY/BY# 将变为低电平。

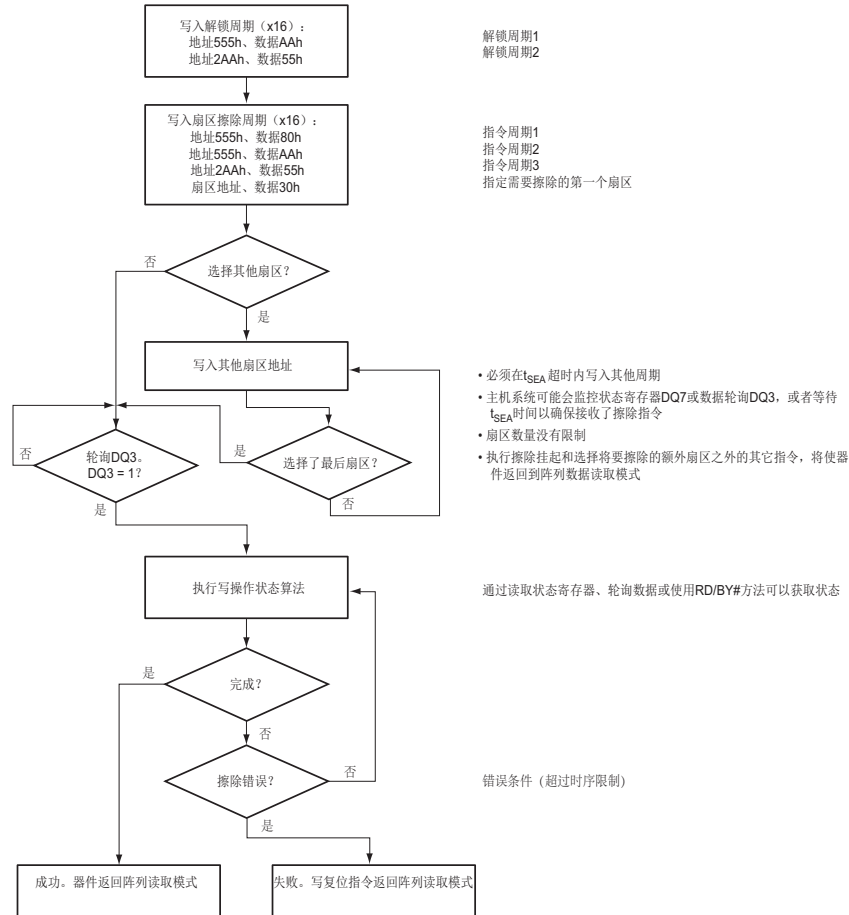
写入指令序列后，会出现 t_{SEA} 的扇区擦除超时。在超时期间，可以写入其他扇区地址以及扇区擦除指令。在超时期间，无效指令将被忽略。加载扇区擦除缓冲区可以通过任意序列完成，扇区数量可以从一个扇区到所有扇区。这些附加周期之间的时间必须小于 t_{SEA} ，否则将开始擦除。超出超时之后的任何扇区擦除地址和指令可能被接受，也可能不被接受。建议在此期间禁止处理器中断，以确保接受所有指令。在写入最后一个扇区擦除指令后，可以再次使能所述中断。请注意，在擦除操作期间，不能使用安全硅扇区、自动选择和 CFI 等功能。系统必须重新写入该指令序列以及其它地址和指令。

系统可以通过读取 RY/BY#、状态寄存器或使用数据轮询来确定擦除操作的状态。有关 RY/BY# 的信息，请参考第 57 页上的就绪/忙碌# (RY/BY#)。有关这些状态位的信息，请参见第 31 页上的状态寄存器。更多有关信息，请参考第 33 页上的数据轮询状态中介绍的内容。

扇区擦除操作开始后，状态寄存器读取和擦除挂起指令均有效。所有其他指令均被忽略。不过，硬件复位会立即终止擦除操作，并在经过 t_{RPH} 时间后返回读取模式。如果扇区擦除操作被终止，一旦器件执行了复位操作，必须重新启动扇区擦除指令序列，以确保数据完整性。

受 ASP DYB 和 PPB 位或密码保护的扇区不会被擦除。请参见第 12 页上的 ASP。如果多扇区擦除操作中包含有被保护的扇区，扇区擦除将跳过受保护的扇区，并继续对下一个扇区执行擦除。如果对受保护扇区的擦除失败，状态寄存器擦除状态位和扇区锁定位将不设置为 1。欲了解参数和时序框图的信息，请参见第 17 页上的嵌入式算法控制器 (EAC)。受 ASP DYB 和 PPB 位保护的扇区不会被擦除。请参见第 12 页上的 ASP。

图 5.4 扇区擦除操作



注意:

1. 请参考 x8 总线周期的指令汇总。

5.3.8 擦除挂起 / 擦除恢复

擦除挂起指令允许系统中断扇区擦除操作，然后从主闪存阵列读取数据或向其编程数据。此指令仅在扇区擦除或编程操作期间有效。如果在芯片擦除操作期间写入擦除挂起指令，该指令将被忽略。

如果在扇区擦除操作期间写入擦除挂起指令，器件需要最多 t_{ESL} （擦除挂起等待时间）时间来挂起擦除操作并更新状态位。

擦除操作被挂起后，该部分进入擦除挂起模式。系统可以从主闪存阵列读取数据或向其编程数据。读取擦除挂起扇区中的地址会生成不确定的数据。系统可通过读取状态寄存器或使用数据轮询来确定扇区是正在擦除还是已挂起。有关这些状态位的信息，请参见第 31 页上的 *状态寄存器*。更多有关信息，请参考第 33 页上的 *数据轮询状态* 中介绍的内容。

擦除挂起的编程操作完成后，EAC 返回擦除挂起状态。系统可以通过读取状态寄存器来确定编程操作的状态，就像在执行标准编程操作一样。

如果在擦除挂起期间编程操作失败，状态寄存器清除或软件复位指令会使器件返回擦除挂起状态。在重新对擦除阵列编程之前，需要恢复并完成擦除操作。

在擦除挂起期间有效的存取和指令包括：

- 读取任何其他非挂起的扇区
- 编程任何其他非挂起的扇区
- 状态寄存器读取
- 状态寄存器清除
- 擦除恢复指令

为恢复扇区擦除操作，系统必须写入擦除恢复指令。器件将返回擦除状态，状态位将更新。进一步写入的恢复指令被忽略。在芯片恢复擦除后，方可写入另一个擦除挂起指令。

进入 ASO 后，不支持擦除挂起和恢复。

5.3.9 进入和退出 ASO

5.3.9.1 ID-CFI ASO

系统可以通过在读取模式期间发出 ID-CFI 进入指令序列来进入 ID-CFI ASO。欲了解详细信息，请参见第 51 页上的表 6.3。

ID-CFI ASO 允许实现下列操作：

- 使用与进入指令中所用 SA 相同的 SA 读取 ID-CFI ASO。
- 读取扇区地址 (SA) + 2h 的扇区保护状态。2h 地址提供与被寻址扇区的当前扇区保护状态相关的易失性信息。2h 位置字的第 0 位显示与被寻址扇区相关的 PPB 和 DYB 位的逻辑与非。因此，如果扇区受到其 PPB=0 或 DYB=0 位的保护，则状态显示为受保护。（1= 扇区受保护，0= 扇区不受保护。）
- ASO 退出。

下面是使用 CFI 进入和退出功能的 C 源代码示例。有关赛普拉斯闪存存储器软件开发指南的通用信息，请参见 *赛普拉斯底层驱动器用户指南*。

```
/* Example: CFI Entry command */
*( (UINT16 *)base_addr + 0x55 ) = 0x0098; /* write CFI entry command */

/* Example: CFI Exit command */
*( (UINT16 *)base_addr + 0x00 ) = 0x00F0; /* write cfi exit command */
```

5.3.9.2 状态寄存器 ASO

状态寄存器 ASO 包含了一个嵌入式算法状态的易失性单字寄存器值。发出状态寄存器读取指令时，寄存器将（在 WE# 的上升沿上）捕获当前的状态，并进入 ASO。状态寄存器内容出现在所有字位置上。第一次读取访问使系统退出状态寄存器 ASO（在 CE# 或 OE# 的上升沿上），并将其返回到发出状态寄存器读取指令时所使用的地址空间映射。发出写入指令不会使器件退出状态寄存器 ASO 状态。

5.3.9.3 安全硅区域 ASO

系统可以通过在读取模式期间发出安全硅区域进入命令序列来存取安全硅区域。此进入指令使用指令中的扇区地址（SA）来确定哪个扇区将被覆盖。

安全硅区域 ASO 允许实现下列操作：

- 读取安全硅区域。
- 通过使用字或写入缓冲区编程指令可以对客户安全硅区域进行编程。不能使用解锁省略指令和 ACC。
- ASO 退出使用传统安全硅区域退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

使用 SSR 区域 3 读密码模式的建议流程如下所示：

- 在 SSR 区域 3 中编程您需要的数据。
- 将锁定寄存器位 10 清除为 0，这样将不能再次执行编程操作。
- 编程 SSR 区域 3 密码。
- 将锁定寄存器位 11 清除为 0，以启用 SSR 区域 3 密码功能。这样，在读取 SSR 区域 3 前，需要应用一个密码。

5.3.9.4 锁定寄存器 ASO

系统可以通过在读取模式期间发出锁定寄存器进入指令序列来存取锁定寄存器。此进入指令不会使用进入指令中的扇区地址。锁定寄存器出现在器件地址空间中的字位置 0。器件地址空间中的所有其他位置均为未定义。

锁定寄存器 ASO 允许下列活动：

- 使用器件地址 0 位置读取锁定寄存器。
- 使用修改版的字编程指令对用户锁定寄存器进行编程。
- ASO 退出使用传统指令集退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

5.3.9.5 密码 ASO

系统可以通过在读取模式期间发出密码进入指令序列来存取密码 ASO。此进入指令不会使用进入指令中的扇区地址。密码的位置是在器件地址空间中字位置 0 到 3。器件地址空间中的所有其他位置均为未定义。

密码 ASO 允许下列操作：

- 使用器件地址位置 0 到 3 读取密码（如果未锁定）。
- 使用修改版的字编程指令对密码进行编程。
- 使用密码解锁指令解锁 PPB 锁定位。
- ASO 退出使用传统指令集退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

5.3.9.6 PPB ASO

系统可以通过在读取模式期间发出 PPB 进入指令序列来存取 PPB ASO。此进入指令不会使用进入指令中的扇区地址。扇区的 PPB 位出现在扇区中所有字位置的位 0。

PPB ASO 允许下列操作：

- 在扇区中任意字的位 0 读取扇区的 PPB 保护状态。
- 使用修改版的字编程指令对 PPB 位进行编程。
- 使用 PPB 擦除指令擦除所有 PPB 位。
- ASO 退出使用传统指令集退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

5.3.9.7 PPB 锁定 ASO

系统可以通过在读取模式期间发出 PPB 锁定进入指令序列来存取 PPB 锁定 ASO。此进入指令不会使用进入指令中的扇区地址。全局 PPB 锁定位出现在器件中所有字位置的位 0。

PPB 锁定 ASO 允许下列操作：

- 在器件地址空间中任意字的位 0 读取 PPB 锁定保护状态。
- 使用修改版的字编程指令设置 PPB 锁定位。
- ASO 退出使用传统指令集退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

5.3.9.8 DYB ASO

系统可以通过在读取模式期间发出 DYB 进入指令序列来存取 DYB ASO。此进入指令不会使用进入指令中的扇区地址。扇区的 DYB 位出现在扇区中所有字位置的位 0。

DYB ASO 允许下列操作：

- 在扇区中任意字的位 0 读取扇区的 DYB 保护状态。
- 使用修改版的字编程指令设置 DYB 位。
- 使用修改版的字编程指令清除 DYB 位。
- ASO 退出使用传统指令集退出指令实现软件向后兼容。
- ASO 退出也可以使用 ASO 通用退出指令，以实现一致的退出方式。

5.3.9.9 软件（指令）复位 /ASO 退出

软件复位是指令集的一部分（参见第 43 页上的表 6.1，*指令定义 x16*），也可使 EAC 返回待机状态，在下列情况下必须使用：

- 退出 ASO 模式
- 发生超时时清除数据轮询的超时位（DQ5）

软件复位不影响 EA 模式。编程或擦除已经开始后，在操作完成前，软件复位指令将一直被忽略。软件复位不影响输出；其主要作用是从 ASO 模式或者从失败的编程或擦除操作返回读取模式。

软件复位可以使未定义的状态（因指令序列无效而造成）返回读取模式。不过，从某些未定义状态返回正常操作可能需要硬件复位。

软件复位没有等待时间要求。软件复位指令在 t_{WPH} 期间执行。

5.3.9.10 连续性检查功能

通过连续性检查功能，可以对从封装连接器到每个晶片焊盘和 DDP 中的各个单独晶片间的连接进行基本测试。该功能是传统的解锁周期序列的扩展，可以在开始执行某些指令时使用该功能。解锁序列包括两个写操作，其中 1 和 0 格式在地址和数据线的低部分上进行交换，而且该格式在第一和第二个写操作之间进行反转。要想执行连续性检查功能，需要扩展这些格式，从而包含所有地址（Amax 转换到 0）和数据线（DQ15 转换到 0）。一个逻辑比较电路寻找在两个写周期期间进行反转的 1 和 0 格式。

在 DDP 情况下，可以使用 A26 输入来选择将要执行写操作的晶片。当检测到正确格式时，状态寄存器位 0 将被设置为 1。状态寄存器清除指令会将状态寄存器位清除为 0。

下表显示的是 x16 模式下单晶片（例如，GL01GT）的连续性检查序列。

相位	访问类型	地址 A26	地址 A25 至 A0	数据	注释
设置	写入	n/a	XXXX555	XX71	清除晶片 0 状态
	写入	n/a	555	XX70	将状态寄存器读取指令写入到晶片 0 内
	读取	n/a	x	RD	读取晶片 0 的状态以确认状态位 0 = 0
连续性模型	写入	n/a	2AAAA55	FF00	第一个连续性周期
	写入	n/a	15555AA	00FF	第二个连续性周期
验证检测到的连续性模型	写入	n/a	555	XX70	将状态寄存器读取指令写入到晶片 0 内
	读取	n/a	x	RD	对于检测到的连续性模型，读取晶片 0 的状态，从而确认状态位 0 为逻辑 1

下表显示的是 x8 模式下单晶片（例如，GL01GT）的连续性检查序列。

相位	访问类型	地址 A26	地址 A25 ~ A-1	数据	注释
设置	写入	n/a	XXXX555	71	清除晶片 0 状态
	写入	n/a	AAA	70	将状态寄存器读取指令写入到晶片 0 内
	读取	n/a	x	RD	读取晶片 0 的状态以确认状态位 0 = 0
连续性模型	写入	n/a	5554AB	FF	第一个连续性周期
	写入	n/a	2AAAB54	00	第二个连续性周期
验证检测到的连续性模型	写入	n/a	555	70	将状态寄存器读取指令写入到晶片 0 内
	读取	n/a	x	RD	对于检测到的连续性模型，读取晶片 0 的状态以确认状态位 0 = 1

5.4 状态监控

EA 状态有三种监控方法。前几代 S29GL 闪存系列使用的方法称为数据轮询和就绪 / 忙碌 # (RY/BY#) 信号。S29GL-T 系列仍支持这些方法。另外一个方法是读取状态寄存器。

5.4.1 状态寄存器

编程和擦除操作的状态由一个 16 位状态寄存器提供。写入状态寄存器读取指令后，将对状态寄存器的内容进行一次读取。发出状态寄存器读取指令时，寄存器将（在 WE# 的上升沿上）捕获当前状态，并进入 ASO。状态寄存器的内容在器件地址空间的所有位置内均为别名（被覆盖）。对状态寄存器 ASO 中进行的有效读（CE# 和 OE# 为低电平）访问可使系统退出 ASO（在 t_{CEPH}/t_{OEPH} 时间内在 CE# 或 OE# 的上升沿上），并将其返回到发出状态寄存器读取指令时所使用的地址空间映射。在 x8 模式下，如果将 CE# 和 OE# 保持为低电平，并且在 A-1 上进行了一次切换，则可以通过一个状态寄存器进入指令读取整个状态寄存器（高位字节和低位字节）。写操作被忽略，并且器件处于状态寄存器 ASO 模式。

状态寄存器包含与最近完成的嵌入式算法（EA）结果相关的位 — 成功或失败：

- 擦除状态（位 5），
- 编程状态（位 4），
- 写入缓冲区异常终止（位 3），
- 扇区锁定状态（位 1），
- 检测到连续性检查模型（位 0）。

以及，与正在执行的 EA 的当前状态相关的位：

- 器件繁忙（位 7），
- 擦除挂起（位 6），
- 编程挂起（位 2），

当前状态位表明 EA 状态：正在进行、挂起或已完成。

高 8 位（位 15:8）保留。它们的值（高或低）不确定，每次状态读取的值不一定相同。软件读取状态时，这些位应忽略，而不用考虑。

如果状态寄存器位 3 为 0，则软复位指令将状态寄存器的位 [5、4、1、0] 清除为 0，但不会影响当前的状态位。

清除状态寄存器指令将状态寄存器的位 [5、4、3、1、0] 清除为 0，但并不影响当前状态位。

表 5.5 状态寄存器

位编号	15:8	7	6	5	4	3	2	1	0
位说明	保留	器件准备就绪位	擦除挂起状态位	擦除状态位	编程状态位	写入缓冲区终止状态位	编程挂起状态位	扇区锁定状态位	连续性检查
位名		DRB	ESSB	ESB	PSB	WBASB	PSSB	SLSB	CC
复位状态	X	1	0	0	0	0	0	0	0
繁忙状态	无效	0	无效	无效	无效	无效	无效	无效	无效
准备就绪状态	X	1	0 = 没有擦除被挂起 1 = 擦除被挂起	0 = 擦除成功 1 = 擦除失败	0 = 编程成功 1 = 编程失败	0 = 编程未异常中止 1 = 编程操作在进行写入缓冲区指令期间被中止	0 = 没有编程被挂起 1 = 编程被挂起	0 = 扇区在操作期间未锁定 1 = 扇区锁定错误	0 = 未检测到连续性检查模型 1 = 检测到连续性检查模型

注意:

1. 位 15 到 8 保留给将来使用，它们可能显示为 0 或 1。检查状态时，应该忽略（屏蔽）这些位。
2. 当器件中没有嵌入式算法正在执行时，位 7 为 1。
3. 仅当位 7 是 1 时，位 6 到 1 才有效。
4. 冷复位或热复位可将所有位置于它们的复位状态。
5. 清除状态寄存器指令或软件复位指令可将位 5、位 4、位 3 和位 1 清除为 0。
6. 发出擦除挂起指令时，用户必须继续读取状态，直至 DRB 变为 1。
7. 擦除恢复指令可将 ESSB 清除为 0。
8. ESB 反映最近擦除操作成功或失败。
9. PSB 反映最近编程操作成功或失败。
10. 在擦除挂起期间，将忽略对挂起扇区或队列中的扇区进行编程操作，而且不会报告错误。
11. 发出编程挂起指令时，用户必须继续读取状态，直到 DRB 变为 1 为止。
12. 编程恢复指令可将 PSSB 清除为 0。
13. SLSB 指明编程或擦除操作因扇区锁定而失败。
14. SLSB 反映最近编程或擦除操作的状态。

5.4.2 数据轮询状态

在活动的嵌入式算法期间，EAC 自动切换到数据轮询 ASO，此时任何读取访问显示的都是 EA 状态。状态信息的单个字在器件地址空间的所有位置均为别名。在状态字中，有多个位用于确定 EA 的状态。这些被看作是 DQ 位，当 EA 正在执行时，它们在读取访问期间出现在数据总线上。DQ 位 15 到 8、DQ4 和 DQ0 被保留，它们提供未定义的数据。状态监控软件必须屏蔽保留的位，而不用考虑它们。在 X8 模式下，当执行数据轮询时，A-1 将被忽略。第 36 页上的表 5.6 和后续小节介绍了剩余位的功能。

5.4.2.1 DQ7: Data# 轮询

Data# 轮询位 DQ7 向主机系统指明嵌入式算法是正在执行中还是已完成。Data# 轮询在编程或擦除指令序列中最后 WE# 脉冲的上升沿之后生效。请注意，对于写入缓冲区编程，Data# 轮询只对编程期间写入缓冲区页面内正在编程的最后一个字有效。如果对写入缓冲区页面中正在编程的最后一个字之外的任何字读取 Data# 轮询状态，将返回错误的状态信息。

在嵌入式编程算法期间，器件在 DQ7 输出为 DQ7 编程的数据位的补码。此 DQ7 状态还应用于擦除挂起期间编程操作的状态检查。当嵌入式编程算法完成时，器件输出为编程的最后一个字的位 7 编程的数据位。在编程挂起情况下，器件只允许读取阵列数据。如果编程地址位于受保护的扇区内，则 DQ7 上的 Data# 轮询在 t_{DP} 时间内有效，然后该器件返回读取阵列数据。

在嵌入式擦除、评估擦除状态或空白检查算法期间，Data# 轮询在 DQ7 上生成 0。当算法完成时，或者如果器件进入擦除挂起模式，Data# 轮询在 DQ7 上生成 1。这与为嵌入式编程算法介绍的补码 / 真数据输出相似：擦除功能将扇区中的所有位变成 1；在此之前，器件输出补码或 0。系统必须提供一个位于所选擦除扇区内的地址，以读取 DQ7 上的有效状态信息。

写入擦除指令序列后，如果所选擦除扇区受到保护，DQ7 上的 Data# 轮询将在 t_{DP} 时间内有效，然后器件返回读取阵列数据。

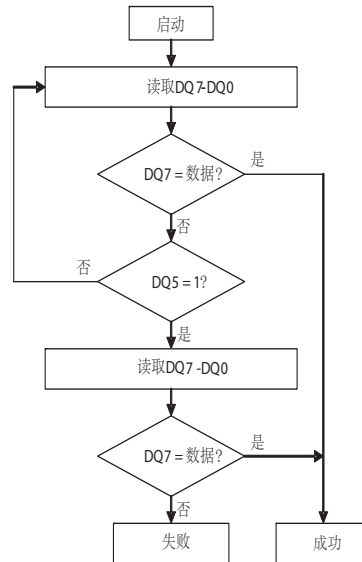
在嵌入式编程或擦除操作完成前，当输出使能 (OE#) 为低电平时，DQ7 可能与 DQ6-DQ0 异步变化。即，器件可能从提供状态信息变为在 DQ7 上提供有效数据。根据系统对 DQ7 输出进行采样的时间，可能读取状态或有效数据。即使器件已完成编程或擦除操作并且 DQ7 为有效数据，DQ6-DQ0 上的数据输出仍可能无效。DQ7-DQ0 上的有效数据出现在后续的读取循环。

当系统检测到 DQ7 已从补码变成真数据时，它可以在随后读取周期在 DQ15-DQ0 (x8 模式下的 Dq7-DQ0) 上读取有效数据。这是因为当输出使能 (OE#) 保持低电平时，DQ7 可能与 DQ6-DQ0 异步变化，如第 81 页上的图 10.17 中所示。第 36 页上的表 5.6 显示的是 DQ7 上 Data# 轮询的输出。第 21 页上的图 5.2 显示的是写入缓冲区编程所使用的 Data# 轮询算法。

有效的 DQ7 数据轮询状态只能从下列各项读取：

- 为写入缓冲区编程操作而加载到写入缓冲区中的最后一个字的地址；
- 单个字编程操作的位置；
- 正在执行擦除、评估擦除状态或空白检查的扇区中的地址；
- 或者芯片擦除期间任何扇区中的地址。

图 5.5 Data# 轮询算法



注意:

1. 由于 DQ7 可能与 DQ5 同时改变，因此即使 DQ5 = 1，也应重新检查 DQ7。

5.4.2.2 DQ6: 切换位 I

DQ6 上的切换位 I 指明嵌入式编程或擦除算法是正在执行还是已完成，或者器件是否已进入编程挂起或擦除挂起模式。切换位 I 可在任意地址读取，在指令序列最后 WE# 脉冲的上升沿之后（编程或擦除操作之前）生效。

在嵌入式编程或擦除算法操作期间，对任何地址的连续读取循环会导致 DQ6 切换。（系统可以使用 OE# 或 CE# 控制读取周期）。当操作完成时，DQ6 停止切换。

写入擦除指令序列后，如果所选擦除扇区受到保护，DQ6 将在 t_{DP} 时间内进行切换，然后 EAC 返回待机（读取模式）。如果所选扇区不受保护，嵌入式擦除算法擦除不受保护的扇区。

系统可以同时使用 DQ6 和 DQ2 来确定扇区是正在擦除还是擦除挂起。当器件正在擦除时（即嵌入式擦除算法正在执行中），DQ6 切换。当器件进入编程挂起模式或擦除挂起模式时，DQ6 停止切换。不过，系统还必须使用 DQ2 来确定哪个扇区正在擦除或擦除挂起。此外，系统还可以使用 DQ7（请参考第 33 页上的 DQ7: Data# 轮询）。

DQ6 在擦除挂起编程模式下也会切换，当嵌入式编程算法完成时停止切换。

第 36 页上的表 5.6 显示的是 DQ6 行上切换位 I 的输出。第 35 页上的图 5.6 以流程图形式显示切换位算法，而第 35 页上的读取切换位 DQ6/DQ2 说明了该算法。第 35 页上的图 5.6 显示的是切换位时序框图。另请参阅第 34 页上的 DQ2: 切换位 II。

5.4.2.3 DQ3: 扇区擦除定时器

写入扇区擦除指令序列后，系统可以读取 DQ3 以确定是否已开始擦除。更多信息，请参考第 26 页上的扇区擦除中介绍的内容。（扇区擦除定时器不适用于芯片擦除指令。）如果选择附加扇区进行擦除，整个超时在每个附加扇区擦除指令后也适用。当超时期间结束时，DQ3 从 0 切换到 1。如果系统附加扇区擦除指令之间的时间小于 t_{SEA}，则系统不需要监控 DQ3。

扇区擦除指令写入后，系统应读取 DQ7（Data# 轮询）或 DQ6（切换位 I）的状态以确保器件已接受指令序列，然后读取 DQ3。如果 DQ3 是 1，则嵌入式擦除算法已开始；所有进一步指令（擦除挂起除外）会被忽略，直至擦除操作完成。如果 DQ3 是 0，器件会接受附加扇区擦除指令。为确保指令已接受，系统软件应在每个后续扇区擦除指令之前和之后检查 DQ3 的状态。如果 DQ3 在第二次状态检查时为高电平，则最后一个指令可能尚未被接受。第 36 页上的表 5.6 显示相对于其他状态位的 DQ3 状态。

5.4.2.4 DQ2: 切换位 II

DQ2 上的切换位 II 与 DQ6 一起使用时，指明特定的扇区是正在擦除（即嵌入式擦除算法正在执行）还是擦除挂起。切换位 II 在指令序列中最后 WE# 脉冲的上升沿之后生效。

当系统读取所选擦除扇区（或在多扇区擦除期间为擦除操作选择的所有扇区）内的地址时，DQ2 将切换。（系统可以使用 OE# 或 CE# 控制读取周期）。但 DQ2 无法区分扇区是正在擦除还是擦除挂起。相对的，DQ6 通过比较，指明器件是正在擦除还是擦除挂起，但无法区分选择的扇区是否正在擦除或擦除挂起的扇区。因此，扇区和模式信息需要两个状态位。请参见第 36 页上的表 5.6 以比较

DQ2 和 DQ6 的输出。第 34 页上的图 5.5 以流程图形式显示切换位算法，并且第 35 页上的读取转换位 DQ6/DQ2 说明了该算法。第 35 页上的图 5.6 显示了切换位的时序框图。

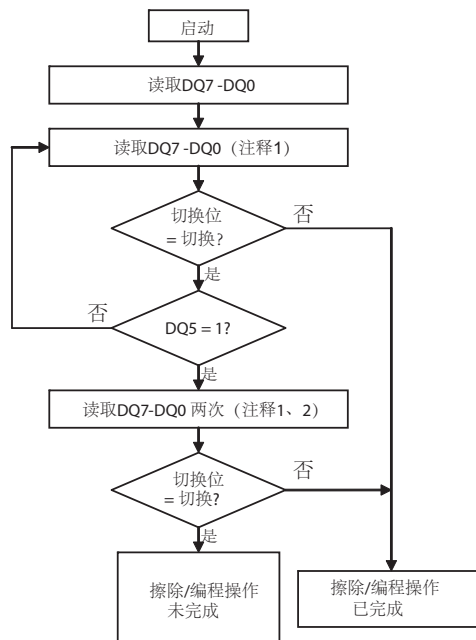
5.4.2.5 读取转换位 DQ6/DQ2

关于下面的讨论，请参见第 34 页上的图 5.5。每当系统开始读取切换位状态时，它必须连续读取 DQ7-DQ0 至少两次，以确定切换位是否切换。通常，系统在第一次读取后会记录和存储切换位的值。第二次读取后，系统会比较切换位的新值和前一个值。如果切换位未切换，则器件已完成编程或擦除操作。系统可以在随后的读取周期读取 DQ15-DQ0（x8 模式下的 DQ7-DQ0）上的阵列数据。

不过，如果最初两个读取循环后，系统确定切换位仍切换，系统也应注意 DQ5 的值是否为高（请参考第 35 页上的 DQ5：超时限制）。若是，系统随后应再次确定切换位是否切换，因为 DQ5 变为高后切换位可能停止切换。如果切换位不再切换，则器件已成功完成编程或擦除操作。如果它仍切换，则器件未成功完成操作，系统必须写入复位指令以返回读取阵列数据。建议将用于轮询目的的数据读取专用于轮询目的。一旦该切换停止，便能够对阵列数据进行后续读取。

其余情况是系统在最初确定切换位正在切换，并且 DQ5 尚未变为高。系统可能通过后续读取循环继续监控切换位和 DQ5，以确定状态，如前一段所述。此外，它可以选择执行其他系统任务。在此情况下，当系统返回以确定操作状态时，必须在算法开始处启动（第 35 页上的图 5.6 的顶部）。

图 5.6 切换位程序



注意:

1. 读取切换位两次可确定其是否切换。参见文字说明。
2. 切换位可能随着 DQ5 变为 1 而停止切换，因此需重新检查。参见文字说明。

5.4.2.6 DQ5：超时限制

DQ5 指明编程或擦除时间是否超出了指定的内部脉冲计数限制。在这些情况下，DQ5 生成“1”，以指明编程或擦除周期未成功完成。系统必须发出复位命令，以使器件返回阵列数据读取模式。

发生超时，软件必须发出软件复位或状态寄存器复位命令以清除超时位（DQ5）并使 EAC 返回初始状态。在此情况下，在复位命令发出后闪存有可能继续通信忙碌最多 t_{TOR} 时间。

5.4.2.7 DQ1：写入到缓冲区终止

DQ1 指明写入到缓冲区操作是否已终止。在这些情况下，DQ1 生成“1”。系统必须发出写入到缓冲区终止复位命令序列或状态寄存器擦除命令，以使 EAC 返回待机（读取模式）并清除状态寄存器失败位。更多信息，请参考第 19 页上的写入缓冲区编程中的内容。

表 5.6 数据轮询状态

操作		DQ7 (2)	DQ6	DQ5 (1)	DQ3	DQ2 (2)	DQ1 (4)	RY/BY#
标准模式	嵌入式编程算法	DQ7#	切换	0	N/A	不切换	0	0
	在擦除扇区的地址内读取 (5)	0	切换	0	1	切换	N/A	0
	在擦除扇区的地址外读取 (5)	0	切换	0	1	不切换	N/A	0
编程挂起模式 (3)	在编程挂起扇区的地址内读取	无效 (不允许)	无效 (不允许)	无效 (不允许)	无效 (不允许)	无效 (不允许)	无效 (不允许)	1
	在非编程挂起扇区的地址内读取	数据	数据	数据	数据	数据	数据	1
擦除挂起模式 (7)	在擦除挂起扇区的地址内读取	1	不切换	0	N/A	切换	N/A	1
	在非擦除挂起扇区的地址内读取	数据	数据	数据	数据	数据	数据	1
	在非擦除挂起的扇区内编程	DQ7#	切换	0	N/A	N/A	N/A	0
写入到缓冲区 (4) (6)	忙碌状态	DQ7#	切换	0	N/A	不切换	0	0
	超时	DQ7#	切换	1	N/A	N/A	0	0
	异常中止状态	DQ7#	切换	0	N/A	N/A	1	0

注意:

1. 当嵌入式编程或嵌入式擦除操作超出了最大时间限制时，DQ5 变为“1”。有关的详细信息，请参见第 35 页上的 DQ5: 超时限制。
2. 当读取状态信息时，DQ7 和 DQ2 需要一个有效的地址。有关的详细信息，请参见相应的小节。
3. 编程挂起行中地址的数据是无效的。可对所有地址（除编程挂起行中的地址之外）进行读取，以获得有效数据。
4. DQ1 指明在写入缓冲区编程操作期间的写入到缓冲区终止状态。
5. 多扇区擦除期间，加载最后的一个扇区后，DQ3 在 50 μs 中保持为“0”。
6. 仅针对编程操作。
7. 当某一挂起扇区处于 SECSI 模式时，如果在该模式下启动了某一编程操作，那么在嵌入式操作期间，DQ6 会切换，DQ2 则不切换。

5.5 错误类型和清除步骤

嵌入式操作状态方法报告的错误类型有三种。根据错误类型，报告的状态和清除错误状态的步骤会有所不同。下面是错误清除后的状态：

- 如果在错误发生之前器件进入了 ASO，它保持进入该状态，等待 ASO 读取或命令写入。
- 如果在错误发生之前一个擦除操作被挂起，器件会返回擦除挂起状态，等待闪存阵列读取或命令写入。
- 否则，器件将处于待机状态，等待闪存阵列读取或命令写入。

5.5.1 嵌入式操作错误

如果在嵌入式操作（编程、擦除、空白检查、或密码解锁）期间发生错误，器件（EAC）保持忙碌状态。RY/BY# 输出保持低电平，所有地址位置继续处于数据轮询状态，状态寄存器显示准备就绪和有效的状态位。器件一直忙碌，直到主机系统状态监控检测到错误状态并且错误状态被清除为止。

在嵌入式算法错误状态期间，数据轮询状态将显示如下：

- DQ7 是加载到写入缓冲区的最后一个字（密码解锁命令时是密码的最后一个字）中 DQ7 位的求反。DQ7 = 0 时，擦除，评估擦除状态或空白检查失败
- DQ6 继续切换
- DQ5 = 1；嵌入式操作失败
- DQ4 是保留，无需关注（屏蔽）
- DQ3 = 1 指明正在进行嵌入式扇区擦除操作；= 0 则指明正在进行嵌入式编程操作
- DQ2 继续切换，与用于读取状态的地址无关
- DQ1 = 0；写入缓冲区终止错误
- DQ0 是保留，无需关注（屏蔽）

在嵌入式算法错误状态期间，状态寄存器显示如下：

- SR[7] = 1；显示有效状态
- SR[6] = X；在 EA 错误期间可能是或不是擦除挂起
- SR[5] = 1；指明擦除或空白检查错误；其他情况 = 0
- SR[4] = 1；指明编程或密码解锁错误；其他情况 = 0
- SR[3] = 0；写入缓冲区终止
- SR[2] = 0；编程挂起
- SR[1] = 0；扇区受保护
- SR[0] = X；保留，无需关注（屏蔽）

当检测到嵌入式算法错误状态时，必须清除错误状态，以便返回正常操作，RY/BY# 为高，为新读取或命令写入做好准备。错误状态可通过写入下列命令来清除：

- 复位命令
- 状态寄存器清除命令

在嵌入式算法错误状态期间可接受的命令包括：

- 状态寄存器读取
- 复位命令
- 状态寄存器清除命令

5.5.2 保护错误

如果嵌入式算法尝试更改受保护区域的数据（编程或擦除受保护的扇区或 OTP 区域），器件（EAC）会进入忙碌状态 t_{DP} 时间，然后返回正常操作。在忙碌期间，RY/BY# 输出保持低电平，所有地址位置继续处于数据轮询状态，状态寄存器显示未准备就绪和无效的状态位（SR[7] = 0）。

在保护错误状态忙碌期间，数据轮询状态将显示如下：

- DQ7 是加载到写入缓冲区的最后一个字中 DQ7 位的求反。DQ7 = 0 表示擦除失败
- DQ6 继续切换，与用于读取状态的地址无关
- DQ5 = 0；指明在忙碌期间嵌入式操作没有失败
- DQ4 是保留，无需关注（屏蔽）
- DQ3 = 1 指明正在执行嵌入式扇区擦除
- DQ2 继续切换，与用于读取状态的地址无关
- DQ1 = 0；写入缓冲区终止错误
- DQ0 是保留，无需关注（屏蔽）

在保护错误状态忙碌期间可接受的命令包括：

- 状态寄存器读取

忙碌期间结束时，器件返回正常操作，数据轮询状态不再存在，RY/BY# 为高，状态寄存器显示准备就绪和有效状态位。器件已准备好对闪存阵列执行新读取或写入命令。

经过保护错误状态忙碌期间后，状态寄存器显示如下：

- SR[7] = 1；显示有效状态
- SR[6] = X；在保护错误忙碌期间后可能是或不是擦除挂起
- SR[5] = 1；指明擦除错误，其他情况 = 0
- SR[4] = 1；指明编程错误，其他情况 = 0
- SR[3] = 0；编程未终止
- SR[2] = 0；没有编程挂起
- SR[1] = 1；因尝试更改受保护的位置而发生错误
- SR[0] = X；保留，无需关注（屏蔽）

经过保护错误状态忙碌期间后可接受的命令包括：

- 任何命令

5.5.3 写入缓冲区终止

如果在执行写入到缓冲区命令期间发生错误，器件（EAC）会保持忙碌状态。RY/BY# 输出保持低电平，所有地址位置继续处于数据轮询状态，状态寄存器显示准备就绪和有效的状态位。器件一直忙碌，直到主机系统状态监控检测到错误状态并且错误状态被清除为止。

在写入到缓冲区终止（WBA）错误状态期间，数据轮询状态将显示如下：

- DQ7 是加载到写入缓冲区的最后一个字中 DQ7 位的求反
- DQ6 继续切换，与用于读取状态的地址无关
- DQ5 = 0；指明编程操作没有失败。WBA 是在编程操作可以开始之前写入到缓冲区命令输入的值中的一个错误
- DQ4 是保留，无需关注（屏蔽）
- 编程操作后，由于没有执行任何擦除操作，无需关注 DQ3。如果擦除操作挂起后，开始执行一个写入缓冲区操作，则 DQ3 将为“1”。如果未执行任何擦除操作，则 DQ3 为无需关注项，并应被屏蔽。
- 由于编程操作后，没有执行任何擦除操作，DQ2 不切换。如果擦除操作挂起后，开始执行一个写入缓冲区编程操作，则 DQ2 将仅在擦除操作挂起的扇区内切换。如果未执行任何擦除操作，则 DQ2 为无需关注，并应被屏蔽。
- DQ1 = 1：写入缓冲区终止错误
- DQ0 是保留，无需关注（屏蔽）

在写入到缓冲区终止（WBA）错误状态期间，状态寄存器将显示如下：

- SR[7] = 1；显示有效状态
- SR[6] = X；在 WBA 错误状态期间可能是或不是擦除挂起
- SR[5] = 0；擦除成功
- SR[4] = 1；与编程相关的错误
- SR[3] = 1；写入缓冲区终止
- SR[2] = 0；没有编程挂起
- SR[1] = 0；在操作期间扇区未锁定
- SR[0] = X；保留，无需关注（屏蔽）

当检测到 WBA 错误状态时，必须清除错误状态，以便返回正常操作，RY/BY# 为高，为新读取 或命令写入做好准备。可通过写入下列命令来清除错误状态，并使器件返回到正常操作：

- 写入缓冲区终止复位命令
- 状态寄存器清除命令

在写入到缓冲区终止（WBA）错误状态期间可接受的命令包括：

- 状态寄存器读取
 - 读取状态寄存器并返回 WBA 忙碌状态
- 写入缓冲区终止复位命令
- 状态寄存器清除命令

5.6 嵌入式算法性能表

联合电子器件工程委员会（JEDEC）标准 JESD22-A117 根据一个合格规范对执行耐久性和保持时间测试的步骤要求进行了定义。该测试的目的在于确定闪存器件在无失败的条件下保持重复的数据更改的能力（编程 / 擦除耐久性）和在预期时间内保持数据的能力（数据保持）。耐久性和数据保持合格规范在 JESD47 中指定，也可以通过知识库方法进行开发，如 JESD94 中所示。

表 5.7 嵌入式算法特性（温度为 -40°C ~ +85°C）

参数		最小值	典型值 (2)	最大值 (3)	单位	注释
扇区擦除（大小为 128 kB）时间			535	3500	ms	包括擦除之前的预编程时间 (7)
芯片擦除时间	GL512T		274	1792 (1)	s	
	GL01GT		548	3584 (1)	s	
单字编程时间 (1)			160	750	µs	
缓冲区编程时间	2 字节 (1)		160	750	µs	
	32 字节 (1)		195	750		
	64 字节 (1)		219	750		
	128 字节 (1)		258	750		
	256 字节 (1)		327	750		
	512 字节 (6)		451	750		
写入缓冲区编程操作对每个字的有效编程时间	512 字节		1.76		µs	
扇区编程（大小为 128 kB）时间 （全缓冲区编程）			115.4	192	ms	(8)
执行擦除挂起操作时的延迟时间 (t _{ESL})				40	µs	
执行编程挂起操作时的延迟时间 (t _{PSL})				40	µs	
从擦除恢复到下一次擦除挂起的时间 (t _{ERS})			100		µs	最少为 60 µs，但 ≥ 擦除完成所需的典型时长
从编程恢复到下一次编程挂起的时间 (t _{PRS})			100		µs	最少为 60 µs，但 ≥ 编程完成所需的典型时长
评估擦除状态时间 (t _{EES})			25	30	µs	
空白检查时间			6.2	8.5	ms	
NOP（每行的编程操作数量）				256		
每个扇区的擦除次数 (1)			100K		周期	

注意：

- 并非 100% 经过了测试。
- 编程和擦除的典型时间假设使用以下条件：温度 = 25°C，V_{CC} = 3.0 V，10,000 次循环，使用交错式数据图案。
- 在最坏条件下：90°C，V_{CC} = 2.7 V，100,000 次循环，使用交错式数据图案。
- 数据保持时间为 20 年（1,000 次擦除循环）。
- 有效的写入缓冲区规范基于 512 字节写入缓冲区操作。
- x8 模式不支持 512 字节的加载。
- 在嵌入式擦除算法的预编程步骤，在扇区和芯片擦除前所有字均编程为 0000h。
- 系统级开销是执行编程命令的总线循环序列所需要的时间。欲了解命令定义的信息，请参见第 43 页上的表 6.1，指令定义 x16。

表 5.8 嵌入式算法特性（温度为 -40°C ~ +105°C）

参数		最小值	典型值 (2)	最大值 (3)	单位	注释
扇区擦除（大小为 128 kB）时间			535	3500	ms	
芯片擦除时间	GL512T		274	1792 (1)	s	包括擦除之前的预编程时间 (7)
	GL01GT		548	3584 (1)		
单字编程时间 (1)			160	1050	µs	
缓冲区编程时间	2 字节 (1)		160	1050	µs	
	32 字节 (1)		195	1050		
	64 字节 (1)		219	1050		
	128 字节 (1)		258	1050		
	256 字节 (1)		327	1050		
	512 字节 (1)		451	1050		
写入缓冲区编程操作对每个字的有效编程时间	512 字节		1.76		µs	
扇区编程（大小为 128 kB）时间 （全缓冲区编程）			115.4	269	ms	(8)
执行擦除挂起操作时的延迟时间 (t _{ESL})				50	µs	
执行编程挂起操作时的延迟时间 (t _{PSL})				50	µs	
从擦除恢复到下一次擦除挂起的时间 (t _{ERS})			100		µs	最少为 60 ns，但 ≥ 擦除完成所需的典型时长。
从编程恢复到下一次编程挂起的时间 (t _{PRS})			100		µs	最少为 60 ns，但 ≥ 编程完成所需的典型时长。
评估擦除状态时间 (t _{EES})			25	30	µs	
空白检查时间			7.6	9.0	ms	
NOP（每行的编程操作数量）				每 16 字进行 一次编程		
每个扇区的擦除次数 (1)			100K		周期	

注意：

- 并非 100% 经过了测试。
- 编程和擦除的典型时间假设使用以下条件：温度 = 25°C，V_{CC} = 3.0 V，10,000 次循环，使用交错式数据图案。
- 在最坏条件下：105°C，V_{CC} = 2.7 V，100,000 次循环，使用交错式数据图案。
- 数据保持时间为 20 年（1,000 次擦除循环）。
- 有效的写入缓冲区规范基于 512 字节写入缓冲区操作。
- x8 模式不支持 512 字节的加载。
- 在嵌入式擦除算法的预编程步骤，在扇区和芯片擦除前所有字均编程为 0000h。
- 系统级开销是执行编程命令的总线循环序列所需要的时间。欲了解命令定义的信息，请参见第 43 页上的表 6.1，指令定义 x16。

表 5.9 嵌入式算法特性 (温度为 -40°C ~ +125°C)

参数		最小值	典型值 (2)	最大值 (3)	单位	注释
扇区擦除 (大小为 128 kB) 时间			535	3500	ms	
芯片擦除时间	GL512T		274	1792 (1)	s	包括擦除之前的预编程时间 (7)
	GL01GT		548	3584 (1)		
单字编程时间 (1)			160	1050	μs	
缓冲区编程时间	2 字节 (1)		160	1050	μs	
	32 字节 (1)		195	1050		
	64 字节 (1)		219	1050		
	128 字节 (1)		258	1050		
	256 字节 (1)		327	1050		
	512 字节 (1)		451	1050		
写入缓冲区编程操作对每个字的有效编程时间	512 字节		1.76		μs	
扇区编程 (128 kB) 时间 (编程整个缓冲区)			115.4	269	ms	(8)
擦除挂起的延迟时间 (t _{ESL})				50	μs	
编程挂起的延迟时间 (t _{PSL})				50	μs	
从擦除恢复到下一次擦除挂起的时间 (t _{ERS})			100		μs	最少为 60 ns, 但 ≥ 擦除完成所需的典型时长。
从编程恢复到下一次编程挂起的时间 (t _{PRS})			100		μs	最少为 60 ns, 但 ≥ 编程完成所需的典型时长。
评估擦除状态时间 (t _{EES})			25	30	μs	
空白检查时间			7.6	9.0	ms	
NOP (每行的编程次数)				每 16 字进行一次编程		
每个扇区的擦除次数 (1)			10K		周期	

注意:

- 并非 100% 经过了测试。
- 编程和擦除的典型时间假设使用以下条件: 温度 = 25°C, V_{CC} = 3.0 V, 10,000 次擦写, 使用随机数据图案。
- 在最坏条件下进行测量: 温度 = 105°C, V_{CC} = 2.7 V, 100,000 次擦写, 使用随机数据图案。
- 数据保持时间为 20 年 (1,000 次擦除循环)。
- 有效的写入缓冲区规范是基于 512 字节写入缓冲区操作。
- x8 模式不支持 512 字节的加载。
- 在嵌入式擦除算法的预编程步骤中, 扇区和芯片擦除前所有字均被编程为 0000h。
- 系统级开销是执行编程命令的总线循环序列所需要的时间。欲了解命令定义的信息, 请参见第 43 页上的表 6.1, 指令定义 x16。

6. 软件接口参考

6.1 命令汇总

表 6.1 指令定义 x16

指令序列 (1)	周期	总线周期 (2-5)													
		第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
读取 (6)	1	RA	RD												
复位 / 退出 ASO (7、18)	1	XXX	F0												
状态寄存器读取	2	555	70	XXX	RD										
状态寄存器清除	1	555	71												
字编程	4	555	AA	2AA	55	555	A0	PA	PD						
写入到缓冲区	6	555	AA	2AA	55	SA	25	SA	WC	WBL	PD	WBL	PD		
编程缓冲区到闪存 (确认)	1	SA	29												
复位 “写缓冲区异常中止” 状态 (13)	3	555	AA	2AA	55	555	F0								
解锁省略	进入	3	555	AA	2AA	55	555	20							
	编程 (9)	2	XXX	A0	PA	PD									
	写入到缓冲区 (9)	4	SA	25	SA	WC	WBL	PD	WBL	PD					
	编程缓冲区到闪存 (确认)	1	SA	29											
	复位 “写缓冲区异常中止” 状态 (13)	3	555	AA	2AA	55	555	F0							
	扇区擦除 (9)	2	XXX	80	SA	30									
	芯片擦除 (9)	2	XXX	80	XXX	10									
退出指令集 (10)	2	XXX	90	XXX	00										
芯片擦除	6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10		
扇区擦除 (20)	6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30		
擦除挂起 / 编程挂起传统方法 (11)	1	XXX	B0												
擦除挂起增强方法															
擦除恢复 / 编程恢复传统方法 (12)	1	XXX	30												
擦除恢复增强方法															
编程挂起增强方法	1	XXX	51												
编程恢复增强方法	1	XXX	50												
评估擦除状态	1	(SA) 555	35												
空白检查	1	(SA) 555	33												
CFI 进入 (8)	1	(SA) 55	98												
连续性检查	7	555	71	555	70	XX	RD	2AAA A55	FF00	15555 AA	00FF	555	70	XX	RD
ID-CFI (自动选择)	ID (自动选择) 进入	3	555	AA	2AA	55	555	90							
	进入 CFI (8)	1	55	98											
	读取 ID-CFI	1	RA	RD											
	退出 CFI	1	XXX	FF											
	复位 / 退出 ASO (7、18)	1	XXX	F0											

表 6.1 指令定义 x16 (续)

指令序列 (1)		周期	总线周期 (2-5)													
			第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
			地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
安全硅区域指令定义																
安全硅区域 (SSR) ASO	SSR 进入	3	555	AA	2AA	55	(SA) 555	88								
	读取 (6)	1	RA	RD												
	字编程	4	555	AA	2AA	55	555	A0	PA	PD						
	写入到缓冲区	6	555	AA	2AA	55	SA	25	SA	WC	WBL	PD	WBL	PD		
	编程缓冲区到闪存 (确认)	1	SA	29												
	复位 “写缓冲区异常中止” 状态 (13)	3	555	AA	2AA	55	555	F0								
	退出 SSR (13)	4	555	AA	2AA	55	555	90	XX	0						
	复位 / 退出 ASO (7、18)	1	XXX	F0												

表 6.1 指令定义 x16 (续)

指令序列 (1)	周期	总线周期 (2-5)														
		第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期		
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	
锁定寄存器指令集定义																
锁定寄存器 ASO	锁定寄存器进入	3	555	AA	2AA	55	555	40								
	编程 (17)	2	XXX	A0	XXX	PD										
	读取 (17)	1	0	RD												
	退出指令集 (14、18)	2	XXX	90	XXX	0										
	复位/退出 ASO (7、18)	1	XXX	F0												
密码保护指令集定义																
密码 ASO	密码 ASO 进入	3	555	AA	2AA	55	555	60								
	编程 (16)	2	XXX	A0	PWAx	PWDx										
	读取 (15)	4	0	PWD0	1	PWD1	2	PWD2	3	PWD3						
	解锁 (15)	7	0	25	0	3	0	PWD0	1	PWD1	2	PWD2	3	PWD3	0	29
	退出指令集 (14、18)	2	XXX	90	XXX	0										
	复位/退出 ASO (7、18)	1	XXX	F0												
非易失性扇区保护指令集定义																
PPB (非易失性扇区保护)	PPB 进入	3	555	AA	2AA	55	555	C0								
	PPB 编程 (19)	2	XXX	A0	SA	0										
	所有 PPB 擦除 (19)	2	XXX	80	0	30										
	PPB 读取 (19)	1	SA	RD (0)												
	退出指令集 (14、18)	2	XXX	90	XXX	0										
	复位/退出 ASO (7、18)	1	XXX	F0												
全局非易失性扇区保护冻结指令集定义																
PPB 锁定位	PPB 锁定进入	3	555	AA	2AA	55	555	50								
	PPB 锁定清除	2	XXX	A0	XXX	0										
	PPB 锁定状态读取 (19)	1	XXX	RD (0)												
	退出指令集 (14、18)	2	XXX	90	XXX	0										
	复位/退出 ASO (18)	1	XXX	F0												

表 6.1 指令定义 x16 (续)

指令序列 (1)	周期	总线周期 (2-5)													
		第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
易失性扇区保护指令集定义															
DYB (易失性扇区保护)	DYB ASO 进入	3	555	AA	2AA	55	555	E0							
	DYB 设置 (19)	2	XXX	A0	SA	0									
	DYB 清除 (19)	2	XXX	A0	SA	1									
	DYB 状态读取 (19)	1	SA	RD (0)											
	退出指令集 (14、18)	2	XXX	90	XXX	0									
	复位 / 退出 ASO (18)	1	XXX	F0											

图标:

X = 无需关注。

RA = 要读取的存储器地址。

RD = 在读取操作期间从位置 RA 读取的数据。

PA = 要编程的存储器位置的地址。

PD = 要在位置 PA 编程的数据。

SA = 所选扇区的地址。地址位 Amax-A16 唯一地选择任一扇区。

WBL = 写入缓冲区位置。地址必须在同一行内。

WC = 字计数，等于要加载的写入缓冲区位置数量减 1。

PWAx = PPB 密码地址，word0 = 00h、word1 = 01h、word2 = 02h 和 word3 = 03h。SSR3 密码地址，word0 = 10h、word1 = 11h、word2 = 12h 和 word3 = 13h。

PWDx = 密码数据 word0、word1、word2 和 word3。

灰色与白色空格 = 读与写操作

注意:

1. 请参见第 58 页上的表 8.1，接口状态，了解总线操作的信息。
2. 所有值均为十六进制数值。
3. 除了下列周期外，所有总线周期均为写入周期；发生于读取过程的读取周期、ID/CFI 读取（制造商 ID/ 器件 ID）、指示位、安全硅区域读取、SSR 锁定读取以及状态寄存器读取的第二个周期。
4. 除了 RD、PD、WC 和 PWD 外，命令序列中的数据位 DQ15-DQ8 无需关注。
5. 除非需要 SA 或 PA，否则在解锁和命令循环期间，地址位 Amax-A11 均为无需关注。（Amax 是最高地址引脚）。
6. 读取阵列的数据时，不需要解锁或命令周期。
7. 在器件处于 ASO 模式或者 DQ5 变为高电平（当器件正在提供状态数据时）的情况下，需要通过复位命令返回阵列数据读取模式。
8. 器件就绪读取阵列数据时，命令将有效。
9. 在解锁省略编程和解锁省略写入到缓冲区命令之前，需要使用解锁省略命令。
10. 当器件处于解锁省略模式时，需要执行解锁省略复位命令，才能返回读取阵列数据。
11. 在擦除挂起模式下，系统可以对非擦除扇区进行读取和编程 / 编程挂起的操作，或者从该模式进入 ID-CFI ASO 模式。擦除挂起命令仅在扇区擦除操作期间有效。
12. 擦除恢复 / 编程恢复命令仅在擦除挂起 / 编程挂起模式下有效。
13. 检测到器件处于写入到缓冲区终止状态后，发出该命令序列以返回到读取状态。请注意，如果进行复位以退出终止状态，则需使用完整的命令序列。
14. 退出命令会使器件返回到读取阵列状态。
15. 密码的各部分可以以任何顺序输入或读取，前提是要输入或读取整个 64 位密码。如果访问 SSR3，则地址为 10h-13h。
16. 对于 PWDx，每个 A0 命令只能编程密码的一部分。密码的各部分必须按连续顺序进行编程（PWD0 - PWD3）。
17. 所有锁定寄存器位都只能编程一次。编程状态 = 0，擦除状态 = 1。此外，持久保护模式锁定位和密码保护模式锁定位不能同时编程，否则将终止锁定寄存器位编程操作，并使器件返回读取模式。保留给未来使用的锁定寄存器位未定义，可能是 0 或 1。
18. 如果发出任何进入命令，则必须发出退出命令，以使器件返回到读取状态。
19. 受保护状态 = 00h，不受保护状态 = 01h。DYB 设置、DYB 擦除或 PPB 编程命令的扇区地址可以是扇区内的任意位置 - 扇区地址的低位无需关注。
20. 请参见第 26 页上的 5.3.7.2 节，扇区擦除，了解多扇区擦除的信息。

表 6.2 指令定义 x8

指令序列 (1)	周期	总线周期 (2-5)													
		第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
读取 (5)	1	RA	RD												
复位 / 退出 ASO (6、17)	1	XXX	F0												
状态寄存器读取	2	AAA	70	XXX	RD										
状态寄存器清除	1	AAA	71												
字编程	4	AAA	AA	555	55	AAA	A0	PA	PD						
写入到缓冲区 (19)	6	AAA	AA	555	55	SA	25	SA	WC	WBL	PD	WBL	PD		
编程缓冲区到闪存 (确认)	1	SA	29												
复位“写缓冲区异常中止”状态 (12)	3	AAA	AA	555	55	AAA	F0								
解锁省略	进入	3	AAA	AA	555	55	AAA	20							
	编程 (8)	2	XXX	A0	PA	PD									
	写入到缓冲区 (8)	4	SA	25	SA	WC	WBL	PD	WBL	PD					
	将缓冲区中的数据编程到闪存 (确认) (8)	1	SA	29											
	复位“写缓冲区异常中止” 状态 (12)	3	AAA	AA	555	55	AAA	F0							
	扇区擦除 (8)	2	XXX	80	SA	30									
	芯片擦除 (8)	2	XXX	80	XXX	10									
	退出指令集 (9)	2	XXX	90	XXX	00									
芯片擦除	6	AAA	AA	555	55	AAA	80	AAA	AA	555	55	AAA	10		
扇区擦除 (19)	6	AAA	AA	555	55	AAA	80	AAA	AA	555	55	SA	30		
擦除挂起 / 编程挂起传统方法 (10)	1	XXX	B0												
擦除挂起增强方法															
擦除恢复 / 编程恢复传统方法 (11)	1	XXX	30												
擦除恢复增强方法															
编程挂起增强方法	1	XXX	51												
编程恢复增强方法	1	XXX	50												
评估擦除状态	1	(SA) AAA	35												
空白检查	1	(SA) AAA	33												
CFI 进入 (7)	1	(SA) AA	98												
连续性检查	7	AAA	71	AAA	70	XX	RD	55554 AB	FF	2AAA B54	00	AAA	70	XX	RD
ID-CFI (自动选择)	ID (自动选择) 进入	3	AAA	AA	555	55	AAA	90							
	CFI 进入 (7)	1	AA	98											
	ID-CFI 读取	1	RA	RD											
	退出 CFI	1	XXX	FF											
	复位 / 退出 ASO (6、17)	1	XXX	F0											

表 6.2 指令定义 x8 (续)

指令序列 (1)		周期	总线周期 (2-5)													
			第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
			地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
安全硅区域指令定义																
安全硅区域 (SSR) ASO	SSR 进入	3	AAA	AA	555	55	(SA) AAA	88								
	读取 (5)	1	RA	RD												
	字编程	4	AAA	AA	555	55	AAA	A0	PA	PD						
	写入到缓冲区 (19)	6	AAA	AA	555	55	SA	25	SA	WC	WBL	PD	WBL	PD		
	编程缓冲区到闪存 (确认)	1	SA	29												
	复位“写缓冲区异常中止” 状态 (12)	3	AAA	AA	555	55	AAA	F0								
	SSR 退出 (12)	4	AAA	AA	555	55	AAA	90	XX	0						
	复位 / 退出 ASO (6、17)	1	XXX	F0												

表 6.2 指令定义 x8 (续)

指令序列 (1)	周期	总线周期 (2-5)														
		第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期		
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	
锁定寄存器指令集定义																
锁定寄存器 ASO	锁定寄存器进入	3	AAA	AA	555	55	AAA	40								
	编程 (16)	2	XXX	A0	XXX	PD										
	读取 (16)	1	0	RD												
	退出指令集 (13、17)	2	XXX	90	XXX	0										
	复位 / 退出 ASO (6、17)	1	XXX	F0												
密码保护指令集定义																
密码 ASO	密码 ASO 进入	3	AAA	AA	555	55	AAA	60								
	编程 (15)	2	XXX	A0	PWAx	PWDx										
	读取 (14)	8	0	PWD0	1	PWD1	2	PWD2	3	PWD3	4	PWD4	5	PWD5	6	PWD6
		7	PWD7													
	解锁 (14)	11	0	25	0	3	0	PWD0	1	PWD1	2	PWD2	3	PWD3	4	PWD4
		5	PWD5	6	PWD6	7	PWD7	0	29							
	退出指令集 (13、17)	2	XXX	90	XXX	0										
复位 / 退出 ASO (6、17)	1	XXX	F0													
非易失性扇区保护指令集定义																
PPB (非易失性扇区保护)	PPB 进入	3	AAA	AA	555	55	AAA	C0								
	PPB 编程 (18)	2	XXX	A0	SA	0										
	所有 PPB 擦除 (18)	2	XXX	80	0	30										
	PPB 读取 (18)	1	SA	RD (0)												
	退出指令集 (13、17)	2	XXX	90	XXX	0										
	复位 / 退出 ASO (6、17)	1	XXX	F0												
	全局非易失性扇区保护冻结指令集定义															
PPB 锁定位	PPB 锁定进入	3	AAA	AA	555	55	AAA	50								
	PPB 锁定清除	2	XXX	A0	XXX	0										
	PPB 锁定状态读取 (18)	1	XXX	RD (0)												
	退出指令集 (13、17)	2	XXX	90	XXX	0										
	复位 / 退出 ASO (17)	1	XXX	F0												

表 6.2 指令定义 x8 (续)

指令序列 (1)		周期	总线周期 (2-5)													
			第一个周期		第二个周期		第三个周期		第四个周期		第五个周期		第六个周期		第七个周期	
			地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
易失性扇区保护指令集定义																
DYB (易失性扇区保护)	DYB ASO 进入	3	AAA	AA	555	55	AAA	E0								
	DYB 设置 (18)	2	XXX	A0	SA	0										
	DYB 清除 (18)	2	XXX	A0	SA	1										
	DYB 状态读取 (18)	1	SA	RD (0)												
	退出指令集 (13、17)	2	XXX	90	XXX	0										
	复位 / 退出 ASO (17)	1	XXX	F0												

图标:

X = 无需关注。

RA = 要读取的存储器地址。

RD = 在读取操作期间从位置 RA 读取的数据。

PA = 要编程的存储器位置的地址。

PD = 要在位置 PA 编程的数据。

SA = 所选扇区的地址。地址位 Amax-A16 唯一地选择任一扇区。

WBL = 写入缓冲区位置。地址必须在同一行内。

WC = 字计数，等于要加载的写入缓冲区位置数量减 1。

PWAx = PPB 密码地址，byte0 = 00h、byte1 = 01h、byte2 = 02h、byte3 = 03h、byte4 = 04h、byte5 = 05h、byte6 = 06h 和 byte7 = 07h。

SSR3 密码地址，byte0 = 20h、byte1 = 21h、byte2 = 22h、byte3 = 23h、byte4 = 24h、byte5 = 25h、byte6 = 26h 和 byte7 = 27h。

PWDx = 密码数据 byte0、byte1、byte2、byte3、byte4、byte5、byte6 和 byte7

灰色与白色空格 = 读与写操作

注意:

1. 请参见第 58 页上的表 8.1，接口状态，了解总线操作的信息。
2. 所有值均为十六进制数值。
3. 除了下列周期外，所有总线周期均为写入周期：发生于读取过程的读取周期、ID/CFI 读取（制造商 ID/ 器件 ID）、指示位、安全硅区域读取、SSR 锁定读取以及状态寄存器读取的第二个周期。
4. 除非需要 SA 或 PA，否则在解锁和命令循环期间，地址位 Amax-A11 均为无需关注。（Amax 是最高地址引脚）。
5. 读取阵列的数据时，不需要解锁或命令周期。
6. 在器件处于 ASO 模式或者 DQ5 变为高电平（当器件正在提供状态数据时）的情况下，需要通过复位命令返回阵列数据读取模式。
7. 器件就绪读取阵列数据时，命令将有效。
8. 在解锁省略编程和解锁省略写入到缓冲区命令之前，需要使用解锁省略命令。
9. 当器件处于解锁省略模式时，需要执行解锁省略复位命令，才能返回读取阵列数据。
10. 在擦除挂起模式下，系统可以对非擦除扇区进行读取和编程/编程挂起的操作，或者从该模式进入 ID-CFI ASO 模式。擦除挂起命令仅在扇区擦除操作期间有效。
11. 擦除恢复/编程恢复命令仅在擦除挂起/编程挂起模式下有效。
12. 检测到器件处于写入到缓冲区终止状态后，发出该命令序列以返回到读取状态。请注意，如果进行复位以退出终止状态，则需使用完整的命令序列。
13. 退出命令会使器件返回到读取阵列状态。
14. 密码的各部分可以以任何顺序输入或读取，前提是要输入或读取整个 64 位密码。如果访问 SSR3，则地址为 20h-27h。
15. 对于 PWDx，每个 A0 命令只能编程密码的一部分。密码的各个部分必须按连续顺序进行编程（PWD0 - PWD7）。
16. 所有锁定寄存器位都只能编程一次。编程状态 = 0，擦除状态 = 1。此外，持久保护模式锁定位和密码保护模式锁定位不能同时编程，否则将终止锁定寄存器位编程操作，并使器件返回读取模式。保留给未来使用的锁定寄存器位未定义，可能是 0 或 1。
17. 如果发出任何进入命令，则必须发出退出命令，以使器件返回到读取状态。
18. 受保护状态 = 00h，不受保护状态 = 01h。DYB 设置、DYB 擦除或 PPB 编程命令的扇区地址可以是扇区内的任意位置 - 扇区地址的低位无需关注。
19. 请参见第 26 页上的 5.3.7.2 节，扇区擦除，了解多扇区擦除的信息。
20. 在 8x 模式，WC 表示 2 x8 WBL/PD 周期（例如，如果 WC = 0，那么第五个总线周期会将数据加载到低字节地址，A-1 为低和第六个总线周期会将数据加载到高字节地址，A-1 为高）。

6.2 器件 ID 和通用闪存接口 (ID-CFI) ASO 映射

ASO 的器件 ID 部分 (字位置 0h 到 0Fh) 提供器件的制造商 ID、器件 ID、扇区保护状态、以及基本功能集信息。

对位置 02h 的读取访问时间总是 t_{ACC} ，对此位置的读取操作需要 CE# 在读取之前变为高，然后返回低以启动读取 (异步读取访问)。不支持位置 02h 和其他 ID 位置之间的页模式读取。支持位置 02h 以外和 ID 位置之间的页模式读取。

在 x8 模式下，地址 A-1 被忽略，两个地址的相应数据的低 8 位被返回 (仅针对 CFI)。x8 模式下，只能读取 CFI 或 Autoselect 数据。x16 模式下，可从任意命令读取两个存储器。

欲了解更多信息，请参见第 28 页上的 *ID-CFI ASO*。

表 6.3 ID (自动选择) 地址映射

说明	地址 (x16)	地址 (x8)	读取数据
制造商 ID	(SA) + 0000h	(SA) + 0000h	0001h
器件 ID	(SA) + 0001h	(SA) + 0002h	227Eh
保护检验	(SA) + 0002h	(SA) + 0004h	扇区保护状态 (1= 扇区受保护, 0= 扇区不受保护)。要想读取其他 SA 保护状态, 只需要提供一个新的 SA 地址。
指示位	(SA) + 0003h	(SA) + 0006h	DQ15-DQ08 = 1 (保留) DQ7 - 工厂锁定的安全硅区域 1 = 锁定, 0 = 未锁定 DQ6 - 用户锁定的安全硅区域 1 = 锁定 0 = 未锁定 DQ5 = 1 (保留) DQ4 - WP# 保护 0 = 最低地址扇区 1 = 最高地址扇区 DQ3 - DQ0 = 1 (保留)
RFU	(SA) + 0004h	(SA) + 0008h	保留
	(SA) + 0005h	(SA) + 000h	保留
	(SA) + 0006h	(SA) + 000Ch	保留
	(SA) + 0007h	(SA) + 000Eh	保留
	(SA) + 0008h	(SA) + 0010h	保留
	(SA) + 0009h	(SA) + 0012h	保留
	(SA) + 000h	(SA) + 0014h	保留
	(SA) + 000h	(SA) + 0016h	保留
低软件位	(SA) + 000Ch	(SA) + 0018h	位 0 - 状态寄存器支持 1 = 支持状态寄存器 0 = 不支持状态寄存器 位 1 - DQ 轮询支持 1 = 支持 DQ 位轮询 0 = 不支持 DQ 位轮询 位 3、位 2 - 指令集支持 11 = 保留 10 = 保留 01 = 精简指令集 00 = 传统指令集 第 4-15 位 - 保留 = 0
高软件位	(SA) + 000Dh	(SA) + 001Ah	保留
器件 ID	(SA) + 000Eh	(SA) + 001Ch	2228h = 1 Gb 2223h = 512 Mb
器件 ID	(SA) + 000Fh	(SA) + 001Eh	2201h

表 6.4 CFI 查询识别字符串

字地址	字节地址	数据	说明
(SA) + 0010h (SA) + 0011h (SA) + 0012h	(SA) + 0020h (SA) + 0022h (SA) + 0024h	0051h 0052h 0059h	查询唯一 ASCII 字符串 “QRY”
(SA) + 0013h (SA) + 0014h	(SA) + 0026h (SA) + 0028h	0002h 0000h	主要 OEM 指令集
(SA) + 0015h (SA) + 0016h	(SA) + 002Ah (SA) + 002Ch	0040h 0000h	主扩展表地址
(SA) + 0017h (SA) + 0018h	(SA) + 002Eh (SA) + 0030h	0000h 0000h	备用 OEM 指令集 (00h = 不存在)
(SA) + 0019h (SA) + 001Ah	(SA) + 0032h (SA) + 0034h	0000h 0000h	备用 OEM 扩展表地址 (00h = 不存在)

表 6.5 CFI 系统接口字符串

字地址	字节地址	数据	说明
(SA) + 001Bh	(SA) + 0036h	0027h	V _{CC} 最小值 (擦除 / 编程) (D7-D4: V, D3-D0: 100 mV)
(SA) + 001Ch	(SA) + 0038h	0036h	V _{CC} 最大值 (擦除 / 编程) (D7-D4: V, D3-D0: 100 mV)
(SA) + 001Dh	(SA) + 003Ah	0000h	V _{PP} 最小电压 (00h = 不存在 V _{PP} 引脚)
(SA) + 001Eh	(SA) + 003Ch	0000h	V _{PP} 最大电压 (00h = 不存在 V _{PP} 引脚)
(SA) + 001Fh	(SA) + 003Eh	0008h	单字写入的典型超时时间为 2 ^N μs
(SA) + 0020h	(SA) + 0040h	0009h	最大多字节编程的 典型超时为 2 ^N μs (00h = 不支持)
(SA) + 0021h	(SA) + 0042h	000Ah	单个块擦除的典型超时时间为 2 ^N ms
(SA) + 0022h	(SA) + 0044h	0014h (1 Gb) 0013h (512 Mb)	整个芯片擦除的典型超时时间为 2 ^N ms (00h = 不支持)
(SA) + 0023h	(SA) + 0046h	0002h (85°C) 0003h (105°C)	单字编程的最大超时时间为典型值的 2 ^N 倍
(SA) + 0024h	(SA) + 0048h	0001h (85°C) 0002h (105°C)	缓冲区写入的最大超时时间为典型值的 2 ^N 倍
(SA) + 0025h	(SA) + 004Ah	0002h	单个块擦除的最大超时时间为典型值的 2 ^N 倍
(SA) + 0026h	(SA) + 004Ch	0002h	整个芯片擦除的最大超时时间为典型值的 2 ^N 倍 (00h = 不支持)

表 6.6 CFI 器件几何定义

字地址	字节地址	数据	说明
(SA) + 0027h	(SA) + 004Eh	001Bh (1 Gb) 001Ah (512 Mb)	器件容量 = 2 ^N 个字节;
(SA) + 0028h	(SA) + 0050h	0002h	闪存器件接口描述 0 = 仅为 x8, 1 = 仅为 x16, 2 = x8/x16 可选
(SA) + 0029h	(SA) + 0052h	0000h	
(SA) + 002Ah	(SA) + 0054h	0009h	多字节写入的最多字节数 = 2 ^N (00 = 不支持)
(SA) + 002Bh	(SA) + 0056h	0000h	
(SA) + 002Ch	(SA) + 0058h	0001h	器件内擦除块区域的数量 1 = 统一器件, 2 = 引导器件
(SA) + 002Dh	(SA) + 005Ah	00Xh	擦除块区域 1 信息 (参见 JEDEC JESD68-01 或 JEP137 规格) 00FFh, 0003h, 0000h, 0002h = 1 Gb 00FFh, 0001h, 0000h, 0002h = 512 Mb
(SA) + 002Eh	(SA) + 005Ch	000Xh	
(SA) + 002Fh	(SA) + 005Eh	0000h	
(SA) + 0030h	(SA) + 0060h	000Xh	
(SA) + 0031h	(SA) + 0062h	0000h	擦除块区域 2 信息 (参见 CFI 出版物 100)
(SA) + 0032h	(SA) + 0064h	0000h	
(SA) + 0033h	(SA) + 0066h	0000h	
(SA) + 0034h	(SA) + 0068h	0000h	
(SA) + 0035h	(SA) + 006Ah	0000h	擦除块区域 3 信息 (参见 CFI 出版物 100)
(SA) + 0036h	(SA) + 006Ch	0000h	
(SA) + 0037h	(SA) + 006Eh	0000h	
(SA) + 0038h	(SA) + 0070h	0000h	
(SA) + 0039h	(SA) + 0072h	0000h	擦除块区域 4 信息 (参见 CFI 出版物 100)
(SA) + 003Ah	(SA) + 0074h	0000h	
(SA) + 003Bh	(SA) + 0076h	0000h	
(SA) + 003Ch	(SA) + 0078h	0000h	
(SA) + 003Dh	(SA) + 007Ah	FFFFh	保留
(SA) + 003Eh	(SA) + 007Ch	FFFFh	
(SA) + 003Fh	(SA) + 007Eh	FFFFh	
(SA) + 003Fh	(SA) + 007Eh	FFFFh	

表 6.7 CFI 主要厂商特定的扩展查询

字地址	字节地址	数据	说明
(SA) + 0040h	(SA) + 0080h	0050h	查询唯一 ASCII 字符串 “PRI”
(SA) + 0041h	(SA) + 0082h	0052h	
(SA) + 0042h	(SA) + 0084h	0049h	
(SA) + 0043h	(SA) + 0086h	0031h	主版本号, ASCII
(SA) + 0044h	(SA) + 0088h	0035h	次版本号, ASCII
(SA) + 0045h	(SA) + 008Ah	0024h	地址敏感解锁 (位 1-0) 00b = 必需 01b = 非必需 工艺技术 (位 5-2) 0000b = 0.23 μm 浮门 0001b = 0.17 μm 浮门 0010b = 0.23 μm MirrorBit 0011b = 0.13 μm 浮门 0100b = 0.11 μm MirrorBit 0101b = 0.09 μm MirrorBit 0110b = 0.09 μm 浮门 0111b = 0.065 μm MirrorBit Eclipse 1000b = 0.065 μm MirrorBit 1001b = 0.045 μm MirrorBit
(SA) + 0046h	(SA) + 008Ch	0002h	擦除挂起 0 = 不支持 1 = 只读 2 = 读 / 写
(SA) + 0047h	(SA) + 008Eh	0001h	扇区保护 00 = 不支持 X = 最小组中的扇区数量
(SA) + 0048h	(SA) + 0090h	0000h	临时取消扇区保护 00 = 不支持 01 = 支持
(SA) + 0049h	(SA) + 0092h	0008h	扇区保护 / 不保护方案 04 = 高电压方法 05 = 软件指令锁定方法 08 = 高级扇区保护方法
(SA) + 004Ah	(SA) + 0094h	0000h	并发操作 00 = 不支持 X = Bank 数量
(SA) + 004Bh	(SA) + 0096h	0000h	突发模式类型 00 = 不支持 01 = 支持
(SA) + 004Ch	(SA) + 0098h	0003h	页面模式类型 00 = 不支持 01 = 4 字页面 02 = 8 字页面 03 = 16 字页面
(SA) + 004Dh	(SA) + 009Ah	00B5h	ACC (加速) 最小供电电压 00 = 不支持 D7-D4: V D3-D0: 100 mV
(SA) + 004Eh	(SA) + 009Ch	00C5h	ACC (加速) 最大供电电压 00 = 不支持 D7-D4: V D3-D0: 100 mV

表 6.7 CFI 主要厂商特定的扩展查询 (续)

字地址	字节地址	数据	说明
(SA) + 004Fh	(SA) + 009Eh	0004h (底部) 0005h (顶部)	WP# 保护 00h = 闪存器件无 WP 保护 (无引导) 01h = 顶部和底部八个 8 kB 扇区, 有 WP (双引导) 02h = 底部引导器件, 有 WP 保护 (底部引导) 03h = 顶部引导器件, 有 WP 保护 (顶部引导) 04h = 统一、底部 WP 保护 (统一底部引导) 05h = 统一、顶部 WP 保护 (统一顶部引导) 06h = 所有扇区都受 WP 保护 07h = 统一、顶部和底部 WP 保护
(SA) + 0050h	(SA) + 00A0h	0001h	编程挂起 00 = 不支持 01 = 支持
(SA) + 0051h	(SA) + 00A2h	0001h	解锁省略 00 = 不支持 01 = 支持
(SA) + 0052h	(SA) + 00A4h	0009h	安全硅扇区 (用户 OTP 区域) 的大小为 2^N (字节)
(SA) + 0053h	(SA) + 00A6h	008Fh	软件特性 位 0: 状态寄存器轮询 (1 = 支持, 0 = 不支持) 位 1: DQ 轮询 (1 = 支持, 0 = 不支持) 位 2: 新编程挂起 / 恢复指令 (1 = 支持, 0 = 不支持) 位 3: 字编程 (1 = 支持, 0 = 不支持) 位 4: 位段编程 (1 = 支持, 0 = 不支持) 位 5: 自动检测编程 (1 = 支持, 0 = 不支持) 位 6: RFU (留给将来使用) 位 7: 每行多次写入 (1 = 支持, 0 = 不支持)
(SA) + 0054h	(SA) + 00A8h	0005h	页面大小 = 2^N 个字节
(SA) + 0055h	(SA) + 00AAh	0006h	最长的擦除挂起超时时间 < 2^N (μ s)
(SA) + 0056h	(SA) + 00ACh	0006h	最长的编程挂起超时时间 < 2^N (μ s)
(SA) + 0057h 至 (SA) + 0077h	(SA) + 00AEh 至 (SA) + 00ACh	FFFFh	保留
(SA) + 0078h	(SA) + 00F0h	0006h	最长的嵌入式硬件复位超时时间 < 2^N (μ s) 通过复位引脚复位
(SA) + 0079h	(SA) + 00F2h	0009h	最长的非嵌入式硬件复位超时时间 < 2^N (μ s) 加电复位

硬件接口

7. 信号描述

7.1 地址和数据配置

地址和数据通过单独的信号输入和 I/O 并行连接（ADP）。

7.2 输入 / 输出简介

表 7.1 I/O 简介

符号	类型	说明
RESET#	输入	硬件复位。在 V_{IL} 时，器件将控制逻辑复位至其待机状态，做好读取阵列数据的准备。
CE#	输入	芯片使能。在 V_{IL} 时，选择要与主存储控制器传输数据的器件。
OE#	输入	输出使能。在 V_{IL} 时，输出被有效驱动。在 V_{IH} 时，输出状态为高阻抗。
WE#	输入	写入使能。在 V_{IL} 时，指明数据从主机传输到器件。在 V_{IH} 时，指明数据从器件传输到主机。
Amax-A0	输入	地址输入。 A25-A0（针对 S29GL01GT） A24-A0（针对 S29GL512T）
DQ14-DQ0	输入 / 输出	数据输入和输出
DQ15/A-1	输入 / 输出	DQ15: 数据输入和输出 A-1: 字节模式下的 LSB 地址输入
WP#/ACC	输入	写保护。在 V_{IL} 时，禁用对器件最低或最高地址 64 kword（128 kB）扇区执行的编程和擦除功能。在 V_{IH} 时，该扇区不受保护。在 V_{HH} 时，器件将自动进入解锁省略模式。WP# 有一个内部上拉电阻；断开连接时，WP# 处于 V_{IH} 状态。
RY/BY#	输出 — 开漏	准备就绪 / 忙碌。指明嵌入式算法是正在执行还是已完成。在 V_{IL} 时，器件正在执行嵌入式算法，如擦除或编程。在高阻态下，器件正准备执行读取操作或正在写入一个新指令。它需要外部上拉电阻来检测高阻态。多个器件可以将它们的 RY/BY# 输出绑定在一起以检测何时所有器件均准备就绪。
BYTE#	输入	选择数据总线宽度。在 V_{IL} 时，器件正在进行字节配置，数据 I/O 引脚 DQ0-DQ7 均有效，DQ15/A-1 作为 LSB 地址输入。在 V_{IH} 时，器件正在进行字配置，数据 I/O 引脚 DQ0-DQ15 均有效。
V_{CC}	电源	内核电源
V_{IO}	电源	多用途 I/O 电源。
V_{SS}	电源	电源接地
NC	无连接	内部无连接。引脚 / 球形焊盘位置在印刷电路板（PCB）中可以用作为路线通道的一部分。
RFU	无连接	保留未来使用。内部未连接，但考虑到未来的兼容性，引脚 / 球形焊盘位置不应连接和被 PCB 路线通道使用。引脚 / 球形焊盘在未来可能供信号使用。
DNU	保留	请勿使用。保留给赛普拉斯使用。引脚 / 球形焊盘内部连接。该输入有一个连接到 V_{SS} 的内部下拉阻抗。引脚 / 球形焊盘可以保持为开路或连接到 PCB 上的 V_{SS} 。

7.3 字 / 字节配置

BYTE# 引脚控制器件数据 I/O 引脚是以字节配置还是字配置运行。如果 BYTE# 引脚设在逻辑“1”，则器件采用字配置，DQ0-DQ15 有效并受 CE# 和 OE# 控制。

如果 BYTE# 引脚设在逻辑“0”，则器件采用字节配置，仅数据 I/O 引脚 DQ0-DQ7 有效并受 CE# 和 OE# 控制。数据 I/O 引脚 DQ8-DQ14 处于三态，DQ15 引脚用作 LSB (A-1) 地址功能的输入。

仅在器件处于待机（读取）模式下，BYTE# 引脚才能切换。

BYTE# 引脚具有一个内部上拉电阻。虽然 x16 系统没有要求，但建议将该引脚与高电压（如 V_{IO} ）相连。

7.4 多用途 I/O 功能

器件可驱动的最高输出电压和可接受的输入电平由 V_{IO} 电源决定。此电源允许器件驱动和接收去往和来自同一总线上接口信号电平与器件内核电压不同的其他器件的信号。

7.5 就绪 / 忙碌 # (RY/BY#)

RY/BY# 是专用的漏极开路输出引脚，指明嵌入式算法、上电复位（POR）或硬件复位是正在执行中还是已完成。RY/BY# 状态在命令序列中最后 WE# 脉冲上升沿之后、当 V_{CC} 在 POR 期间高于 V_{CC} 最小值时、或者在 RESET# 下降沿之后有效。由于 RY/BY# 是漏极开路输出，多个 RY/BY# 引脚可通过上拉电阻并行连接到 V_{IO} 。

如果输出是低（忙碌），器件正在执行擦除、编程或复位。（这包括擦除挂起模式下的编程）。如果输出为高（就绪），则器件已准备好读取数据（包括在擦除挂起模式期间），或者处于待机模式。

第 36 页上的表 5.6，*数据轮询状态* 显示了 RY/BY# 在每一个操作中的输出。

如果嵌入式算法失败（编程 / 擦除由于最大脉冲或编程终止而失败），则 RY/BY# 将为低电平，直到状态寄存器位 4 和 5 被清除并且复位命令被发出为止。如果嵌入式算法失败（扇区被锁定），则 RY/BY# 将转为高电平（就绪）状态。这包括对锁定的扇区执行的擦除或编程。

7.6 硬件复位

RESET# 输入提供一种硬件复位方法，可使器件返回待机状态。当 RESET# 保持低至少 t_{RP} 时间时，器件立即：

- 终止任何正在执行的操作，
- 退出所有 ASO，
- 使所有输出三态化，
- 复位状态寄存器，
- 将 EAC 复位到待机状态。
- CE# 在复位操作期间 (t_{RPH}) 被忽略。
- 为符合复位电流规范 (I_{CC5})，CE# 必须保持高电平状态。

为确保数据完整性，一旦器件准备好接受另一个命令序列，应重新启动被中断的操作。

8. 信号协议

以下部分介绍 29GL-T 系列闪存器件的主机系统接口信号工作方式和时序。

8.1 接口状态

表 8.1 描述了每个接口状态所需的接口信号值。

表 8.1 接口状态

接口状态	V _{CC}	V _{IO}	RESET#	CE#	OE#	WE#	BYTE# (6)	WP#/ACC	Amax-A0 (1)	DQ0-DQ7	DQ8-DQ15	
											BYTE# = V _{IH}	BYTE# = V _{IL}
关闭电源 (硬件数据保护)	< V _{LKO}	≤ V _{CC}	X	X	X	X	L 或 H	X	X	高阻态	高阻态	高阻态
加电 (冷) 复位	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	X	X	X	X	L 或 H	X	X	高阻态	高阻态	高阻态
硬件 (暖) 复位	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	L	X	X	X	L 或 H	X	X	高阻态	高阻态	高阻态
接口待机	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	H	H	X	X	L 或 H	H	X	高阻态	高阻态	高阻态
自动睡眠 (2、4)	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	H	L	X	X	L 或 H	H	有效	输出可用	输出可用	DQ8-DQ14 = 高阻态, DQ15 = A-1
读取 (输出禁用) (3)	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	H	L	H	H	L 或 H	X	有效	高阻态	高阻态	高阻态
随机读取	≥ V _{CC} 最小值	≥ V _{IO} 最小值	H	L	L	H		X	有效	输出有效的 时间	输出有效的 时间	DQ8-DQ14 = 高阻态, DQ15 = A-1
页读取操作	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	H	L	L	H	L 或 H	X	Amax-A4 有效 A3-A0 (或 A3-A-1) 被修改	输出有效的 时间	输出有效的 时间	DQ8-DQ14 = 高阻态, DQ15 = A-1
写入	≥ V _{CC} 最小值	≥ V _{IO} 最小值 ≤ V _{CC}	H	L	H	L	L 或 H	(5)	有效	输入有效	输入有效	DQ8-DQ14 = 高阻态, DQ15 = A-1

图标:

L = V_{IL}

H = V_{IH}

X = V_{IL} 或 V_{IH}

L/H = 上升沿

H/L = 下降沿

有效 = 所有总线信号都有稳定的低或高电平

修改 = 有效状态不同于之前的有效状态

可用 = 读取数据在内部存储, 输出驱动由 OE# 控制

注意:

1. 地址分别是 Amax:A0 (字模式); Amax:A-1 (字节模式)。

2. WE# 和 OE# 不能同时等于 V_{IL}。

3. 读取 (输出禁用) 是由 OE# 为高电平时启动的读取。

4. 自动睡眠是一个读/写操作: 发生该操作时, 数据在一段扩展时间内输入到总线上, CS# 不转为高电平, 并且器件的内部逻辑进入待机模式, 从而节约功耗。

5. 如果 WP# = V_{IL}, 最外面的扇区保持受保护。如果 WP# = V_{IH}, 最外面的扇区不受保护。WP# 有一个内部上拉电阻; 当断开连接时, WP# 将等于 V_{IH}。

6. V_{IL} = V_{SS} 和 V_{IH} = V_{IO}。

8.2 关闭电源（硬件数据保护）

当内核供电电压（ V_{CC} ）下降到锁定电压（ V_{LKO} ）以下时，存储器被视为断电。当 V_{CC} 低于 V_{LKO} 时，整个存储器阵列受保护，不能进行写入或擦除操作。这样可防止在电源转换期间存储内容发生篡改。在电源切换到关闭期间， V_{IO} 应保持不大于 V_{CC} 。

如果 V_{CC} 降到 V_{RST} （最小值）以下，然后又升高，超过 V_{RST} （最小值）并达到 V_{CC} （最小值），器件将进入上电复位接口状态，且 EAC 开始冷复位嵌入式算法。

8.3 节能模式

8.3.1 接口待机

待机是在主机没有选择器件传输数据时（ $CE\# =$ 高）的一种默认、低功耗接口状态。在此状态下，所有输入均被忽略，所有输出（ $RY/BY\#$ 除外）均为高阻抗。 $RY/BY\#$ 是 EAC 的直接输出，不受主机接口的控制。

8.3.2 自动睡眠

在经过随机读取访问时间后，自动睡眠模式将器件接口能耗降低到睡眠级别（ I_{CC6} ）。地址在 $t_{ACC} + 30\text{ ns}$ 期间保持稳定时，器件将自动进入该模式。在自动睡眠模式下，输出数据处于锁存状态，系统可随时使用。数据输出取决于 $OE\#$ 信号的电平，但自动睡眠模式的电流与 $OE\#$ 信号电平无关。当地址发生变化时，标准地址访问时序（ t_{ACC} 或 t_{PACC} ）将提供新数据。有关自动睡眠模式的电流规范 I_{CC6} ，请参考第 64 页上的直流特性。

自动睡眠有助于降低电流消耗，特别是当主机系统时钟减慢以降低功耗时。与系统高速运行时相比，在系统时钟减慢期间，读取和写入循环时间可能增加数倍。即使在延长的数据传输周期内 $CE\#$ 可能为低，存储器件主机接口也会在 $t_{ACC} + 30\text{ ns}$ 后切换到自动睡眠电流。器件将保持自动睡眠电流 t_{ASSB} 时间。随后器件将切换到待机电流级别。这使得存储器在较长数据传输期间的大部分时间都能保持自动睡眠或待机功率级别，避免了在主机系统选择了存储器件的所有时间都消耗全部读取功率。

然而，EAC 的运行与主机接口的自动睡眠模式无关，它会在嵌入式算法执行期间继续消耗电流。只有当主机接口和 EAC 都处于其待机状态时，电流才能达到待机级别。

8.4 读取

8.4.1 读取（输出禁用）

当 $CE\#$ 为低电平时，主机系统存储控制器开始读取或写入数据传输。在数据传输开始时经常有一段时间 $CE\#$ 为低、地址为有效、 $OE\#$ 为高、 $WE\#$ 为高。在此状态下，会假定是一个读取访问并启动随机读取进程，同时数据输出保持高阻抗。如果 $OE\#$ 信号变为低，接口将转换到随机读取状态，数据输出被驱动。如果 $WE\#$ 信号保持低，接口将转换到写入状态。注意， $OE\#$ 和 $WE\#$ 不应同时为低，以确保主机系统和存储器之间没有数据总线冲突。

8.4.2 随机（异步）读取

当主机系统接口通过驱动 $CE\#$ 低来选择存储器件时，器件接口将退出待机状态。如果 $CE\#$ 变为低时 $WE\#$ 为高，会启动随机读取访问。数据输出与地址映射模式以及在读取访问启动时提供的地址有关。

当 $CE\#$ 为低、 $OE\#$ 为低、 $WE\#$ 保持高、地址保持稳定、且满足异步访问时间时，数据出现在 $DQ15-DQ0$ （x8 模式下为 $DQ7-DQ0$ ）上。地址访问时间（ t_{ACC} ）等于从稳定地址到有效输出数据的延迟。芯片使能访问时间（ t_{CE} ）是从稳定 $CE\#$ 到有效输出数据的延迟。为了将读取数据驱动到数据输出， $OE\#$ 信号必须在有效数据可用之前保持低至少输出使能时间（ t_{OE} ）。

$CE\#$ 有效（ t_{CE} ）、地址稳定（ t_{ACC} ）或 $OE\#$ 有效（ t_{OE} ）后开始的随机访问时间结束时，无论哪个先发生，数据输出都提供来自当前有效的地址映射模式的读取数据。如果 $CE\#$ 保持低并且 A_{max} 到 $A4$ 的任何地址信号变为新值，会开始新的随机读取访问。如果 $CE\#$ 保持低并且 $OE\#$ 变为高，接口转换到读取（输出禁用）状态。如果 $CE\#$ 保持低、 $OE\#$ 变为高并且 $WE\#$ 变为低，接口转换到写入状态。如果 $CE\#$ 返回高，接口进入待机状态。相邻访问（在访问之间 $CE\#$ 保持低）要求更改地址以启动第二个访问操作。

请参见第 71 页上的异步读取操作。

8.4.3 页读取操作

随机读取访问结束后，如果 $CE\#$ 保持低、 $OE\#$ 保持低、 A_{max} 到 $A4$ 地址信号保持稳定、并且任意 $A3$ 到 $A0$ 地址信号变化，会在同一页面内开始新访问。在 x8 模式下，每当任意 $A3$ 到 $A-1$ 地址信号变化时，都会在同一页面内开始进行新的访问。与随机读取访问相比，页面读取完成得更快（ t_{PACC} ）。

8.5 写入

8.5.1 异步写入

如果 CE 为低后 WE# 变为低，会从一个读取状态转换到写入状态。如果在 CE# 变为低之前 WE# 为低，会从待机状态直接转换到写入状态，而不开始读取访问。

当 CE# 为低、OE# 为高、并且 WE# 变为低时，开始写入数据传输。注意，OE# 和 WE# 不应同时为低，以确保主机系统和存储器之间没有数据总线冲突。当异步写入循环时序要求得到满足时，WE# 可以变为高以将地址和数据值捕获到 EAC 命令存储器中。

地址在 WE# 或 CE#（两者中后到来的一个）下降沿上捕获。数据在 WE# 或 CE#（两者中先到来的一个）上升沿上捕获。

如果 CE# 在 WE# 变为低之前为低，并且在 WE# 变为高之后保持低，则访问称为 WE# 控制的写入。当 WE# 为高且 CE# 变为高时，将转换到待机状态。如果 CE# 保持低并且 WE# 变为高，将转换到读取（输出禁用）状态。

如果 WE# 在 CE# 变为低之前为低，并且在 CE# 变为高之后保持低，则访问称为 CE# 控制的写入。CE# 控制的写入转换到待机状态。

如果 WE# 在 CE# 变为低之前为低，则 CE# 变为低时将启动写入传输。如果 WE# 在 CE# 变为高之后为低，则在 CE# 的上升沿上捕获地址和数据。这些情况被称为 CE# 控制的写入状态转换。

由读取访问随后的写入操作（在访问之间 CE# 保持低）要求更改地址以启动后面的读取访问。

相邻访问（在访问之间 CE# 保持低）要求更改地址以启动第二个访问操作。

EAC 命令存储器阵列没有 ASO，也不能被主机系统读取。EAC 检查每个写入传输中的地址和数据，以确定写入是否是合法命令序列的一部分。如果合法命令序列是完整的，EAC 将启动相应的 EA。

8.5.2 写入脉冲“假信号”保护

WE# 上短于 5 ns（典型值）的噪声脉冲不会启动写入循环。

8.5.3 逻辑禁止

通过将 OE# 设为 V_{IL} ，或 CE# 为 V_{IH} ，或 WE# 为 V_{IH} ，可禁止写入周期。要启动写入周期，CE# 和 WE# 必须为低电平（ V_{IL} ），并且 OE# 为高电平（ V_{IH} ）。

9. 电气规范

9.1 最大绝对额定值

表 9.1 最大绝对额定值

存储温度塑料包装	-65°C ~ +150°C
通电状态下的环境温度	-65°C ~ +125°C
对地电压	
所有引脚 (RESET# 除外) (1)	-0.5 V ~ (V _{IO} + 0.5 V)
RESET# (1)	-0.5 V ~ (V _{CC} + 0.5 V)
输出短路电流 (2)	100 mA
V _{CC}	-0.5 V ~ +4.0 V
V _{IO}	-0.5 V ~ +4.0 V
ACC	-0.5 V ~ +12.5 V

注意:

1. 输入或 I/O 引脚上的最小直流电压是 -0.5 V。在电压转换期间, 输入或 I/O 引脚可能低于 V_{SS}, 降到 -2.0 V, 时间最长为 20 ns。请参见第 63 页上的图 9.3。输入或 I/O 引脚上的最大直流电压是 V_{CC}+0.5 V。在电压转换期间, 输入或 I/O 引脚可能升到 V_{CC}+2.0 V, 时间最长为 20 ns。请参见第 63 页上的图 9.4。
2. 每一次只能有一个输出对地短接。短接时间不能超过一秒。
3. 如果使用大于**最大绝对额定值**中所列出的数值, 可能造成永久性损害。这只是压力额定值; 并不暗示器件在这些值或者在此数据手册操作部分所示值之上的任何其他情形下能正常运行。如果让器件长时间在绝对最大额定值情况下运行, 会影响器件的可靠性。

9.2 闩锁特性

此产品符合 JEDEC 标准 JESD78C 闩锁测试要求。

9.3 工作范围

9.3.1 温度范围

参数	符号	条件	规范		单位
			最小值	最大值	
环境温度	T _A	工业级 (I) 器件	-40	+85	°C
		扩展的工业级 (V) 器件	-40	+105	°C
		扩展级 (N) 器件	-40	+125	°C

9.3.2 供电电压

V _{CC}	2.7 V 至 3.6 V
V _{IO}	1.65 V 至 V _{CC} + 200 mV

注意:

1. 运行范围定义了一些限值, 在这些限值之间可保证器件正常运行。

9.3.3 上电和断电

上电或断电期间, V_{CC} 必须不小于 V_{IO} (V_{CC} ≥ V_{IO})。在 V_{CC} 和 V_{IO} 增加到 V_{CC} 和 V_{IO} 最小阈值以上, 并一直保持该状态后的 t_{VCS} 延迟时间内, 器件会忽略所有输入。在 t_{VCS} 期间, 器件会执行上电复位操作。

在断电期间或当电压降到 V_{CC} 锁定最大值 (V_{LKO}) 以下, V_{CC} 和 V_{IO} 电压必须保证在 t_{PD} 的时间内低于 V_{CC} 复位 (V_{RST}) 最低值, 以确保当 V_{CC} 和 V_{IO} 重新返回到其运行范围时, 器件正确进行初始化。请参见第 62 页上的图 9.2。如果在电压下降过程中, V_{CC} 保持为高于 V_{LKO} 最大值的状态, 那么当 V_{CC} 再次超过 V_{CC} 最小值时, 器件将被初始化, 并正常运行。如果器件因不正确的初始化而被锁定, 可以使用硬件复位来正确初始化器件。

为了确保稳定的 V_{CC} 和 V_{IO} 电源, 必须采取正常的预防措施进行电源去耦。系统中每个器件的 V_{CC} 和 V_{IO} 电源应使用一个大小合适的电容器进行去耦 (此电容器需与封装接近, 且通常为 0.1 μF)。V_{IO} 始终不能比 V_{CC} 大 200 mV (V_{CC} ≥ V_{IO} - 200 mV)。

表 9.2 打开电源 / 关闭电源电压和时序

符号	参数	最小值	最大值	单位
V_{CC}	V_{CC} 电源	2.7	3.6	V
V_{LKO}	如果 V_{CC} 电平低于该电平, 则需要重新执行初始化操作 (1)		2.5	V
V_{RST}	要求 V_{CC} 和 V_{IO} 为低电压, 以保证能够进行初始化 (1)	1.0		V
t_{VCS}	从 V_{CC} 和 $V_{IO} \geq$ 最小值到开始第一次访问的时间 (1)	300		μs
t_{PD}	$V_{CC} \leq V_{RST}$ (最小值) 的时间 (1)	15		μs

注意:

- 并非 100% 经过了测试。

图 9.1 上电

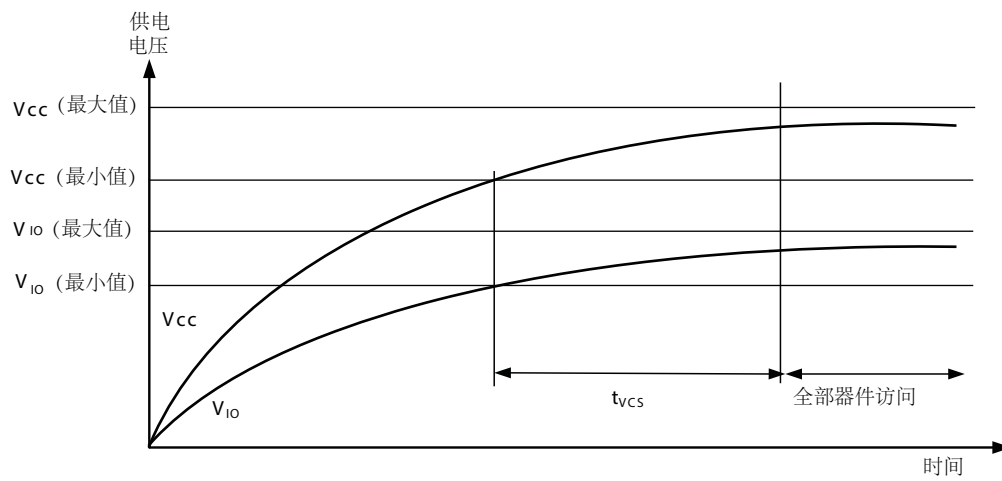
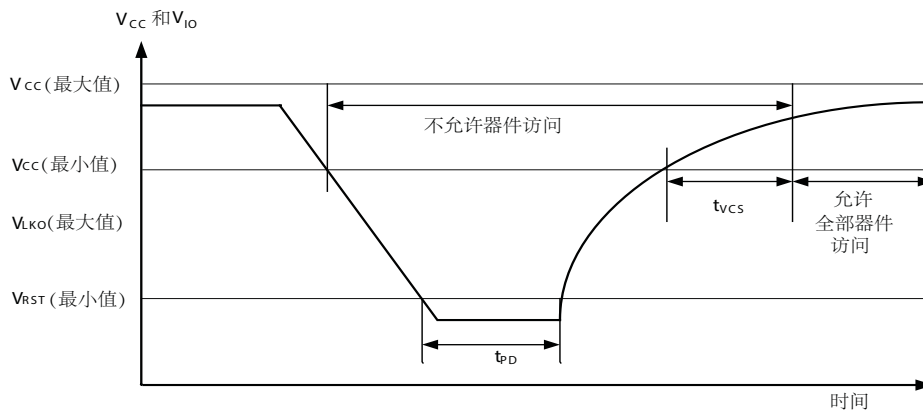


图 9.2 断电或电压下降



9.3.4 输入信号过冲

图 9.3 最大负过冲波形

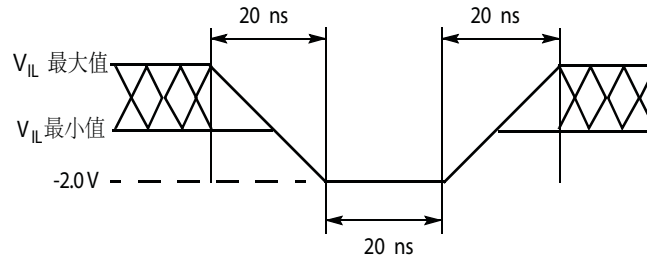
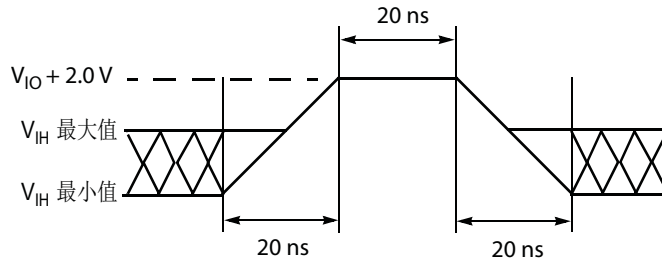


图 9.4 最大正过冲波形



9.4 直流特性

表 9.3 直流特性（温度为 -40°C ~ +85°C）

参数	说明	测试条件	最小值	典型值 (2)	最大值	单位
I_{LI}	输入负载电流	$V_{IN} = V_{SS} - V_{CC}$, $V_{CC} = V_{CC}$ 最大值 所有其他信号 (WP#, BYTE# 除外)		± 0.02	± 1.0	μA
			WP#, BYTE#		± 0.5	
I_{LO}	输出漏电流	$V_{OUT} = V_{SS} - V_{CC}$, $V_{CC} = V_{CC}$ 最大值		± 0.02	± 1.0	μA
I_{CC1}	V_{CC} 有效读取电流	CE# = V_{IL} , OE# = V_{IH} , f = 5 MHz 时地址切换, $V_{CC} = V_{CC}$ 最大值		55	60	mA
I_{CC2}	相对于 V_{CC} 的页面内读取电流	CE# = V_{IL} , OE# = V_{IH} , f = 33 MHz 时地址切换, $V_{CC} = V_{CC}$ 最大值		9	25	mA
I_{CC3}	相对于 V_{CC} 的有效擦除 / 编程电流 (1)、(2)	CE# = V_{IL} , OE# = V_{IH} , $V_{CC} = V_{CC}$ 最大值		45	100	mA
I_{CC4}	相对于 V_{CC} 的待机电流	CE#, RESET#, OE# = V_{IH} , $V_{IH} = V_{IO}$ $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值		70	100	μA
I_{CC5}	相对于 V_{CC} 的复位电流 (2)、(7)	CE# = V_{IH} , RESET# = V_{IL} , $V_{CC} = V_{CC}$ 最大值		10	20	mA
I_{CC6}	自动睡眠模式 (3)	$V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值, $t_{ACC} + 30$ ns		3	6	mA
		$V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值, t_{ASSB}		100	150	μA
I_{CC7}	上电期间相对于 V_{CC} 的电流 (2)、(6)	RESET# = V_{IO} , CE# = V_{IO} , OE# = V_{IO} , $V_{CC} = V_{CC}$ 最大值		53	80	mA
V_{IL}	输入低电平电压 (4)		-0.5		$0.3 \times V_{IO}$	V
V_{IH}	输入高电平电压 (4)		$0.7 \times V_{IO}$		$V_{IO} + 0.4$	V
V_{HH}	ACC 加速编程电压	$V_{CC} = 2.7 - 3.6$ V	11.5		12.5	V
V_{OL}	输出低电平电压 (4)、(8)	$I_{OL} = 100 \mu A$ (DQ15-DQ0) ; $I_{OL} = 2$ mA (RY/BY#)			$0.15 \times V_{IO}$	V
V_{OH}	输出高电平电压 (4)	$I_{OH} = 100 \mu A$	$0.85 \times V_{IO}$			V
V_{LKO}	低电平 V_{CC} 锁定电压 (2)		2.25		2.5	V
V_{RST}	低电平 V_{CC} 上电复位电压 (2)			1.0		V

注意:

1. 在执行嵌入式算法期间, I_{CC} 有效。
2. 并非 100% 经过了测试。
3. 地址在指定的时间内保持稳定时, 通过自动睡眠模式可实现低功耗模式。
4. $V_{IO} = 1.65$ V 到 V_{CC} 还是 2.7 V 到 V_{CC} , 取决于不同的型号。
5. $V_{CC} = 3$ V 和 $V_{IO} = 3$ V 或 1.8 V。 $V_{IO} = 1.8$ V 时, I/O 引脚不可在高于 1.8 V 的电压下运行。
6. 在上电期间, 有电流波峰需求, 要求系统能够提供该电流, 以保证器件正确初始化。
7. 如果复位开始时嵌入式操作正在执行中, 则电流消耗将保持在嵌入式操作规范, 直到嵌入式操作被复位停止。如果复位开始时没有执行任何嵌入式操作, 或者在嵌入式操作停止之后, 在 t_{RPH} 剩余期间消耗的电流为 I_{CC5} 。在 t_{RPH} 结束时, 器件会进入待机模式, 直到进行下一个读取或写入操作为止。
8. 推荐的 RY/BY# 输出上拉电阻范围为 5 K Ω -10 K Ω 。

表 9.4 直流特性 (温度为 -40°C ~ +105°C)

参数	说明	测试条件	最小值	典型值 (2)	最大值	单位
I_{LI}	输入负载电流	$V_{IN} = V_{SS} \sim V_{CC}$, $V_{CC} = V_{CC}$ 最大值	所有其他信号 (WP#、 BYTE# 除外)	± 0.02	± 1.0	μA
			WP#、 BYTE#	± 0.5	± 2.0	
I_{LO}	输出漏电流	$V_{OUT} = V_{SS} \sim V_{CC}$, $V_{CC} = V_{CC}$ 最大值		± 0.02	± 1.0	μA
I_{CC1}	V_{CC} 有效读取电流	$CE\# = V_{IL}$, $OE\# = V_{IH}$, $f = 5 \text{ MHz}$ 时地址切换, $V_{CC} = V_{CC}$ 最大值		55	60	mA
I_{CC2}	相对于 V_{CC} 的页面内读取电流	$CE\# = V_{IL}$, $OE\# = V_{IH}$, $f = 33 \text{ MHz}$ 时地址切 换, $V_{CC} = V_{CC}$ 最大值		9	25	mA
I_{CC3}	相对于 V_{CC} 的有效擦除 / 编程电流 (1)、(2)	$CE\# = V_{IL}$, $OE\# = V_{IH}$, $V_{CC} = V_{CC}$ 最大值		45	100	mA
I_{CC4}	相对于 V_{CC} 的待机电流	$CE\#$, $RESET\#$, $OE\# = V_{IH}$, $V_{IH} = V_{IO}$ $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值		70	200	μA
I_{CC5}	相对于 V_{CC} 的复位电流 (2)、(7)	$CE\# = V_{IH}$, $RESET\# = V_{IL}$, $V_{CC} = V_{CC}$ 最大值		10	20	mA
I_{CC6}	自动睡眠模式 (3)	$V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值, $t_{ACC} + 30 \text{ ns}$		3	6	mA
			$V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ 最大值, t_{ASSB}		100	200
I_{CC7}	上电期间相对于 V_{CC} 的电流 (2)、(6)	$RESET\# = V_{IO}$, $CE\# = V_{IO}$, $OE\# = V_{IO}$, $V_{CC} = V_{CC}$ 最大值		53	80	mA
V_{IL}	输入低电平电压 (4)		-0.5		$0.3 \times V_{IO}$	V
V_{IH}	输入高电平电压 (4)		$0.7 \times V_{IO}$		$V_{IO} + 0.4$	V
V_{HH}	ACC 加速编程电压	$V_{CC} = 2.7 - 3.6 \text{ V}$	11.5		12.5	V
V_{OL}	输出低电平电压 (4)、(8)	$I_{OL} = 100 \mu A$ (DQ15-DQ0); $I_{OL} = 2 \text{ mA}$ (RY/BY#)			$0.15 \times V_{IO}$	V
V_{OH}	输出高电平电压 (4)	$I_{OH} = 100 \mu A$	$0.85 \times V_{IO}$			V
V_{LKO}	低电平 V_{CC} 锁定电压 (2)		2.25		2.5	V
V_{RST}	低电平 V_{CC} 上电复位电压 (2)			1.0		V

注意:

- 在执行嵌入式算法期间, I_{CC} 有效。
- 并非 100% 经过了测试。
- 地址在指定的时间内保持稳定时, 通过自动睡眠模式可实现低功耗模式。
- $V_{IO} = 1.65 \text{ V}$ 到 V_{CC} 还是 2.7 V 到 V_{CC} , 取决于不同的型号。
- $V_{CC} = 3 \text{ V}$ 和 $V_{IO} = 3 \text{ V}$ 或 1.8 V 。 $V_{IO} = 1.8 \text{ V}$ 时, I/O 引脚不可在高于 1.8 V 的电压下运行。
- 在上电期间, 有电流波峰需求, 要求系统能够提供该电流, 以保证器件正确初始化。
- 如果复位开始时嵌入式操作正在执行中, 则电流消耗将保持在嵌入式操作规范, 直到嵌入式操作被复位停止。如果复位开始时没有执行任何嵌入式操作, 或者在嵌入式操作停止之后, 在 t_{RPH} 剩余期间消耗的电流为 I_{CC7} 。在 t_{RPH} 结束时, 器件会进入待机模式, 直到进行下一个读取或写入操作为止。
- 推荐的 RY/BY# 输出上拉电阻范围为 $5 \text{ K}\Omega$ - $10 \text{ K}\Omega$ 。

表 9.5 直流电特性 (温度为 -40°C ~ +125°C)

参数	说明	测试条件	最小值	典型值 (2)	最大值	单位
I _{LI}	输入负载电流	V _{IN} = V _{SS} ~ V _{CC} , V _{CC} = V _{CC} 最大值	所有其他信号 (WP#、 BYTE# 除外)	±0.02	±1.0	μA
			WP#、 BYTE#	±0.5	±2.0	
I _{LO}	输出漏电流	V _{OUT} = V _{SS} ~ V _{CC} , V _{CC} = V _{CC} 最大值		±0.02	±1.0	μA
I _{CC1}	V _{CC} 有效读取电流	CE# = V _{IL} , OE# = V _{IH} , f = 5 MHz 时地址切换, V _{CC} = V _{CC} 最大值		55	60	mA
I _{CC2}	相对于 V _{CC} 的页面内读取电流	CE# = V _{IL} , OE# = V _{IH} , f = 33 MHz 时地址切换, V _{CC} = V _{CC} 最大值		9	25	mA
I _{CC3}	相对于 V _{CC} 的有效擦除 / 编程电流 (1)、(2)	CE# = V _{IL} , OE# = V _{IH} , V _{CC} = V _{CC} 最大值		45	100	mA
I _{CC4}	相对于 V _{CC} 的待机电流	CE#, RESET#, OE# = V _{IH} , V _{IH} = V _{IO} V _{IL} = V _{SS} , V _{CC} = V _{CC} 最大值		70	215	μA
I _{CC5}	相对于 V _{CC} 的复位电流 (2)、(7)	CE# = V _{IH} , RESET# = V _{IL} , V _{CC} = V _{CC} 最大值		10	20	mA
I _{CC6}	自动睡眠模式 (3)	V _{IH} = V _{IO} , V _{IL} = V _{SS} , V _{CC} = V _{CC} 最大值, t _{ACC} + 30 ns		3	6	mA
		V _{IH} = V _{IO} , V _{IL} = V _{SS} , V _{CC} = V _{CC} 最大值, t _{ASSB}		100	215	μA
I _{CC7}	上电期间相对于 V _{CC} 的电流 (2)、(6)	RESET# = V _{IO} , CE# = V _{IO} , OE# = V _{IO} , V _{CC} = V _{CC} 最大值		53	80	mA
V _{IL}	输入低电平电压 (4)				0.3 x V _{IO}	V
V _{IH}	输入高电平电压 (4)				V _{IO} + 0.4	V
V _{HH}	ACC 加速编程电压	V _{CC} = 2.7 - 3.6 V			12.5	V
V _{OL}	输出低电平电压 (4)、(8)	I _{OL} = 100 μA (DQ15-DQ0); I _{OL} = 2 mA (RY/BY#)			0.15 x V _{IO}	V
V _{OH}	输出高电平电压 (4)	I _{OH} = 100 μA				V
V _{LKO}	低电平 V _{CC} 锁定电压 (2)				2.5	V
V _{RST}	低电平 V _{CC} 上电复位电压 (2)			1.0		V

注意:

1. 在执行嵌入式算法期间, I_{CC} 有效。
2. 并非 100% 经过了测试。
3. 地址在指定的时间内保持稳定时, 通过自动睡眠模式可实现低功耗模式。
4. V_{IO} = 1.65 V 到 V_{CC} 还是 2.7 V 到 V_{CC}, 取决于不同的型号。
5. V_{CC} = 3 V 和 V_{IO} = 3 V 或 1.8 V。V_{IO} = 1.8 V 时, I/O 引脚不可在高于 1.8 V 的电压下运行。
6. 在上电期间, 有电流波峰需求, 要求系统能够提供该电流, 以保证器件正确初始化。
7. 如果复位开始时嵌入式操作正在执行中, 则电流消耗将保持在嵌入式操作规范, 直到嵌入式操作被复位停止。如果复位开始时没有执行任何嵌入式操作, 或者在嵌入式操作停止之后, 在 t_{RPH} 剩余期间消耗的电流为 I_{CC7}。在 t_{RPH} 结束时, 器件会进入待机模式, 直到进行下一个读取或写入操作为止。
8. 推荐的 RY/BY# 输出上拉电阻范围为 5 KΩ-10 KΩ。

9.5 电容特性

表 9.6 FBGA (LAA) 封装中的连接器电容

参数符号	参数说明	测试设置	典型值	最大值	单位
C_{IN}	输入电容	$V_{IN} = 0$	4	5.5	pF
C_{OUT}	输出电容	$V_{OUT} = 0$	3.5	5	pF
C_{IN2}	控制引脚电容	$V_{IN} = 0$	4	8	pF
RY/BY#	输出电容	$V_{OUT} = 0$	3	4	pF
RESET#	复位输入电容	$V_{IN} = 0$	21	23	pF

注意:

1. 采样值, 并非 100% 经过了测试。
2. 测试条件: $T_A = 25^{\circ}\text{C}$, $f = 1.0\text{ MHz}$ 。

表 9.7 FBGA (LAE) 封装的连接器电容

参数符号	参数说明	测试设置	典型值	最大值	单位
C_{IN}	输入电容	$V_{IN} = 0$	3.5	5	pF
C_{OUT}	输出电容	$V_{OUT} = 0$	3.5	5	pF
C_{IN2}	控制引脚电容	$V_{IN} = 0$	3.5	7	pF
RY/BY#	输出电容	$V_{OUT} = 0$	2.5	3.5	pF
RESET#	复位输入电容	$V_{IN} = 0$	20	22	pF

注意:

1. 采样值, 并非 100% 经过了测试。
2. 测试条件: $T_A = 25^{\circ}\text{C}$, $f = 1.0\text{ MHz}$ 。

表 9.8 FBGA (VBU) 封装的连接器电容

参数符号	参数说明	测试设置	典型值	最大值	单位
C_{IN}	输入电容	$V_{IN} = 0$	3.5	5	pF
C_{OUT}	输出电容	$V_{OUT} = 0$	3.5	5	pF
C_{IN2}	控制引脚电容	$V_{IN} = 0$	3.5	7	pF
RY/BY#	输出电容	$V_{OUT} = 0$	3	4	pF
RESET#	复位输入电容	$V_{IN} = 0$	20	22	pF

注意:

1. 采样值, 并非 100% 经过了测试。
2. 测试条件: $T_A = 25^{\circ}\text{C}$, $f = 1.0\text{ MHz}$ 。

表 9.9 TSOP 封装连接器电容

参数符号	参数说明	测试设置	典型值	最大值	单位
C_{IN}	输入电容	$V_{IN} = 0$	3	5	pF
C_{OUT}	输出电容	$V_{OUT} = 0$	3	4.5	pF
C_{IN2}	控制引脚电容	$V_{IN} = 0$	3.5	7	pF
RY/BY#	输出电容	$V_{OUT} = 0$	2.5	3.5	pF
RESET#	复位输入电容	$V_{IN} = 0$	20	22	pF

注意:

1. 采样值, 并非 100% 经过了测试。
2. 测试条件: $T_A = 25^{\circ}\text{C}$, $f = 1.0\text{ MHz}$ 。

10. 时序规范

10.1 波形切换

波形	输入	输出
		稳定
		从 H 变为 L
		从 L 变为 H
	不用考虑，任何更改都可以	更改，状态未知
	不适用	中间行是高阻抗状态 (High-Z)

10.2 交流测试条件

图 10.1 测试设置

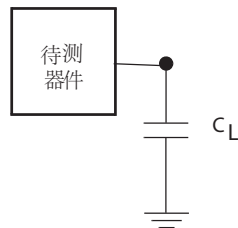


表 10.1 测试规范

参数	所有速度	单位
输出负载电容, C_L	30	pF
输入上升和下降时间 (1)	1.5	ns
输入脉冲电平	$0.0-V_{IO}$	V
输入时序测量参考电平	$V_{IO}/2$	V
输出时序测量参考电平	$V_{IO}/2$	V

注意:

1. 在 V_{IL} 最大值和 V_{IH} 最小值之间进行测试。

图 10.2 输入波形和测量电平



10.3 上电复位 (POR) 和热复位

为了确保稳定的 V_{CC} 和 V_{IO} 电源, 必须采取正常的预防措施进行电源去耦。系统中每个器件的 V_{CC} 和 V_{IO} 电源应使用一个大小合适的电容器进行去耦 (此电容器需与封装接近, 且通常为 $0.1 \mu F$)。

表 10.2 加电和复位参数

参数	说明	限制	数值	单位
t_{VCS}	从 V_{CC} 建立到开始第一次访问的时间 (1)、(2)	最小值	300	μs
t_{VIOS}	从 V_{IO} 建立到开始第一次访问的时间 (1)、(2)	最小值	300	μs
t_{RPH}	从 $RESET\#$ 为低电平到 $CE\#$ 为低电平的时间	最小值	35	μs
t_{RP}	$RESET\#$ 脉冲宽度	最小值	200	ns
t_{RH}	$RESET\#$ (高电平) 和 $CE\#$ (低电平) 之间的时间	最小值	50	ns
t_{CEH}	$CE\#$ 脉冲宽度高电平	最小值	20	ns

注意:

- 并非 100% 经过了测试。
- 从 V_{CC} 达 V_{CC} 最小值和 V_{IO} 达 V_{IO} 最小值到 $RESET$ 为 V_{IH} 和 $CE\#$ 为 V_{IL} 之间进行时序测量。
- $RESET\#$ 低在 POR 期间可选。如果在 POR 期间激活 $RESET$, t_{RPH} , t_{VIOS} 或 t_{VCS} 中的最后发生的时间确定 $CS\#$ 何时变为低电平。如果 $RESET\#$ 在 t_{VIOS} 或 t_{VCS} 后保持低的要求得到满足, t_{RPH} 将从 t_{VIOS} 或 t_{VCS} 结束时开始计算。 $CS\#$ 变为低电平前, $RESET$ 必须在 t_{RH} 时间内保持为高电平。
- 上电期间, $V_{CC} \geq V_{IO} - 200 mV$ 。
- V_{CC} 和 V_{IO} 的升降速率可能是非线性的。
- $t_{RP} + t_{RH}$ 的总和不能小于 t_{RPH} 。

10.3.1 上电 (冷) 复位 (POR)

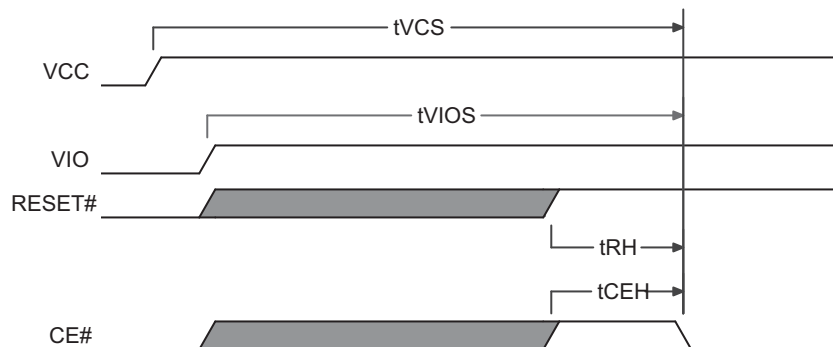
在电源上升期间, V_{IO} 电源电压不可大于 V_{CC} 电源电压。 V_{IH} 同样不能大于 V_{IO} 供电电压。

冷复位嵌入式算法需要一个长达数百 μs 的时间 (t_{VCS}) 从非易失性存储器加载所有 EAC 算法和默认状态。在冷复位期间, 所有控制信号 (包括 $CE\#$ 和 $RESET\#$) 均被忽略。如果 $CE\#$ 在 t_{VCS} 期间保持为低电平, 器件在 t_{VCS} 时间内消耗的电流可能超过典型 POR 电流, 但 $CE\#$ 的电平不会影响冷复位 EA。在 t_{VCS} 期间, $RESET\#$ 可能为高或低电平。如果 $RESET\#$ 在 t_{VCS} 期间为低电平, 它在 t_{VCS} 结束时可能保持该状态, 以使器件保持硬件复位状态。如果 $RESET\#$ 在 t_{VCS} 结束时为高电平, 器件将进入待机状态。

第一次上电时, 电源电压先低于 V_{RST} , 然后逐渐上升, 以达到运行范围的最小值, 内部器件配置和冷复位操作被启动。 $CE\#$ 在 POR 操作期间 (t_{VCS} 或 t_{VIOS}) 被忽略。 $RESET\#$ 低在此 POR 期间可选。如果在 POR 期间将 $RESET\#$ 置低, 则必须满足硬件复位参数 t_{RP} 和 t_{RPH} 。这样, 复位操作会在 t_{VCS} 或 t_{VIOS} 或 t_{RPH} 结束后完成。 $CE\#$ 、 $OE\#$ 或地址转换会启动第一个读取操作。如果 $CE\#$ 在 POR 期间保持为低电平, 那么当前地址将自动被读取。

在冷复位期间, 器件消耗的电流为 I_{CC7} 。

图 10.3 上电图



10.3.2 硬件（热）复位

在硬件复位期间 (t_{RP})，器件消耗的电流为 I_{CC5} 。

RESET# 保持为 V_{SS} 电压时，器件消耗的电流为 CMOS 待机电流 (I_{CC4})。如果 RESET# 保持为 V_{IL} 而不是 V_{SS} ，则待机电流会更大。

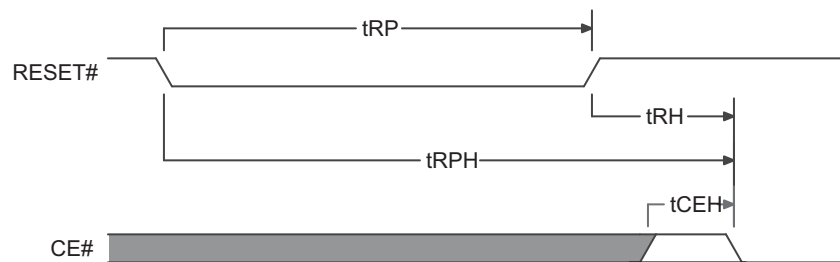
在 t_{VCS} 之后，如果器件尚未完成冷复位，而 RESET# 已被激活，将执行 Cold Reset# EA 而不是 Warm RESET#，并且需要 t_{VCS} 时间完成。请参见第 70 页上的图 10.4，硬件复位。

器件完成 POR 并进入待机状态后，以后转换到硬件复位状态时将启动热复位嵌入式算法。热复位的时间比冷复位的时间短得多，它只需要几十 μs (t_{RPH}) 便能完成。在热复位 EA 期间，正在执行的所有嵌入式算法都停止，EAC 返回其 POR 状态，而不会从非易失性存储器重新加载 EAC 算法。热复位 EA 完成后，如果 RESET# 保持低，接口将保持在硬件复位状态。当 RESET# 返回高时，接口将转换到待机状态。如果 RESET# 在热复位 EA 结束时为高，接口将直接转换到待机状态。如果 CE# 在热复位期间保持低电平，那么当前地址将自动被读取。

如果在 t_{VCS} 结束时尚未正确完成 POR，以后转换到硬件复位状态时将导致转换到上电复位接口状态，并启动冷复位嵌入式算法。这样可确保器件能够完成冷复位，即使系统上电电压升高的相关问题可导致 POR 不能正确启动或完成。RY/BY# 引脚在冷复位或热复位期间为低，以指明器件正在执行复位操作。

硬件复位由 RESET# 信号变为 V_{IL} 启动。

图 10.4 硬件复位



10.4 交流特性

10.4.1 异步读取操作

表 10.3 读取操作 $V_{IO} = V_{CC} = 2.7\text{ V} \sim 3.6\text{ V}$ (温度为 $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$)

参数		说明	测试设置		速度选项	单位
JEDEC	标准				100	
t_{AVAV}	t_{RC}	读周期的时长 (1)	512 Mb, 1 Gb	最小值	100	ns
t_{AVQV}	t_{ACC}	地址到输出延迟时间 CE# = V_{IL} OE# = V_{IL}	512 Mb, 1 Gb	最大值	100	ns
t_{ELQV}	t_{CE}	片选到输出延迟时间 OE# = V_{IL}	512 Mb, 1 Gb	最大值	100	ns
	t_{PACC}	页面访问时间	512 Mb, 1 Gb	最大值	15	ns
t_{GLQV}	t_{OE}	输出使能到输出延迟时间	读取	最大值	25	ns
			轮询	最大值	35	
t_{AXQX}	t_{OH}	自地址 CE# 或 OE# (先发生者) 的输出保持时间		最小值	0	ns
t_{EHQZ}	t_{DF}	片选使能或输出使能到输出为高阻态的时间 (1)		最大值	15	ns
	t_{OEH}	输出使能保持时间 (1)	读取	最小值	0	ns
			轮询	最小值	10	ns
	t_{ASO}	地址建立时间	轮询	最小值	15	ns
	t_{ASH}	地址保持时间	轮询	最小值	0	ns
	t_{CEPH}	CE# 为高电平的时间	轮询	最小值	20	ns
	t_{OEP}	OE# 为低电平的时间	轮询	最小值	25	ns
	t_{OEPH}	OE# 为高电平的时间	轮询	最小值	20	ns
	t_{OEC}	OE# 周期时间	轮询	最小值	60	ns
	t_{ASSB}	从自动睡眠模式切换为待机模式的时间 (1)	CE# = V_{IL} , 地址稳定	典型值	5	μs
				最大值	8	μs
t_{BLEL}	t_{FLEL}	从 BYTE# 为低电平到 CE# 为低电平的时间		最小值	10	ns
t_{BHEL}	t_{FHEL}	从 BYTE# 为高电平到 CE# 为低电平的时间		最小值	10	ns
t_{BLQV}	t_{FLQV}	从 BYTE# 为低电平到输出为高阻态的时间 (1)		最大值	1	μs
t_{BHQV}	t_{FHQV}	从 BYTE# 为高电平到输出的延迟时间		最大值	1	μs

注意:

- 并非 100% 经过了测试。

表 10.4 读取操作 $V_{IO} = 1.65\text{ V} \sim V_{CC}$, $V_{CC} = 2.7\text{ V} \sim 3.6\text{ V}$ (温度为 $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$)

参数		说明	测试设置		速度选项	单位	
JEDEC	标准				110		
t_{AVAV}	t_{RC}	读周期的时长 (1)		512 Mb, 1 Gb	最小值	110	ns
t_{AVQV}	t_{ACC}	地址到输出延迟时间	$CE\# = V_{IL}$ $OE\# = V_{IL}$	512 Mb, 1 Gb	最大值	110	ns
t_{ELQV}	t_{CE}	片选到输出延迟时间	$OE\# = V_{IL}$	512 Mb, 1 Gb	最大值	110	ns
	t_{PACC}	页面访问时间		512 Mb, 1 Gb	最小值	25	ns
t_{GLQV}	t_{OE}	输出使能到输出延迟时间		读取和轮询	最大值	35	ns
t_{AXQX}	t_{OH}	自地址 $CE\#$ 或 $OE\#$ (先发生者) 的输出保持时间			最小值	0	ns
t_{EHQZ}	t_{DF}	片选使能或输出使能到输出为高阻态的时间 (1)			最大值	20	ns
	t_{OEh}	输出使能保持时间 (1)		读取	最小值	0	ns
				轮询	最小值	10	ns
	t_{ASO}	地址建立时间		轮询	最小值	15	ns
	t_{ASH}	地址保持时间		轮询	最小值	0	ns
	t_{CEPH}	$CE\#$ 为高电平的时间		轮询	最小值	20	ns
	t_{OEP}	$OE\#$ 为低电平的时间		轮询	最小值	25	ns
	t_{OEPH}	$OE\#$ 为高电平的时间		轮询	最小值	20	ns
	t_{OEC}	$OE\#$ 周期时间		轮询	最小值	60	ns
	t_{ASSB}	从自动睡眠模式切换为待机模式的时间 (1)		$CE\# = V_{IL}$, 地址稳定	典型值	5	μs
					最大值	8	μs
t_{BLEL}	t_{FLEL}	从 $BYTE\#$ 为低电平到 $CE\#$ 为低电平的时间			最小值	10	ns
t_{BHEL}	t_{FHEL}	从 $BYTE\#$ 为高电平到 $CE\#$ 为低电平的时间			最小值	10	ns
t_{BLQV}	t_{FLQV}	从 $BYTE\#$ 为低电平到输出为高阻态的时间 (1)			最大值	1	μs
t_{BHQV}	t_{FHQV}	从 $BYTE\#$ 为高电平到输出的延迟时间			最大值	1	μs

注意:

1. 并非 100% 经过了测试。

表 10.5 读取操作 $V_{IO} = V_{CC} = 2.7\text{ V} \sim 3.6\text{ V}$ (温度为 $-40^{\circ}\text{C} \sim +105^{\circ}\text{C}$)

参数		说明	测试设置		速度选项	单位
JEDEC	标准				110	
t_{AVAV}	t_{RC}	读周期的时长 (1)	512 Mb, 1 Gb	最小值	110	ns
t_{AVQV}	t_{ACC}	地址到输出延迟时间 CE# = V_{IL} OE# = V_{IL}	512 Mb, 1 Gb	最大值	110	ns
t_{ELQV}	t_{CE}	片选到输出延迟时间 OE# = V_{IL}	512 Mb, 1 Gb	最大值	110	ns
	t_{PACC}	页面访问时间	512 Mb, 1 Gb	最大值	15	ns
t_{GLQV}	t_{OE}	输出使能到输出延迟时间	读取	最大值	25	ns
			轮询	最大值	35	
t_{AXQX}	t_{OH}	自地址 CE# 或 OE# (先发生者) 的输出保持时间		最小值	0	ns
t_{EHQZ}	t_{DF}	片选使能或输出使能到输出为高阻态的时间 (1)		最大值	15	ns
	t_{OEHL}	输出使能保持时间 (1)	读取	最小值	0	ns
			轮询	最小值	10	ns
	t_{ASO}	地址建立时间	轮询	最小值	15	ns
	t_{ASH}	地址保持时间	轮询	最小值	0	ns
	t_{CEPH}	CE# 为高电平的时间	轮询	最小值	20	ns
	t_{OEP}	OE# 为低电平的时间	轮询	最小值	25	ns
	t_{OEPH}	OE# 为高电平的时间	轮询	最小值	20	ns
	t_{OEC}	OE# 周期时间	轮询	最小值	60	ns
	t_{ASSB}	从自动睡眠模式切换为待机模式的时间 (1)	CE# = V_{IL} , 地址稳定	典型值	5	μs
				最大值	8	μs
t_{BLEL}	t_{FLEL}	从 BYTE# 为低电平到 CE# 为低电平的时间		最小值	10	ns
t_{BHEL}	t_{FHEL}	从 BYTE# 为高电平到 CE# 为低电平的时间		最小值	10	ns
t_{BLQV}	t_{FLQV}	从 BYTE# 为低电平到输出为高阻态的时间 (1)		最大值	1	μs
t_{BHQV}	t_{FHQV}	从 BYTE# 为高电平到输出的延迟时间		最大值	1	μs

注意:

- 并非 100% 经过了测试。

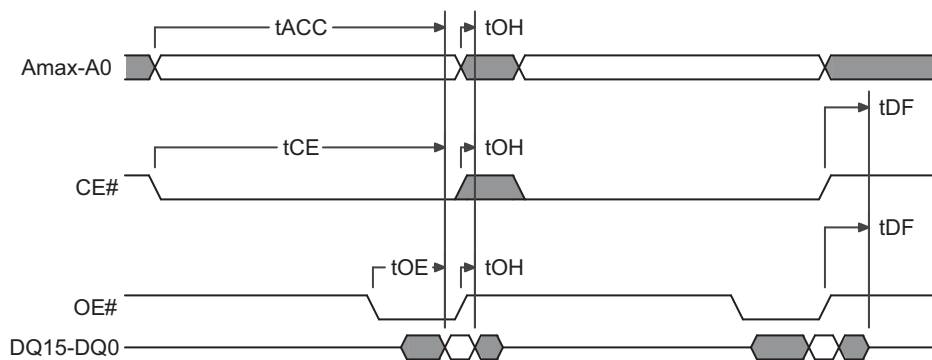
表 10.6 读取操作 $V_{IO} = 1.65\text{ V} \sim V_{CC}$, $V_{CC} = 2.7\text{ V} \sim 3.6\text{ V}$ (温度为 $-40^{\circ}\text{C} \sim +105^{\circ}\text{C}$)

参数		说明	测试设置		速度选项	单位	
JEDEC	标准				120		
t_{AVAV}	t_{RC}	读周期的时长 (1)		512 Mb, 1 Gb	最小值	120	ns
t_{AVQV}	t_{ACC}	地址到输出延迟时间	$CE\# = V_{IL}$ $OE\# = V_{IL}$	512 Mb, 1 Gb	最大值	120	ns
t_{ELQV}	t_{CE}	片选到输出延迟时间	$OE\# = V_{IL}$	512 Mb, 1 Gb	最大值	120	ns
	t_{PACC}	页面访问时间		512 Mb, 1 Gb	最大值	25	ns
t_{GLQV}	t_{OE}	输出使能到输出延迟时间		读取和轮询	最大值	35	ns
t_{AXQX}	t_{OH}	自地址 $CE\#$ 或 $OE\#$ (先发生者) 的输出保持时间			最小值	0	ns
t_{EHQZ}	t_{DF}	片选使能或输出使能到输出为高阻态的时间 (1)			最大值	15	ns
	t_{OEHL}	输出使能保持时间 (1)		读取	最小值	0	ns
				轮询	最小值	10	ns
	t_{ASO}	地址设置时间		轮询	最小值	15	ns
	t_{ASH}	地址保持时间		轮询	最小值	0	ns
	t_{CEPH}	$CE\#$ 为高电平的时间		轮询	最小值	20	ns
	t_{OEP}	$OE\#$ 为低电平的时间		轮询	最小值	25	ns
	t_{OEPH}	$OE\#$ 为高电平的时间		轮询	最小值	20	ns
	t_{OEC}	$OE\#$ 周期时间		轮询	最小值	60	ns
	t_{ASSB}	从自动睡眠模式切换为待机模式的时间 (1)		$CE\# = V_{IL}$, 地址稳定	典型值	5	μs
					最大值	8	μs
t_{BLEL}	t_{FLEL}	从 $BYTE\#$ 为低电平到 $CE\#$ 为低电平的时间			最小值	10	ns
t_{BHEL}	t_{FHEL}	从 $BYTE\#$ 为高电平到 $CE\#$ 为低电平的时间			最小值	10	ns
t_{BLQV}	t_{FLQV}	从 $BYTE\#$ 为低电平到输出为高阻态的时间 (1)			最大值	1	μs
t_{BHQV}	t_{FHQV}	从 $BYTE\#$ 为高电平到输出的延迟时间			最大值	1	μs

注意:

- 并非 100% 经过了测试。

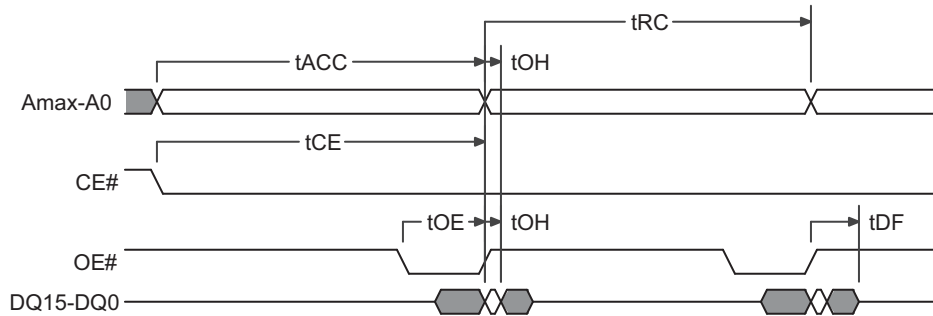
图 10.5 相邻读取 (t_{ACC}) 操作时序图



注意:

- 地址为 $Amax:A0$ (字模式); $Amax:A-1$ (字节模式), 数据为 $DQ15-DQ0$ (字模式); $DQ7-DQ0$ (字节模式)。

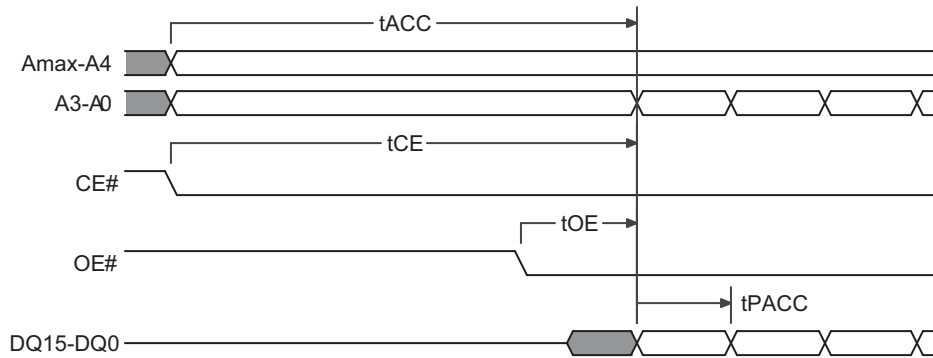
图 10.6 相邻读取操作 (t_{RC}) 时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。
2. 相邻操作 (访问之间 CE# 保持低) 要求更改地址以启动第二个访问。

图 10.7 页面读取时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。
2. 切换 A3:A0 (字模式); A3:A-1 (字节模式)。

10.4.2 异步写入操作

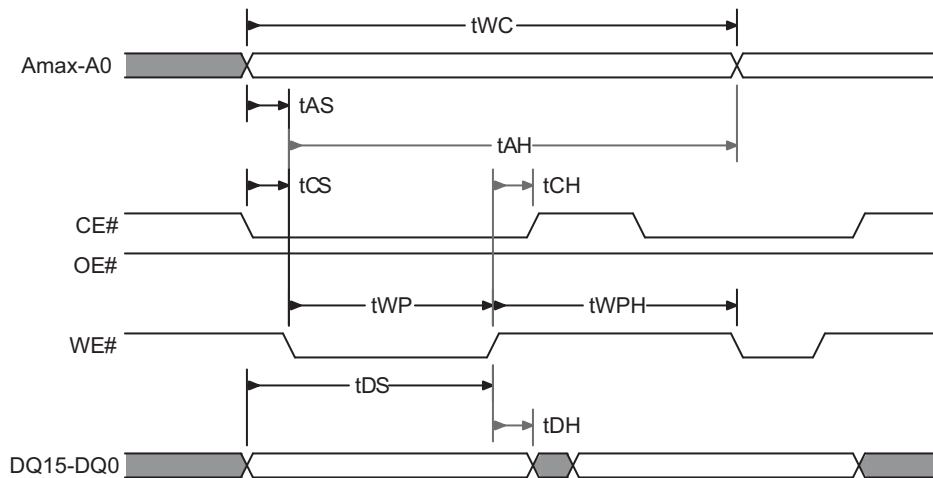
表 10.7 写入操作

参数		说明		$V_{IO} = 2.7 V \sim V_{CC}$	$V_{IO} = 1.65 V \sim V_{CC}$	单位
JEDEC	标准					
t_{AVAV}	t_{WC}	写周期的时长 (1)	最小值	60		ns
t_{AVWL}	t_{AS}	地址建立时间	最小值	0		ns
	t_{ASO}	切换位轮询期间从地址建立到 OE# 为低电平的时间	最小值	15		ns
t_{WLAX}	t_{AH}	地址保持时间	最小值	45		ns
	t_{AHT}	切换位轮询期间自 CE# 或 OE# 为高电平的地址保持时间	最小值	0		ns
t_{DVWH}	t_{DS}	数据建立时间	最小值	30		ns
t_{WHDX}	t_{DH}	数据保持时间	最小值	0		ns
t_{GHWL}	t_{GHWL}	写入之前的读取恢复时间 (OE# 为高电平到 WE# 为低电平)	最小值	0		ns
t_{ELWL}	t_{CS}	CE# 建立时间	最小值	0		ns
t_{WHEH}	t_{CH}	CE# 保持时间	最小值	0		ns
t_{WLWH}	t_{WP}	WE# 脉冲宽度	最小值	25		ns
t_{WHWL}	t_{WPH}	WE# 脉冲宽度高电平	最小值	20		ns
	t_{SEA}	扇区擦除超时时间	最小值	50		μs

注意:

- 并非 100% 经过了测试。

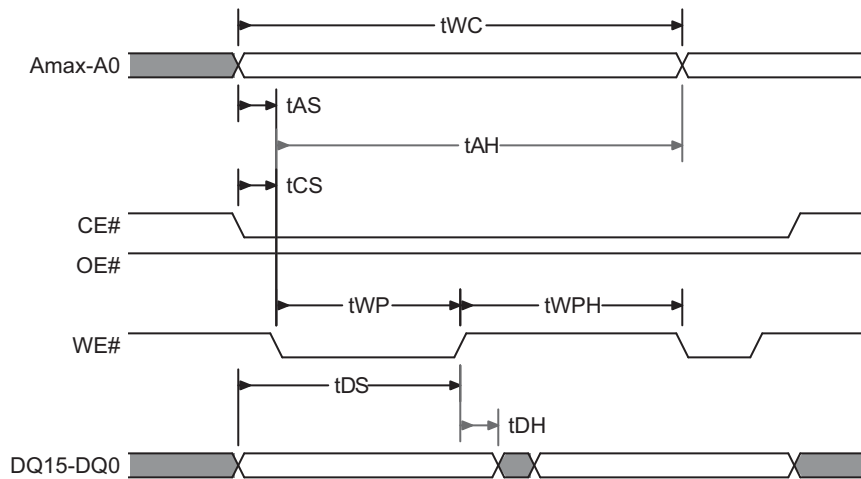
图 10.8 相邻写入操作时序图



注意:

- 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

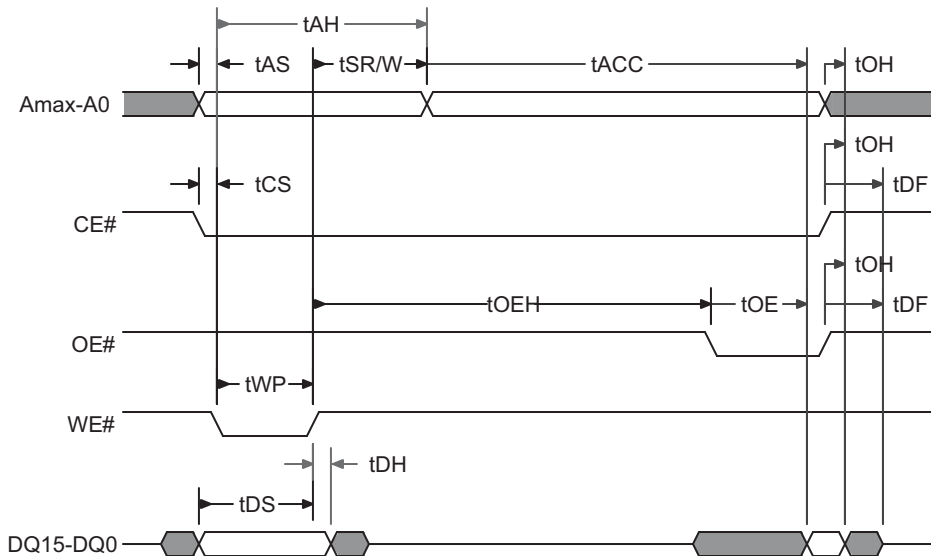
图 10.9 相邻 (CE#VIL) 写入操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

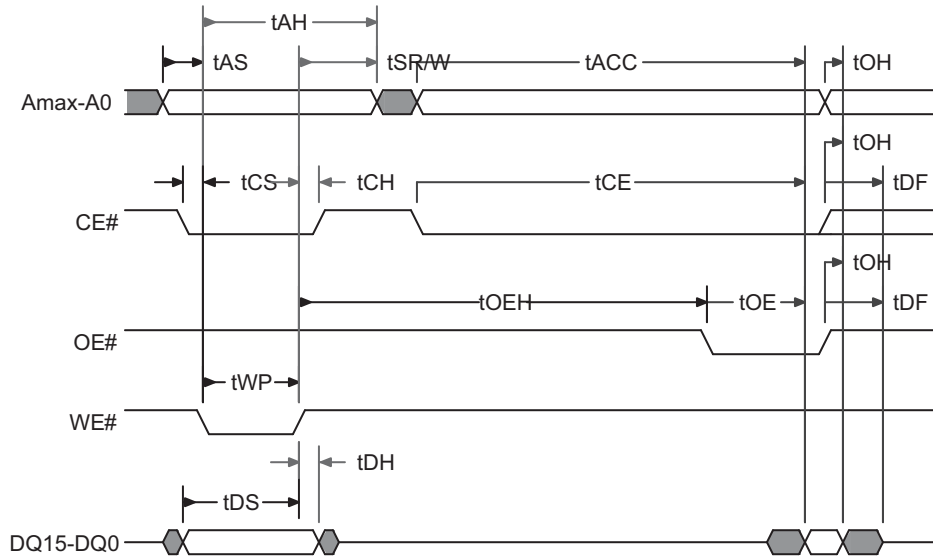
图 10.10 写入到读取 (t_{ACC}) 操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

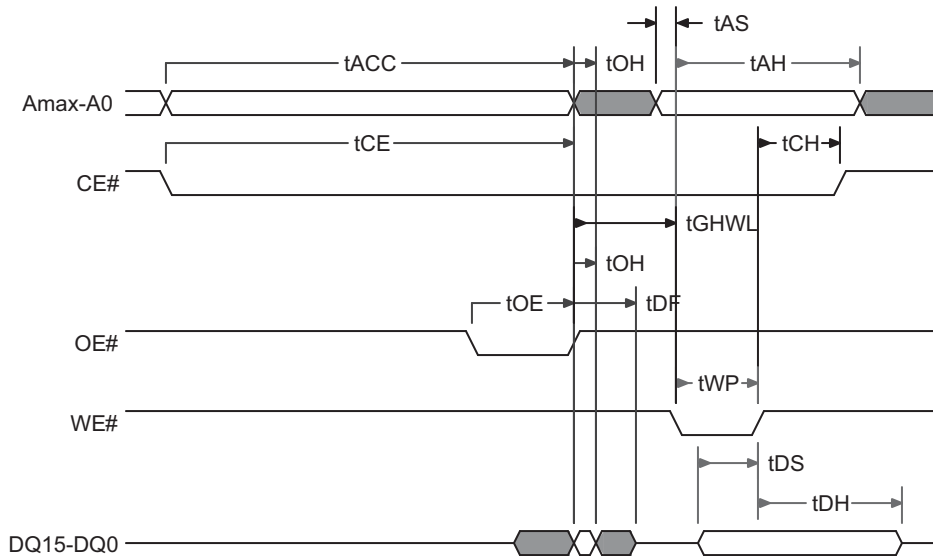
图 10.11 写入到读取 (t_{CE}) 操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

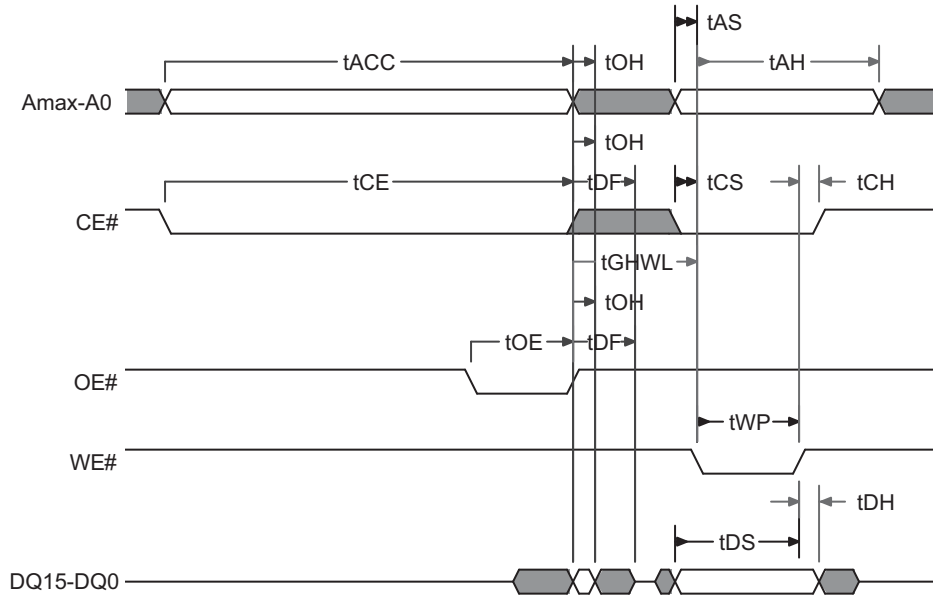
图 10.12 读取到写入 ($CE\# V_{IL}$) 操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

图 10.13 读取到写入（CE# 切换）操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

表 10.8 擦除 / 编程操作

参数		说明		$V_{IO} = 2.7 V \sim V_{CC}$	$V_{IO} = 1.65 V \sim V_{CC}$	单位
JEDEC	标准					
t_{WHWH1}	t_{WHWH1}	执行写入缓冲区编程操作的时间	典型值	(3)		μs
		有效对每个字进行写入缓冲区编程操作的时间	典型值	(3)		μs
		对字或页面进行编程操作的时间	典型值	(3)		μs
t_{WHWH2}	t_{WHWH2}	执行扇区擦除操作的时间 (1)	典型值	(3)		ms
	t_{BUSY}	擦除 / 编程有效到 RY/BY# 延迟时间	最大值	80		ns
	t_{SRW}	读取和写入操作之间的等待时间 (2)	最小值	10		ns
	t_{ESL}	擦除挂起等待时间	最大值	(3)		μs
	t_{PSL}	编程挂起等待时间	最大值	(3)		μs
	t_{RB}	RY/BY# 恢复时间	最小值	0		μs
	t_{PPB}	PPB 锁定 / 解锁时间	最小值	80		μs
			最大值	120		
	t_{DP}	数据轮询到受保护扇区 (编程) 的时间	最小值	3	μs	
			最大值	20		
		数据轮询到受保护的扇区 (擦除) 的时间	最小值	3		
			最大值	100		
	t_{VHH}	V_{HH} 上升和下降时间 (1)	最小值	250		ns
	t_{TOR}	清除超出时序错误的时间 (DQ5)	最小值	100		ns

注意:

- 并非 100% 经过了测试。
- 在 WE# 上升沿上, 必须等待 t_{SRW} 时间, 然后才切换到另一个地址。
- 请参考第 40 页上的表 5.7 和第 41 页上的表 5.8, 了解特定值信息。

图 10.14 加速编程操作时序图

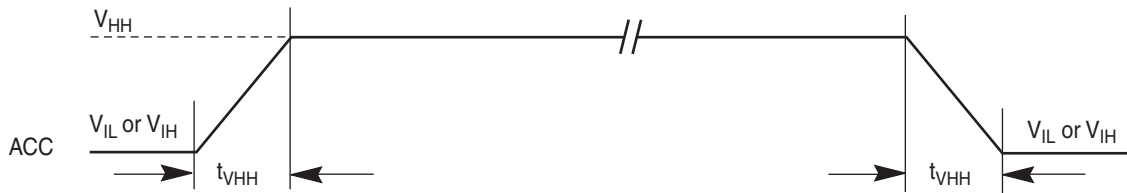
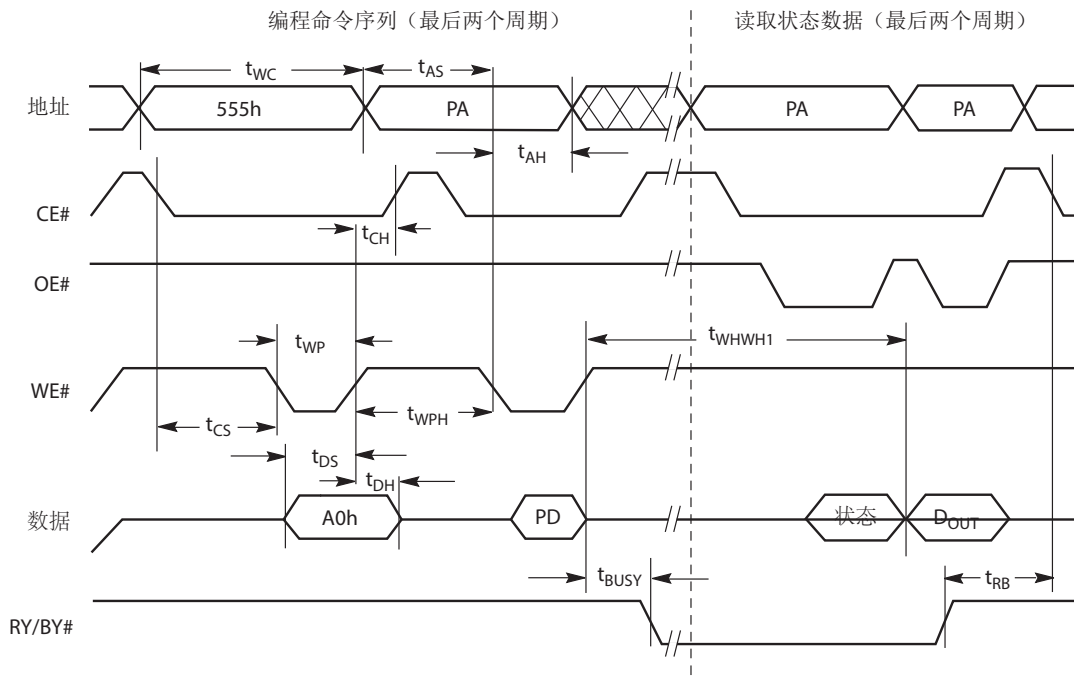


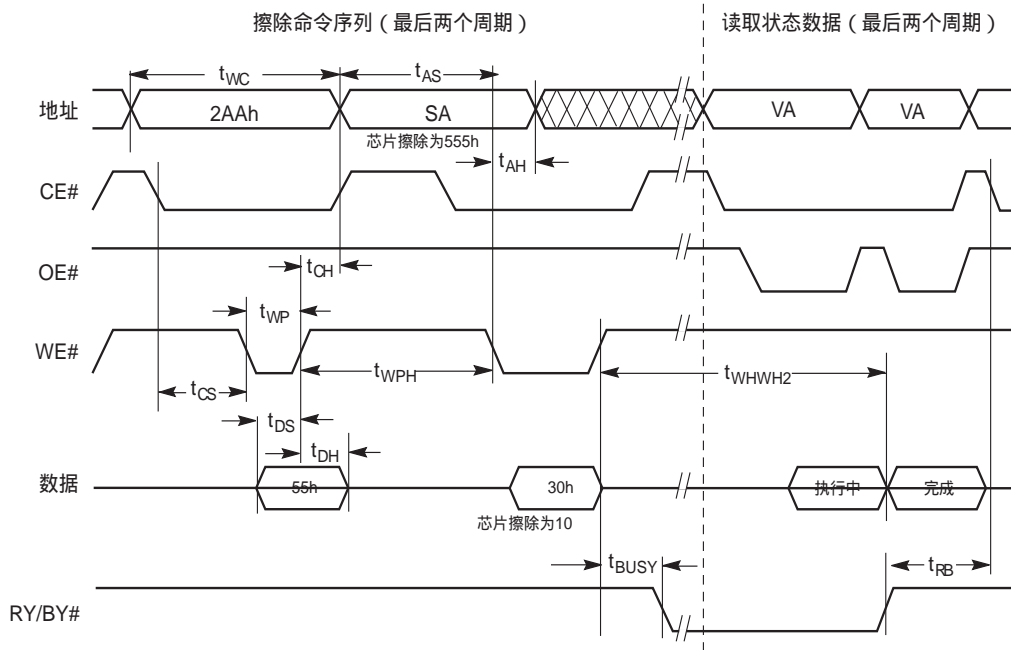
图 10.15 编程操作时序图



注意:

1. 地址为 $A_{max}:A_0$ (字模式); $A_{max}:A-1$ (字节模式), 数据为 $DQ_{15}-DQ_0$ (字模式); DQ_7-DQ_0 (字节模式)。
2. PA = 编程地址, PD = 编程数据, D_{OUT} 是位于编程地址的真实数据。

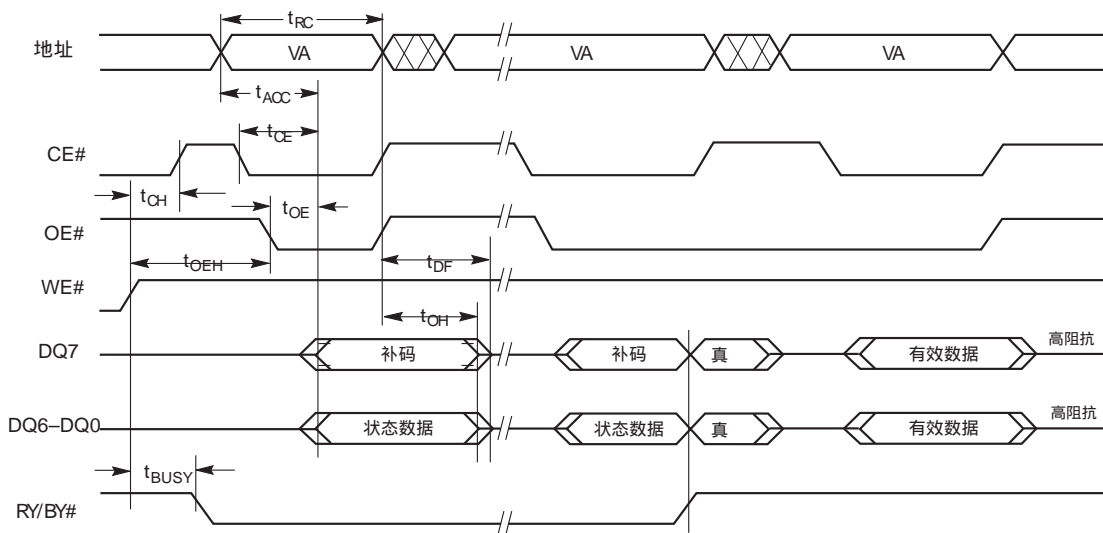
图 10.16 芯片 / 扇区擦除操作时序图



注意:

1. 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。
2. SA = 扇区地址 (对于扇区擦除), VA = 用于读取状态数据的有效地址。

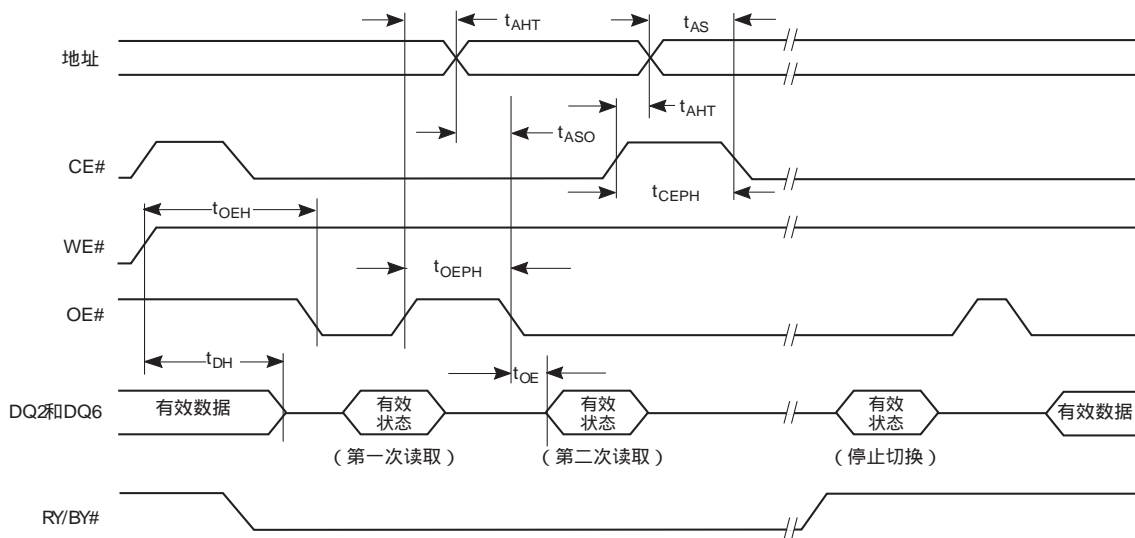
图 10.17 Data# 轮询时序图 (在执行嵌入式算法期间)



注意:

1. VA = 有效地址。图例显示命令序列之后的第一个状态循环, 最后一个状态读取循环, 以及阵列数据读取循环。

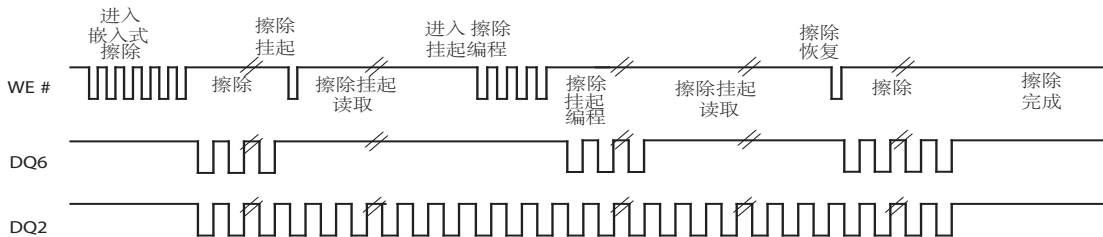
图 10.18 切换位时序图（在执行嵌入式算法期间）



注意:

1. 器件忙碌时，DQ6 将在任一读取地址切换。如果地址在当前擦除扇区之内，DQ2 将切换。

图 10.19 DQ2 与 DQ6 关系图



注意:

1. 系统可以使用 OE# 或 CE# 切换 DQ2 和 DQ6。仅当读取的地址在擦除挂起扇区内时，DQ2 才切换。

10.4.3 备用 CE# 控制的写入操作

表 10.9 备用 CE# 控制的写入操作

参数		说明		V _{IO} = 2.7 V ~ V _{CC}	V _{IO} = 1.65 V ~ V _{CC}	单位
JEDEC	标准					
t _{AVAV}	t _{WC}	写周期的时长 (1)	最小值	60		ns
t _{AVWL}	t _{AS}	地址建立时间	最小值	0		ns
	t _{ASO}	切换位轮询期间从地址建立到 OE# 为低电平的时间	最小值	15		ns
t _{WLAX}	t _{AH}	地址保持时间	最小值	45		ns
	t _{AHT}	切换位轮询期间自 CE# 或 OE# 为高电平的地址保持时间	最小值	0		ns
t _{DVWH}	t _{DS}	数据建立时间	最小值	30		ns
t _{WHDX}	t _{DH}	数据保持时间	最小值	0		ns
	t _{CEPH}	在切换位轮询期间 CE# 为高电平的时间	最小值	20		ns
	t _{OEPH}	切换位轮询期间 OE# 为高电平的时间	最小值	20		ns
t _{GHEK}	t _{GHEL}	写入之前的读取恢复时间 (OE# 为高电平到 WE# 为低电平)	最小值	0		ns

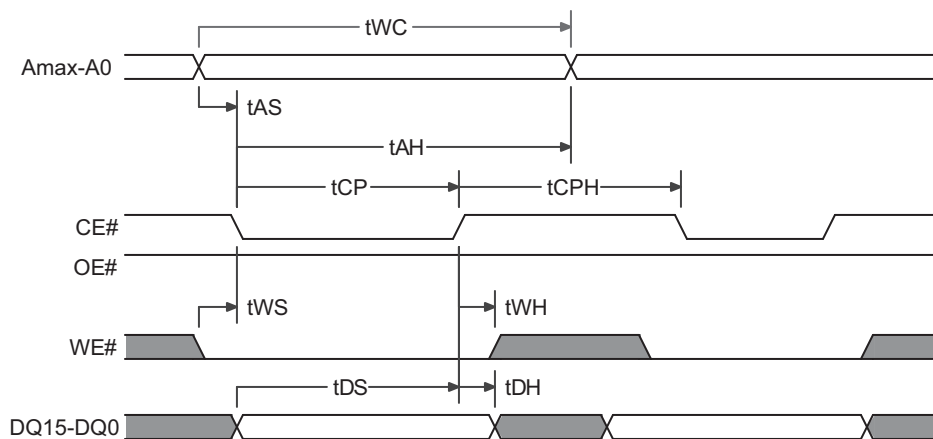
表 10.9 备用 CE# 控制的写入操作 (续)

参数		说明		$V_{IO} = 2.7 V \sim V_{CC}$	$V_{IO} = 1.65 V \sim V_{CC}$	单位
JEDEC	标准					
t_{WLEL}	t_{WS}	WE# 建立时间	最小值	0		ns
t_{ELWH}	t_{WH}	WE# 保持时间	最小值	0		ns
t_{ELEH}	t_{CP}	CE# 脉冲宽度	最小值	25		ns
t_{EHEL}	t_{CPH}	CE# 脉冲宽度高电平	最小值	20		ns
	t_{SEA}	扇区擦除超时时间	最小值	50		μs

注意:

- 并非 100% 经过了测试。

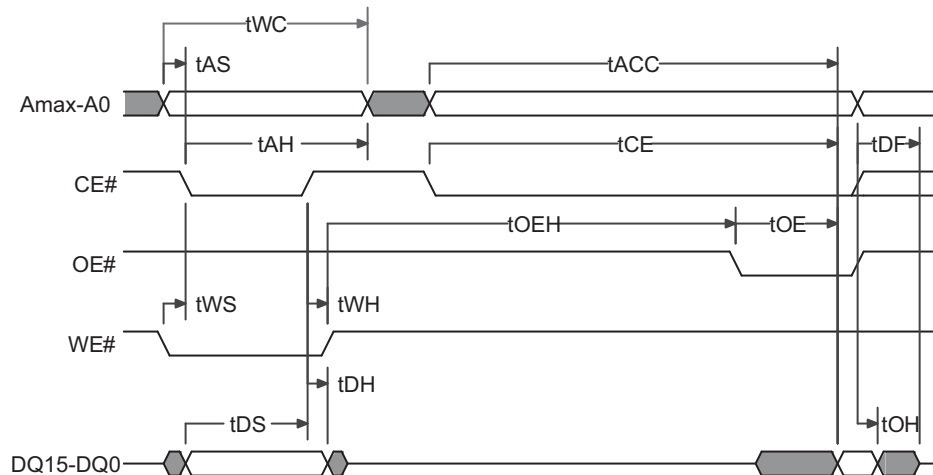
图 10.20 相邻 (CE#) 写入操作时序图



注意:

- 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

图 10.21 (CE#) 写入到读取操作时序图



注意:

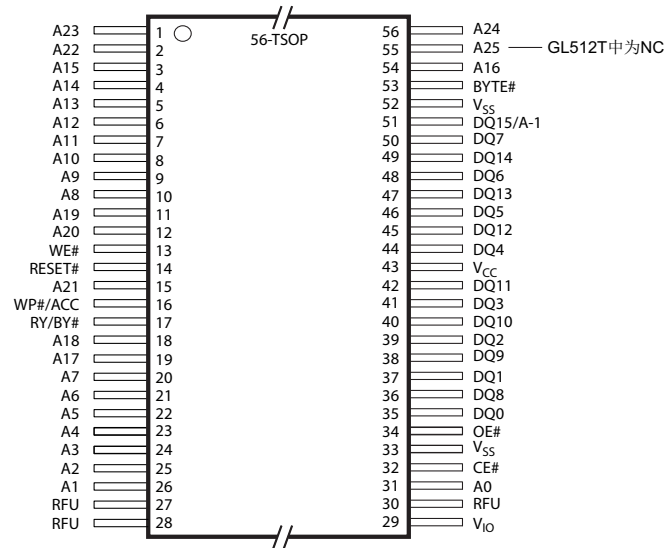
- 地址为 Amax:A0 (字模式); Amax:A-1 (字节模式), 数据为 DQ15-DQ0 (字模式); DQ7-DQ0 (字节模式)。

11. 物理接口

11.1 56-TSOP

11.1.1 连接图

图 11.1 标准 56-TSOP

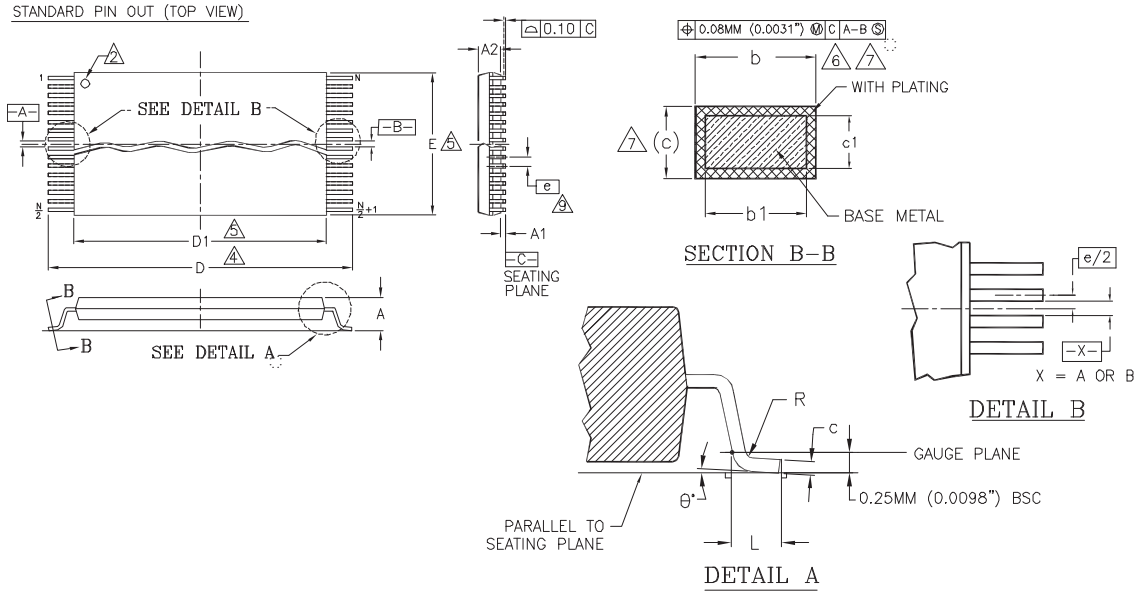


注意:

1. 引脚 27、28 和 30 保留给未来使用 (RFU)。

11.1.2 物理图

图 11.2 56 引脚薄小外形封装 (TSOP), 14 x 20 mm



PACKAGE	TS 56		
JEDEC	MO-142 (B) EC		
SYMBOL	MIN.	NOM.	MAX.
A	---	---	1.20
A1	0.05	---	0.15
A2	0.95	1.00	1.05
b1	0.17	0.20	0.23
b	0.17	0.22	0.27
c1	0.10	---	0.16
c	0.10	---	0.21
D	19.80	20.00	20.20
D1	18.30	18.40	18.50
E	13.90	14.00	14.10
e	0.50 BASIC		
L	0.50	0.60	0.70
∅	0*	-	8*
R	0.08	---	0.20
N	56		

NOTES:

- 1 CONTROLLING DIMENSIONS ARE IN MILLIMETERS (mm). (DIMENSIONING AND TOLERANCING CONFORMS TO ANSI Y14.5M-1982.)
- 2 PIN 1 IDENTIFIER FOR STANDARD PIN OUT (DIE UP).
- 3 TO BE DETERMINED AT THE SEATING PLANE -C-. THE SEATING PLANE IS DEFINED AS THE PLANE OF CONTACT THAT IS MADE WHEN THE PACKAGE LEADS ARE ALLOWED TO REST FREELY ON A FLAT HORIZONTAL SURFACE.
- 4 DIMENSIONS D1 AND E DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.15 mm PER SIDE.
- 5 DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 mm TOTAL IN EXCESS OF b DIMENSION AT MAX MATERIAL CONDITION. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD TO BE 0.07 mm.
- 6 THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 mm AND 0.25 mm FROM THE LEAD TIP.
- 7 LEAD COPLANARITY SHALL BE WITHIN 0.10 mm AS MEASURED FROM THE SEATING PLANE.
- 8 DIMENSION "e" IS MEASURED AT THE CENTERLINE OF THE LEADS.

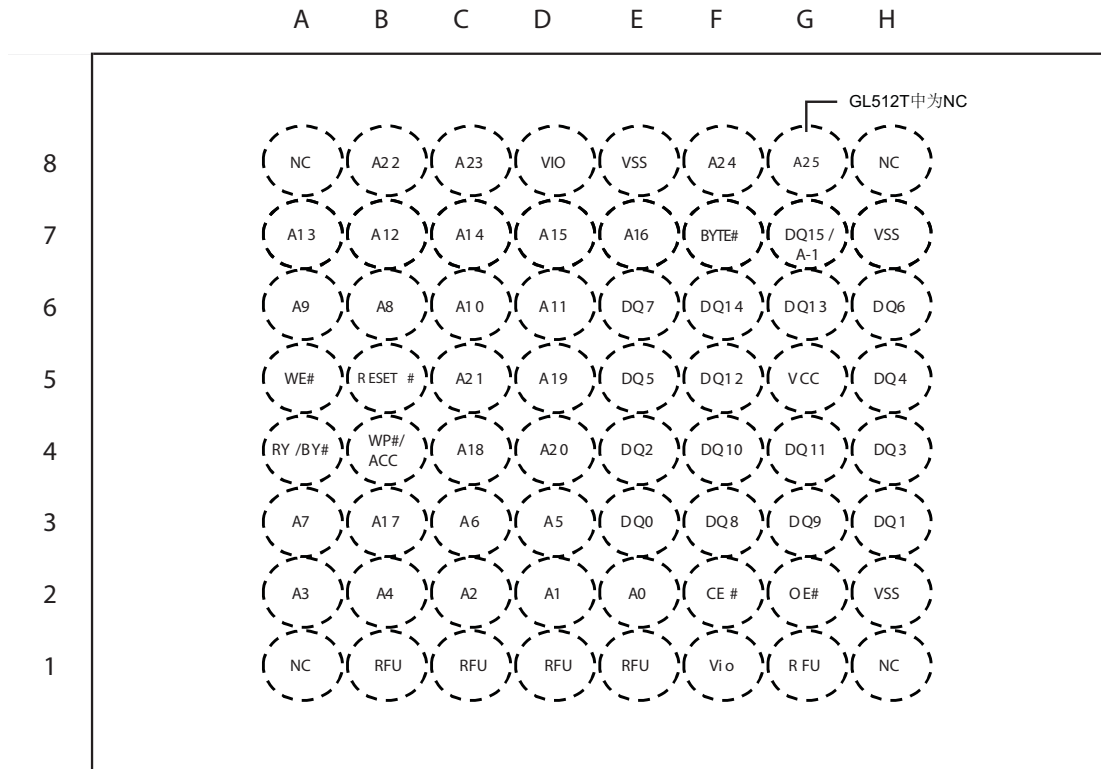
316038.10A

11.2 64 球 FBGA

11.2.1 连接图

图 11.3 64 球加固球栅阵列
顶视图

产品 引脚分布

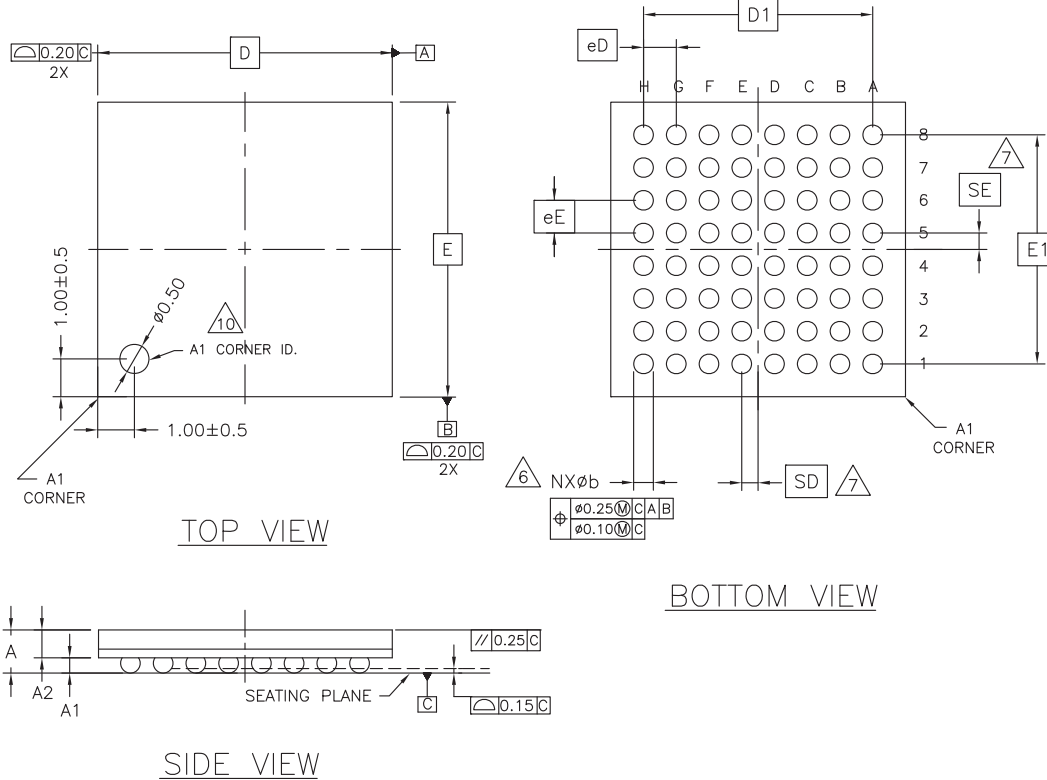


注意:

1. 球 A1、A8、H1 和 H8 是无连接 (NC)。
2. 球 B1、C1、D1、E1 和 G1 是保留给未来使用 (RFU)。

11.2.2 物理图 — LAE064

图 11.4 LAE064 — 64 球加固球栅阵列 (FBGA), 9 x 9 mm



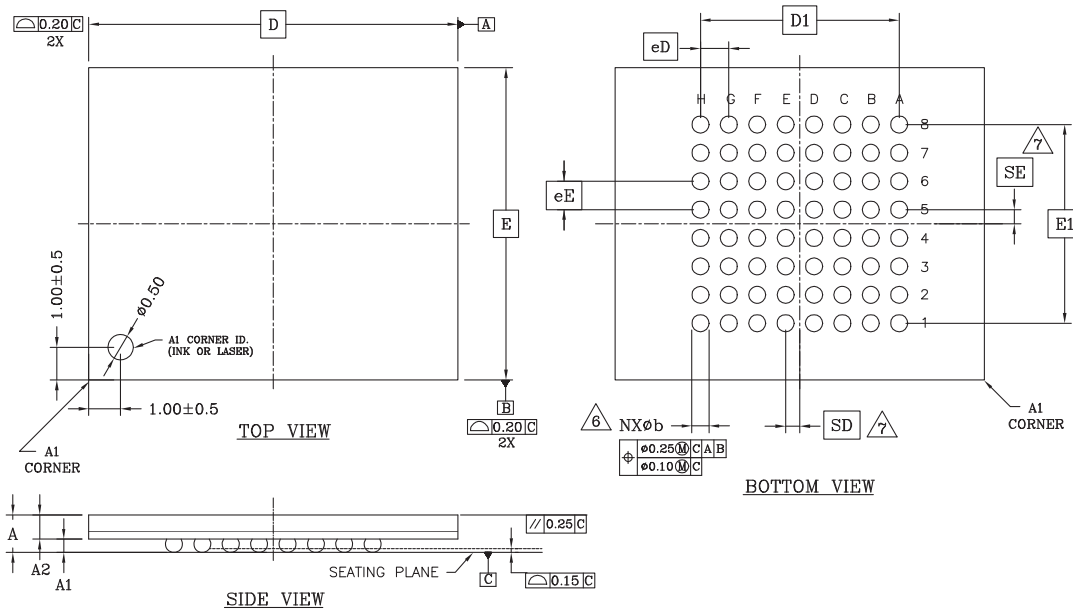
PACKAGE	LAE 064			
JEDEC	N/A			
9.00 mm x 9.00 mm PACKAGE				
SYMBOL	MIN	NOM	MAX	NOTE
A	---	---	1.40	PROFILE HEIGHT
A1	0.40	---	---	STANDOFF
A2	0.60	---	---	BODY THICKNESS
D	9.00 BSC.			BODY SIZE
E	9.00 BSC.			BODY SIZE
D1	7.00 BSC.			MATRIX FOOTPRINT
E1	7.00 BSC.			MATRIX FOOTPRINT
MD	8			MATRIX SIZE D DIRECTION
ME	8			MATRIX SIZE E DIRECTION
N	64			BALL COUNT
fb	0.50	0.60	0.70	BALL DIAMETER
eD	1.00 BSC.			BALL PITCH - D DIRECTION
eE	1.00 BSC.			BALL PITCH - E DIRECTION
SD / SE	0.50 BSC.			SOLDER BALL PLACEMENT
	NONE			DEPOPULATED SOLDER BALLS

NOTES:

- DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
- ALL DIMENSIONS ARE IN MILLIMETERS.
- BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 (EXCEPT AS NOTED).
- [e] REPRESENTS THE SOLDER BALL GRID PITCH.
- SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
N IS THE TOTAL NUMBER OF SOLDER BALLS.
- DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
- SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.
WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = $\frac{e}{2}$
- NOT USED.
- "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.

3623 \ 16-038.12 \ 1.16.07

11.2.3 物理图 — LAA064



PACKAGE	LAA 064			
JEDEC	N/A			
	13.00 mm x 11.00 mm PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	---	---	1.40	PROFILE HEIGHT
A1	0.40	---	---	STANDOFF
A2	0.60	---	---	BODY THICKNESS
D	13.00 BSC.			BODY SIZE
E	11.00 BSC.			BODY SIZE
D1	7.00 BSC.			MATRIX FOOTPRINT
E1	7.00 BSC.			MATRIX FOOTPRINT
MD	8			MATRIX SIZE D DIRECTION
ME	8			MATRIX SIZE E DIRECTION
N	64			BALL COUNT
φb	0.50	0.60	0.70	BALL DIAMETER
eD	1.00 BSC.			BALL PITCH - D DIRECTION
eE	1.00 BSC.			BALL PITCH - E DIRECTION
SD / SE	0.50 BSC.			SOLDER BALL PLACEMENT
	NONE			DEPOPULATED SOLDER BALLS

NOTES:

- DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
- ALL DIMENSIONS ARE IN MILLIMETERS.
- BALL POSITION DESIGNATION PER JESD 95-1, SPP-010 (EXCEPT AS NOTED).
- \square REPRESENTS THE SOLDER BALL GRID PITCH.
- SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
N IS THE TOTAL NUMBER OF SOLDER BALLS.
- \triangle DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
- \triangle SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.
WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = $\frac{e}{2}$
- NOT USED.
- "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.

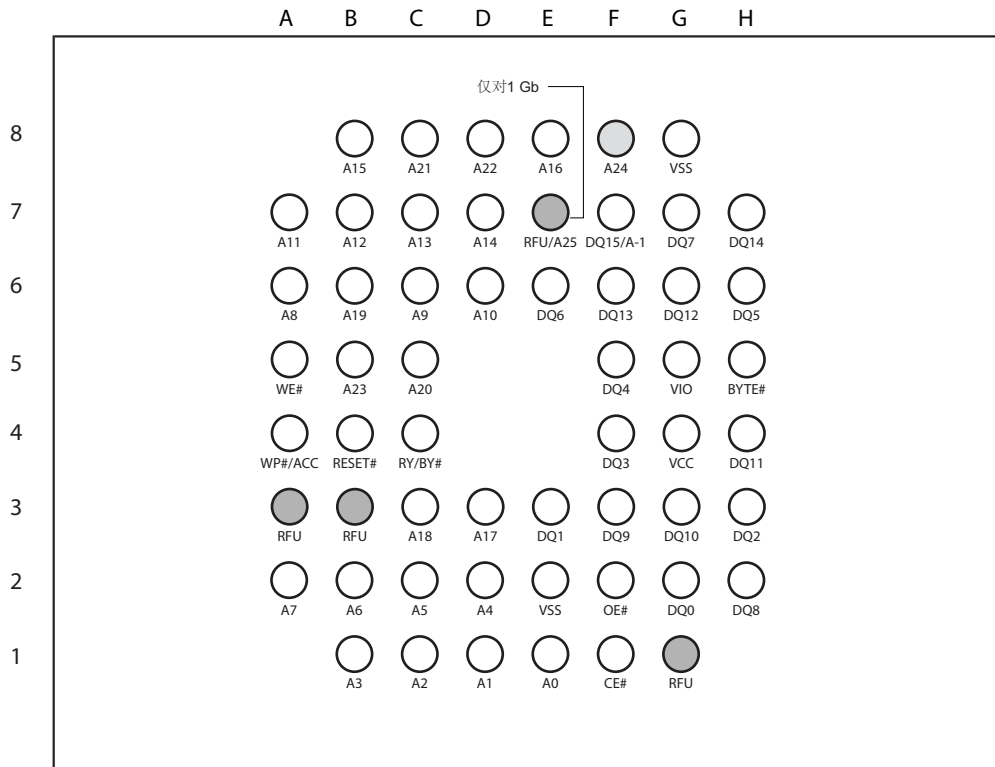
3354\16-038.12d

11.3 56 球 FBGA

11.3.1 连接图

图 11.5 56 球加固球栅阵列
顶视图

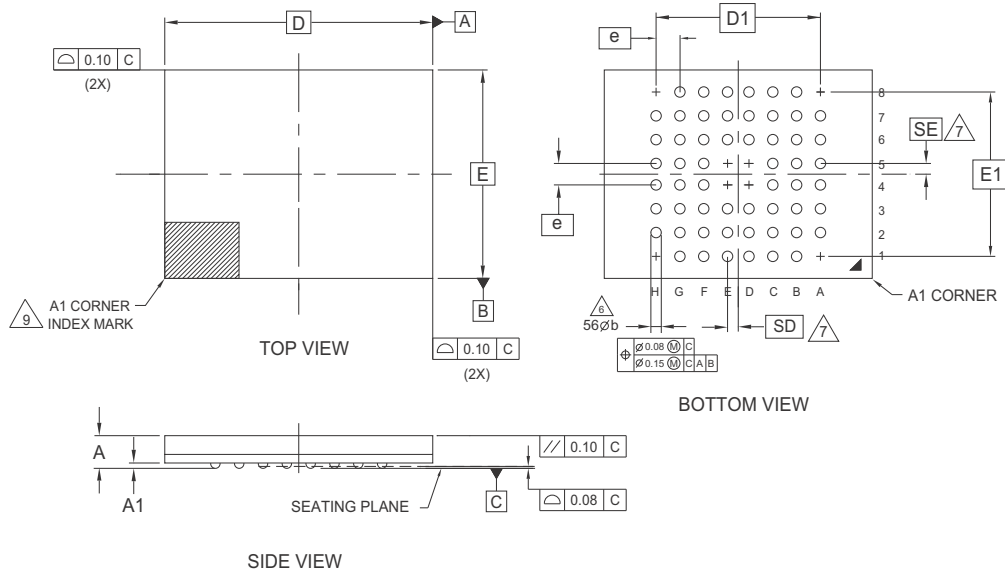
产品引脚分布



注意:

1. 球 A3、B3 和 G1 是保留给未来使用 (RFU)。

11.3.2 物理图 — VBU 056



PACKAGE	VBU 056			
JEDEC	N/A			
	9.00 mm x 7.00 mm NOM PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	---	---	1.00	OVERALL THICKNESS
A1	0.17	---	---	BALL HEIGHT
D	9.00 BSC.			BODY SIZE
E	7.00 BSC.			BODY SIZE
D1	5.60 BSC.			BALL FOOTPRINT
E1	5.60 BSC.			BALL FOOTPRINT
MD	8			ROW MATRIX SIZE D DIRECTION
ME	8			ROW MATRIX SIZE E DIRECTION
N	56			TOTAL BALL COUNT
∅ b	0.35	0.40	0.45	BALL DIAMETER
e	0.80 BSC.			BALL PITCH
SD/SE	0.40 BSC.			SOLDER BALL PLACEMENT
	A1,A8,D4,D5,E4,E5,H1,H8			DEPOPULATED SOLDER BALLS

NOTES:

- DIMENSIONING AND TOLERANCING METHODS PER ASME Y14.5M-1994.
- ALL DIMENSIONS ARE IN MILLIMETERS.
- BALL POSITION DESIGNATION PER JEP95, SECTION 4.3, SPP-010.
- [e] REPRESENTS THE SOLDER BALL GRID PITCH.
- SYMBOL "MD" IS THE BALL MATRIX SIZE IN THE "D" DIRECTION.
SYMBOL "ME" IS THE BALL MATRIX SIZE IN THE "E" DIRECTION.
N IS THE TOTAL NUMBER OF POPULATED SOLDER BALL POSITIONS FOR MATRIX SIZE MD X ME.
- DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
- SD AND SE ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW SD OR SE = 0.000.
WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, SD OR SE = [e/2]
- "*" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
- A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.
- OUTLINE AND DIMENSIONS PER CUSTOMER REQUIREMENT.

g10591 16-038 25 1 01.26.12

12. 关于 FBGA 封装的特殊处置说明

FBGA 封装的闪存产品需要特殊处置。

如果使用超声波清洁方法，FBGA 封装的闪存器件可能被损坏。如果封装体长时间放置在温度高于 150°C 的地方，封装和 / 或数据完整性可能受损。

13. 订购信息

有效组合

推荐组合表列出了计划大量供应的配置。下表会随着新组合的推出而更新。如要确认特定组合的可用性并了解最新推出的组合，请咨询您当地的销售代表。

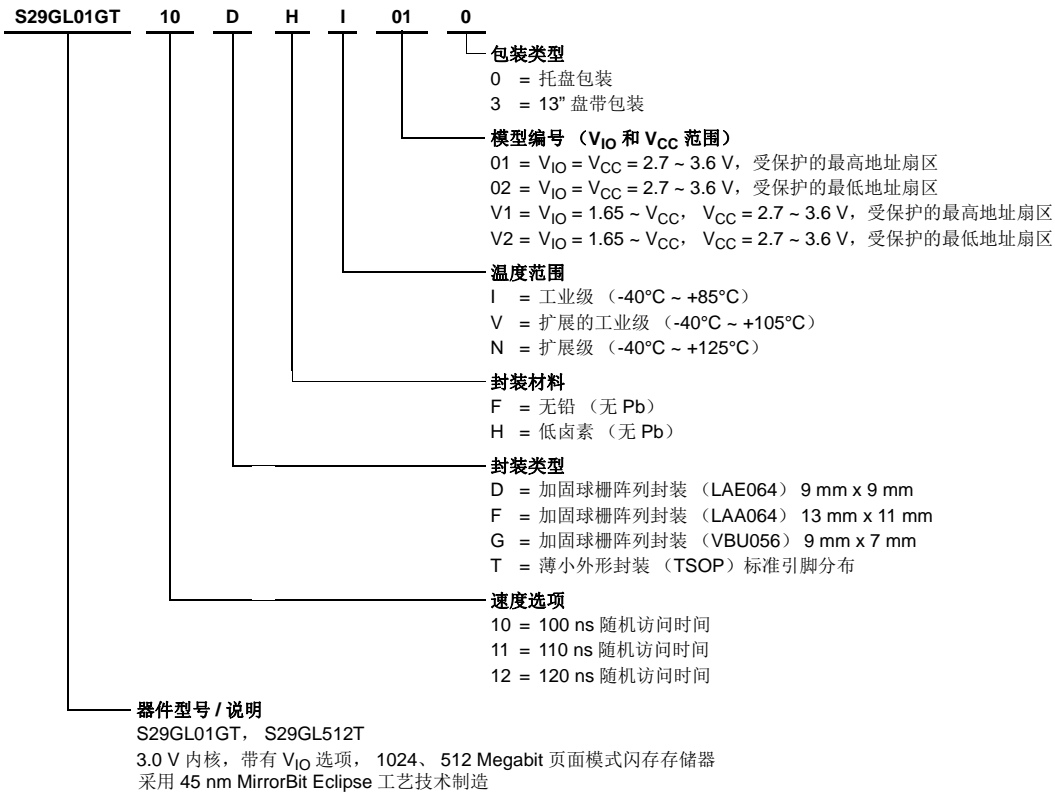
表 13.1 S29GL-T 有效组合

S29GL-T 有效组合					
基极 OPN	速度 (ns)	封装和温度 (1)	型号	封装类型 (2)	订购器件型号 (yy = 模型编号, x = 包装类型)
S29GL01GT	100	DHI, FHI, GHI, TFI	01, 02	0, 3	S29GL01GT10DHlyyx S29GL01GT10FHlyyx S29GL01GT10GHlyyx S29GL01GT10TFlyyx
	110	DHI, FHI, TFI	V1, V2		S29GL01GT11DHlyyx S29GL01GT11FHlyyx S29GL01GT11TFlyyx
	110	FHV, DHV, TFV	01, 02		S29GL01GT11DHVyyx S29GL01GT11TFVyyx S29GL01GT11FHVyyx
	120	DHV, TFV	V1, V2		S29GL01GT12DHVyyx S29GL01GT12TFVyyx
	120	DHN, TFN	01, 02		S29GL01GT12DHNyyxx S29GL01GT12TFNyyxx
	130	DHN, TFN	V1, V2		S29GL01GT13DHNyyxx S29GL01GT13TFNyyxx
S29GL512T	100	DHI, FHI, GHI, TFI	01, 02	0, 3	S29GL512T10DHlyyx S29GL512T10FHlyyx S29GL512T10GHlyyx S29GL512T10TFlyyx
	110	DHI, FHI, TFI	V1, V2		S29GL512T11DHlyyx S29GL512T11FHlyyx S29GL512T11TFlyyx
	110	DHV, TFV	01, 02		S29GL512T11DHVyyx S29GL512T11TFVyyx
	120	DHV, TFV	V1, V2		S29GL512T12DHVyyx S29GL512T12TFVyyx
	120	DHN, TFN	01, 02		S29GL512T12DHNyyxx S29GL512T12TFNyyxx
	130	DHN, TFN	V1, V2		S29GL512T13DHNyyxx S29GL512T13TFNyyxx

注意:

1. 更多速度、封装和温度选项会在以后提供。请咨询您当地的销售代表，以了解可用的选项。
2. 封装类型 0 是标准选项。

一般市场器件的订购器件型号是如下形式的有效组合：



14. 其他资源

www.spansion.com 提供下列相关文档：

14.1 软件链接

可从以下网站下载闪存器件的支持信息：

<http://www.spansion.com/Support/Pages/DriversSoftware.aspx>

- Spansion 底层驱动程序
- 增强型闪存驱动程序
- 闪存文件系统

从下面的网站下载仿真模型和 CAD 模型的支持信息：

<http://www.spansion.com/Support/Pages/SimulationModels.aspx>

- VHDL 和 Verilog
- IBIS
- ORCAD

14.2 应用笔记链接

下面列出了与该产品有关的应用笔记。可以从

<http://www.spansion.com/Support/TechnicalDocuments/Pages/ApplicationNotes.aspx> 网站上下载所有应用笔记。

- 通用闪存接口版本 1.5 供应商专用扩展
- 通用闪存接口规范
- 连接不使用的 MirrorBit 闪存数据线
- 开发系统级确认例程
- 闪存：概述
- 将 i.MX3x 连接到 S29GL MirrorBit NOR 闪存
- 将 S29GL-S 连接到 Freescale Coldfire 处理器
- 将 Spansion GL MirrorBit 系列连接到 Freescale i.MX31 处理器
- MirrorBit 闪存写入缓冲区编程和页缓冲区读取
- 优化编程 / 擦除次数
- 耐久性和数据保持时间的实践指南
- Spansion 65 nm GL-S Eclipse 系列架构的编程者指南
- MirrorBit 闪存的复位电压和时序要求
- 信号完整性和可靠的闪存操作
- 了解交流特性
- 了解负载电容和访问时间
- 了解页面模式闪存器件
- 使用 CFI 读取和调试系统
- 多用途 I/O: DQ 和增强型
- 平均磨损

14.3 规范公告

请联系您当地的销售办事处，以了解详细信息。

15. 文档修订记录

文档标题: S29GL01GT/S29GL512T, 1 Gbit (128 MByte)/512 Mbit (64 MByte), GL-T MirrorBit® Eclipse™ 闪存非易失性存储器系列				
文档编号: 002-03718				
版本	ECN 编号	变更者	提交日期	变更说明
**	4966571	YOQI	10/16/2015	本档版本号为 Rev. **, 译自英文版 002-00247 Rev. *C。
*A	5062011	YOQI	12/23/2015	本档版本号为 Rev. *A, 译自英文版 002-00247 Rev. *E。
*B	5376429	NFB	07/28/2016	本档版本号为 Rev. *B, 译自英文版 002-00247 Rev. *F。

销售、解决方案和法律信息

全球销售和设计支持

赛普拉斯公司具有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想找到离您最近的办事处，请访问赛普拉斯所在地。

产品

ARM® Cortex® 微控制器 cypress.com/arm
汽车级产品 cypress.com/automotive
时钟与缓冲器 cypress.com/clocks
接口 cypress.com/interface
照明与电源控制 cypress.com/powerpsoc
存储器 cypress.com/memory
PSoC cypress.com/psoc
触摸感应产品 cypress.com/touch
USB 控制器 cypress.com/USB
无线 / 射频 cypress.com/wireless

PSoC® 解决方案

psoc.cypress.com/solutions
PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

赛普拉斯开发者社区

[社区](#) | [论坛](#) | [博客](#) | [视频](#) | [培训](#)

技术支持

cypress.com/go/support

© 赛普拉斯半导体公司，2015-2016。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不会根据专利权或其他权利以明示或暗示方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯不保证产品能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途外，未经赛普拉斯明确的书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能因为发生功能异常和故障，而对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受赛普拉斯软件许可协议的限制。