

概述

赛普拉斯FL-L器件系列是非易失性闪存存储器产品，它们采用了以下两种技术：

- 浮栅技术
- 65 nm光刻技术

FL-L系列产品通过串行外设接口（SPI）连接主机系统。该产品支持传统SPI的一比特串行输入和输出（即单线I/O或SIO）、可选的两比特（即双线I/O或DIO）、四比特（四线I/O或QIO）以及四线外设接口（QPI）命令。另外，它还为QIO和QPI提供了双倍数据速率（DDR）命令支持，在时钟的双边沿上传送地址和读取数据。

该产品具有页编程缓冲区，允许在一个操作中最多编程256字节，并提供单独的4 KB大小扇区、32 KB大小半块、64 KB大小块或全芯片擦除能力。

FL-L系列器件支持更高的时钟频率和四线命令，因此指令读取的传输率能够等于或超过传统的并行接口、异步、NOR闪存存储器，同时可明显降低信号连接的计数。

FL-L系列产品为各种移动或嵌入式应用提供了大容量、灵活性高和速度快等性能。该产品系列为空间、信号连接，功耗有限的系统提供了一个理想的存储解决方案。这些存储器的灵活性和性能比普通串行闪存器件好很多。最适合将它们用于传送代码到RAM，直接执行代码（XIP）和存储可重新编程的数据。

特性

■ 具有多个I/O的串行外设接口（SPI）

- 支持时钟极性以及相位模式0和3
- 具有双倍数据速率（DDR）选项
- 具有四线外设接口（QPI）选项
- 具有扩展地址：24或32位地址选项
- 串行指令子集和封装兼容S25FL-A、S25FL1-K、S25FL-P、S25FL-S以及S25FS-S的SPI系列
- 多个I/O指令子集和封装兼容S25FL-A、S25FL1-K、S25FL-P、S25FL-S以及S25FS-S的SPI系列

■ 读取

- 指令：普通、快速、双线I/O、四线I/O、双线输出、四线输出、DDR四线I/O。
- 模式：回卷突发、持续突发（XIP）、QPI
- 串行闪存可发现参数（SFDP）用于配置信息

■ 编程结构

- 含有256字节页编程缓冲区的3.0 V FL-L闪存存储器
- 编程挂起和恢复

■ 擦除结构

- 统一的4KB扇区擦除
- 统一的32KB半块擦除
- 统一的64KB块擦除
- 芯片擦除
- 擦除挂起和恢复

■ 具有100,000次编程/擦除周期

■ 20年的数据保留时间

■ 安全特性

- 具有状态和配置寄存器保护

- 主闪存阵列外有四个安全区域，每个大小为256字节。

- 传统块保护：块范围

- 单独和区域保护

- 单独块锁定：易失性单独扇区/块保护

- 指针区域：非易失性扇区/块范围

- 安全区域2和3以及指针区域的电源锁定、密码和永久保护

■ 技术

- 65 nm浮栅技术

■ CMOS I/O的单个供电电压

- 2.7 V到3.6 V

■ 温度范围/级别

- 工业级（-40°C ~ +85°C）

- 扩展的工业级（-40°C ~ +105°C）

- 扩展范围（-40°C ~ +125°C）

- 汽车级，AEC-Q100等级3（-40°C ~ +85°C）

- 汽车级，AEC-Q100等级2（-40°C ~ +105°C）

- 汽车级，AEC-Q100等级1（-40°C ~ +125°C）

■ 封装类型（所有均无铅）

- 8-SOIC 208 mil（SOC008）— 仅S25FL128L提供

- WSON 5 × 6 mm（WND008）— 仅S25FL128L提供

- WSON 6 × 8 mm（WNG008）— 仅S25FL256L提供

- 16-SOIC 300 mil（SO3016）— 仅S25FL256L提供

- BGA-24 6 × 8 mm

- 5 × 5 ball（FAB024）封装

- 4 × 6 ball（FAC024）封装

性能摘要

最大读取速度SDR

命令	时钟速率 (MHz)	MBps
读取	50	6.25
快速读取	133	16.5
双线读取	133	33
四线读取	133	66

最大读取速度DDR

命令	时钟速率 (MHz)	MB/s
DDR四线读取	66	66

典型的编程和擦除速率

操作	KB/s
页编程	854
4 KB扇区擦除	80
32 KB半块擦除	168
64 KB块擦除	237

典型的电流消耗, -40°C ~ +85°C

操作	典型电流	单位
快速读取5 MHz	10	mA
快速读取10 MHz	10	mA
快速读取20 MHz	10	mA
快速读取50 MHz	15	mA
快速读取108 MHz	25	mA
快速读取133 MHz	30	mA
四线I/O / QPI读取108 MHz	25	mA
四线I/O / QPI读取133 MHz	30	mA
四线I/O / QPI DDR读取33 MHz	15	mA
四线I/O / QPI DDR读取66 MHz	30	mA
编程	40	mA
擦除	40	mA
待机SPI	20	μA
待机QPI	60	μA
深度掉电	2	μA

目录

1. 产品概述	4	8.7 安全区域阵列命令	98
1.1 移植时注意事项.....	4	8.8 单块锁定命令	100
2. 连接图	5	8.9 指针区域命令	104
2.1 SOIC 16引脚.....	5	8.10 单独和区域保护 (IRP) 命令	105
2.2 包含8个连接器的封装.....	5	8.11 复位命令	110
2.3 BGA球型焊盘封装.....	6	8.12 深度掉电命令	111
2.4 FBGA封装的特殊处理指令说明.....	7	9. 数据完整性	114
3. 信号说明	8	9.1 擦除次数.....	114
3.1 输入/输出概要.....	8	9.2 数据保持时间.....	114
3.2 多路输入/输出 (MIO).....	9	10. 软件接口参考	115
3.3 串行时钟 (SCK).....	9	10.1 JEDEC JESD216B串行闪存可发现参数.....	115
3.4 芯片选择 (CS#).....	9	10.2 器件ID地址映射.....	122
3.5 串行输入 (SI) /IO0.....	9	10.3 初始供应状态.....	122
3.6 串行输出 (SO) /IO1.....	9	11. 电气规范	123
3.7 写保护 (WP#) /IO2.....	9	11.1 绝对最大额定值.....	123
3.8 IO3/RESET#.....	9	11.2 闭锁特性.....	123
3.9 RESET#.....	10	11.3 热阻.....	123
3.10 电源电压 (V _{DD}).....	10	11.4 工作范围.....	124
3.11 电源和信号接地 (V _{SS}).....	10	11.5 上电和下电.....	125
3.12 无连接 (NC).....	10	11.6 直流特性.....	127
3.13 保留供将来使用 (RFU).....	10	12. 时序规范	130
3.14 请勿使用 (DNU).....	10	12.1 切换波形的关键.....	130
4. 框图	11	12.2 交流测试条件.....	130
4.1 系统框图.....	11	12.3 复位.....	131
5. 信号协议	14	12.4 SDR交流特性.....	134
5.1 SPI时钟模式.....	14	12.5 DDR交流特性.....	137
5.2 命令协议.....	15	12.6 嵌入式算法性能表.....	139
5.3 接口状态.....	20	13. 订购信息	140
5.4 数据保护.....	24	13.1 订购器件型号.....	140
6. 地址空间映射	25	14. 物理框图	143
6.1 概述.....	25	14.1 SOIC 16引脚 (300 mil) 封装 (SO3016).....	143
6.2 闪存存储器阵列.....	25	14.2 8引脚SOIC (208 mil) 封装 (SOC008).....	144
6.3 ID地址空间.....	26	14.3 无引脚并包含8个连接器的WSON, 其大小为5 x 6 mm (WND008).....	145
6.4 JEDEC JESD216串行闪存可发现参数 (SFDP) 空间.....	26	14.4 无引脚并包含8个连接器的WSON, 其大小为6 x 8 mm (WNG008).....	146
6.5 安全区域地址空间.....	26	14.5 球栅阵列24球型焊盘, 其大小为6 x 8 mm (FAB024).....	147
6.6 寄存器.....	27	14.6 球栅阵列24球型焊盘, 其大小为6 x 8 mm (FAC024).....	148
7. 数据保护	43	15. 其他资源	149
7.1 安全区域.....	43	15.1 术语表.....	149
7.2 深度掉电.....	43	15.2 赛普拉斯产品路线图链接.....	150
7.3 写入使能指令.....	44	15.3 软件链接.....	150
7.4 写入保护信号.....	45	15.4 应用笔记链接.....	150
7.5 状态寄存器保护 (SRP1、SRP0).....	45	13. 文档修订记录	151
7.6 阵列保护.....	46		
7.7 单独和区域保护.....	52		
8. 命令	58		
8.1 命令集汇总.....	58		
8.2 标识命令.....	64		
8.3 寄存器访问命令.....	67		
8.4 读取闪存阵列命令.....	80		
8.5 编程闪存阵列命令.....	89		
8.6 擦除闪存阵列命令.....	91		

1. 产品概述

1.1 移植时注意事项

1.1.1 特性比较

FL-L系列的指令集和封装与先前的FL-S、FL1-K和FL-P系列相兼容。

表 1.1 赛普拉斯 SPI 系列比较

参数	FL-L	FL-S	FL1-K	FL-P
技术节点	65 nm	65 nm	90 nm	90 nm
架构	浮栅	MirrorBit® Eclipse™	浮栅	MirrorBit®
出厂日期		量产中	量产中	量产中
大小	256 Mb	128 Mb - 1 Gb	4 Mb - 64 Mb	32 Mb - 256 Mb
总线宽度	x1、x2、x4	x1、x2、x4	x1、x2、x4	x1、x2、x4
供电电压	2.7 V - 3.6 V	2.7 V - 3.6 V / 1.65 V - 3.6 V V _{IO}	2.7 V - 3.6 V	2.7 V - 3.6 V
正常读取速度	6 MB/s (50 MHz)	6 MB/s (50 MHz)	6 MB/s (50 MHz)	5 MB/s (40 MHz)
快速读取速度	16.5 MB/s (133 MHz)	17 MB/s (133 MHz)	13 MB/s (108 MHz)	13 MB/s (104 MHz)
双线读取速度	33 MB/s (133 MHz)	26 MB/s (104 MHz)	26 MB/s (108 MHz)	20 MB/s (80 MHz)
四线读取速度	66 MB/s (133 MHz)	52 MB/s (104 MHz)	52 MB/s (108 MHz)	40 MB/s (80 MHz)
四线读取速度 (DDR)	66 MB/s (66 MHz)	80 MB/s (80 MHz)	-	-
编程缓冲区大小	256 B	256 B/512 B	256 B	256 B
擦除扇区/块大小	4 KB/32 KB/64 KB	64 KB/256 KB	4 KB/64 KB	64 KB/256 KB
参数扇区大小	-	4 KB (可选)	-	4 KB
扇区/块擦除速率 (典型值)	80 KB/s (4 KB) 168 KB/s (32 KB) 237 KB/s (64 KB)	500 KB/s	136 KB/s (4 KB) 437 KB/s (64 KB)	130 KB/s
页编程速率 (典型值)	854 KB/s (256 B)	1.2 MB/s (256 B) 1.5 MB/s (512 B)	365 KB/s	170 KB/s
安全区域/OTP	1024 B	1024 B	768 B (3 × 256 B)	506 B
单独和区域保护或高级的扇区保护	有	有	无	无
擦除挂起/恢复	有	支持	有	无
编程挂起/恢复	有	支持	有	无
工作温度范围	-40°C到+85°C -40°C到+105°C -40°C到+125°C	-40°C到+85°C -40°C到+105°C	-40°C到+85°C	-40°C到+85°C -40°C到+105°C

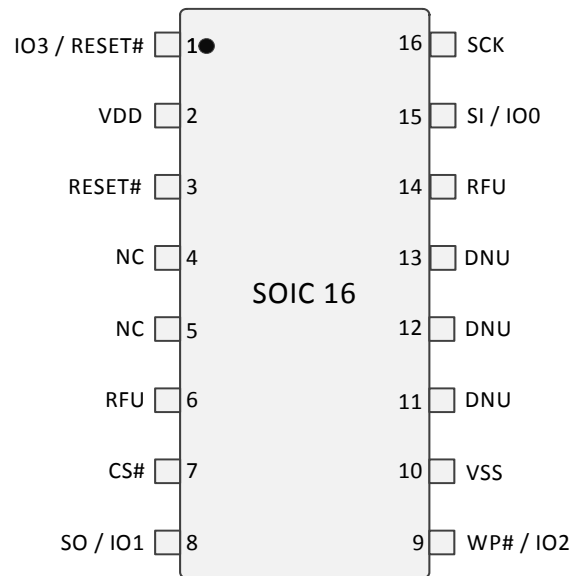
注意:

1. 欲了解详细信息, 请参考单独的数据手册。

2. 连接图

2.1 SOIC 16引脚

图2.1 16引脚SOIC封装（SO3016），顶视图



注意:

1. RESET#和IO3/RESET#输入间连接着一个内部上拉电阻。如果未使用四线模式或进行硬件复位，那么系统中的这些输入会保持为未连接状态。

2.2 包含8个连接器的封装

图2.2 8引脚塑料小外型封装（SOIC8）

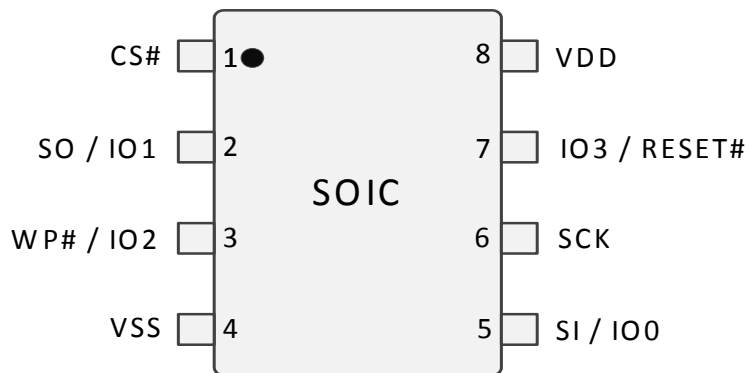
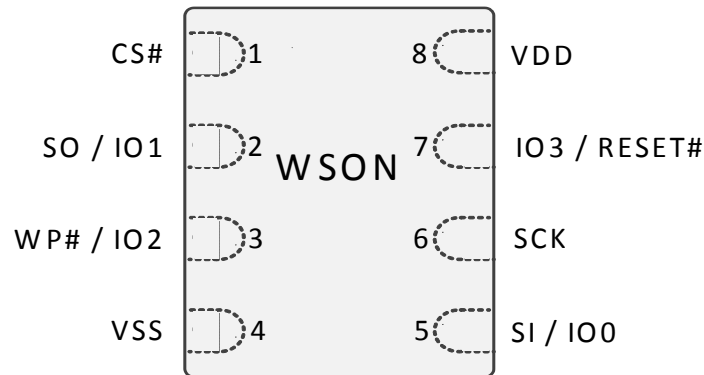


图2.3 包含8个管脚的封装（WSO6 6x8）（WSO6 5x6），顶视图

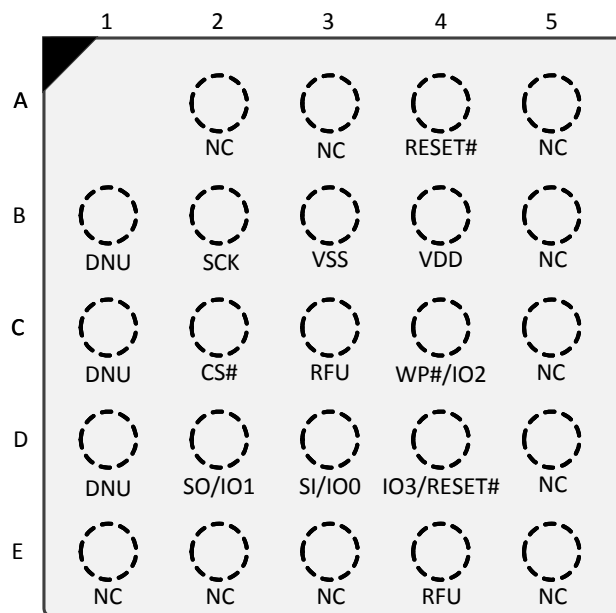


注意:

1. RESET#输入端连接着一个内部上拉电阻。如果未使用四线模式或进行硬件复位，那么系统中该输入会保持为未连接状态。

2.3 BGA球型焊盘封装

图2.4 24-BGA，5x5球型焊盘封装（FAB024），顶视图

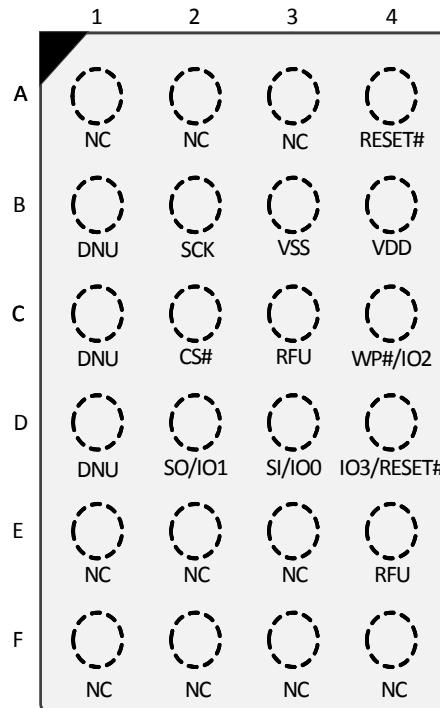


注意:

1. 信号连接位于FAC024 BGA相对应的位置，因此单PCB封装可以使用两种封装中的某一种。

2. RESET#输入端连接着一个内部上拉电阻。如果未使用四线模式或进行硬件复位，那么系统中该输入会保持为未连接状态。

图2.5 24-Ball BGA, 4x6球型焊盘封装 (FAC024), 顶视图



注意:

1. RESET#输入上连接着一个内部上拉电阻。如果未使用四线模式或进行硬件复位, 那么系统中的该输入会保持为未连接状态。

2.4 FBGA封装的特殊处理指令说明

如果使用超声波清洁方法, BGA封装中的闪存存储器器件可能被损坏。如果将封装长期放置在温度高于150°C的地方, 封装和/或数据的完整性可能受到损坏。

3. 信号说明

支持多输入/输出的串行外设接口（SPI-MIO）

很多具有独立的并行控制、地址和数据信号的存储器件与其主机系统连接时，需要大量的信号连接和更大的封装。由于大量信号交换，因此信号连接的数量越多，功耗也越高；较大的封装尺寸会增加成本。

FL-L系列通过6个信号将所有控制、地址和数据信息串行传输给主机系统，因此可以减少信号连接的数量。这样可以降低存储器封装的成本和信号交换的功耗，减少主机信号连接的数量，主机可以节省连接以供其他功能使用。

FL-L系列使用了业界标准的单比特SPI，同时支持可选的扩展命令，用于进行两比特（双线）或4比特（四线）宽的串行传输。这种多宽度接口被称为SPI Multi-I/O或SPI-MIO。

3.1 输入/输出概要

表 3.1 信号表

信号名称	类型	说明
RESET#	输入	硬件复位 ：该信号为低电平时，器件复位并返回待机状态，准备接收命令。该信号具有一个内部上拉电阻；在主机系统中如果不使用该信号，可以保持它为未连接状态。
SCK	输入	串行时钟
CS#	输入	芯片选择
SI/I00	I/O	单比特数据命令的 串行输入 或双线/四线命令的I00。
SO/I01	I/O	单比特数据命令的 串行输出 。双线或四线命令的I01。
WP#/I02	I/O	不在四线模式（CR1V[1] = 0和SR1NV[7] = 1）下时，为 写保护 。 在四线模式（CR1V[1] = 1）下，为 I02 。 该信号具有一个内部上拉电阻；在主机系统中，如果不使用四线模式或不用于写保护，可以保持未连接状态。如果通过设置SR1NV[7] = 1和CR1V[1] = 0使能了写保护功能，那么在执行WRR或WRAR命令期间，主机系统需要将WP#驱动为高电平或低电平。
I03/RESET#	I/O	在四线I/O模式下，当配置寄存器1的QUAD位CR1V[1] = 1时，或在QPI模式下，当CS#为低电平并且配置好寄存器2的QPI位CR2V[3] = 1时，该信号将为 I03 。 当CR2V[7] = 1，并且处于非四线I/O模式（CR1V[1] = 0）时，或在四线模式下（CR1V[1] = 1）CS#为高电平时，该信号为 RESET# 。 该信号具有一个内部上拉电阻；在主机系统如果不使用四线模式或不用于RESET#，可以保持未连接状态。
V _{DD}	电源	供电电源 。
V _{SS}	电源	接地电源 。
NC	未使用	未连接 。没有任何器件内部信号连接到封装连接器，将来也不计划用于信号连接。该连接可以安全地用于印刷电路板（PCB）上的信号路由空间。然而，连接到NC的任何信号电压电平不能大于V _{DD} 。
RFU	保留	保留供将来使用 。当前没有任何器件内部信号连接到封装连接器，但是将来可能使用。不建议将RFU连接到PCB电路板上的任何信号，以便PCB可以在尺寸兼容的器件中实现未来的增强型特性。
DNU	保留	请勿使用 。器件的内部信号可能被连接到封装连接器。赛普拉斯可能使用该连接进行测试或其他用途，不应使该信号与任何主机系统信号连接。DNU信号处于低电平（V _{IL} ）时，与该信号有关的所有功能均无效。该信号使用了一个内部下拉电阻，在主机系统中可以保持为未连接状态或被连接到V _{SS} 。请勿将这些连接用于PCB信号路由通道。请勿将主机系统的任何信号连接到该信号。

注意：

1. 使用内部上拉或下拉电阻的输入驱动电流低于2 μA。只有在上电过程中，该电流才最大（在4 μS内为150 mA）。如果采用普通的工艺技术、Vcc = 3.3 V和温度 = -40°C，则上拉或下拉电阻的阻值为~4.5 MΩ；温度 = 90°C时，该值为~6.6 MΩ。

3.2 多路输入/输出 (MIO)

传统SPI单比特命令（即单IO或SIO）只能通过串行输入（SI）信号将主机的数据发送到存储器内。数据会通过串行输出（SO）信号从存储器内串行回送到主机。

双线或四线输入/输出（I/O）命令只能通过SI/IO0信号将指令发送给存储器。地址或数据可按两比特一组通过IO0和IO1从主机发送到存储器内，也可以按照四比特（半字节）一组通过IO0、IO1、IO2和IO3发送。数据同样可以按照两比特一组通过IO0和IO1从存储器回送给主机，也可以按照四比特（半字节）一组通过IO0、IO1、IO2和IO3被回送。

QPI模式允许以四比特（半字节）一组的方式将所有指令、地址和数据通过IO0、IO1、IO2和IO3从主机传送到存储器内。数据同样可以按四比特（半字节）一组的方式通过IO0、IO1、IO2和IO3回送到主机内。

3.3 串行时钟 (SCK)

该输入信号为SPI接口提供了同步参考。可以在SCK信号的上升沿上锁存指令、地址或数据。在SDR命令中，数据输出会在SCK的下降沿后发生变化。

3.4 芯片选择 (CS#)

芯片选择信号指出某条命令正在向器件或从器件传输信息，并指示存储器件的相关信号。

当CS#信号为高逻辑状态时，不会选择芯片，所有输入信号均被忽略，并且所有输出信号均为高阻抗。除非正在进行某个内部嵌入式操作，否则器件会处于待机模式。嵌入式操作从状态寄存器1的正在写入位（SR1V[0]）被置为1开始，直到完成该操作为止。嵌入式操作的示例包括：编程、擦除或写入寄存器（WRR）等操作。

将CS#输入驱动为低逻辑状态可以启用器件，使器件进入工作模式。上电后，必须等到CS#的下降沿到来后才能执行其他命令。

3.5 串行输入 (SI) /IO0

该输入信号用于将数据串行传输给器件。它接收指令、地址和要写入闪存的数据。各个数值在串行SCK时钟信号的上升沿上被锁存。在执行双线和四线命令过程中，SI变成IO0（一个输入和输出），用于接收指令、地址和数据（数值在串行SCK时钟信号的上升沿上被锁存），并在SCK下降沿上（若执行SDR命令）或在任意一个边沿上（若执行DDR命令）移出数据。

3.6 串行输出 (SO) /IO1

该输出信号用于串行传输器件中的数据。在串行SCK时钟信号的下降沿上移出数据。在执行双线和四线命令过程中，SO变成IO1（一个输入和输出），用于接收地址和数据（数值在串行SCK时钟信号的上升沿上被锁存），并在SCK下降沿上（若执行SDR命令）或在任意边沿上（若执行DDR命令）移出数据。

3.7 写保护 (WP#) /IO2

将WP#驱动为低电平（ V_{IL} ）时，如果将状态寄存器1（SR1NV[7]）或（SR1V[7]）的状态寄存器保护位0（SRP0_NV）或（SRP0）设置为1，那么不能写入到状态寄存器、配置寄存器或DLR寄存器内。在这种情况下，会忽略选择SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的命令，并且不会报告任何错误。

这样可以防止传统的块保护设置发生改变。因此，在执行更改状态寄存器、配置寄存器或DLR寄存器（通过将SRP0_NV设置为1）的命令时，如果WP#为低电平，则不能改变存储器中受传统块保护功能保护的所有数据字节。同样，也不能编程安全区域锁定位（LB3-LB0）。

使能四线模式（CR1V[1] = 1）或QPI模式（CR2V[3] = 1）时，会禁用WP#的功能。当使能四线模式或QPI模式（CR2V[3] = 1）时，WP#将被IO2输入/输出替换，用于接收地址和数据（数值在串行SCK时钟信号的上升沿上被锁存），并在SCK下降沿上（若执行SDR命令）或在任意边沿上（若执行DDR命令）移出数据。

WP#具有一个内部上拉电阻；未连接时，WP#为高电平（ V_{IH} ），如果四线模式、QPI模式或保护功能未使用该引脚，则在主机系统中它可以被保持为不连接状态。

3.8 IO3/RESET#

当使能四线模式（CR1V[1] = 1）或QPI模式（CR2V[3] = 1）时，会将IO3作为输入/输出使用，用于接收地址和数据（数值在串行SCK时钟信号的上升沿上被锁存），并在SCK下降沿上（若执行SDR命令）或在任意边沿上（若执行DDR命令）移出数据。

通过写入配置寄存器2的非易失性位7 (CR2NV[7] = 1) 来启用IO3/RESET#功能时, IO3/RESET#也可用于初始化硬件复位功能。如果器件并非处于四线模式 (114、144、444), 即CR1V[1]=0; CS#为高电平时, 则只能将该输入作为RESET#使用。如果使能了四线模式 (CR1V[1] = 1) 或QPI模式 (CR2V[3] = 1), 或通过将CS#设置为低电平来选择器件, 则只能将IO3/RESET#作为IO3传送信息。CS#为高电平时, IO3/RESET#不能传送信息, 只能将它作为复位输入使用。在四线模式 (114、144、444) 下, 如果CS#为高电平时, 则复位功能在这些模式下仍可用。

当系统进入复位状态时, 必须将CS#信号驱动为高电平 (作为复位过程的一部分), 并将IO3/RESET#信号驱动为低电平。CS#变为高电平时, IO3/RESET#输入便从IO3切换为复位输入。当CS#保持为高电平和IO3/RESET#信号保持为低电平 (t_{RP}) 时, 会检测到复位条件。如果不需要进行复位操作, 则系统必须在向存储器发送数据的操作结束时将IO3/RESET#和CS#驱动为高电平。将数据发送给主机系统后, 存储器会驱动IO3为高电平, 使之保持高电平 t_{CS} 的时长。这样可确保IO3/RESET#不进入悬空状态, 也不会被内部或外部被动上拉电阻慢慢地上拉到高电平。因此, t_{RP} 时间结束前IO3/RESET#不会被视为高电平, 从而也不会触发意外的复位事件。

CR2V[7] = 0时, 将禁用IO3/RESET#输入的复位功能。

IO3/RESET#输入具有一个内部上拉电阻; 如果四线模式或复位功能未使用该输入, 则在主机系统中它可以保持为不连接状态。如果主机系统将IO3/RESET#信号驱动为高电平, 然后停止驱动该信号, 则通过使用内部上拉电阻将它保持为高电平。

请注意, 如果多个SPI-MIO存储器中有任何一个工作于四线I/O模式, 那么它们不能共享IO3/RESET#输入, 因为来自一个选定存储器或向其发送的IO3可用作第二个非选定存储器 (其共享了同一个IO3/RESET#信号) 的复位信号。

3.9 RESET#

RESET#输入提供了一种用于硬件复位的方法, 通过将器件返回到待机状态可准备接收命令。如果将RESET#驱动为逻辑低 (V_{IL}) 的时间不小于 t_{RP} 时长, 则器件将启动硬件复位过程。

给器件上电并经过 t_{PU} 时间后, RESET#也会导致相同的初始化过程。

此外, 可随时将RESET#置为低电平。为确保数据的完整性, 一旦器件准备好接受某个命令序列时, 应重新对由硬件复位中断的操作进行初始化。

RESET#具有一个内部上拉电阻; 如果不使用该引脚, 则在主机系统中它可以保持为不连接状态。如果主机系统将复位信号驱动为高电平有效, 然后停止驱动该信号, 则通过使用内部上拉电阻将该信号保持为高电平。

RESET#输入在所有封装中均不可用。器件的RESET#输入不可用时, 它将被锁定为非活动状态。

3.10 电源电压 (V_{DD})

V_{DD} 是所有器件内部逻辑的电源电压。这种单个电源电压为器件的所有内部功能 (包括读取、编程和擦除操作) 供电。

3.11 电源和信号接地 (V_{SS})

V_{SS} 是器件内核、输入信号接收器和输出驱动器的公共电压开漏和接地参考。

3.12 无连接 (NC)

没有任何器件内部信号被连接到封装连接器, 将来也不计划用于信号连接。该连接可能安全地用作印刷电路板 (PCB) 上的信号路由空间。

3.13 保留供将来使用 (RFU)

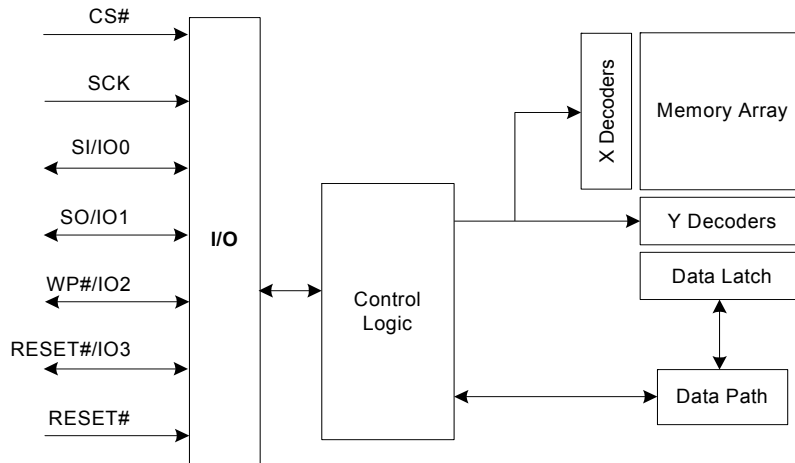
当前没有任何器件内部信号被连接到封装连接器, 但该连接器可能留给将来使用的。不建议将RFU连接到PCB电路板上的任何信号, 以便PCB可以在尺寸兼容的器件中实现未来的增强型特性。

3.14 请勿使用 (DNU)

器件的内部信号被连接到封装连接器。赛普拉斯可能使用该连接进行测试或其他用途, 不应使该信号与任何主机系统信号连接。DNU信号处于低电平 (V_{IL}) 时, 与该信号有关的所有功能均无效。该信号使用了一个内部下拉电阻, 在主机系统中可以保持为未连接状态或被连接到 V_{SS} 。请勿将该DNU信号连接到PCB板上的任意信号。请勿将主机系统的任何信号连接到该信号。

4. 框图

图4.1 逻辑框图



4.1 系统框图

图4.2 SPI总线上的总线主设备和存储器件之间的关系 — 单比特数据路径

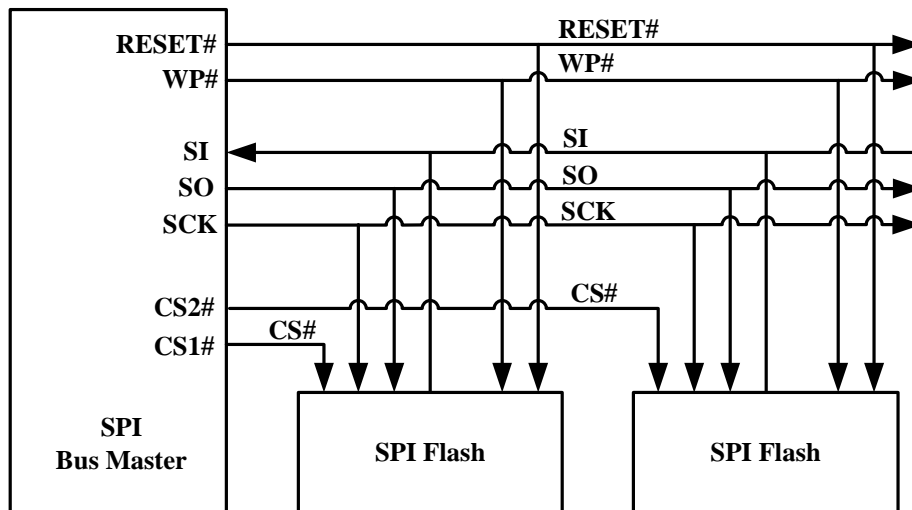


图4.3 SPI总线上的总线主设备和存储器件之间的关系 — 双比特数据路径

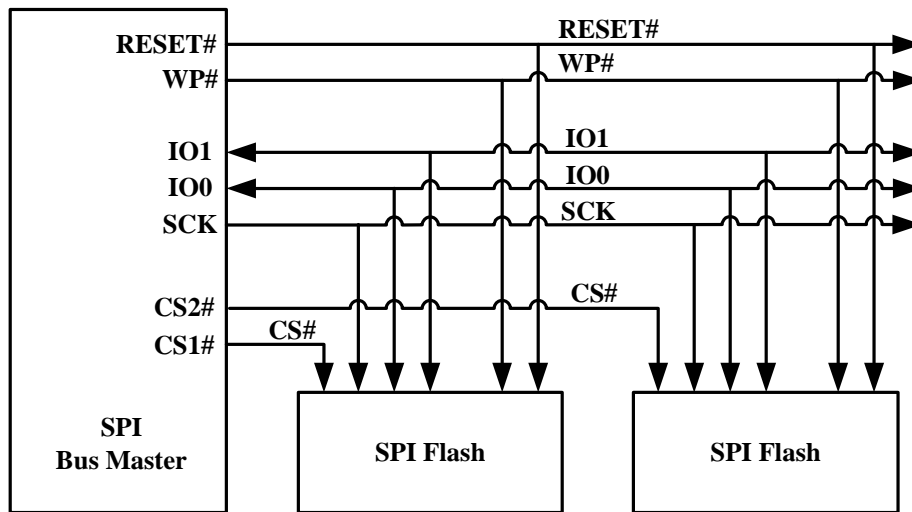


图4.4 SPI总线上的总线主设备和存储器件之间的关系 — 四比特数据路径 — 单独RESET#

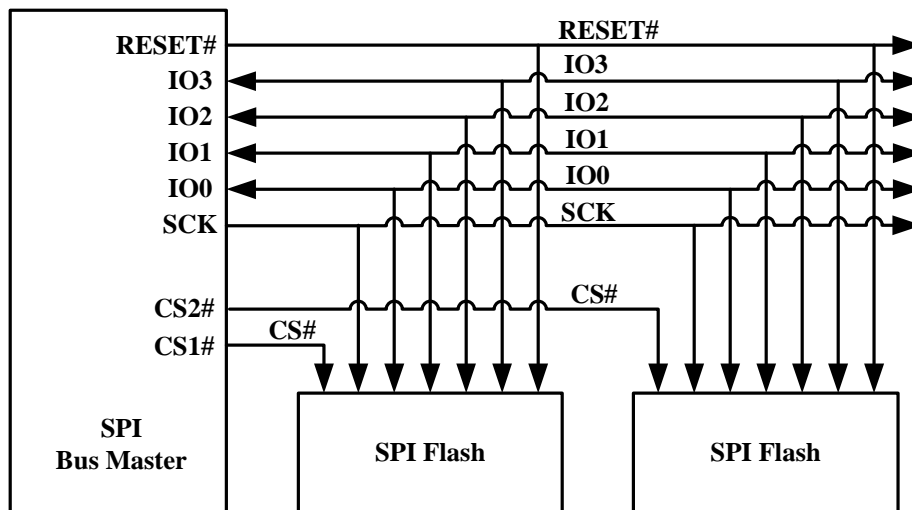
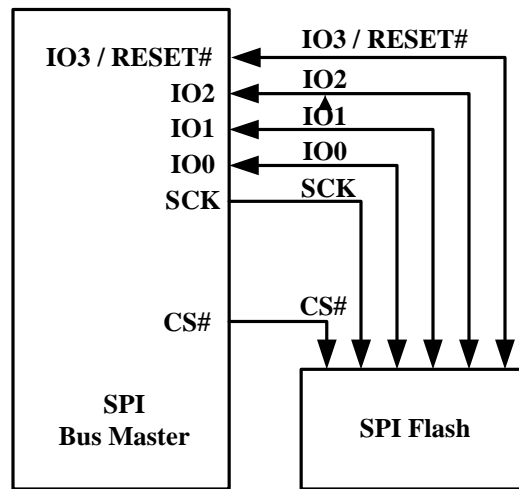


图4.5 SPI总线上的总线主设备和存储器件之间的关系 — 四比特数据路径 — I/O3_RESET#



5. 信号协议

5.1 SPI时钟模式

5.1.1 单倍数据速率 (SDR)

通过嵌入式微控制器（总线主设备）可以将FL-L系列驱动到以下任意一种时钟模式。

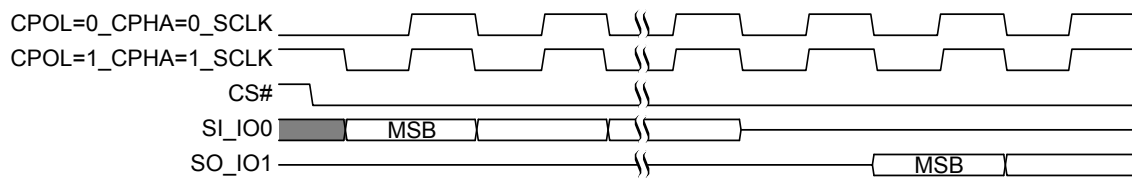
- **模式0:** 时钟极性 (CPOL) = 0、时钟相位 (CPHA) = 0
- **模式3:** CPOL = 1、CPHA = 1

在这两种模式中，器件的输入数据总是在SCK信号的上升沿上被锁存，输出数据总是在SCK时钟信号的下降沿上可用。

两种模式间的区别主要是：当总线主设备处于待机模式并且不传输数据时的时钟极性。

- CPOL = 0、CPHA = 0时，SCK将处于逻辑低状态
- CPOL = 1、CPHA = 1时，SCK将处于逻辑高状态

图5.1 支持SPI SDR模式



整个文档其余部分的时序图通常都是通过显示SCK在CS#下降沿时的高和低电平来显示模式0和模式3的。在某些情况下，时序图可能通过显示SCK在CS#下降沿时的低电平来显示唯一的模式0。而唯一的模式3的时序图只是表示在CS#的下降沿上时钟信号为高电平，因此模式3不需要从CS#下降沿后到SCK上升沿前的建立和保持时间。

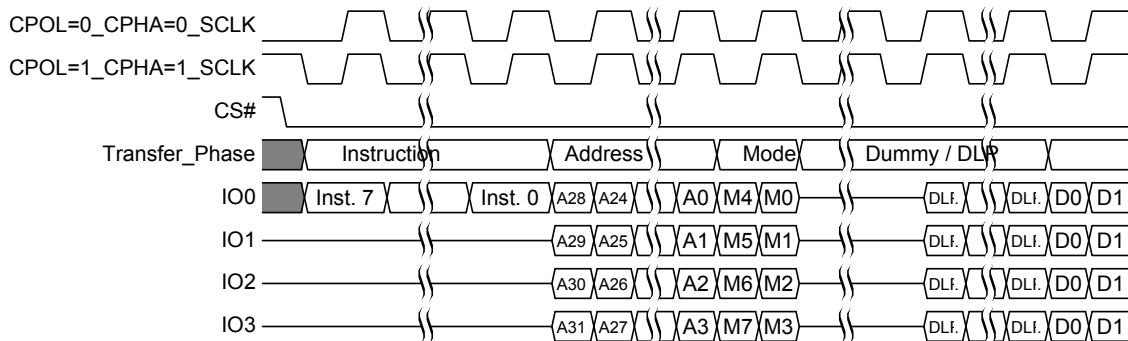
SCK周期是从一个SCK下降沿到下一个SCK下降沿进行测量（计数）得到的。在模式0下，由于开始执行某个命令时SCK已经处于低电平状态，因此执行命令时的第一个SCK周期指的是从CS#的下降沿到第一个SCK下降沿。

5.1.2 双倍数据速率 (DDR)

DDR命令也支持模式0和模式3。与SDR命令相同，在DDR命令中，指令位也是在时钟的上升沿上被锁存的。但是指令后的地址和输入数据则在SCK的双边沿上被锁存。最后指令位在SCK的下降沿上被锁存之后，第一个地址位会在随后的第一个SCK上升沿上被锁存。第一个输出数据位则在最后访问延迟（虚拟）周期结束后的下降沿上被发送。

与SDR命令相同，SCK周期也是从一个SCK下降沿到下一个SCK下降沿进行测量（计数）得到的。在模式0下，由于开始执行某个命令时SCK已经处于低电平状态，因此执行命令时的第一个SCK周期指的是从CS#的下降沿到第一个SCK下降沿。

图5.2 支持SPI DDR模式



5.2 命令协议

主机系统和FL-L存储器件系列之间都是以单位（unit）形式（称为命令）进行通信的。请参考第58页上的8. 命令一节，了解所有命令的定义和详情。

所有命令都以一个8位指令开始，该指令用于选择信息传输类型或器件需要执行的操作。命令可能也具有一个地址、指令修饰符、延迟周期、传送到存储器，或者是从存储器传送的数据。主机系统和存储器件之间的所有指令、地址和数据信息都是连续传输的。

通过一个使用3个数字的数字命令法来划分命令协议，以便能参考三个命令阶段的发送宽度。

- 指令；
- 地址和指令修饰符（连续读取模式位）；
- 数据。

单比特宽的命令只通过SI信号依次传输指令和地址或数据。数据会通过SO信号从存储器串行回送到主机内。对于单比特宽指令、单比特宽地址和修饰符、单比特宽数据的结构，被称为1-1-1命令协议。

双线输出或四线输出命令从主机端通过SI（IO0）将地址和虚拟周期依次进行发送。数据可以按照两比特一组通过IO0和IO1从存储器回送到主机，或按照四比特（半字节）一组通过IO0、IO1、IO2和IO3回送。对于双线输出和四线输出命令，这种结构分别被称为1-1-2和1-1-4命令协议。

双线或四线输入/输出（I/O）命令会按照两比特一组通过IO0和IO1将主机的数据传输给存储器，或按照四比特（半字节）一组通过IO0、IO1、IO2和IO3从主机发送，然后传送虚拟周期。数据也可以按照同样的方式回送给主机。对于双线I/O和四线I/O命令，这种结构分别被称为1-2-2和1-4-4命令协议。

FL-L系列也支持QPI模式，在该模式下所有信息（包括指令、地址、修饰符和数据）都以4比特的宽度传输。这种形式被称为4-4-4命令协议。

命令的结构如下：

- 每个命令都从CS#的下降沿开始执行，并在CS#的上升沿结束。主机通过一个命令将芯片选择（CS#）信号驱动为低电平，从而能够选择器件。
- 串行时钟（SCK）标记了主机和存储器之间所传输的单个位或一组位。
- 每个命令都以一个8位（一个字节）的指令开始。该指令用于选择信息传输类型或需要执行的器件操作。该指令在SCK的上升沿上进行传送。但某些读命令被前一个读命令修改，因此该指令要从更早的命令中显示。这种形式被称为连续读取模式。器件处于连续读取模式时，由于该指令和用于初始化连续读取模式的读取命令相同，所以各个指令位不会在开始执行命令时被传输。在该模式下，命令会从读取地址开始。因此，连续读取模式会将处于相同类型的读取命令中的每个读命令内删除8个指令位。
- 该指令可以单独在设备的某些地址空间中选择一个位置，或者跟着地址位一起选择。该指令用于确定所使用的地址空间。地址可能是24位或32位（字节边界）的地址。地址会在SCK的上升沿（若执行SDR命令）或在SCK双边沿上（若执行DDR命令）得到传送。
- 在传统的SPI模式下，传输指令后所有传输宽度都由已发送的指令确定。以下表示的可能是其中一种传输类型：仅通过串行输入（SI）或串行输出（SO）信号传输单一比特；通过IO0和IO1信号每一（双线）传输以两比特形式进行发送；或通过IO0-IO3信号每一（四线）传输以4比特形式进行。在双线或四线组中，最低有效位是通过IO0信号发送的。其他更高有效位则按照权重大小通过编号更高的IO信号发送。单一位或并行位组则按照从最高有效位到最低有效位的顺序进行传输。

- 在QPI模式下，所有信息都按照4比特宽（四线传输）通过IO0-IO3信号进行传输。
- 双线和四线I/O读命令在发送地址后将发送一个指令修饰符（称为连续读取模式位），以指明下一个命令的类型与之前的命令相同，而它并不是明确的命令。这些模式位用于初始化或结束连续读取模式。因此，下一条命令只会提供新地址和模式位，并不提供指令字节。当某个命令序列中重复了相同的命令类型时，会减少发送每一个命令所需要的时间。模式位会在SCK的上升沿（若执行SDR命令）或在SCK双边沿上（若执行DDR命令）进行发送。
- 地址或模式位发送之后可以接着发送将被写入到存储器的数据，可以是一个读延迟周期（数据返回给主机前）。
- 写入数据位传输会在SCK的上升沿（若执行SDR命令）或在SCK双边沿上（若执行DDR命令）锁存。
- SCK会在读访问的延迟周期内继续进行切换。该延迟（又称为虚拟周期）可能为0到几个SCK周期。最后的读取延迟周期结束后，第一次读取的数据位会在SCK下降沿上被输出。第一次读取的数据位被认为是在下一个SCK下降沿上传输给主机的。将在SCK的下一个上升沿（若执行SDR命令）或在SCK的任一沿上（若执行DDR命令）进行下一次传输操作。
- 如果命令将读数据返回给主机，则存储器件会继续发送数据，直到主机将CS#信号驱动为高电平为止。结束读取数据序列中任一个传输操作后，都可以将CS#信号驱动为高电平。这样会终止命令。
- 命令被终止时，不会再返回数据，此时主机可将CS#输入信号驱动为高电平。传输单独指令的第八位或最后写数据字节的第八位后，CS#信号必须变为高电平。这样，当位数量在CS#被驱动为低电平后是8位的整数倍时，必须将CS#信号驱动为高电平。如果CS#信号在指令或写入数据的8位边界后没有变为高电平，该命令将被拒绝，并且不能执行该命令。
- 所有指令、地址和模式位均被移入到器件内，并且先发送最高有效位（MSB）。数据位都被移入到器件，或从器件移出，并且要先传输MSB。所有数据都以字节为单位进行传输，并且先发送最低地址字节。随后的数据字节按照从最低到最高字节地址的顺序（即字节地址递增）进行发送。
- 在编程、擦除或写周期（嵌入式操作）中，尝试读取闪存阵列的操作均被忽略。嵌入式操作仍继续进行而不受任何影响。执行嵌入式操作时只接受少数命令集。这些信息会在单独命令的说明部分进行介绍。
- 执行时间会因命令不同而发生变化。提供用于读取某个执行命令的状态信息的命令，可确定命令结束的时间，并且确定该命令是否被成功执行。

5.2.1 命令顺序示例

图5.3 单独指令的命令

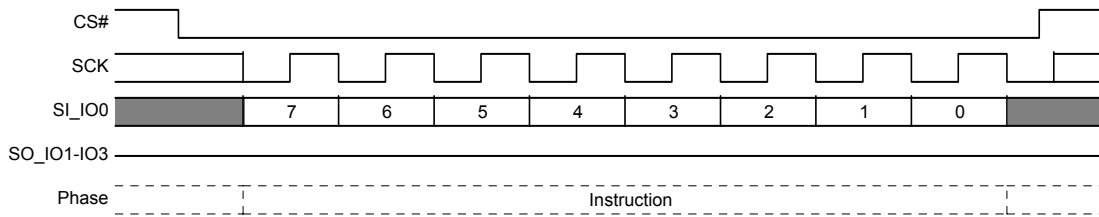


图5.4 单线输入命令

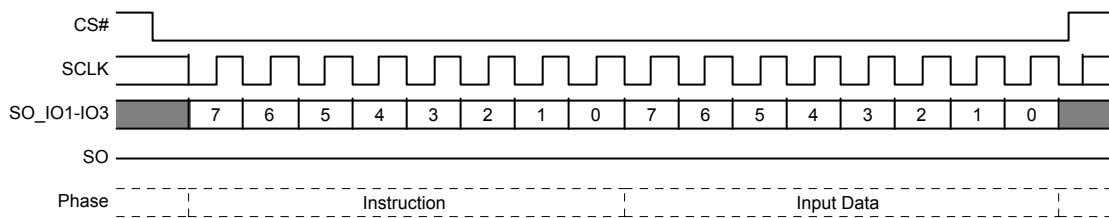


图5.5 单线输出命令（没有延迟）

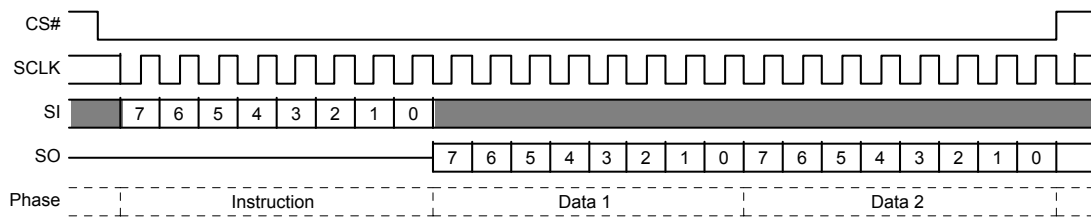


图5.6 单线I/O命令（有延迟）

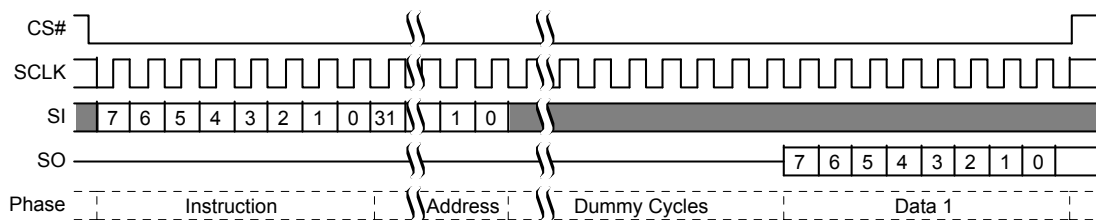


图5.7 双线输出读命令

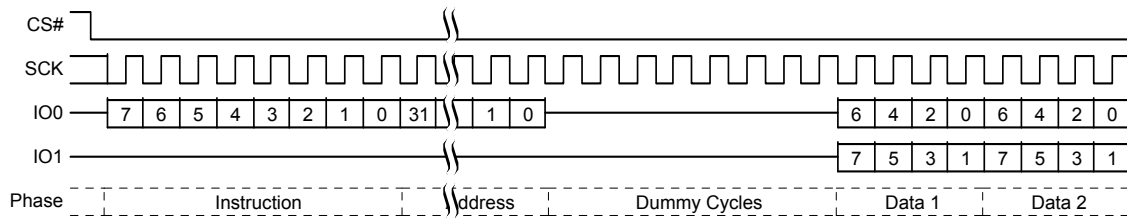


图5.8 四线输出读命令

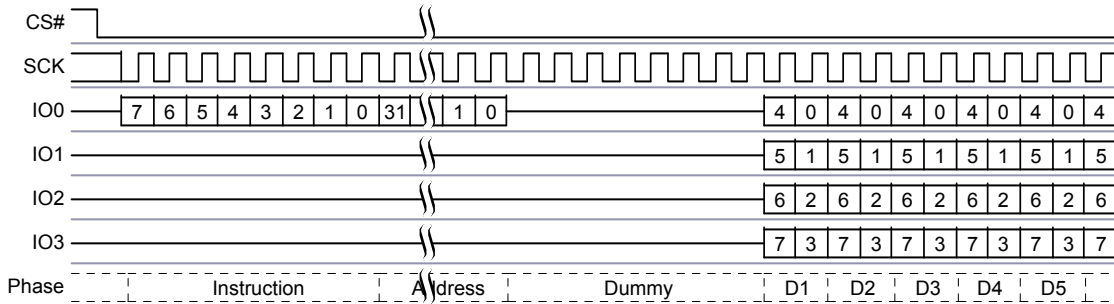


图5.9 双线I/O命令

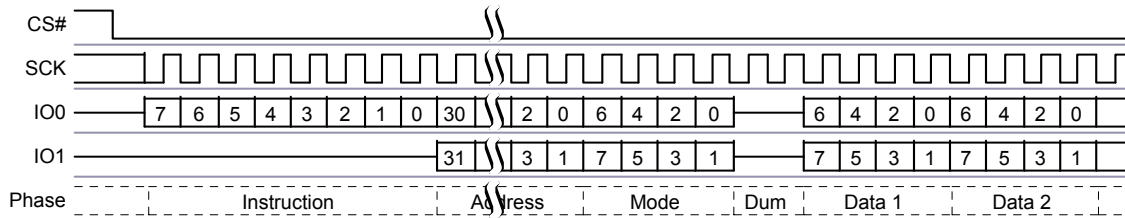
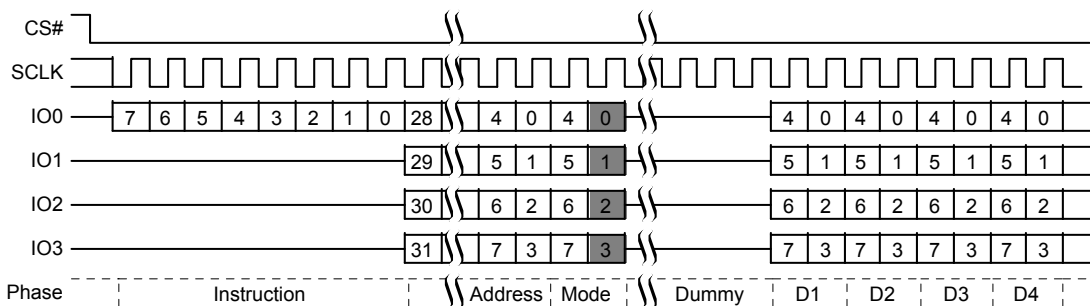
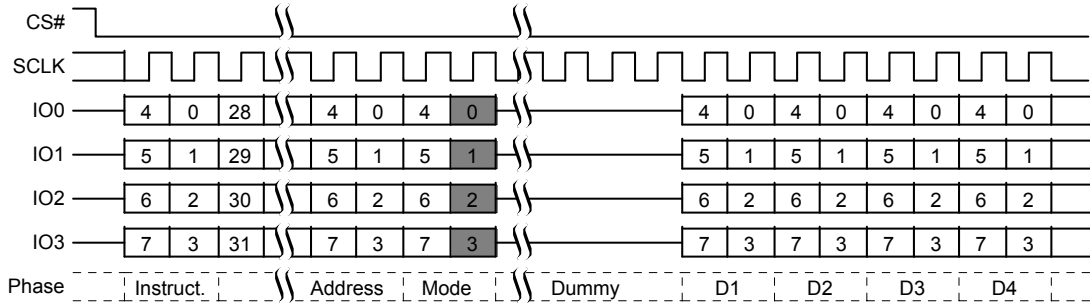


图5.10 四线I/O命令



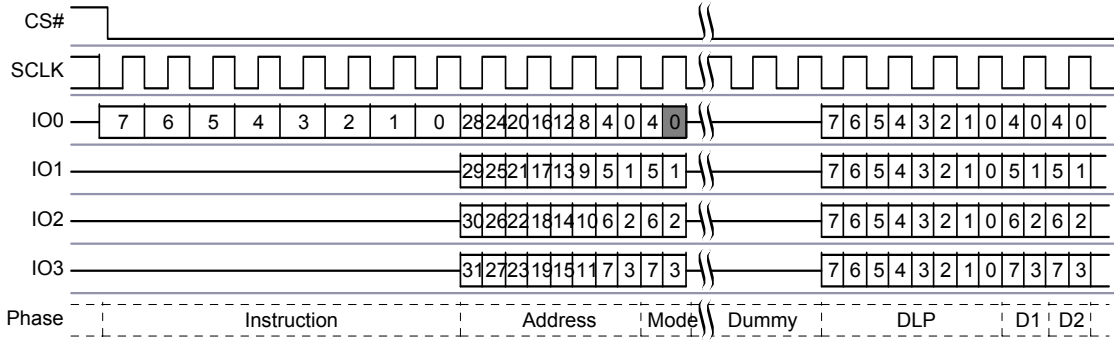
注意：灰色显示的是可选位，在该周期中主机无需发送这些位。

图5.11 在QPI模式下的四线I/O读命令



注意：灰色显示的是可选位，在该周期中主机无需发送这些位。

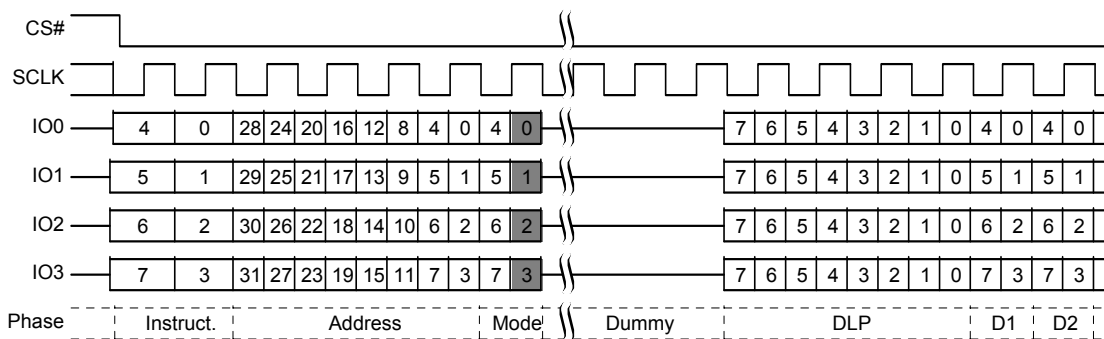
图5.12 DDR四线I/O读命令



注意：

1. 灰色显示的是可选位，在该周期中主机无需发送这些位。

图5.13 在QPI模式下的DDR四线I/O读命令



注意：

1. 灰色显示的是可选位，在该周期中主机无需发送这些位。

有关每个命令的其他序列框图，请参考第58页上的8. 命令一节中的内容。

5.3 接口状态

本节说明了与SPI接口状态相应的输入和输出信号电平。

表 5.1 接口状态汇总

接口状态	V_{DD}	SCK	CS#	RESET#	IO3/RESET#	WP#/IO2	SO/IO1	SI/IO0
下电状态	$< V_{DD}(\text{low})$	X	X	X	X	X	Z	X
低电压 硬件数据保护	$< V_{DD}(\text{cut-off})$	X	X	X	X	X	Z	X
上电(冷)复位	$\geq V_{DD}(\text{min})$	X	HH	X	X	X	Z	X
引脚(热)复位非四线模式	$\geq V_{DD}(\text{min})$	X	X	HL	HL	X	Z	X
硬件(热)复位四线模式	$\geq V_{DD}(\text{min})$	X	HH	HL	HL	X	Z	X
接口待机	$\geq V_{DD}(\text{min})$	X	HH	HH	HH	X	Z	X
指令周期 (传统SPI)	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	HV	Z	HV
单线输入周期 从主机到存储器的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	Z	HV
单线延迟(虚拟)周期	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	Z	X
单线输出周期 从存储器到主机的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	MV	X
双线输入周期 从主机到存储器的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	HV	HV
双线延迟(虚拟)周期	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	X	X
双线输出周期 从存储器到主机的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HH	X	MV	MV
四线输入周期 从主机到存储器的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HV	HV	HV	HV
四线延迟(虚拟)周期	$\geq V_{DD}(\text{min})$	HT	HL	HH	X	X	X	X
四线输出周期 从存储器到主机的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	MV	MV	MV	MV
DDR四线输入周期 从主机到存储器的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	HV	HV	HV	HV
DDR延迟(虚拟)周期	$\geq V_{DD}(\text{min})$	HT	HL	HH	X	X	X	X
DDR四线输出周期 从存储器到主机的传输	$\geq V_{DD}(\text{min})$	HT	HL	HH	MV	MV	MV	MV

注意:

- Z = 无驱动信号 — 悬空信号
- HL = 由主机驱动的 V_{IL}
- HH = 由主机驱动的 V_{IH}
- HV = HL或HH
- X = HL或HH或Z
- HT = 在HL和HH间切换
- ML = 由存储器驱动的 V_{IL}
- MH = 由存储器驱动的 V_{IH}
- MV = ML或MH

5.3.1 断电状态

当内核供电电压不大于 $V_{DD}(\text{Low})$ 电压时, 器件被视为断电。这时, 器件不对外部信号做出响应, 防止对该器件执行的所有编写或擦除操作。

5.3.2 低功耗时的硬件数据保护

如果 V_{DD} 小于 $V_{DD}(\text{Cut-off})$ ，那么存储器器件会忽略各条命令，以保证内核供电电压在工作范围外时不会启动编写和擦除操作。如果内核电压在 $\geq t_{PD}$ 时间内保持不超过 $V_{DD}(\text{Low})$ ，然后上升到 $\geq V_{DD}(\text{Minimum})$ 值时，器件将开始执行加电复位（POR）程序。继续执行POR程序，直到 t_{PU} 时间结束为止。在 t_{PU} 期间，器件不会对外部输入信号做出响应，也不会驱动任何输出。经过 t_{PU} 时间后，器件将切换到接口待机状态，并能够接受各条命令。欲了解有关POR的更多信息，请参见第131页上的12.3.1 上电（冷）复位一节。

5.3.3 硬件（暖）复位状态

这是一个配置选项，器件并不处于任何一种四线或QPI模式时，或者器件处于四线模式或QPI模式并且CS#为高电平时，则通过该选项，可以将IO3/RESET#作为硬件复位输入使用。在四线模式或QPI模式下，某些封装中会提供单独的复位输入（RESET#）。在 t_{RP} 期间，如果IO3/RESET#或RESET#被置为低电平，则器件将启动硬件复位程序。该程序将在 t_{RPH} 时间内继续执行。经过 t_{RPH} 时间以及在RESET#上升后的复位保持时间（ t_{RH} ）后，器件将切换到接口待机状态，并能够接受各条命令。欲了解有关硬件复位的更多信息，请参见第131页上的12.3 复位一节。

5.3.4 接口待机状态

如果CS#为高电平，则SPI接口处于待机状态。其他输入（RESET#除外）均被忽略。接口会保持等待状态，直到开始执行新命令为止。当CS#转为低电平以启动一条新命令时，下一个接口状态是指令周期。

在接口待机状态下，如果不再执行任何嵌入式算法，存储器器件将消耗待机电流（ I_{SB} ）。如果正在执行某个嵌入式算法，那么将消耗相应的电流，直到完成该算法为止，此时整个器件将消耗待机电流。

5.3.5 指令周期状态（传统的SPI模式）

当主机驱动了一个指令的最高有效位（MSB）且CS#转为低电平时，在SCK的下一个上升沿上，器件将捕获指令（开始新命令）的最高有效位。在后面每一个SCK的上升沿上，器件都将捕获8位指令的下一个较低有效位。主机保持CS#为低电平并驱动写保护（WP#）和IO3/RESET#信号以供指令使用。不过，仅在处理WRR或WRAR命令或任何影响状态寄存器、配置寄存器和DLR寄存器命令的指令周期内，WP#才被验证。除非执行这些指令周期期间，否则WP#将被忽略。如果器件未处于四线模式（ $CR1V[1] = 0$ ）或QPI模式（ $CR2V[3] = 0$ ），并且不需要硬件复位，则IO3/RESET#被置为高电平。

每个指令选择了用于操作的地址空间和在执行命令的其余部分期间使用的传输格式。传输格式可能为单线、双线O、四线O、双线I/O、四线I/O或DDR四线I/O。下一个接口的状态取决于所接收到的指令。

一些命令可以独立使用，无需将地址或数据传输到存储器或传输存储器的地址或数据。经过SCK（在这种命令中指令的第八位）上升沿后，主机将CS#返回高电平状态。在这种情况下，下一个接口状态是接口待机。

5.3.6 指令周期状态（QPI模式）

在QPI模式下，如果 $CR2V[3] = 1$ ，则在每个周期内各条指令能够传输4比特。在该模式下，指令周期与四线输入周期相同。请参见第22页上的5.3.13 QPP或QOR地址输入周期状态一节。

5.3.7 单线输入周期状态 — 从主机到存储器的传输

将单线串行输入（SI）信号上的指令从主机发送到存储器器件后，一些命令会传输信息。主机保持RESET#为高电平、CS#为低电平，并驱动SI以供命令使用。存储器不会驱动串行输出（SO）信号。

下一个期望接口的状态取决于该指令。通过使用额外的单线输入周期，某些指令会连续将地址或数据发送给存储器。其他命令可能会切换到单线延迟状态，或直接转到单线、双线或四线输出周期状态。

5.3.8 单线延迟（虚拟）周期状态

读取命令可能没有延迟周期，也可能会有若干个延迟周期，在这些周期内，直到发送到主机前，主闪存存储器阵列内的数据都被读取。配置寄存器（ $CR3V[3:0]$ ）中的延迟代码决定了延迟周期的数量。在延迟周期内，主机将保持RESET#和IO3/RESET#为高电平、CS#为低电平，SCK会进行切换。写保护（WP#）信号被忽略。在这些周期内，主机可能会驱动SI信号，或者将SI保持为悬空状态。在延迟周期内，存储器不会使用在SO或其他I/O信号上所驱动的任何数据。在延迟周期内，存储器不会驱动串行输出（SO）或其他I/O信号。

下一个接口状态取决于命令结构，例如，延迟周期数量以及读取方式是单线、双线或四线宽度。

5.3.9 单线输出周期状态 — 从存储器到主机的传输

通过单线串行输出（SO）信号，若干个命令会将信息返回到主机。主机将保持RESET#，并且IO3/RESET#为高电平、CS#为低电平。写保护（WP#）信号被忽略。存储器忽略了串行输入（SI）信号。存储器使用数据驱动SO信号。

下一个接口状态仍是单线输出周期，直到主机通过将CS#返回到高电平状态来结束命令为止。

5.3.10 双线输入周期状态 — 从主机到存储器的传输

每个周期内，双线读取I/O命令都会将两个地址或模式位传输给存储器。主机保持RESET#和IO3/RESET#为高电平、CS#为低电平。在该过程中，写保护（WP#）信号将被忽略。主机将驱动SI / IO0和SO / IO1上的地址。

传输地址和模式位后，下一个接口状态是双线延迟周期（若需要延迟周期）或双线输出周期（若无需延迟周期）。

5.3.11 双线延迟（虚拟）周期状态

读取命令可能没有延迟周期，也可能会有若干个延迟周期。在这些周期内，在发送到主机前，主闪存存储器阵列内的数据都被读取。配置寄存器（CR3V[3:0]）中的延迟代码决定了延迟周期的数量。在延迟周期内，主机会保持RESET#和IO3/RESET#为高电平、CS#为低电平，SCK会持续切换。写保护（WP#）信号被忽略。在这些周期内，主机可能驱动SI / IO0和SO / IO1信号，或者将这些信号保持为悬空状态。在延迟周期内，存储器不会使用在SI / IO0和SO / IO1或其他I/O信号上所驱动的任意数据。最后的延迟周期结束时，主机必须停止驱动SCK的下降沿上SI / IO0和SO / IO1信号。在所有延迟周期内，建议停止主机对这些信号的驱动操作，以便在延迟周期结束时存储器开始驱动前，有足够时间使主机驱动器停止传输操作。这样做，如果信号方向发生改变，则能够防止主机与主机和存储器间发生冲突。在延迟周期内，存储器不会驱动SI / IO0和SO / IO1信号。

最后延迟周期结束后，下一个接口状态是双线输出周期。

5.3.12 双线输出周期状态 — 从存储器到主机的传输

在每个周期内，双线读取输出和双线读取I/O命令会将数据（2位）返回到主机内。主机保持RESET#和IO3/RESET#为高电平、CS#为低电平。在该过程中，写保护（WP#）信号将被忽略。在双线输出周期内，在SCK的下降沿上，存储器将驱动SI / IO0和SO / IO1信号上的数据。

下一个接口状态仍是双线输出周期，直到主机通过使CS#返回到高电平状态来结束命令为止。

5.3.13 QPP或QOR地址输入周期状态

四线页编程和四线输出读取命令仅在IO0信号上将地址发送到存储器。其他IO信号均被忽略。主机将保持RESET#和IO3/RESET#为高电平、CS#为低电平，并驱动IO0。

QPP命令发送地址后，下一个接口状态将是四线输入周期。QOR传输地址后，下一个接口状态是四线延迟周期（若需要延迟周期）或四线输出周期（若无需延迟周期）。

5.3.14 四线输入周期状态 — 从主机到存储器的传输

每个周期内，四线I/O读取命令都将四个地址位或模式位传输到存储器中。在QPI模式下，每个周期（包括指令周期）内，四线I/O读取和页编程命令都会将四个数据位传输到存储器内。主机会保持CS#为低电平，并驱动IO信号。

四线I/O读取命令传输地址和模式位后，下一个接口状态是四线延迟周期（若需要延迟周期）或四线输出周期（若无需延迟周期）。在QPI模式下，页编程命令传输需要编程的数据后，主机会使CS#返回到高电平状态，接口会返回到待机状态。

5.3.15 四线延迟（虚拟）周期状态

读取命令可能没有延迟周期，也可能会有若干个延迟周期。在这些周期内，在发送到主机前，主闪存存储器阵列内的数据都被读取。配置寄存器（CR3V[3:0]）中的延迟代码决定了延迟周期的数量。在延迟周期内，主机会保持CS#为低电平，并持续切换SCK。在这些周期内，主机可能驱动IO信号，或者将IO保持为悬空状态。在延迟周期内，存储器不会使用IO信号上所驱动的任意数据。最后的延迟周期结束时，主机必须停止驱动下降沿上的IO信号。在所有延迟周期内，建议停止主机对这些信号的驱动操作，以便在延迟周期结束时存储器开始驱动前，有足够时间使主机驱动器停止传输操作。这样做，如果信号方向发生改变，则能够防止主机与主机和存储器间发生冲突。在延迟周期内，存储器不会驱动IO信号。

最后延迟周期结束后，下一个接口状态是四线输出周期。

5.3.16 四线输出周期状态 — 从存储器到主机的传输

每个周期内，四线O和四线I/O读取命令会将4位的数据返回给主机。主机保持CS#为低电平。在四线输出周期期间，存储器会驱动IO0-IO3信号上的数据。

下一个接口状态仍是四线输出周期，直到主机使CS#返回高电平状态而使命令结束为止。

5.3.17 DDR四线输入周期状态 — 从主机到存储器的传输

DDR四线I/O读取命令会通过所有IO信号将地址和模式位发送给存储器。每个周期内，在SCK的上升沿上传输4比特，在下降沿上传输4比特。主机保持CS#为低电平。

传输地址和模式位后，下一个接口状态是DDR延迟周期。

5.3.18 DDR延迟周期状态

DDR读取命令可能具有一个或若干个延迟周期。在这些周期内，在发送给主机前，主闪存存储器阵列内的数据将被读取。配置寄存器（CR3V[3:0]）中的延迟代码决定了延迟周期的数量。在延迟周期内，主机会保持CS#为低电平。此外，主机不会驱动IO信号。因此，在存储器开始驱动前，有足够的时间使主机驱动器停止传输操作。这样做，如果信号方向发生改变，便能够防止主机与主机和存储器间发生冲突。在最后4个延迟周期内，存储器可以选择使用一个数据学习模式（DLP）驱动所有IO信号。如果延迟周期数小于5，则不应该使能DLP选项。因此，在存储器开始驱动DLP前，至少有一个高阻抗的周期以等待IO信号经过处理并返回。如果有多于4个延迟周期，则存储器不会驱动IO信号，直到最后四个延迟周期为止。

最后延迟周期结束后，下一个接口状态是DDR四线输出周期，具体情况取决于指令。

5.3.19 DDR四线输出周期状态 — 从存储器到主机的传输

DDR四线I/O读取命令会通过所有IO信号将数据位发送给主机。每个周期内，在SCK的上升沿上传输4比特，在下降沿上传输4比特。主机保持CS#为低电平。

下一个接口状态仍是DDR四线输出周期，直到主机使CS#返回到高电平状态来结束命令为止。

5.4 数据保护

硬件设计提供并管理着某些基本的保护功能，用于防止对存储数据进行意外的更改。具体内容如下。其它软件管理的保护方式将在本文档的软件章节中进行讨论。

5.4.1 上电

当内核供电电压不大于 $V_{DD(Low)}$ 电压时，器件被视为断电。这时，器件不对外部信号做出响应，防止对该器件执行的所有编写或擦除操作。在tPU期间，用户不能输入任何有效命令。

5.4.2 低功耗

如果 V_{DD} 小于 $V_{DD(Cut-off)}$ ，那么存储器器件会忽略各条命令，以保证内核供电电压在工作范围外时不会启动编写和擦除操作。

5.4.3 时钟脉冲数量

器件会验证所有修改非易失性存储器和寄存器数据的命令是否包含了一个时钟脉冲数量。执行这些命令前，该时钟脉冲数量是8的倍数（字节边界）。不在8位（字节）边界上结束的命令将被忽略，该命令不存在任何错误状态。

5.4.4 深度下电（DPD）

在DPD模式下，器件仅对从DPD模式恢复的命令（RESABh）发出响应。在该模式下，所有其他命令均被忽略，从而防止对存储器执行编程和擦除操作。如果使能了IO3/RESET#功能（CR2V[7] = 1）或者RESET#有效，那么通过将IO3/RESET#或RESET#设置为低电平状态便可以启动硬件复位并使器件退出DPD模式。

6. 地址空间映射

6.1 概述

6.1.1 扩展地址

FL-L系列支持32位（4个字节）地址。这样，同前一代（旧版）SPI器件相比，器件容量更高（前代SPI器件仅支持24位（3个字节）地址。对于24位的字节分辨率系统，地址寄存器允许对16 MB（128 Mb）的地址空间进行访问。对于32位的字节分辨率系统，地址寄存器允许对4 GB（32 Gb）的地址空间直接进行访问。

传统的命令仍支持24位地址，以实现软件向后兼容。通过以下两种方法可以获取扩展的32位地址：

- 扩展地址模式 — 一个易失性配置寄存器位，它更改了所有传统命令，以获取由主机系统提供的32位地址。
- 4字节地址命令 — 执行传统的和新的功能，从而能够获取32位地址。

上电或复位后，扩展地址模式的默认条件是由非易失性配置位控制的。可将默认的扩展地址模式设置为24或32位地址。这样可以对器件的前128 Mb空间进行传统的软件兼容访问，并允许器件在32位地址模式下直接启动。

6.1.2 多地址空间

许多命令在主闪存存储器阵列上运行。某些命令在独立于主闪存阵列的地址空间内运行。每个单独地址空间使用了整个24或32位地址，但只定义可用地址空间的一小部分。

6.2 闪存存储器阵列

主闪存阵列被分为大小统一的擦除单位，这些单位被称为物理块（64 KB）、半块（32 KB）和扇区（4 KB）。

表 6.1 S25FL256L 扇区地址映射

块大小 (KB)	块数量	块范围	半块大小 (KB)	半块数量	半块范围	扇区大小 (KB)	扇区数量	扇区范围	地址范围 (字节地址)	注意
64	1	BA00	32	1	HBA00	4	1	SA00	000000h-000FFFh	扇区起始地址 — 扇区结束地址
			:	:	:	:	:	:	:	
			32	2	HBA01	4	16	SA15	000F000h-000FFFFh	
64	512	BA511	32	1023	HBA1022	4	8176	SA8175	1FF0000h-1FF0FFFh	
			:	:	:	:	:	:	:	
			32	1024	HBA1023	4	8192	SA8191	1FFF000h-1FFFFFh	

表 6.2 S25FL128L 扇区地址映射

块大小 (KB)	块数量	块范围	半块大小 (KB)	半块数量	半块范围	扇区大小 (KB)	扇区数量	扇区范围	地址范围 (字节地址)	注意
64	1	BA00	32	1	HBA00	4	1	SA00	000000h-000FFFh	扇区起始地址 — 扇区结束地址
			:	:	:	:	:	:	:	
			32	2	HBA01	4	16	SA15	00F000h-00FFFFh	
64	256	BA255	32	511	HBA510	4	4080	SA4079	FF0000h-FF0FFFh	
			:	:	:	:	:	:	:	
			32	512	HBA511	4	4096	SA4095	FFF000h-FFFFFFh	

6.3 ID地址空间

RDID命令（9Fh）从单独的闪存存储器地址空间内读取信息，以获取器件标识（ID）。请参考第122页上的10.2 [器件ID地址映射一节](#)，了解定义ID地址空间内容的表格。该ID地址空间由赛普拉斯编程，主机系统只能对其执行读取操作。

6.3.1 器件的唯一ID

64位唯一编号位于唯一器件ID地址空间中的8个字节内，请参考第122页上的表10.6. [器件的唯一ID](#)。该唯一ID可以作为每个器件的唯一软件可读序列号。

6.4 JEDEC JESD216串行闪存可发现参数（SFDP）空间

根据串行闪存可发现参数的JEDEC JESD216标准，RSFDP命令（5Ah）从单独闪存存储器地址空间读取信息，以获取器件标识、特性和配置信息。该ID地址空间作为SFDP参数中的一个。请参考第115页上的10.1 [JEDEC JESD216B串行闪存可发现参数一节](#)，了解定义SFDP地址空间内容的表格。该SFDP地址空间由赛普拉斯编程，主机系统只能对其进行读取操作。

6.5 安全区域地址空间

每个FL-L系列存储器器件具有一个独立于主闪存阵列的1024字节安全区域地址空间。该安全区域分为四个可独立锁定的256字节区域。安全区域存储器空间包含某些信息，这些信息暂时被保护或长期锁定，因此无法对其进行其他的编程或擦除操作。

从赛普拉斯出厂时，这些区域的数据字节被擦除为FFh。当该区域不被保护或锁定时，可以对其进行编程和擦除操作，就像对其他闪存存储器地址空间进行的操作一样。可以独立擦除每个区域。安全区域锁定位（CR1NV[5:2]）位于配置寄存器1内。安全区域锁定位是一次性可编程（OTP）位。编程这些位后（设置为1），一个锁定位将长期防止对相关区域进行其他的擦除或编程操作。

保护寄存器（PR）NVLock位暂时防止对区域2和区域3进行编程或擦除操作。NVLock位是易失性位，通过使用IRP逻辑和各种命令可以设置或清除它。请参见第40页上的[保护寄存器（PR）](#)。

IRP寄存器（IRP[2]）中的安全区域密码保护位用于防止对区域2和区域3进行编程和擦除操作，直到提供一个密码为止。IRP寄存器（IRP[6]）中的安全区域读取保护位用于防止对区域3进行读取操作，直到提供一个密码为止。尝试读取防止读操作的区域时，将返回无效和未定义的数据。请参考第39页上的[单独和区域保护寄存器（IRP）](#)。

尝试擦除或编程受保护或被锁定的区域时，会导致结果失败，并且SR2V[6:5]中的P_ERR或E_ERR位被设置为1。（请参考第31页上的[易失性状态寄存器2（SR2V）](#)了解详细说明）。

表 6.3 安全区域地址映射

区域	字节地址范围（十六进制）	初始供应状态（十六进制）
区域0	000到0FF	所有字节 = FF
区域1	100到1FF	所有字节 = FF
区域2	200到2FF	所有字节 = FF
区域3	300到3FF	所有字节 = FF

6.6 寄存器

寄存器是存储器单元的小组，用于配置FL-L系列存储器器件的运行方式以及报告器件的运行状态。通过使用特殊命令可以访问这些寄存器。使用于每个寄存器的命令（和十六进制指令代码）被注释在每个寄存器的说明内。

在旧版的SPI存储器器件中，独立寄存器位可以是同一个寄存器中易失性、非易失性或一次性可编程（OTP）位的混合体。在某些配置选项中，寄存器位的类型会发生改变，例如，从非易失性变为易失性。

FL-L系列使用单独的非易失性或易失性存储器单元组（区域）来实现不同类型的寄存器位。然而，FL-L系列兼容了传统寄存器和各种命令并且它们始终有效。当传统寄存器具有易失性位或读取传统寄存器的命令具有零读取延迟时，每个传统寄存器有非易失性和易失性版本。在上电复位（POR）或硬件复位或软件复位期间，寄存器的非易失性版本被复制为易失性版本，以提供易失性寄存器的默认状态。当写入非易失性寄存器位时，寄存器的非易失性版本将被擦除并编程新的位值，而且寄存器的易失性版本将更新非易失性版本的新内容。无论读取非易失性寄存器还是易失性寄存器，都会提供该寄存器的易失性版本。当编程OTP位时，寄存器的非易失性版本将被编程，而且寄存器易失性版本中的相应位被更新。当写入易失性寄存器位时，只有寄存器易失性版本中的相应位被更新。

每一位的类型注释均位于每个寄存器的说明内。每一位的默认状态指的是执行上电复位、硬件复位或软件复位后的状态（如果它是易失性位）。如果该位是非易失性或OTP位，那么其默认状态便是器件从赛普拉斯出厂时该位的值。

6.6.1 状态寄存器1

6.6.1.1 非易失性状态寄存器1（SR1NV）S25FL256L

相关命令：非易失性寄存器的写使能（WREN 06h）、写禁用（WRDI 04h）、写入寄存器（WRR 01h）、读取任何寄存器（RDAR 65h）、写入任何寄存器（WRAR 71h）。

表 6.4 非易失性状态寄存器 1（SR1NV）

位	字段名称	功能	类型	默认状态	说明
7	SRP0_NV	状态寄存器保护0 默认设置	非易失性	0	为SRP0提供默认状态。
6	TBPROT_NV	TBPROT默认设置	非易失性	0	为TBPROT提供默认状态。
5	BP_NV3	传统块保护默认设置	非易失性	0000b	为BP位提供默认状态。
4	BP_NV2				
3	BP_NV1				
2	BP_NV0				
1	WEL_D	WEL的默认设置	非易失性 只读	0	为WEL状态位提供默认状态。用户不能编程该位。
0	WIP_D	WIP的默认设置	非易失性 只读	0	为WIP状态位提供默认状态。用户不能编程该位。

状态寄存器保护非易失性位（SRP0_NV） SR1NV[7]: 为SRP0提供默认状态。请参见第45页上的**状态寄存器保护（SRP1、SRP0）**。

顶部或底部保护位（TBPROT_NV） SR1NV[6]: 为TBPROT提供默认状态。

传统模块保护位（BP_NV3、BP_NV2、BP_NV1、BP_NV0） SR1NV[5:2]: 为BP_3到BP_0位提供默认状态。

写使能锁存的默认位（WEL_D） SR1NV[1]: 为SR1V[1]中的WEL状态提供默认状态。该位由赛普拉斯编程，用户不能编程它。

正在写入的默认位（WIP_D） SR1NV[0]: 为SR1V[0]中的WIP状态提供默认状态。该位由赛普拉斯编程，用户不能编程它。

6.6.1.2 易失性状态寄存器1 (SR1V) S25FL256L

相关命令：读取状态寄存器1 (RDSR105h)、易失性寄存器的写使能 (WRENV50h)、写入寄存器 (WRR01h)、清除状态寄存器 (CLSR 30h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。通过使用RDSR1命令可以显示该寄存器。

表 6.5 S25FL256L 易失性状态寄存器 1 (SR1V)

位	字段名称	功能	类型	默认状态	说明
7	SRP0	状态寄存器保护0	易失性		1 = WP#为低电平时，如果不执行影响SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的任何命令，可以锁定SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的当前状态 0 = 不进行寄存器保护，即使WP#为低电平。
6	TBPROT	顶部或底部的保护	易失性		1 = 从底部（低地址）开始执行BP 0 = 从顶部（高地址）开始执行BP
5	BP3	传统块保护易失性	易失性	SR1NV	防止对所选的扇区（块）范围进行编程或擦除操作。
4	BP2				
3	BP1				
2	BP0				
1	WEL	写使能锁存	易失性只读		0 = 不使能写入，不可执行嵌入式操作；1 = 使能写入，可执行嵌入式操作 该位不受WRR或WRAR的影响，但受WREN、WRENV、WRDI和CLSR命令的影响。
0	WIP	正在执行写入操作	易失性只读		1 = 器件处于忙碌状态，正在执行编程或擦除等嵌入式操作 0 = 器件已经处于待机模式，可以接收命令 该位不受WRR或WRAR的影响，它仅提供WIP状态。

状态寄存器保护0 (SRP0) SR1V[7]: 当该位为1并且WP#输入为低电平时，器件将处于硬件保护模式。在这种模式下，由于这些寄存器都被配置为只读的，所以更改状态寄存器或配置寄存器的所有命令均被忽略，并且不被执行。因此，可有效地锁存状态寄存器和配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV位。如果WP#为高电平，便能够更改状态寄存器和配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV。如果SRP0为0，WP#无效，则可能更改状态寄存器和配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV。WP#不会影响任何其他寄存器的写操作。SRP0跟踪该位 (SRP0_NV) 非易失性版本的更改。当使能QPI或QIO模式 (CR2V[3]或CR1V[1] = 1) 时，内部WP#信号电平为1。这是因为使能其中某个模式时，WP#外部输入将作为IO2使用。这样可以有效地关闭硬件保护。这时，寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被解锁，您可以对这些寄存器进行写操作。请参见第45页上的**状态寄存器保护 (SRP1、SRP0)**。

TBPROT SR1V[6]: 该位定义了状态寄存器中传统模块保护位BP3、BP2、BP1和BP0的参考点。如状态寄存器部分中所述，通过使用BP3-0位，用户可以选择保护阵列的某一部分 (1/64、1/4、1/2等等) 或整个阵列。将TBPROT设置为0时，传统模块保护被定义为从该阵列的顶部（最大地址）开始。将TBPROT设置为1时，传统模块保护被定义为从该阵列的底部（0地址）开始。TBPROT跟踪该位 (TBPROT_NV) 非易失性版本的更改。

传统模块保护 (BP3、BP2、BP1、BP0) SR1V[5:2]: 这些位定义了防止编程和擦除操作的主闪存阵列区域。第46页上的**7.6.1 传统块保护**一节说明了BP位值如何选择受保护的存储器阵列区域。

写使能锁存 (WEL) SR1V[1]: 必须将WEL位设置为1，这样才能使能编程、写入或擦除操作，并能够防止对存储器或寄存器值进行意外更改。执行写使能 (WREN) 命令会将写使能锁存位置1，这样便能执行后面所有的编程、擦除或写命令。通过写禁用 (WRDI) 命令，将写使能锁存位置0，这样可以防止执行编程、擦除和写命令。编程、写入或擦除操作成功后，WEL位将被清除为0。如果这些操作失败，WEL位将保持设置状态，应该使用CLSR命令清除它。断电/上电序列、硬件复位或软件复位发生后，写使能锁存位将被设置为WEL_D。WRR或WRAR命令不会影响该位的状态。

正在写入 (WIP) SR1V[0]: 表示器件是否正在执行编程、写入、擦除或其他操作，器件在这段时间内会忽略新的操作命令。当该位被设置为1时，器件会执行一个操作。当WIP为1时，仅接收读取状态寄存器 (RDSR1、RDSR2)、读取任何寄存器 (RDAR)、擦除/编程暂停 (EPS)、清除状态寄存器 (CLSR)、读取配置寄存器 (RDCR1、RDCR2、RDCR3) 和软件复位 (RST 99h前面的RSTEN 66h) 等命令。只有正在对存储器阵列进行擦除或编程操作时，才能接受EPS命令。当WIP = 1时，状态寄存器E_ERR和P_ERR位将被更新。当P_ERR或E_ERR位为1时，WIP位将保持设置为1，以表示器件的繁忙状态并且无法接收新的操作命令。必须接收清除状态寄存器 (CLSR) 命令，这样可以使器件返回到待机模式。WIP位被清零时，表示无操作正在执行。它是一个只读位。

6.6.1.3 易失性状态寄存器1 (SR1NV) S25FL128L

相关命令：非易失性寄存器的写使能 (WREN 06h)、写禁用 (WRDI 04h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。

表 6.6 S25FL128L 非易失性状态寄存器 1 (SR1NV)

位数	字段名称	功能	类型	默认状态	说明
7	SRP0_NV	状态寄存器保护0默认设置	非易失性	0	为SRP0提供默认状态。
6	SEC_NV	扇区/块保护	非易失性	0	为SEC提供默认状态
5	TBPROT_NV	TBPROT默认设置	非易失性	0	为TBPROT提供默认状态。
4	BP_NV2	传统块保护默认设置	非易失性	000b	为BP位提供默认状态。
3	BP_NV1				
2	BP_NV0				
1	WEL_D	WEL的默认设置	非易失性 只读	0	为WEL状态提供默认状态。用户不能编程该位。
0	WIP_D	WIP的默认设置	非易失性 只读	0	为WIP状态提供默认状态。用户不能编程该位。

状态寄存器保护非易失性位 (SRP0_NV) SR1NV[7]: 为SRP0提供默认状态。

请参见第45页上的**状态寄存器保护 (SRP1、SRP0)**。

扇区/块保护位 (SEC_NV) SR1NV[6]: 为SEC提供默认状态。

顶部或底部保护位 (TBPROT_NV) SR1NV[5]: 为TBPROT提供默认状态。

传统块保护位 (BP_NV3、BP_NV2、BP_NV1、BP_NV0) SR1NV[4:2]: 为BP_2到BP_0位提供默认状态。

写使能锁存的默认位 (WEL_D) SR1NV[1]: 为SR1V[1]中的WEL状态提供默认状态。该位由赛普拉斯编程，用户不能编程它。

正在写入的默认位 (WIP_D) SR1NV[0]: 为SR1V[0]中的WIP状态提供默认状态。该位由赛普拉斯编程，用户不能编程它。

6.6.1.4 易失性状态寄存器1 (SR1V) S25FL128L

相关命令：读取状态寄存器1 (RDSR105h)、易失性寄存器的写使能 (WRENV50h)、写入寄存器 (WRR01h)、清除状态寄存器 (CLSR 30h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。通过使用RDSR1命令可以显示该寄存器。

表 6.7 S25FL128L 易失性状态寄存器 1 (SR1V)

位数	字段名称	功能	类型	默认状态	说明
7	SRP0	状态寄存器保护0	易失性	SR1NV	1 = WP#为低电平时, 如果不执行影响SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的任何命令, 可以锁定SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的当前状态 0 = 不进行寄存器保护, 即使WP#为低电平。
6	SEC	扇区/块保护	易失性		0 = BP2-BP0保护64 kB块 1 = BP2-BP0保护4 kB扇区
5	TBPROT	顶部或底部的保护	易失性		1 = 从底部(低地址)开始执行BP 0 = 从顶部(高地址)开始执行BP
4	BP2	传统块保护易失性	易失性		防止对所选的扇区(块)范围进行编程或擦除操作。
3	BP1				
2	BP0				
1	WEL	写使能锁存	易失性 只读		0 = 不使能写入, 不可执行嵌入式操作; 1 = 使能写入, 可执行嵌入式操作 该位不受WRR或WRAR的影响, 但受WREN、WRENV、WRDI和CLSR命令的影响。
0	WIP	正在执行写入操作	易失性 只读		1 = 器件处于忙碌状态, 正在执行编程或擦除等嵌入式操作 0 = 器件已经处于待机模式, 可以接受命令 该位不受WRR或WRAR的影响, 它仅提供WIP状态。

状态寄存器保护0 (SRP0) SR1V[7]: 当该位为1并且WP#输入为低电平时, 器件将处于硬件保护模式。在这种模式下, 由于这些寄存器都被配置为只读, 所以更改状态寄存器或配置寄存器的所有命令均都被忽略, 并且不被执行。因此, 可有效地锁存状态寄存器和配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV位。如果WP#为高电平, 则可能更改状态寄存器和配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV, 并能够更改配置寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV。WP#不会影响任何其他寄存器的写操作。SRP0跟踪该位 (SRP0_NV) 非易失性版本的更改。当使能QPI或QIO模式 (CR2V[3]或CR1V[1] = 1) 时, 内部WP#信号电平为1。这是因为当使能其中的某个模式时, WP#外部输入将作为IO2使用。这样可以有效地关闭硬件保护。这时, 寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被解锁, 您可以对这些寄存器进行写操作。请参见第45页上的**状态寄存器保护 (SRP1、SRP0)**。

扇区/块保护 (SEC) SR1V[6]: 该位决定了块保护位 (BP2、BP1、BP0) 将保护4 kB的扇区 (SEC = 1) 还是64 kB的块 (SEC = 0)。请参考第46页上的**7.6.1 传统块保护**一节, 了解SEC位的值如何选择受保护的存储器阵列区域。

TBPROT SR1V[5]: 该位定义了状态寄存器中传统块保护位BP2、BP1和BP0的参考点。如状态寄存器部分中所述, 通过使用BP2-0位, 用户可以选择保护阵列的某一部分 (1/64、1/4、1/2等等) 或整个阵列。将TBPROT设置为0时, 传统模块保护被定义为从该阵列的顶部 (最大地址) 开始。将TBPROT设置为1时, 传统模块保护被定义为从该阵列的底部 (0地址) 开始。TBPROT跟踪该位 (TBPROT_NV) 非易失性版本的更改。

传统块保护位 (BP2、BP1、BP0) SR1V[4:2]: 这些位定义了防止编程和擦除操作的主闪存阵列区域。第46页上的**7.6.1 传统块保护**一节说明了BP位值如何选择受保护的存储器阵列区域。

写使能锁存位 (WEL) SR1V[1]: 必须将WEL位设置为1, 这样才能使能编程、写入或擦除操作, 并能够防止对存储器或寄存器值进行意外更改。执行写使能 (WREN) 命令会将写使能锁存位设置为1, 这样便能执行后面所有的编程、擦除或写命令。通过写禁用 (WRDI) 命令, 将写使能锁存位设置为0, 这样可以防止执行编程、擦除和写命令。编程、写入或擦除操作成功后, WEL位将被清除为0。如果这些操作失败, WEL位将保持设置状态, 应该使用CLSR命令清除它。断电/上电序列、硬件复位或软件复位发生后, 写使能锁存位将被设置为WEL_D。WRR或WRAR命令不会影响该位的状态。

正在写入位 (WIP) SR1V[0]: 表示器件是否正在执行编程、写入、擦除或其他操作，器件在这段时间内会忽略新的操作命令。当该位被设置为1时，器件会执行一个操作。当WIP为1时，仅接收读取状态寄存器 (RDSR1、RDSR2)、读取任何寄存器 (RDAR)、擦除/编程挂起 (EPS)、清除状态寄存器 (CLSR) 和软件复位 (RSTEN 66h后面紧跟着RST 99h) 等命令。只有正在对存储器阵列进行擦除或编程操作时，才能接受EPS命令。当WIP=1时，状态寄存器E_ERR和P_ERR位将被更新。当P_ERR或E_ERR位为1时，WIP位将保持设置为1，以表示器件的繁忙状态并且无法接收新的操作命令。必须接收清除状态寄存器 (CLSR) 命令，这样可以使器件返回到待机模式。将WIP位清除为0时，表示没有任何操作正在进行。这是一个只读位。

6.6.2 易失性状态寄存器2 (SR2V)

相关命令: 读取状态寄存器2 (RDSR207h)、读取任何寄存器 (RDAR65h)。状态寄存器2中没有用户可编程的非易失性位，它的所有定义位都是易失性只读状态位。这些位的默认状态是由硬件设置的。

表 6.8 易失性状态寄存器 2 (SR2V)

位	字段名称	功能	类型	默认状态	说明
7	RFU	保留		0	保留供将来使用
6	E_ERR	发生了擦除错误	易失性 只读	0	1 = 发生了错误 0 = 无错误
5	P_ERR	发生了编程错误	易失性 只读	0	1 = 发生了错误 0 = 无错误
4	RFU	保留		0	保留供将来使用
3	RFU	保留		0	保留供将来使用
2	RFU	保留		0	保留供将来使用
1	ES	擦除挂起	易失性 只读	0	1 = 处于擦除挂起模式。 0 = 未处于擦除挂起模式。
0	PS	编程挂起	易失性 只读	0	1 = 处于编程挂起模式。 0 = 未处于编程挂起模式。

擦除错误位 (E_ERR) SR2V[6]: 擦除错误位用于指出擦除操作的成功或失败。当擦除错误位被设置为1时，则表示最后擦除操作有错误。当用户尝试擦除受保护的单独主存储器扇区或一个被锁定的安全区域时，该位将被设置。如果在执行命令期间找到受保护的扇区时，芯片擦除命令将设置E_ERR。将擦除错误位设置为1时，通过使用清除状态寄存器 (CLSR) 命令可以将该位清除为0。这是一个只读位，它不受WRR或WRAR命令的影响。

编程错误 (P_ERR) SR2V[5]: 编程错误位用于指出编程操作的成功或失败。将编程错误位设置为1时，则表示最后编程操作有错误。当用户尝试对受保护的主存储器扇区或一个被锁定的安全区域进行编程时，该位将被置位。将编程错误位设置为1时，通过使用清除状态寄存器 (CLSR) 命令可以将该位清除为0。这是一个只读位，它不受WRR或WRAR命令的影响。

擦除挂起位 (ES) SR2V[1]: 擦除挂起位用于指出器件是否进入擦除挂起模式。用户不能对该状态位进行写操作。当擦除挂起位为1时，器件将处于擦除挂起模式。当擦除挂起位为0时，则器件不处于擦除挂起模式。请参考第94页上的8.6.5 [编程或擦除挂起 \(PES 75h\)](#) 一节，了解擦除挂起/恢复命令的详细内容。

编程挂起 (PS) SR2V[0]: 编程挂起位用于指出器件是否处于编程挂起模式。用户不能对该状态位进行写操作。当编程挂起位为1时，则器件处于编程挂起模式。当编程挂起位为0时，则器件不处于编程挂起模式。更多信息，请查阅第94页上的8.6.5 [编程或擦除挂起 \(PES 75h\)](#) 一节。

6.6.3 配置寄存器1

配置寄存器1用于控制某些接口和数据保护功能。通过使用带有十六个输入周期的WRR命令或通过使用WRAR命令可以更改这些寄存器位。

6.6.3.1 非易失性配置寄存器1 (CR1NV)

相关命令：非易失性寄存器的写使能 (WREN 06h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。

表 6.9 非易失性配置寄存器 1 (CR1NV)

位	字段名称	功能	类型	默认状态	说明
7	SUS_D	挂起状态默认设置	非易失性只读	0	为挂起状态提供默认状态。用户不能编程该位。
6	CMP_NV	补充保护默认设置	非易失性	0	为CMP提供默认状态。
5	LB3	安全区域锁定位	OTP	0	安全区域3:0的OTP锁定位3:0 0 = 安全区域未被锁定 1 = 安全区域永久性被锁定
4	LB2			0	
3	LB1			0	
2	LB0			0	
1	QUAD_NV	四线默认设置	非易失性	0	为QUAD提供默认状态。
0	SRP1_D	状态寄存器保护1默认设置	OTP	0	当IRP[2:0] = “111”时，可以编程SRP1_D位。 锁定SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的当前状态 1 = 寄存器永久锁定 0 = POR发生后，寄存器不受SRP1的保护

挂起擦除/编程状态位 (SUS_D) CR1NV[7]: 为CR1V[7]中的SUS位提供默认状态。用户不能编程该位。

补充保护位 (CMP_NV) CR1NV[6]: 为CR1V[6]中的CMP位提供默认状态。

安全区域锁定位 (LB3、LB2、LB1、LB0) CR1NV[5:2]: 提供安全区域的OTP写保护控制。将LB位设置为1时，将不能对安全区域进行其他编程或擦除操作。

四线数据宽度非易失性位 (QUAD_NV) CR1NV[1]: 为CR1V[1]中的QUAD位提供默认状态。WRR或WRAR命令会影响该位的状态。CR1NV[1] = 1时，在上电或复位时将默认运行四线数据带宽的命令。

状态寄存器保护1默认位 (SRP1_D) CR1NV[0]: 为CR1V[0]中的SRP1位提供默认状态。当IRP[2:0] = “111”时，用户可以对SRP1_DOTP位进行编程。当SRP1_D = “1”时，寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被永久性锁定。请参见第45页上的**状态寄存器保护 (SRP1、SRP0)**。

6.6.3.2 易失性配置寄存器1 (CR1V)

相关命令：读取配置寄存器1 (RDCR135h)、易失性寄存器的写使能 (WRENV50h)、写入寄存器 (WRR01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。通过使用RDCR1命令可以显示该寄存器。

表 6.10 易失性配置寄存器 1 (CR1V)

位	字段名称	功能	类型	默认状态	说明
7	SUS	挂起状态	易失性 只读	CR1NV	1 = 擦除/编程挂起 0 = 擦除/编程不挂起
6	CMP	补充保护	易失性		0 = 正常保护映射 1 = 转换保护映射
5	LB3	安全区域锁定位的易失性 副本	易失性 只读		用户不可写入 请参见CR1NV[5:2] 安全区域3:0的OTP锁定位3:0 0 = 安全区域未被锁定 1 = 安全区域永久性被锁定
4	LB2				
3	LB1				
2	LB0				
1	QUAD	四线I/O模式	易失性		1 = 四线 0 = 双线或串行
0	SRP1	状态寄存器保护1	易失性	锁定SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、 DLRNV和DLRV的当前状态 1 = 寄存器被锁定 0 = 寄存器未被锁定	

挂起状态位 (SUS) CR1V[7]: 挂起状态位用于指出器件是否进入擦除或编程挂起模式。用户不能对该状态位进行写操作。将挂起状态位置为“1”时，器件将进入擦除或编程挂起模式。将挂起状态位清除为“0”时，器件不处于擦除或编程挂起模式。请参考第94页上的8.6.5 编程或擦除挂起 (PES 75h) 一节，了解擦除/编程挂起/恢复命令的详细信息。**补充保护位 (CMP) CR1V[6]:** CMP与TBPROT、BP3、BP2、BP1和BP0位一起使用，用于为阵列保护映射提供更高的灵活性，这样可以为从半个阵列到整个阵列的空间提供保护。

LB[3:0] CR1V[5:2]: 这些位是CR1NV的相关OTP位的易失性复制。它们跟踪相关OTP版本的更改。

四线数据宽度 (QUAD) CR1V[1]: 设置为1时，这些位将器件的数据宽度切换到4比特四线模式。当CS#为低电平时，WP#变为IO2，IO3/RESET#变为有效的I/O信号。当CS#为高电平时，IO3/RESET#变为RESET#输入。WP#输入不被监控，因此它会正常运行，并且它被内部设置为高电平（无效）。可以正常执行串行和双线I/O读取命令，但在使用不同的数据路径宽度切换各个命令时不用为这些命令驱动WP#输入。同样，在执行这些命令期间，不需要驱动IO3/RESET#（CS#为低电平）。当使用读取四线输出、读取四线I/O和读取DDR四线I/O命令时，必须将QUAD位设置为1。通过执行易失性寄存器写操作，可在较短且准确的时间段内切换器件接口。这些时间段分别为 t_{QEN} （切换到QIO模式）和 t_{QEX} （返回到SPI模式）。因此，能立即以QIO协议方式发送以下命令。通过QPIEN和QPIEX命令进入/退出QPI模式，或将CR2V[3]设置为1时，将使用四线数据宽度模式（无论QUAD位的状态如何）。

状态寄存器保护1 (SRP1) CR1V[0]: 将SRP1位设置为1时，将通过防止对SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV寄存器进行的写操作保护这些寄存器的状态。请参见第45页上的状态寄存器保护 (SRP1、SRP0)。

如果SRP1位保持清除为0，SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV寄存器不受SRP1的保护。然而，这些寄存器可能受SRP0（SR1V[7]）和WP#输入的保护。

将一个逻辑1写入到SRP1位后，只有通过下电到上电周期或一次硬件复位才能将其清除为逻辑0。软件复位将不影响到SRP1位的状态。

CR1V[0] SRP1位是易失性的。上电后，SRP1的默认状态由CR1NV[0]中的SRP1_D决定。通过使用单个WRR或WRAR命令可以同时设置SRP1位并更新CR1V中的其他值。

6.6.4 配置寄存器2

配置寄存器2控制某些接口功能。通过使用读取任何寄存器和写入任何寄存器命令可以读取和更改这些寄存器位。该寄存器的非易失性版本允许为这些控制设置POR、硬件复位或软件复位状态。在正常操作期间，该寄存器的易失性版本控制着功能状态。

6.6.4.1 非易失性配置寄存器2 (CR2NV)

相关命令：非易失性寄存器的写使能 (WREN 06h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。

表 6.11 非易失性配置寄存器 2 (CR2NV)

位	字段名称	功能	类型	默认状态	说明
7	IO3R_NV	IO3_Reset	非易失性	0	1 = 使能 — 当CS#为高电平或四线模式被禁用 (CR1V[1]=0) 或QPI被禁用 (CR3V[3] = 0) 时, IO3_RESET作为IO3/RESET#输入使用 0 = 禁用 — IO3没有其他功能, 硬件复位被禁用。
6	OI_NV	输出阻抗		1	提供默认输出阻抗状态。请参见第34页上的表6.12. <i>输出阻抗控制</i>
5				1	
4	RFU	保留		0	保留供将来使用
3	QPI_NV	QPI		0	1 = 使能 — 使用QPI (4-4-4) 协议 0 = 禁用 — 使用传统SPI协议, 指令始终在SI线上串行传输为QPI模式提供默认状态。
2	WPS_NV	写保护选择		0	为WPS提供默认状态 0 = 传统保护 1 = 单块锁定
1	ADP_NV	上电时的地址长度		0	为地址长度提供默认状态 1 = 4字节地址 0 = 3字节地址
0	RFU	保留		0	保留供将来使用

IO3_Reset非易失性位CR2NV[7]: 该位控制IO3信号性能的POR、硬件复位或软件复位状态。由于传统SPI器件封装中的信号数量和连接有限, 所以几乎所有传统SPI器件都没有硬件复位输入信号。如果没有将IO3信号用于传输主机系统和存储器之间的信息, 那么FL-L器件系列允许将IO3信号作为硬件复位输入使用。通过使用该非易失性IO3_Reset配置位, 可以立即启动 (引导) 器件, 并且IO3作为RESET#信号使用。

输出阻抗非易失性位CR2NV[6:5]: 这些位控制IO信号输出阻抗 (驱动强度) 的POR、硬件复位或软件复位状态。通过多个驱动强度可以使输出阻抗与系统印刷电路板环境相互匹配, 从而能够尽量降低过冲和振铃。这些非易失性输出阻抗配置位允许器件使用合适的驱动强度立即启动 (引导)。

表 6.12 输出阻抗控制

CR2NV[6:5] 阻抗选项	对V _{SS} 的典型阻抗 (Ω)	对V _{DD} 的典型阻抗 (Ω)	注释
00	18	21	
01	26	28	
10	47	45	
11	71	64	工厂默认设置

QPI非易失性位CR2NV[3]: 该位为所有命令的预期指令宽度控制POR、硬件复位或软件复位状态。传统SPI命令始终在SI (IO0) 信号上以1位宽度 (串行I/O) 发送指令。FL-L器件系列也支持QPI模式。在该模式下, 主机系统和存储器之间所有的传输都是在IO0到IO3上实现的, 传输宽度为4比特, 包括所有指令。通过该非易失性QPI配置位可使器件立即在QPI模式下启动 (引导), 而不需在传统串行指令模式下进行。转到QPI模式的建议流程是先使用QPIEN (38h) 命令, WRR或WRAR命令也可以将CR2V[3]置1 (QPI模式)。QIO模式的易失性寄存器写操作拥有 (t_{QEN}) 时间, 用于将器件接口切换到QIO模式和 (t_{QEX}) 时间, 从而能够将器件切换到SPI模式。因此, 能以QPI协议方式立即发送以下命令。通过使用WRAR命令将CR2NV[3]编程为1, 然后轮询SR1V[0], 以确定编程操作是否完成。同样, 使用QPIEX (F5h) 命令退出QPI模式。也可以使用WRR或WRAR命令将CR2V[3]清除为0。

写保护选择非易失性CR2NV[2]: 该位控制写保护方法的POR、硬件复位或软件复位状态。在使能该非易失性配置位的情况下, 当器件启动 (引导) 时, 将使能单独模块锁存保护 (而不是传统的模块保护)。

上电时的地址长度、非易失性CR2NV[1]: 该位为所有命令的预期地址长度控制POR、硬件复位或软件复位状态。这些命令需要地址而且这些地址不被固定为3字节或4字节地址。多数传统的SPI命令需要一个地址, 这些命令通常使用3字节 (24位) 地址。如果器件大小超过128 Mb, 将需要一个4字节 (32位) 地址来访问整个存储器阵列。通过使用地址长度配置位可以将所有3字节地址命令更改为所期待的4字节地址。请参考第60页上的表8.3. *FL-L系列命令集 (按功能排列)*, 了解命令地址长度。该非易失性地址长度配置位允许器件在4字节地址模式立即启动 (引导), 而不是在传统的3字节地址模式下进行。

6.6.4.2 易失性配置寄存器2 (CR2V)

相关命令: 读取配置寄存器2 (RDCR2 15h)、读取任何寄存器 (RDAR65h)、易失性寄存器的写使能 (WRENV50h)、写入寄存器 (WRR01h)、写入任何寄存器 (WRAR71h)、进入4字节地址模式 (4BENB7h)、退出4字节地址模式 (4BEXE9h)、进入QPI (38h)、退出QPI (F5h)。通过使用RDCR2命令可以显示该寄存器。

表 6.13 易失性配置寄存器 2 (CR2V)

位	字段名称	功能	类型	默认状态	说明		
7	IO3R	IO3_Reset	易失性	CR2NV	1 = 使能 — 当CS#为高电平或四线模式被禁用 (CR1V[1] = 0) 或QPI被禁用 (CR3V[3] = 0) 时, IO3作为RESET#输入使用。 0 = 禁用 — IO3没有其他功能, IO3/RESET#输入的硬件复位功能被禁用。		
6	OI	输出阻抗			易失性	CR2NV	请参见第34页上的表6.12. <i>输出阻抗控制</i>
5							保留供将来使用
4	RFU	保留			易失性	CR2NV	1 = 使能 — 使用QPI (4-4-4) 协议 0 = 禁用 — 使用传统SPI协议, 指令始终在SI线上串行传输
3	QPI	QPI					0 = 传统块保护 1 = 单块锁定
2	WPS	写保护选择	易失性 只读	CR2NV	只读状态位 1 = 4字节地址 0 = 3字节地址		
1	ADP	上电时的地址长度			当前地址模式 1 = 4字节地址 0 = 3字节地址		
0	ADS	地址长度状态	易失性	CR2NV[1]	当前地址模式 1 = 4字节地址 0 = 3字节地址		

IO3复位CR2V[7]: 该位控制IO3/RESET#信号的状态。当CS#为高电平或四线模式 (CR1V[1] = 0) 被禁用或QPI (CR3V[3] = 0) 被禁用时, 在正常操作期间, 通过易失性IO3复位配置位, 可将IO3作为RESET#输入使用。

输出阻抗位CR2V[6:5]: 这些位控制IO信号输出阻抗 (驱动强度)。通过使用该易失性输出阻抗配置位, 用户可以在正常操作期间调整驱动强度。

QPICR2V[3]: 该位控制所有命令的预期指令宽度。通过使用易失性QPI配置位, 在正常操作期间, 器件可以进入和退出QPI模式。将该位设置为QPI模式时, 无论QIO模式 (CR1V[1]) 的设置如何, 四线模式都有效。将该位清除为传统的SPI模式时, QUAD位不受影响。通过使用QPIEN (38h) 命令可以将QPI CR2V[3]位设置为1, 通过使用QPIEX (F5h) 命令可以将其设置为0。

表 6.14 QPI 和 QIO 模式控制位

QPI CR2V[3]	QUAD CR1V[1]	说明
0	0	SIO模式: 单线和双线读取, WP#/IO2输入作为WP#引脚使用, IO3/RESET#输入作为RESET#引脚使用
0	1	QIO模式: 单线、双线和四线读取, WP#/IO2输入作为IO2使用, IO3/RESET#输入作为IO3或RESET#引脚使用
1	X	QPI模式: 四线读取, WP#/IO2输入作为IO2使用, IO3/RESET#输入作为IO3或RESET#引脚使用

写保护选择位CR2V[2]: 通过该位可以选择阵列保护方法: 第46页上的 *传统块保护* 或第51页上的 *单块锁定 (IBL) 保护*。通过易失性配置位, 用户可以在正常操作期间更改保护方法。

上电 (ADP) CR2V[1]时的地址长度: 该位为只读位, 它显示了上电复位、硬件复位或软件复位后各条命令的地址长度。这些命令使用长度不固定的地址 (3字节或4字节)

地址长度状态 (ADS) CR2V[0]: 该位控制所有命令的预期地址长度。这些命令需要地址, 这些地址不被固定为3字节或4字节地址。请参考第60页上的表8.3. *FL-L系列命令集 (按功能排列)*, 了解命令地址长度。通过易失性地址长度配置位, 用户可以在正常操作期间更改地址的长度。4字节地址模式 (4BEN) 命令直接将该位设置为4字节地址模式, (4BEX) 退出命令将该位设置回3字节地址模式。更新地址长度非易失性CR2NV[1]位时, 该位也被更新。

6.6.5 配置寄存器3

配置寄存器3控制着主闪存阵列读命令的连续回卷状态和读取延迟。连续回卷配置不会影响读取非主闪存阵列的区域的命令, 如: 读取寄存器或安全区域的命令。由于在POR、硬件复位或软件复位期间寄存器中的内容被复制到寄存器的易失性版本内, 所以该寄存器的非易失性版本允许设置该控制操作的启动 (引导) 状态。正常操作期间, 该寄存器的易失性版本控制功能状态。

通过使用读取配置3 (RDCR3 33h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h) 命令可以对这些寄存器位进行读取和更改操作。通过使用设置连续长度 (77h) 命令也可以写入该寄存器的易失性版本。

6.6.5.1 非易失性配置寄存器3 (CR3NV)

相关命令: 非易失性寄存器的写使能 (WREN 06h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)。

表 6.15 非易失性配置寄存器 3 (CR3NV)

位	字段名称	功能	类型	默认状态	说明
7	RFU	保留	非易失性	0	保留供将来使用
6	WL_NV	回卷长度默认设置		1	00 = 8字节回卷 01 = 16字节回卷 10 = 32字节回卷 11 = 64字节回卷
5				1	
4				WE_NV	回卷使能默认设置
3	RL_NV	读取延迟默认设置		1	0到15个延迟 (虚拟) 周期 (随后读地址或连续模式位)。
2				0	
1				0	
0				0	

回卷长度非易失性位CR3NV[6:5]: 这些位控制着回卷读取长度和对齐的POR、硬件复位或软件复位状态。

回卷使能非易失性位CR3NV[4]: 该位控制着回卷使能的POR、硬件复位或软件复位状态。受回卷使能影响的命令包括: 读取四线I/O、读取QPI、读取DDR四线I/O和读取DDRQPI。该配置位允许器件在连续回卷读取模式下立即启动 (引导), 而不是在传统的连续读取模式下进行。

读取延迟非易失性CR3NV[3:0]: 这些位控制着所有可变延迟读命令中读取延迟 (虚拟周期) 的POR、硬件复位或软件复位状态。以下读命令在地址或模式的结束和读取数据开始返回到主机之间有一个可变的延迟周期:

- 对于以下命令, 所有时钟频率的延迟均为一个虚拟周期。如果默认延迟代码为0, 该延迟为一个虚拟周期。
 - 数据学习模式读取DLPRD (1-1-1) 或 (4-4-4)
 - IRP读取IRPRD (1-1-1) 或 (4-4-4)
 - 保护寄存器读取PRRD (1-1-1) 或 (4-4-4)
 - 密码读取PASSRD (1-1-1) 或 (4-4-4)
- 以下命令的延迟/时钟频率如表 6.16和表 6.17所示。如果默认延迟代码为0, 该延迟为8个虚拟周期。
 - 快速读取FAST_READ (1-1-1)
 - 四线O读取QOR、4QOR (1-1-4)
 - 双线O读取DOR、4DOR (1-1-2)
 - 双线I/O读取DIOR、4DIOR (1-2-2)
 - 四线I/O读取QIOR、4QIOR (1-4-4) 或 (4-4-4)

- DDR四线I/O读取DDRQIOR、4DDRQIOR (1-4-4)
- 安全区域读取SECR (1-1-1) 或 (4-4-4)
- 读取任何寄存器RDAR (1-1-1) 或 (4-4-4)
- 读取串行闪存可发现参数RSFDP (1-1-1) 或 (4-4-4)

非易失性读延迟配置位设置需要使用的读延迟（虚拟周期）数量，因此可以使用主机系统的合适读延迟立即启动（引导）器件

表 6.16 延迟代码（周期）与频率

延迟代码 0	读取命令最大频率 (MHz)						
	快速读取 (1-1-1)	双线输出读取 (1-1-2)	双线I/O读取 (1-2-2)	四线输出读取 (1-1-4)	四线I/O读取 (1-4-4)	四线I/O读取 QPI (4-4-4)	DDR 四线I/O (1-4-4) QPI (4-4-4)
	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 4 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 2 虚拟周期 = 8	模式周期 = 2 虚拟周期 = 8	模式周期 = 1 虚拟周期 = 8
1	50	50	75	35	35	35	20
2	65	65	85	45	45	45	25
3	75	75	95	55	55	55	35
4	85	85	108	65	65	65	45
5	95	95	108	75	75	75	55
6	108	105	108	85	85	85	60
7	108	108	133	95	95	95	66
8	108	108	133	108	108	108	66
9	133	133	133	115	115	115	66
10	133	133	133	115	115	115	66
11	133	133	133	120	120	120	66
12	133	133	133	120	120	120	66
13	133	133	133	133	133	133	66
14	133	133	133	133	133	133	66
15	133	133	133	133	133	133	66

表 6.17 延迟代码（周期）与频率

延迟代码 0	读取命令最大频率 (MHz)					
	读取任何寄存器 (1-1-1)	读取任何寄存器 QPI (4-4-4)	安全区域读取 (1-1-1)	安全区域读取QPI (4-4-4)	读取SFDP RSFDP (1-1-1)	读取SFDP RSFDP QPI (4-4-4)
	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8	模式周期 = 0 虚拟周期 = 8
1	50	15	50	15	50	15
2	65	25	65	25	65	25
3	75	35	75	35	75	35
4	85	45	85	45	85	45
5	95	55	95	55	95	55
6	108	65	108	65	108	65
7	108	75	108	75	108	75
8	108	85	108	85	108	85
9	133	95	133	95	133	95
10	133	108	133	108	133	108
11	133	115	133	115	133	115
12	133	115	133	115	133	115
13	133	120	133	120	133	120
14	133	120	133	120	133	120
15	133	133	133	133	133	133

注意:

1. SCK频率 > 133 MHz SDR, 或66MHz DDR不受该器件系列的支持。
2. 双线I/O、四线I/O、QPI、DDR四线I/O和DDR QPI命令协议包括位于地址后面的连续读取模式位。这些位的时钟周期不计入下表中的延迟周期。例如：传统的四线I/O命令在地址后面有两个连续的读取模式周期。因此，对于无其他读取延迟的传统四线I/O命令，受支持的频率如下表所示（读取延迟为0）。通过递增可变读取延迟，可增大四线I/O命令的频率，从而允许执行受支持的最大133 MHz频率和QPI受支持的最大133 MHz频率的操作。
3. 其他命令具有固定的延迟。例如，读取始终有0读取延迟，读取唯一ID具有32个虚拟周期，从深度断电释放则具有24个虚拟周期。

6.6.5.2 易失性配置寄存器3 (CR3V)

相关命令：读取配置3 (RDCR3 33h)、易失性寄存器的写使能 (WRENV 50h)、写入寄存器 (WRR 01h)、读取任何寄存器 (RDAR 65h)、写入任何寄存器 (WRAR 71h)、设置连续长度 (SBL 77h)。通过使用RDCR3命令可以显示该寄存器。

表 6.18 易失性配置寄存器 3 (CR3V)

位	字段名称	功能	类型	默认状态	说明
7	RFU	保留	易失性	CR3NV	保留供将来使用
6	WL	回卷长度			00 = 8字节回卷
5					01 = 16字节回卷
					10 = 32字节回卷
					11 = 64字节回卷
4	WE	回卷使能			0 = 回卷使能
3	RL	读延迟			1 = 回卷禁用
2					0到15个延迟（虚拟）周期（随后读地址或连续模式位）
1					
0					

回卷长度CR3V[6:5]: 在正常操作期间，这些位控制着回卷读取长度和对齐。通过使用这些易失性配置位，用户可以在正常操作期间调整连续回卷读取长度。

回卷使能CR3V[4]: 这些位控制着连续回卷功能。通过使用该易失性配置位，在正常操作期间，器件可以进入和退出连续回卷读取模式。

CR3V[4] = 1时，回卷模式被禁用，而且可以连续读取无限长度的数据。

CR3V[4] = 0时，将使能回卷模式而且只能从读命令提供的字节地址开始读取固定长度和对齐的8、16、32或64字节组，并且在该字节组边界上回卷。

读延迟CR3V[3:0]: 这些位设置可变延迟读命令中的读延迟（虚拟周期）。通过使用这些易失性配置位，用户可以在正常操作期间调整读取延迟，从而按需要优化不同命令的延迟或不同频率的延迟。

6.6.6 单独和区域保护寄存器（IRP）

相关命令：IRP读取（IRPRD 2Bh）和IRP编程（IRPP 2Fh）、读取任意寄存器（RDAR 65h）、写入任意寄存器（WRAR 71h）。

IRP寄存器指的是一个16位的OTP存储位置，用于永久地配置单独和区域保护（IRP）特性的状态。IRP中没有用户可编程的易失性位，它的所有定义位都是OTP（一次性编程）。

各IRP位的默认状态是由赛普拉斯编程的。

表 6.19 IRP 寄存器（IRP）

位	字段名称	功能	类型	默认状态	说明
15到7	RFU	保留	OTP	所有位为1	保留供将来使用
6	SECRRP	安全区域3读取密码模式使能位	OTP	1	0 = 选择安全区域3读取密码模式 1 = 不选择安全区域3读取密码模式 如果IRP[2:0] = “111”，则IRP[6]可编程
5	RFU	保留	OTP	1	保留供将来使用
4	IBLLBB	IBL锁定启动位	OTP	1	0 = 所有单独IBL位在上电时均处于不受保护状态，并被设为“1” 1 = 所有的单独IBL位在上电时均处于受保护状态，并被设为“0” 如果IRP[2:0] = “111”，IRP[4]可编程
3	RFU	保留	OTP	1	保留供将来使用
2	PWDMLB	密码保护模式锁定位	OTP	1	0 = 密码保护模式永久被使能。 1 = 密码保护模式并非永久被使能。 如果IRP[2:0] = “111”，IRP[2]可编程
1	PSLMLB	电源锁定保护模式锁定位	OTP	1	0 = 电源锁定保护模式被永久使能。 1 = 电源锁定保护模式没有被永久使能。 如果IRP[2:0] = “111”，IRP[1]可编程
0	PERMLB	永久保护锁定位	OTP	1	0 = 永久保护模式被永久使能。 1 = 永久保护模式没有被永久使能。 如果IRP[2:0] = “111”，IRP[0]可编程

安全区域读取密码模式使能（SECRRP）IRP[6]: 写入数值0时，SECRRP会使能安全区域3读取密码模式，而PWDMLB位IRP[2]将同时或稍后被编程。仅在IRP[2:0] = 111时，才能对SECRRP位进行编程；否则，编程操作会失败，并且P_ERR的值被设置为1。请参考第57页上的7.7.4 安全区域读取密码保护一节。

IBL锁定启动位（IBLLBB）IRP[4]: 默认状态下，它的值为1；在上电、硬件复位或软件复位后的保护状态，所有单独的IBL位均被设置为0。要想编程或擦除该阵列，必须在执行编程或擦除指令前发送全局IBL解锁或扇区/块IBL解锁指令。所有单独的IBL位被编程为0时，在上电、硬件复位或软件复位后，各个扇区都进入不受保护状态。仅在IRP[2:0] = 111时，才能对IBLLBB位进行编程；否则，编程操作失败，并且P_ERR的值被设置为1。请参见第51页上的7.6.2 单块锁定（IBL）保护一节。

密码保护模式锁定位（PWDMLB）IRP[2]: 被设置为0时，将永久地选中密码保护模式，以保护安全区域2、3和指针区域。仅在IRP[2:0] = 111时，才能对SECRRP位进行编程；否则，编程操作失败，并且P_ERR的值被设置为1。请参考第56页上的7.7.3 密码保护模式一节。

在通过设置IRP[2] = “0”选中密码保护模式后，所有IRP位将被锁定，并永久地禁止对其进行的任意编程。尝试对任意IRP位进行编程会导致编程错误，并且P_ERR的值被设置为1。

设置密码模式（IRP[2] = 0）之前，必须提前对密码进行编程并验证。

电源锁定保护模式锁定位 (PSLMLB) IRP[1]: 该位被设置为0时, 会永久地选中电源锁定保护模式。仅在IRP[2:0] = 111时, 才能对IRP[1]位进行编程; 否则, 编程操作会失败, 并且P_ERR的值被设置为1。

在通过设置IRP[1] = 0选中电源锁定保护模式后, 所有IRP位将被锁定, 并永久地禁止对其进行的任意编程。尝试对任意IRP位进行编程会导致编程错误, 并且P_ERR的值被设置为1。请参见第55页上的7.7.1 IRP寄存器一节。

永久性保护锁定位 (PERMLB) IRP[0]: 该位被设置为0时, 将一直保护指针区域和安全区域2和3。它向指针区域和安全区域2、3提供的永久性的保护方式更为简单(无需使用密码或PRL命令)。请参见第55页上的7.7.1 IRP寄存器一节。

PWDMLB (IRP[2])、PSLMLB (IRP[1]) 和PERMLB (IRP[0]) 之间是相互排除的, 其中只能有一位被设置为0。仅在IRP[2:0] = 111时, 才能对各IRP位进行编程。当IRP[2:0]的值不等于111时, 尝试对IRP位进行编程会引起编程错误, 并且P_ERR的值被设置为1。在系统配置期间, 应一直选中IRP保护模式, 这样可以确保恶意程序以后不会选中意外的保护模式。在通过IRP模式选择来锁定所有保护配置后, 恶意程序在后续时间内将不能对保护模式进行任意更改。

6.6.7 密码寄存器 (PASS)

相关命令: 密码读取 (PASSRDE7h) 和密码编程 (PASSPE8h)、读取任意寄存器 (RDAR65h) 写入任意寄存器 (WRAR71h)。PASS (密码) 寄存器指的是64位的OTP存储器位置, 该寄存器用于永久性定义单独和区域保护 (IRP) 特性的密码。PASS中没有用户可编程的易失性位, 它的所有定义位都是OTP。可以使用PASS的一个易失性副本来满足易失性寄存器的读取延迟要求, 但该易失性寄存器并不是用户可写的空间, 或没有详细的说明。在设置IRP[2] = 0后, 不能读取或编程密码。请参见第39页上的表6.19. IRP寄存器 (IRP)。

表 6.20 密码寄存器 (PASS)

位	字段名称	功能	类型	默认状态	说明
63到0	PWD	隐藏密码	OTP	FFFFFFFF-FFFFFFFFh	64位密码的非易失性OTP存储。通过将IRP寄存器位2设为0选择密码保护模式后, 密码将不再可读。

6.6.8 保护寄存器 (PR)

相关命令: 保护寄存器读取 (PRRD A7h)、保护寄存器锁定 (PRL A6h) 或读取任意寄存器 (RDAR 65h)。

PR中没有用户可编程的非易失性位, 它的所有定义位都是易失性只读状态位。RFU位的默认状态是由硬件设置的。PR寄存器没有非易失性版本。

NVLOCK位用于保护安全区域2、3和指针区域保护。当NVLOCK[0] = 0时, 不能对安全区域2、3和指针区域保护进行任何更改。

表 6.21 保护状态寄存器 (PR)

位	字段名称	功能	类型	默认状态	说明
7	RFU	保留	易失性 只读	00h	保留供将来使用
6	SECRRP	安全区域读取密码		IRP[6]	0 = 当NVLOCK = 0时, 安全区域3受密码保护, 不能对其执行读取操作 1 = 安全区域3不受密码保护, 可以对其进行读取操作
5	RFU	保留		0	保留供将来使用
4	RFU	保留		0	保留供将来使用
3	RFU	保留		0	保留供将来使用
2	RFU	保留		0	保留供将来使用
1	RFU	保留		0	保留供将来使用
0	NVLOCK	保护非易失性配置		IRP[2]和IRP[0]	0 = 安全区域2和3以及指针区域受保护, 不能对其进行写入操作 1 = 可以对安全区域2和3以及指针区域进行写入操作。 1

注意:

1. 保护寄存器锁定 (PRL) 命令将NVLOCK的值设置为1。

6.6.9 单独块锁定访问寄存器 (IBLAR)

相关命令：IBL读取 (IBLRD 3Dh或4IBLRD E0h)、IBL锁定 (IBL 36h或4IBL E1h)、IBL解锁 (IBLUL 39h或4IBUL E2h)、全局IBL锁定 (GBL 7Eh)、全局IBL解锁 (GBUL 98h)。

IBLAR中没有用户可编程的非易失性位，它的所有位是IBL阵列中易失性位的副本。IBL阵列的默认状态是由硬件设置的。IBLAR寄存器没有非易失性版本。

表 6.22 IBL 访问寄存器 (IBLAR)

位	字段名称	功能	类型	默认状态	说明
7到0	IBL	对单独扇区/块的IBL进行读取/写入操作	易失性	如果IRP[4] = 1, 为00h 否则, 为FFh	00h = 通过IBL、4IBL和GBL命令将所寻址的扇区/块的IBL设置为“0”，这样便不能对扇区进行编程或擦除操作。 FFh = 通过IBUL、4IBUL和GBUL命令将所寻址的扇区/块的IBL设置为“1”，这样能够对扇区进行编程或擦除操作。

注意

1. 请参见第51页上的图7.2. 单块锁定/指针区域保护控制。
2. 可以使用IBLRD和4IBLRD命令读取IBL位的值。

6.6.10 指针区域保护寄存器 (PRPR)

相关命令：设置指针区域 (SPRP FBh或4SPRP E3h)、读取任意寄存器 (RDAR 65h)、写入任意寄存器 (WRAR 71h)。

PRPR包含用户可编程的非易失性位。PRPR位的默认状态是由硬件设置的。PRPR寄存器没有易失性版本。更多有关信息，请参见第52页上的7.6.3 指针区域保护 (PRP) 一节。

表 6.23 PRP 寄存器 (PRPR)

位	字段名称	功能	类型	默认状态	说明
A31到A25	RFU	保留	非易失性	11111111b	保留供将来使用
A24	PRPAD	PRP地址		1	指针地址 在S25FL256L中为A24 在S25FL128L中为保留供将来使用的位
A23到A16				FFh	指针地址A23到A16
A15到A12				Fh	指针地址A15到A12
A11	PRPALL	PRP保护所有扇区		1	0 = 保护指针区域中的所选扇区 1 = 保护所有扇区
A10	PRPEN	PRP使能		1	0 = 使能指针区域保护 1 = 禁用指针区域保护
A9	PRPTB	PRP顶部/底部		1	0 = 指针区域保护从顶部 (高地址) 开始 1 = 指针区域保护从底部 (低地址) 开始
A8	RFU	保留		1	保留供将来使用
A7到A0	RFU	保留		FFh	保留供将来使用

6.6.11 DDR数据学习寄存器

相关命令：编程DLRNV (PDLRNV 43h)、写入DLRV (WDLRV 4Ah)、读取数据学习模式 (DLPRD 41h)、读取任意寄存器 (RDAR 65h)、写入任意寄存器 (WRAR 71h)。

数据学习模式 (DLP) 位于一个8位非易失性数据学习寄存器 (DLRNV) 内，它也被称为一个8位易失性数据学习寄存 (DLRV)。从赛普拉斯出厂时，DLRNV的默认值为00h。一旦对DLRNV进行了编程，便不能重新编程或擦除它；DLRNV中数据模式的副本也会被写入到DLRV内。可以随时向DLRV进行写操作，但如果正在进行硬件和软件复位或上电周期时，数据模式将返回到它在DLRNV中的状态。在执行SPI DDR模式中所描述的学习阶段内，DLP将位于DLRV中。在每个时钟沿上，各个IO均输出相同的DLP值。例如：如果DLP为34h (或二进制00110100)，那么在第一个时钟边沿上，所有IO都输出数值0；然后在第二个时钟边沿上，所有IO都输出数值0，在第三个时钟边沿上都输出数值1，并以此类推。

DLRV的值为00h时，DDR指令中的虚拟阶段不会显示前导码数据模式。

表 6.24 非易失性数据学习寄存器（DLRNV）

位	字段名称	功能	类型	默认状态	说明
7到0	NVDLP	非易失性数据学习模式	OTP	00h	在DDR读取命令延迟（虚拟）周期中传输到主机的OTP值，用以为主机提供一个培训模式，从而准确地确定所收数据位中的中心点。

表 6.25 易失性数据学习寄存器（DLRV）

位	字段名称	功能	类型	默认状态	说明
7到0	VDLP	易失性数据学习模式	易失性	在POR或复位期间提取DLRNV的值	NVDLP的易失性副本，用于使能并将数据学习模式（DLP）传送到输出端。系统操作期间，主机可以修改VDLP。

7. 数据保护

7.1 安全区域

该器件具有一个独立于主闪存阵列的1024字节的地址空间。此空间分为四个独立锁定长度为256字节的区域。请参见第26页上的6.5 [安全区域地址空间](#)一节。

安全区域存储器空间用于增强系统安全性。可以使用各个数据值将闪存组件与系统CPU/ASIC“配对”起来，用于防止设备替换。安全区域地址空间是由安全区域锁定或保护寄存器NVLOCK位（PR[0]）保护。请参见第43页上的7.1.4 [安全区域锁定](#)（LB3、LB2、LB1、LB0）一节。

7.1.1 读取安全区域存储器空间。

安全区域读取命令（SECRR）使用与快速读取使用了相同的协议。超过安全区域地址有效范围（1024字节）的任意读取操作会引起不正确的数据。请参见第99页上的8.7.3 [安全区域读取](#)（SECRR 48h）一节。

当NVLOCK = 0时，通过进行以下设置：PWDMLB位IRP[2] = 0和SECRRP位IRP[6] = 0，可使安全区域3进入密码保护模式，禁止发生读操作。

7.1.2 编程安全区域

安全区域编程指令（SECRP）使用与页面编程指令相同的协议。请参见第98页上的8.7.2 [安全区域编程](#)（SECRP 42h）一节。

第26页上的表6.3中显示的是安全区域编程的有效地址范围。超出该有效地址范围的任意安全区域编程操作都会被忽略，并且SR2V[5]中的P_ERR不被设置为1。

通过将PWDMLB位IRP[2]设置为0，可将安全区域2和3进入密码保护模式，并禁止编程操作。

7.1.3 擦除安全区域

安全区域擦除指令（SECRE）使用与扇区擦除指令相同的协议。请参见第98页上的8.7.1 [安全区域擦除](#)（SECRE 44h）一节。

第26页上的表6.3中显示的是安全区域擦除的有效地址范围。超出该有效地址范围的任意安全区域擦除操作将被忽略，并且SR2V中的E_ERR不被设置为1。

通过将PWDMLB位IRP[2]设置为0，可使安全区域2和3进入密码保护模式，并禁止发生擦除操作。

7.1.4 安全区域锁定（LB3、LB2、LB1、LB0）

安全区域锁定（LB3、LB2、LB1、LB0）是配置寄存器1（CR1NV[5:2]）中非易失性的一次性编程位，用于控制各安全区域的写保护功能，并显示其状态。安全区域0至3的默认状态为解锁。通过使用写入状态寄存器命令或写入任意寄存器命令，可单独将LB[3:0]的值设置为1。LB[3:0]是一次性可编程（OTP）位。一旦被设置为1，其相应的256字节安全区域将永久性变成“只读”空间。

7.2 深度掉电

深度掉电（DPD）指令能够提供另一种数据保护方式，使得所有指令（深度掉电（RES ABh）和硬件复位过后的释放指令除外）被忽略于深度掉电（DPD）模式因此，在DPD模式期间，该指令会禁止所有编程或擦除操作。

7.3 写入使能指令

7.3.1 写入使能 (WREN)

执行用于指定非易失性数据的任意指令前，先要写好写使能 (WREN) 指令。WREN指令用于设置写使能锁存 (WEL) 位。执行上电、硬件和软件复位期间或在完成下面指令后，WEL位将被清零 (禁用写操作)：

- 复位
- 页面编程 (PP或4PP)
- 四线页编程 (QPP或4QPP)
- 扇区擦除 (SE或4SE)
- 半块擦除 (HBE或4HBE)
- 块擦除 (BE或4BE)
- 芯片擦除 (CE)
- 写入禁用 (WRDI)
- 写入寄存器 (WRR)
- 写入任意寄存器 (WRAR)
- 安全区域擦除 (SECRE)
- 安全区域字节编程 (SECRP)
- 单独和区域保护寄存器 (IRPP)
- 密码编程 (PASSP)
- 清除状态寄存器 (CLSR)
- 设置指针区域保护 (SPRP或4SPRP)
- 编程非易失性数据学习寄存器 (PDLRNV)
- 写入易失性数据学习寄存器 (WDLRV)

7.3.2 易失性寄存器的写使能 (WRENV)

执行用于指定易失性寄存器数据的任意写入寄存器 (WRR) 指令之前，先要编写好写使能易失性 (WRENV) 指令。

7.4 写入保护信号

当不处于四线模式 (CR1V[1] = 0) 或QPI模式 (CR2V[3] = 0) 时, 写入保护 (WP#) 输入和状态寄存器保护0 (SRP0) 位 (SR1NV[7]) 一起提供由硬件输入信号控制的保护。当WP#为低电平和SRP0的值为1时, 状态寄存器1 (SR1NV和SR1V)、配置寄存器 (CR1NV、CR1V、CR2NV、CR2V、CR3NV和CR3V) 以及DDR数据学习寄存器 (DLRNV和DLRV) 的状态不会发生改变。这样会防止禁用或更改由传统的块保护位或安全区域锁定位指定的保护。请参见第27页上的6.6.1 状态寄存器1一节。

7.5 状态寄存器保护 (SRP1、SRP0)

配置寄存器和状态寄存器 (CR1V[0]和SR1V[7]) 中的状态寄存器保护位 (SRP1和SRP0) 都是易失性的。SRP位规定SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的写保护方式: 软件保护、硬件保护或电源锁定保护。

表 7.1 状态寄存器保护位 (高安全性)

SRP1_D CR1NV[0]	SRP1 CR1V[0]	SRP0 SR1V[7]	WP#	状态寄存器	说明
0	0	0	X	软件保护	WP#引脚没有任何控制。可以对SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV进行写操作。(工厂默认设置)
0	0	1	0	硬件保护	当WP#引脚为低电平时, SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被锁定, 不能对它们进行写操作。(1) (4)
0	0	1	1	无硬件保护	当WP#引脚为高电平时, SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被解锁, 可以对它们进行写操作。(1)
0	1	X	X	电源锁定	SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV受保护, 不能对它们进行写操作, 直到下一个下电和上电周期到来为止。(2)
1	1	X	X	一次性编程	当SRP1_D CR1NV[0] = 1时, SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV永久性受保护, 不能对它们进行写操作。(3)

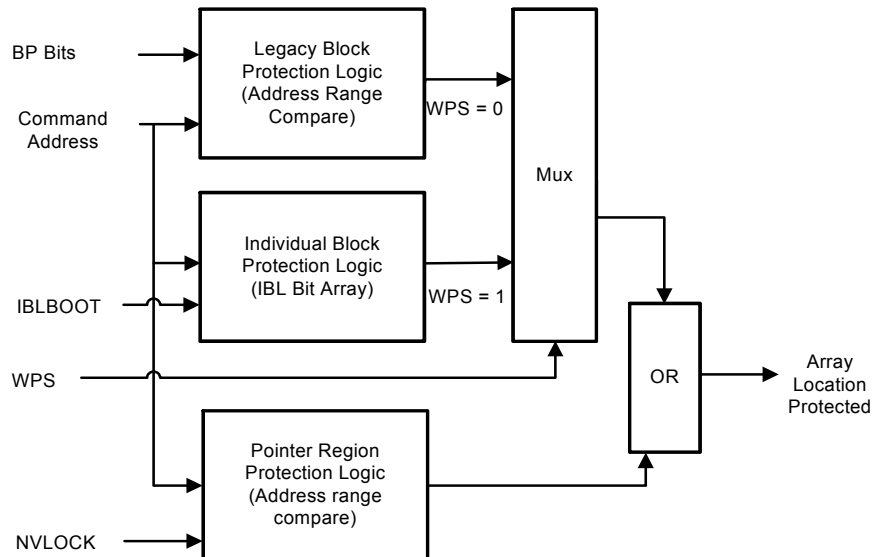
注意:

- SRP0是在下电、上电周期、软件或硬件复位过后SRP0_NV (SR1NV[7]) 的默认状态中被重新加载的。上电时, 若想要使用WP#引脚使能硬件保护模式, 需要将SRP0_NV位设置为1。
- 当SRP1 = 1时, 如果发生下电、上电周期或硬件复位事件, SRP1的值会变为0 (因为该位是从SRP1_D中重新加载的)。
- 只有在IRP[2:0] = 111时, 才能写入SRP1_D。SRP1_D CR1NV[0] = 1时, 下电、上电周期或硬件复位事件会从SRP1_D中将SRP1位重新加载为1。由于易失性位SRP1是不可写位, 因此它会提供OTP保护功能。当SRP1_D的值被设置为1时, 应将SRP0_NV设置为1, 以表示正在使用OTP保护。
- 使能QPI或QIO模式 (CR2V[3]或CR1V[1] = 1) 时, 内部WP#信号为1。这是因为当使能其中任意一种模式时, WP#外部输入都将作为IO2使用。当SRP1-SRP0 = 01b时, 可有效关闭硬件保护。寄存器SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV被解锁, 可以对这些寄存器进行写操作。
- WIP、WE和SUS (SR1[1:0]和CR1[7]) 是易失性的只读状态位, 它们一直不受写入状态寄存器指令的影响。
- 由SRP位和WP#保护 (如表中所示) 时, 将不能向SR1NV、CR1NV、CR2NV和CR3NV的非易失性版本执行写操作。如果在执行写入状态寄存器 (01h) 指令或写入任意寄存器 (71h) 指令之前先执行写使能 (06h) 指令, 将选用这些状态寄存器位的非易失性版本执行写操作。
- 在由SRP位和WP#保护 (如表中所示) 时, 将不能向SR1V、CR1V和CR2V的易失性版本执行写操作。如果在执行写入状态寄存器 (01h) 指令前先执行了写使能易失性状态寄存器 (50h) 指令; 或在执行写入任意寄存器 (71h) 指令前先执行了写使能 (06h) 指令, 那么会选择这些状态寄存器位的易失性版本来执行写操作。
- 易失性CR3V位不受SRP位的保护, 并且在执行写入状态寄存器 (01h) 指令前, 可通过易失性 (50h) 写使能指令随时写入这些位。另外, 还可以使用WRAR (71h) 或SBL (77h) 指令对CR3V寄存器中的位执行写操作。
- 在系统上电和引导代码执行期间: 通过可靠的引导代码能够判定是否需要修改以下寄存器的值: SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV的值。无需任何更改时, 可以将SRP1位 (CR1V[0]) 的值设置为1, 以确保系统在上电时保持正常操作期间, SR1NV、SR1V、CR1NV、CR1V、CR2NV、CR2V、CR3NV、DLRNV和DLRV寄存器能够维持其状态。

7.6 阵列保护

存储器阵列保护分成三种类型：传统块（LBP）、单独块锁定（IBL）和指针区域（PRP）。用户可以使用写保护选择（WPS）位来使能下面两种保护机制中的一种：传统块（LBP）保护（WPS CR2V[2] = 0）或单独块锁定（IBL）保护（WPS CR2V[2] = 1）。请参见第35页上的易失性配置寄存器2（CR2V）。一次只能使能一种保护机制。传统块保护是默认的保护机制，它与IBL保护方案相互独立。通过执行设置指针区域保护指令或在A10 = 0时执行WRAR指令，可以使能指针区域保护。请参考第104页上的指针区域命令。被使能时，指针区域保护与传统块保护或单独块锁定保护之间是逻辑OR关系。

图7.1 LBP、IBL和PRP阵列保护的WPS选择



7.6.1 传统块保护

可以使用传统块保护位（S25FL256L），状态寄存器位BP3、BP2、BP1、BP0--SR1V[5:2]（S25FL128L），状态寄存器位BP2、BP1、BP0 -- SR1V[4:2]以及配置寄存器TBPROT位（SR1V[6] S25FL256L）（SR1V[5] S25FL128L），CMP（CR1V[6]位和SEC（SR1V[5] S25FL128L）来禁止对主闪存阵列中某个地址范围进行的所有编程和擦除操作。该范围的大小由BP位的值决定，并且该范围的上限或下限起始点由配置寄存器（SR1V[6]S25FL256L）（SR1V[5]S25FL128L，）的TBPROT位决定。CMP位（CR1V[6]）被置1时，将执行互补保护。

被使能时，指针区域保护会与传统块保护区域进行逻辑OR运算。

表 7.2 S25FL128L 块保护 (CMP = 0)

状态寄存器					128L (128 Mb) 块保护 (CMP = 0)			
SEC	TBPROT	BP2	BP1	BP0	受保护块的数量	受保护的地址范围	受保护的空間	受保护的比率
X	X	0	0	0	无	无	无	无
0	0	0	0	1	252 ~ 255	FC0000h - FFFFFFFh	256 kB	高层1/64
0	0	0	1	0	248 ~ 255	F80000h - FFFFFFFh	512 kB	高层1/32
0	0	0	1	1	240 ~ 255	F00000h - FFFFFFFh	1 MB	高层1/16
0	0	1	0	0	224 ~ 255	E00000h - FFFFFFFh	2 MB	高层1/8
0	0	1	0	1	192 ~ 255	C00000h - FFFFFFFh	4 MB	高层1/4
0	0	1	1	0	128 ~ 255	800000h - FFFFFFFh	8 MB	高层1/2
0	1	0	0	1	0 ~ 3	000000h - 03FFFFFFh	256 kB	低层1/64
0	1	0	1	0	0 ~ 7	000000h - 07FFFFFFh	512 kB	低层1/32
0	1	0	1	1	0 ~ 15	000000h - 0FFFFFFh	1 MB	低层1/16
0	1	1	0	0	0 ~ 31	000000h - 1FFFFFFh	2 MB	低层1/8
0	1	1	0	1	0 ~ 63	000000h - 3FFFFFFh	4 MB	低层1/4
0	1	1	1	0	0 ~ 127	000000h - 7FFFFFFh	8 MB	低层1/2
X	X	1	1	1	0 ~ 255	000000h - FFFFFFFh	16 MB	全部
1	0	0	0	1	255	FFF000h - FFFFFFFh	4 kB	高层 1/4096
1	0	0	1	0	255	FFE000h - FFFFFFFh	8 kB	高层1/2048
1	0	0	1	1	255	FFC000h - FFFFFFFh	16 kB	高层1/1024
1	0	1	0	X	255	FF8000h - FFFFFFFh	32 kB	高层1/512
1	0	1	1	0				
1	1	0	0	1	0	000000h - 000FFFh	4 kB	低层1/4096
1	1	0	1	0	0	000000h - 001FFFh	8 kB	低层1/2048
1	1	0	1	1	0	000000h - 003FFFh	16 kB	低层1/1024
1	1	1	0	X	0	000000h - 007FFFh	32 kB	低层1/512
1	1	1	1	0				

注意:

1. X = 无需关注。

表 7.3 S25FL128L (128 Mb) 块保护 (CMP = 1)

状态寄存器					128L传统块保护 (CMP = 1)			
SEC	TBPORT	BP2	BP1	BP0	受保护块的数量	受保护的地址范围	受保护的空間	受保护的的比例
X	X	0	0	0	0 ~ 255	000000h - FFFFFFFh	16 MB	全部
0	0	0	0	1	0 ~ 251	000000h - FBFFFFh	16,128 kB	低层63/64
0	0	0	1	0	0 ~ 247	000000h - F7FFFFh	15,872 kB	低层31/32
0	0	0	1	1	0 ~ 239	000000h - EFFFFFFh	15 MB	低层15/16
0	0	1	0	0	0 ~ 223	000000h - DFFFFFFh	14 MB	低层7/8
0	0	1	0	1	0 ~ 191	000000h - BFFFFFFh	12 MB	低层3/4
0	0	1	1	0	0 ~ 127	000000h - 7FFFFFFh	8 MB	低层1/2
0	1	0	0	1	4 ~ 255	040000h - FFFFFFFh	16,128 kB	高层63/64
0	1	0	1	0	8 ~ 255	080000h - FFFFFFFh	15,872 kB	高层31/32
0	1	0	1	1	16 ~ 255	100000h - FFFFFFFh	15 MB	高层15/16
0	1	1	0	0	32 ~ 255	200000h - FFFFFFFh	14 MB	高层7/8
0	1	1	0	1	64 ~ 255	400000h - FFFFFFFh	12 MB	高层3/4
0	1	1	1	0	128 ~ 255	800000h - FFFFFFFh	8 MB	高层1/2
X	X	1	1	1	无	无	无	无
1	0	0	0	1	0 ~ 255	000000h - FFEFFFFh	16,380 kB	低层4095/4096
1	0	0	1	0	0 ~ 255	000000h - FFDFFFFh	16,376 kB	低层2047/2048
1	0	0	1	1	0 ~ 255	000000h - FF8FFFFh	16,368 kB	低层1023/1024
1	0	1	0	X	0 ~ 255	000000h - FF7FFFFh	16,352	低层511/512
1	0	1	1	0				
1	1	0	0	1	0 ~ 255	001000h - FFFFFFFh	16,380 kB	高层4095/4096
1	1	0	1	0	0 ~ 255	002000h - FFFFFFFh	16,376 kB	高层2047/2048
1	1	0	1	1	0 ~ 255	004000h - FFFFFFFh	16,368 kB	高层1023/1024
1	1	1	0	X	0 ~ 255	008000h - FFFFFFFh	16,352 kB	高层511/512
1	1	1	1	0				

注意:

1. X = 无需关注。

表 7.4 S25FL256L (256 Mb) 上部阵列补充传统块保护 (TBPROT = 0、CMP = 1)

状态寄存器内容				S25FL256L传统块保护 (TBPROT = 0、CMP = 1)			
BP3	BP2	BP1	BP0	受保护块的数量	受保护块	受保护空间 (KB)	受保护比例
0	0	0	0	512	0-511	32768	所有块
0	0	0	1	511	0-510	32704	低层511/512
0	0	1	0	510	0-509	32640	低层255/256
0	0	1	1	508	0-507	32512	低层127/128
0	1	0	0	504	0-503	32256	低层63/64
0	1	0	1	496	0-495	31744	低层31/32
0	1	1	0	480	0-479	30720	低层15/16
0	1	1	1	448	0-447	28672	低层7/8
1	0	0	0	384	0-383	24576	低层3/4
1	0	0	1	256	0-255	16384	低层1/2
1	0	1	0	0	无	0	无
1	0	1	1	0	无	0	无
1	1	0	0	0	无	0	无
1	1	0	1	0	无	0	无
1	1	1	0	0	无	0	无
1	1	1	1	0	无	0	无

表 7.5 S25FL256L (256 Mb) 下部阵列补充传统块保护 (TBPROT = 1、CMP = 1)

状态寄存器内容				S25FL256L传统块保护 (TBPROT = 1、CMP = 1)			
BP3	BP2	BP1	BP0	受保护块的数量	受保护块	受保护空间 (KB)	受保护比例
0	0	0	0	512	0-511	32768	所有块
0	0	0	1	511	1-511	32704	高层511/512
0	0	1	0	510	2-511	32640	高层255/256
0	0	1	1	508	4-511	32512	高层127/128
0	1	0	0	504	8-511	32256	高层63/64
0	1	0	1	496	16-511	31744	高层31/32
0	1	1	0	480	32-511	30720	高层15/16
0	1	1	1	448	64-511	28672	高层7/8
1	0	0	0	384	128-511	24576	高层3/4
1	0	0	1	256	256-511	16384	高层1/2
1	0	1	0	0	无	0	无
1	0	1	1	0	无	0	无
1	1	0	0	0	无	0	无
1	1	0	1	0	无	0	无
1	1	1	0	0	无	0	无
1	1	1	1	0	无	0	无

表 7.6 S25FL256L (256 Mb) 上部阵列传统块保护 (TBPROT = 0、CMP = 0)

状态寄存器内容				S25FL256L传统块保护 (TBPROT = 0、CMP = 0)			
BP3	BP2	BP1	BP0	受保护块的数量	受保护块	受保护空间 (KB)	受保护比例
0	0	0	0	0	无	0	无
0	0	0	1	1	511	64	高层1/512
0	0	1	0	2	510-511	128	高层1/256
0	0	1	1	4	508-511	256	高层1/128
0	1	0	0	8	504-511	512	高层1/64
0	1	0	1	16	496-511	1024	高层1/32
0	1	1	0	32	480-511	2048	高层1/16
0	1	1	1	64	448-511	4096	高层1/8
1	0	0	0	128	384-511	8192	高层1/4
1	0	0	1	256	256-511	16384	高层1/2
1	0	1	0	512	0-511	32768	全部
1	0	1	1	512	0-511	32768	全部
1	1	0	0	512	0-511	32768	全部
1	1	0	1	512	0-511	32768	全部
1	1	1	0	512	0-511	32768	全部
1	1	1	1	512	0-511	32768	全部

7.6.2 单块锁定 (IBL) 保护

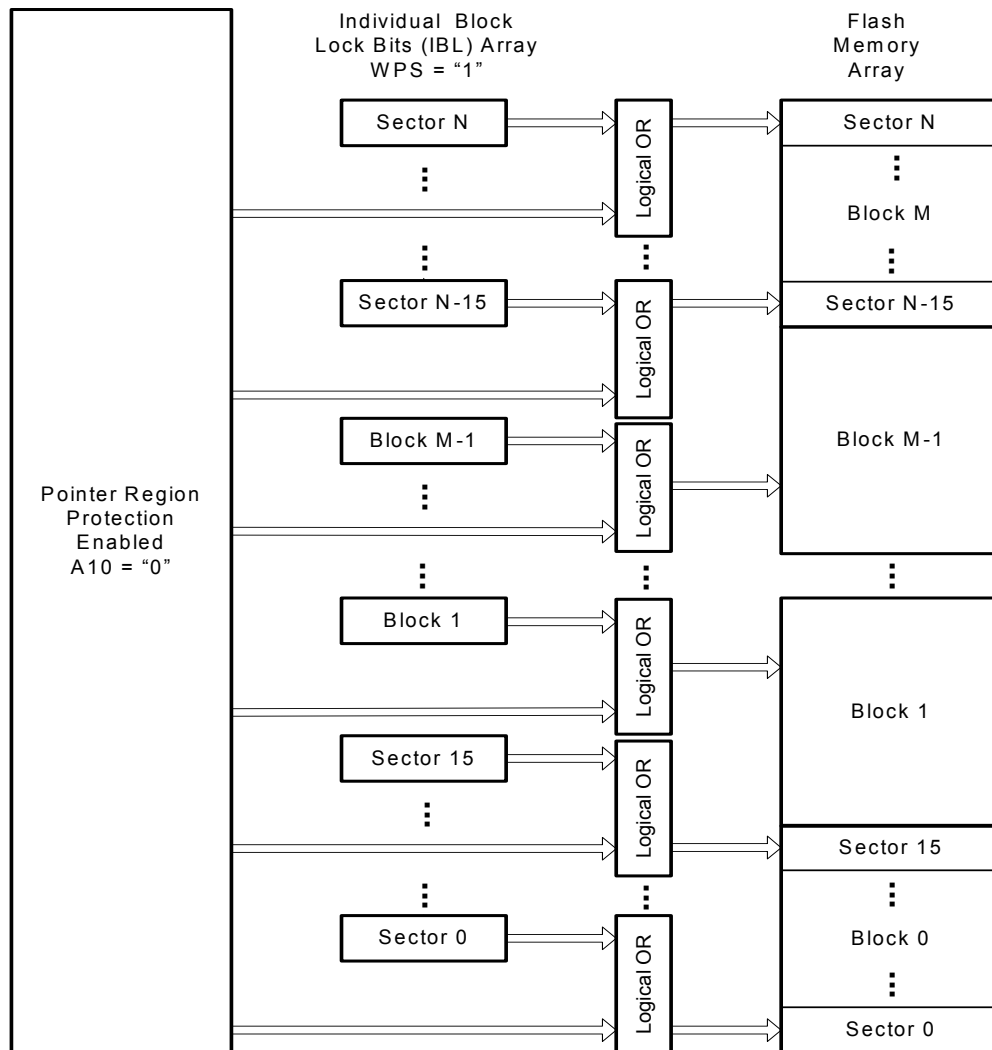
单块定位 (IBL) 是易失性的, 其中每一位用于锁定一个扇区/块, 并且可以被单独修改。通过发送IBL或GBL命令, 可以将IBL位设置为0, 从而保护每个相关的扇区/块。通过发送IBUL或GUL命令, 可以将IBL位清除为1, 从而解除每个相关的扇区/块的保护。通过发送IBLRD命令, 可以读取每个IBL位的状态。该功能允许软件方便地保护单个扇区/块, 以免发生意外更改; 但在需要更改时仍能方便地去解除保护。由于各IBL位都是易失性位, 因此可以按要求对它们进行设置或清除。

每个主64 KB 块和在底部和顶部块中的各4 KB扇区都有相应的易失性的单独块定位 (IBL)。如果某个扇区/块的IBL位为0, 则相关的扇区/块受到保护, 不能对其执行编程和擦除操作。

如果使能了指针区域保护, 则该保护区域会与IBL位进行逻辑OR运算。

在上电、硬件复位或软件复位后, IBLBB处于默认状态 (IBLBB = 1) (请参考第39页上的表6.19. IRP寄存器 (IRP)), 所有单独的IBL位在保护状态下都被设置为0。要想编程或擦除该阵列, 必须在执行编程或擦除命令前先发送全局IBL解锁或扇区/块IBL解锁命令。[IBLBB = 0]时, 在上电、硬件复位或软件复位过后的不受保护的状态中, 所有单独的IBL位都被设置为1。

图7.2 单块锁定/指针区域保护控制



- 注意:**
1. "M" 指的是顶部64 KB的块。
 2. "N" 指的是顶部4 KB扇区。

7.6.3 指针区域保护 (PRP)

指针区域保护由一个非易失性地址指针指定，该地址指针将选择其中一个4 KB扇区作为存储器中保护区域和不受保护区域间的边界。这样可以确保各单独扇区间的间隔在整个电源周期和复位期间一直位于有效范围内。轮到下一个电源周期，PRP设置仍处于保护状态，直到提供了某个密码为止；该位也可能永久性被锁定。可以将PRP与传统块保护或单块锁定保护方式结合起来使用。使能PRP保护时，它与由WPS位 (CR2V[2]) 指定的保护方式之间是逻辑OR关系。

通过发送设置指针区域保护 (SPRP FBh或4SPRP E3h) 命令 (请参考第104页上的8.9节)，或使用写入任意寄存器 (WRAR 71h) 命令写入到PRPR寄存器 (请参考第77页上的8.3.15节) 内，可以使能或禁用PRP，并设置指针的值。

S25FL256L 器件必须进行4字节的地址使能 (CR2V[0] = 1)，以设置指针区域保护寄存器PRPR (请参考第41页上的6.6.10节)，这样才能确保A24和A25被正确设置。

在发送设置块/指针保护命令或者将写入任意寄存器 (WRAR 71h) 命令写入PRPR寄存器后，A10的值会使能或禁用指针保护机制。如果A10 = 1，则禁用指针保护区域。这是默认的状态，其他指针的值不需关注。如果A10 = 0，则使能指针保护区域。A10的值被写入到PRPR寄存器中非易失性的指针位。RFU位的指针地址值是无需关注内容，但读取这些位将返回到数值1。更多有关PRPR的信息，请参考第41页上的6.6.10节。

如果使能了指针保护机制，则指针值会指定寄存器中保护区域和不受保护区域间的块边界。指针边界是由被写入到PRPR寄存器内的非易失性指针值中的三个 (A23-A12) 或四个 (A31-A12) 地址字节设置的。不受保护的区域包括由指针值规定的4 KB扇区。

A9的值用于确定不受保护的区域是否从存储器阵列的顶部 (最高地址) 或底部 (最低地址) 延长到指针所指位置。发送SPRP或4SPRP命令后再发送某个地址，如果A9 = 0，则包含该地址的4 KB扇区以及从底部朝上 (从零地址到更高地址) 的所有扇区不再受保护。如果A9 = 1，则包含该地址的4 KB扇区以及从顶部朝下 (从最高地址到更低地址) 的所有扇区将不受保护。A9的值位于PRPR寄存器中非易失性指针值。

可以使用A11位来保护所有扇区。如果A11 = 1，则所有扇区都受保护。如果A11 = 0，则不受保护的Amax-A12指定。A11的值位于PRPR寄存器中非易失性指针值。

在挂起状态期间，SPRP或4SPRP命令被忽略。这是因为在挂起状态下不能擦除或重新编程指针值。

如果NVLOCK PR[0] = 0，则SPRP或4SPRP被忽略。

读取任意寄存器 (65h) 命令 (请参考第75页上的8.3.14节) 可读取PRP访问寄存器的内容。这样允许读取指针内容，用于进行测试和验证。

表 7.7 PRP 表

A11	A10	A9	受保护的地址范围	不受保护的地址范围	注释
x	1	x	无	全部	A10 = 1时，PRP被禁用 (这是默认状态，剩下所有指针值均为无需关注项)。
0	0	0	1FFFFFF到 (A[31:12]+1)	A[31:12] 到 0000000	包含了该地址的4 kB扇区以及从底部向上 (从零地址到更高地址) 的所有扇区都不受保护。
0	0	1	(A[31:12]-1) 到 0000000	1FFFFFF到A[31:12]	包含了该地址的4 kB扇区以及从顶部向下 (从最高地址到低地址) 的所有扇区都不受保护。
1	0	x	1FFFFFF到 0000000	不适用	A10 = 0和A11 = 1表示保护所有扇区，Amax-A12为无需关注项。

如果使能了指针保护方案 (A10 = 0)，并且该指针保护某个地址空间的所有位置时，向该空间发送的擦除命令将不能被执行。例如：可执行块擦除命令的某个4 KB的阵列，一旦采用了指针保护，所有擦除命令都会失败。使能PRP (A10 = 0) 时，芯片擦除CEh命令被忽略，从而设置 E_ERR状态位。

被使能时，指针区域保护与传统块保护区域 (WPS CR2V[2] = 0) 或单独块锁定保护 (WPS CR2V[2] = 1) 之间是逻辑OR关系 (请参考第46页上的图7.1. LBP、IBL和PRP阵列保护的WPS选择)。

7.7 单独和区域保护

单独和区域保护 (IRP) 指的是一套独立的硬件和软件方案，用于禁用或使能安全区域2、3和指针区域保护寄存器中的编程或擦除操作。

其中每种方法都可监控NVLOCK位的状态 (PR[0])。NVLOCK = 1时，能够编程和擦除安全区域2、3和指针区域保护寄存器 (PRPR)。NVLOCK = 0时，不可编程或擦除安全区域2、3和PRPR。请注意，安全区域2、3还分别受LB2或LB3 = 1 (CR1NV[4:5]) 的保护。

电源锁定保护是默认的方法。该方法在POR或硬件复位期间将NVLOCK位设置为1，因此NVLOCK相关的区域和寄存器不受器件复位的保护。PRL (A6h) 命令将NVLOCK位清除为0，以保护NVLOCK相关的区域和寄存器。电源锁定保护方法没有任何命令能够将NVLOCK位设置为1，因此NVLOCK位一直为0，直到下一次断电或硬件复位为止。电源锁定方法允许引导代码选择修改安全区域2和3或者PRPR中的值（通过编程或擦除这些非易失性区域），然后通过将NVLOCK位清除为0来禁止对这些区域进行的所有更改，以确保系统正常操作。这种情况有时被称为引导代码控制的保护。

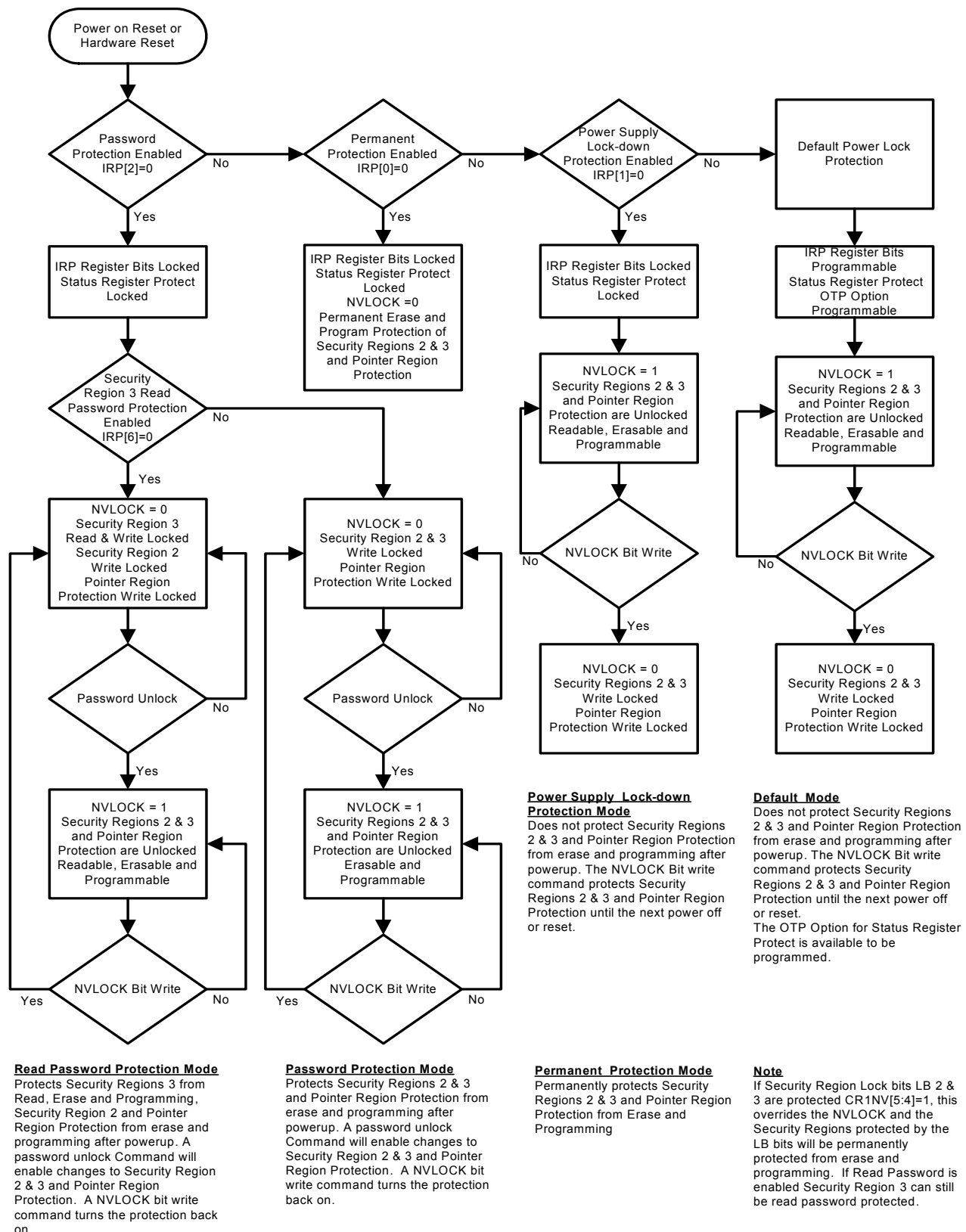
在POR或硬件复位期间，密码保护方法将保护寄存器NVLOCK位被清除为0，并将SECRRP位设置为IRP[6]，以保护NVLOCK相关的区域和寄存器。SECRRP位指定安全区域3是否可读。采用密码保护方法时，可永久地编程和隐藏某个64位的密码。可以使用PASSU (EAh) 命令提供密码，然后将其同隐藏密码进行比较。如果密码匹配，NVLOCK位将被设置为1，从而能够给NVLOCK相关的区域和寄存器解锁。通过PRL (A6h) 命令，可以将NVLOCK位清除为0，再次激活密码保护。

采用永久性保护方法时，会永久性地将SECRRP位设置为1，并清除NVLOCK位为0。这样可永久性保护安全区域2、3和PRPR。

NVLOCK位管理方法通过编程IRP寄存器（IRP[2或1或0]中的OTP位来选择，以便永久性选择所使用的方法。

第54页上的图7.3. *永久性保护、密码保护和电源锁定保护概览*中显示的是这些方法的概述。

图7.3 永久性保护、密码保护和电源锁定保护概览



7.7.1 IRP寄存器

IRP寄存器用于永久地配置单独和区域保护（IRP）特性的状态。请参见第39页上的表6.19. *IRP寄存器（IRP）*。

从工厂出厂时，所有器件默认使电源锁定保护模式，加电时所有区域均不受保护。

器件程序员或主机系统必须通过编程各相应的一次性可编程位来选择想要使用的保护模式：永久性保护、电源锁定保护或密码保护模式。编程其中某位会使器件被永久性锁定为所选定的模式：

出厂默认IRP寄存器

- IRP[6] = 1：读取密码保护模式不被使能。
- IRP[4] = 1：IBL位都处于上电保护状态。
- IRP[2] = 1：密码保护模式不被使能。
- IRP[1] = 1：电源锁定保护模式不被使能，但作为默认模式。
- IRP[0] = 1：永久性保护模式不被使能。

IRP寄存器编程规则：

- 如果选中了读取密码保护模式，则必须在设置密码保护模式锁定位IRP[2]前编写SECRRP位，或同时进行。
- 如果在不受保护模式下初始化各IBL位，则必须在设置保护模式锁定位IRP[2:0]中的某一位前编写IBLLBB位，或者同时进行。
- 如果选中了密码模式，则必须在设置密码保护模式的锁定位IRP[2]前编写密码。
- 各保护模式是相互排除的，一次只能选择其中一种模式。一旦选中了某种保护模式IRP[2:0]，将不能编写IRP寄存器位，并禁止对OTP寄存器进行任何更改。如果尝试修改上述任何寄存器位，则在完成选择保护模式后，该操作会失败，并且P_ERR（SR2V[5]）被设置为1。

IRP寄存器的编程时间与典型的页编程时间相同。系统可以通过读取状态寄存器中的WIP位来确定IRP寄存器的状态。请参考第27页上的6.6.1 *状态寄存器1*一节，了解WIP的详细信息。

请参见第56页上的7.7.3 *密码保护模式*一节。

7.7.1.1 IBL锁定启动位

如果默认的IBL锁定位IRP[4] = 1，则上电或发生复位时（在硬件复位或软件复位后的状态），所有IBL位都会进入“受保护状态”。如果IBL锁定位IRP[4] = 0（编程），正在处于上电或复位模式的所有IBL位都会进入“不受保护状态”。

7.7.2 保护寄存器 (PR)

7.7.2.1 NVLOCK位 (PR[0])

NVLOCK位是一个易失性位，用于保护：

- 指针区域保护寄存器
- 安全区域2和3

该位被清除为0时，NVLOCK将锁定相关区域。它被设置为1时，可修改这些区域。相关的详细信息，请参见第40页上的6.6.8 [保护寄存器 \(PR\)](#) 一节。

PRL命令用于将NVLOCK位清除为0。将所有相关区域配置为所需设置后，需要将NVLOCK位清除为0。

在电源锁定保护模式下，NVLOCK位在POR或硬件复位期间被设置为1。软件复位命令不会对NVLOCK位产生影响。将该位清除为0时，所有软件命令序列都不能将NVLOCK置1，只有通过硬件复位或上电才能设置NVLOCK位。

在密码保护模式下，NVLOCK位在POR或硬件复位期间被清除为0。只有通过密码解锁命令才能将NVLOCK位置1。

在永久性保护模式下，会将NVLOCK位永久性清除为0。这样可一直保护安全区域2和3和PRPR。

7.7.2.2 安全区域读取密码锁定位 (SECRP, PR[6])

SECRP位是一个易失性位，用于防止对安全区域3进行读取操作。SECRP[6]=0时，无法读取安全区域3。更多信息，请参见第40页上的6.6.8 [保护寄存器 \(PR\)](#) 一节。

在密码保护模式下，SECRP位在POR、软件复位或硬件复位期间与IRP[6]相同。只有通过密码解锁命令才能将NVLOCK位设为1。软件复位不会对NVLOCK位产生影响。

在永久性保护模式下，SECRP位永久性被设置为1。这样可以持久维持安全区域3的可读性。

7.7.3 密码保护模式

密码保护模式在解锁NVLOCK位时要求64位密码，因此可比电源锁定保护模式提供级别更高的安全性。除了密码要求外，在上电和复位发生后，NVLOCK位还被清除为0，以确保在上电或复位后提供保护。通过输入整个密码并成功执行密码解锁命令后，NVLOCK位将被设置为1，从而允许修改与扇区NVLOCK相关的区域和寄存器。

密码保护注意事项：

- 编程和验证密码后，必须设置密码模式 (IRP[2] = 0)，以防止读取密码。
- 密码编程命令只能编程各个“0”位。若在一个比特位编程为0后编程1，该位将保留0，并且不会设置编程错误。
- 从赛普拉斯出厂时，密码全部是1。它位于自己的存储空间中，可通过使用密码编程、密码读取、RDAR和WRAR命令进行访问。
- 所有64位密码组合均为有效密码。
- 一旦编程了密码保护模式，便能够防止对64位密码进行读取和编程操作。对密码区域的所有额外编程和读取命令均被禁止，这些命令被忽略或返回未定义数据。选择了密码模式锁定位后，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。
- 保护模式锁定位不可擦除。
- 只有输入准确的密码，才能解锁。如果密码解锁命令提供的密码与隐藏的内部密码不匹配，解锁操作会失败，与对受保护的扇区执行编程操作相似。P_ERR位被设置为1，WIP位保持设置状态，并且NVLOCK位保持清零状态。
- 每经过100 μs ± 20 μs时间，只能接收一条密码解锁命令。这样，黑客要想尝试所有64位组合以找到正确匹配的密码，就要经过不可思议的超长时间（5800万年）。读取状态寄存器1命令可用于读取WIP位，从而可以指出器件完成了密码解锁命令还是准备好接收新的密码命令。提供有效的密码时，密码解锁命令会立即将WIP位清零，不需要经过100 μs的延迟时间。
- 如果在选择密码模式后丢失了密码，则无法将NVLOCK位设置为1。

7.7.4 安全区域读取密码保护

安全区域读取密码保护允许防止对安全区域3进行读取、编程和擦除操作。

- 安全区域读取密码保护是密码保护模式的附加选项（如上所述）。用户将SECRRP位IRP[6]编程为0时，安全区域读取密码保护被使能。设置密码保护模式锁定位IRP[2]时/前，必须编程SECRRP位IRP[6]。

仅在编程密码并且将IRP[2]清零后，安全区域读取密码保护才会生效。

SECRRP（PR[6]）位被设为0时，安全区域3成为不可读取的空间。如果读取这些区域，返回的是无效和未定义的数据。

7.7.5 建议的IRP保护流程

在系统制造中，闪存器件的配置如下：

1. 根据需要编程安全区域
2. 根据需要设置指针区域保护寄存器
3. 若使用密码保护，则需要编程密码寄存器（PASS）。
4. 根据需要编程IRP寄存器，包括通过IRP[2:0]选择永久性保护、电源锁定或密码IRP保护模式。明确指定一种保护模式非常重要，这样可以防止以后对IRP寄存器的意外或特意编程。这样可确保只有特定的保护功能被使能。编程IRP寄存器前/时：
 - a. 通过IBLLBB位（IRP[4]）可以在无保护状态下为所有IBL位供电。
 - b. 通过编程SECRRP位（IRP[6]），可以使安全区域读取密码保护使用密码来控制对安全区域3的读取访问。

在系统上电和引导代码执行过程中：如果正在使用电源锁定保护模式，则通过可靠引导代码能够确定是否需要修改与NVLOCK相关的区域或寄存器。如果无需修改，则可以使用PRL命令将NVLOCK位清零，以便在通电时防止在正常系统操作期间对与NVLOCK相关的区域或寄存器进行修改。

8. 命令

主机系统和FL-L存储器件系列之间都是以命令为单位形式进行通信。有关命令协议的详细信息，请参见第15页上的5.2 命令协议一节。

虽然在某些情况下可以使用主机软件直接控制SPI接口信号，但是信号关系和时序详情则通常由主机系统和存储器设备的硬件接口处理。本节内容着重于软件接口，所以没有详细描述信号关系和时序。它的重点内容是每个命令中位传输的逻辑序列。下面是一些需要记住的通用信号关系说明。更多有关命令的比特级别格式和信号时序关系的信息，请参见第15页上的5.2 命令协议一节。

- 主机总是控制片选（CS#）、串行时钟（SCK）和单比特传输中的串行输入（SI）。存储器则驱动单比特读取传输的串行输出（SO）。主机和存储器在双比特和四比特传输期间交替驱动IO0-IO3信号。
- 所有命令的启动方式均相同，都是主机在SCK的第一个上升沿前将CS#驱动为低电平，以选择存储器。执行命令过程中，CS#一直保持低电平状态。CS#返回高电平状态时，命令结束。一般情况下，CS#在八位传输的倍数期间保持低电平状态，用以传输字节粒度的信息。如果CS#在非8位对齐位置上返回高电平状态，则不接受任何命令。

8.1 命令集汇总

8.1.1 扩展寻址

要想寻址大于128 Mb空间的地址，可以使用以下两个选项：

1. 始终要求一个4字节地址的指令，用于访问高达32 Gb的存储器：

表 8.1 扩展型地址：4 字节地址命令

命令名称	功能	指令（十六进制）
4READ	读取	13
4FAST_READ	快速读取	0C
4DOR	双线输出读取	3C
4QOR	四线输出读取	6C
4DIOR	双线I/O读取	BC
4QIOR	四线I/O读取	EC
4DDRQIOR	DDR四线I/O读取	EE
4PP	页编程	12
4QPP	四线页编程	34
4SE	扇区擦除	21
4HBE	半块擦除	53
4BE	块擦除	DC
4IBLRD	IBL读取	E0
4IBL	IBL锁定	E1
4IBUL	IBL解锁	E2
4SPRP	设置指针区域保护	E3

2. 4字节地址模式与3 字节地址指令向后兼容。标准的3 字节指令可以同地址长度配置位（CR2V[0]）控制的4字节地址模式结合使用。CR2V[0]的默认值可从CR2NV[1]中加载（在执行上电、硬件复位或软件复位后），实现默认的3字节（24位）或4字节（32位）寻址。地址长度（CR2V[0]）被设为1时，将修改传统命令，要求地址字段中的4字节（32位）地址。以下指令可以同4 字节地址模式配置结合起来使用，以便从3字节地址字段转为4字节地址字节。

表 8.2 扩展型地址：4 字节地址模式下的 3 字节地址命令

命令名称	功能	指令（十六进制）
RSFDP	读取SFDP	5A
READ	读	03
FAST_READ	快速读取	0B
DOR	双线输出读取	3B
QOR	四线输出读取	6B
DIOR	双线I/O读取	BB
QIOR	四线I/O读取	EB
DDRQIOR	DDR四线I/O读取	ED
PP	页编程	02
QPP	四线页编程	32
SE	扇区擦除	20
HBE	半块擦除	52
BE	块擦除	D8
RDAR	读取任何寄存器	65
WRAR	写入任何寄存器	71
SECRE	安全区域擦除	44
SECRP	安全区域编程	42
SECRR	安全区域读取	48
IBLRD	IBL读取	3D
IBL	IBL锁定	36
IBUL	IBL解锁	39
SPRP	设置指针区域保护	FB

8.1.2 按功能分类的命令汇总

表 8.3 FL-L 系列命令集（按功能排列）

功能	命令名称	命令说明	指令值 (十六进制)	最大频率 (MHz)	地址长度 (字节)	QPI
读取器件ID	RDID	读取ID (JEDEC制造商ID)	9F	108	0	有
	RSFDP	读取JEDEC串行闪存可发现参数	5A	133	3或4	有
	RDQID	四线读取ID	AF	108	0	有
	RUID	读取唯一ID	4B	133	0	有
寄存器访问	RDSR1	读取状态寄存器1	05	108	0	有
	RDSR2	读取状态寄存器2	07	108	0	无
	RDCR1	读取配置寄存器1	35	108	0	无
	RDCR2	读取配置寄存器2	15	108	0	无
	RDCR3	读取配置寄存器3	33	108	0	无
	RDAR	读取任何寄存器	65	133	3或4	有
	WRR	写入寄存器 (状态寄存器1和配置寄存器1、2、3)	01	133	0	有
	WRDI	写禁用	04	133	0	有
	WREN	写使能, 用于修改非易失性数据	06	133	0	有
	WRENV	写使能, 用于修改易失性状态和配置寄存器的数据	50	133	0	有
	WRAR	写入任何寄存器	71	133	3或4	有
	CLSR	清除状态寄存器	30	133	0	有
	4BEN	进入4字节地址模式	B7	133	0	有
	4BEX	退出4字节地址模式	E9	133	0	有
	SBL	设置连续长度	77	133	0	有
	QPIEN	进入QPI模式	38	133	0	无
	QPIEX	退出QPI模式	F5	133	0	有
	DLPRD	数据学习模式读取	41	133	0	有
	PDLRNV	编程非易失性数据学习寄存器	43	133	0	有
WDLRV	写入易失性数据学习寄存器	4A	133	0	有	
读取闪存阵列	READ	读	03	50	3或4	无
	4READ	读取	13	50	4	无
	FAST_READ	快速读取	0B	133	3或4	无
	4FAST_READ	快速读取	0C	133	4	无
	DOR	双线输出读取	3B	133	3或4	无
	4DOR	双线输出读取	3C	133	4	无
	QOR	四线输出读取	6B	133	3或4	无
	4QOR	四线输出读取	6C	133	4	无
	DIOR	双线I/O读取	BB	133	3或4	无
	4DIOR	双线I/O读取	BC	133	4	无
	QIOR	四线I/O读取 (CR1V[1] = 1) 或CR2V[3] = 1	EB	133	3或4	有
	4QIOR	四线I/O读取 (CR1V[1] = 1) 或CR2V[3] = 1	EC	133	4	有
	DDRQIOR	DDR四线I/O读取 (CR1V[1] = 1) 或CR2V[3] = 1	ED	66	3或4	有
4DDRQIOR	DDR四线I/O读取 (CR1V[1] = 1) 或CR2V[3] = 1	EE	66	4	有	
编程闪存阵列	PP	页编程	02	133	3或4	有
	4PP	页编程	12	133	4	有
	QPP	四线页编程	32	133	3或4	无
	4QPP	四线页编程	34	133	4	无

表 8.3 FL-L 系列命令集（按功能排列）（续）

功能	命令名称	命令说明	指令值 (十六进制)	最大频率 (MHz)	地址长度 (字节)	QPI
擦除闪存阵列	SE	扇区擦除	20	133	3或4	有
	4SE	扇区擦除	21	133	4	有
	HBE	半块擦除	52	133	3或4	有
	4HBE	半块擦除	53	133	4	有
	BE	块擦除	D8	133	3或4	有
	4BE	块擦除	DC	133	4	有
	CE	芯片擦除	60	133	0	有
	CE	芯片擦除（备用指令）	C7	133	0	有
擦除/编程挂起/恢复	EPS	擦除/编程挂起	75	133	0	有
	EPR	擦除/编程恢复	7A	133	0	有
安全区域阵列	SECRE	安全区域擦除	44	133	3或4	有
	SECRP	安全区域编程	42	133	3或4	有
	SECRR	安全区域读取	48	133	3或4	有
阵列保护	IBLRD	IBL读取	3D	133	3或4	有
	4IBLRD	IBL读取	E0	133	4	有
	IBL	IBL锁定	36	133	3或4	有
	4IBL	IBL锁定	E1	133	4	有
	IBUL	IBL解锁	39	133	3或4	有
	4IBUL	IBL解锁	E2	133	4	有
	GBL	全局IBL锁定	7E	133	0	有
	GBUL	全局IBL锁定	98	133	0	有
	SPRP	设置指针区域保护	FB	133	3或4 (2)	有
单独和区域保护	4SPRP	设置指针区域保护	E3	133	4	有
	IRPRD	IRP寄存器读取	2B	133	0	有
	IRPP	IRP寄存器编程	2F	133	0	有
	PRRD	保护寄存器读取	A7	133	0	有
	PRL	保护寄存器锁定（NVLOCK位写入）	A6	133	0	有
	PASSRD	密码读取	E7	133	0	有
	PASSP	密码编程	E8	133	0	有
复位	PASSU	密码解锁	EA	133	0	有
	RSTEN	软件复位使能	66	133	0	有
	RST	软件复位	99	133	0	有
深度下电	MBR	模式位复位	FF	133	0	有
	DPD	深度下电	B9	133	0	有
	RES	退出深度下电模式/器件ID	AB	133	0	有
RFU	Reserved-18	保留	18			
RFU	Reserved-41	保留	41			
RFU	Reserved-43	保留	43			
RFU	Reserved-4A	保留	4A			
RFU	Reserved-ED	保留	ED			
RFU	Reserved-EE	保留	EE			

注意:

1. 若在QPI模式下发送不受支持的命令，这些命令将发生未定义行为。
2. 对于S25FL256L器件，SPRP命令必须使用4字节地址模式（CR2V[0] = 1）。

8.1.3 读取器件标识

有多条用于读取有关器件制造商、器件类型和器件特性等信息的命令。来自不同供应商的SPI存储器使用了不同的命令和格式来读取存储器信息。FL-L系列产品支持三种器件信息命令。

8.1.4 寄存器读取或写入

有多种寄存器用于报告嵌入式操作状态或控制器件配置选项。可以通过命令对这些寄存器进行读取或写入操作。寄存器包括易失性和非易失性位。寄存器中的非易失性位自动被擦除和编程，如单个（写）操作一样。

8.1.4.1 监控操作状态

主机系统通过监控状态寄存器中的“正在写入”（WIP）位，可以指出是否完成了写入、编程、擦除、挂起或其它嵌入式操作。读取状态寄存器1命令或读取任意寄存器命令可提供WIP位的状态。读取状态寄存器2或读取任意寄存器命令将提供状态寄存器中的编程错误（P_ERR）和擦除错误（E_ERR）位的状态，指明最新的编程或擦除命令是否完成。当P_ERR或E_ERR位为1时，WIP位将保持为1，以表示器件的繁忙状态并且无法接收新的操作命令。P_ERR或E_ERR被设为1时，只有状态读取（RDSR105h、RDSR207h）、读取任意寄存器（RDAR 65h）、读取配置RDCR1和RDCR3、状态清除（CLSR 30h）和软件复位（RSTEN 66h随后是RST 99h）为有效命令。并且必须发送清除状态寄存器（CLSR）命令，才能使器件返回待机状态。另外，也可以通过硬件复位或软件复位（RSTEN 66h随后是RST 99h），使器件返回待机状态。

8.1.4.2 配置

可以使用一些命令来读取、写入和保护寄存器，从而控制接口路径宽度、接口时序、接口地址长度以及数据保护相关内容。

8.1.5 读取闪存阵列

可以从存储器的任意字节边界上读取数据。数据字节从低位字节地址到高位字节地址被连续读取，直到主机将CS#置高（表示结束数据传输）为止。如果字节地址达到存储器阵列的最大地址，读取操作将从阵列的地址0重新开始。

可以通过设置突发长度（SBL 77h）命令并按照所要求的循环读取长度和对齐来实现连续回卷读取，请参见第78页上的8.3.16 [设置连续长度（SBL 77h）](#)一节。连续回卷读取仅适用于四线I/O和QPI模式。

有些不同的读取命令用于指定不同的访问延迟和数据路径宽度。双倍数据速率（DDR）命令还定义了地址和数据位与两个SCK边沿的关系：

- 读取命令在每个SCK上升沿上通过SI/IO0信号提供一个地址位，并在每个SCK下降沿上通过SO/IO1信号返回一个数据位。使用该命令，地址和返回数据间并没有延迟，但是最大SCK速率被限于50 MHz。
- 使用其它读取命令，地址和返回数据间有一定的延迟，但能够以更高的SCK频率进行操作。延迟时间取决于配置寄存器的读取延迟值。
- 快速读取命令在每个SCK上升沿上通过SI/IO0信号提供一个地址位，并在每个SCK下降沿上通过SO/IO1信号返回一个数据位。
- 双线或四线输出读取命令在每个SCK上升沿上通过SI/IO0引脚提供地址，并在每个SCK下降沿上通过IO0 - IO3信号返回两个或四个数据位。
- 双线或四线I/O读取命令在每个SCK上升沿上提供两个或四个地址位，并在每个SCK下降沿上通过IO0-IO3信号返回两个或四个数据位。如果模式位值为Axh，则连续读取功能会被使能。
- 四线双倍数据速率读取命令在每个SCK边沿上提供了四个地址位，并在每个SCK边沿上通过IO0 - IO3信号返回四个数据位。模式位值为Axh时，连续读取功能会被使能。

8.1.6 编程闪存阵列

编程数据需要两条命令：写使能（WREN）和页编程（PP、4PP、QPP、4QPP）。通过页编程命令，可以在一次操作中编程从1字节到连续256个字节的数据。编程操作可以使位值保持为1，或从1改为0。要想将位值从0变为1，需要使用擦除操作。

8.1.7 擦除闪存阵列

扇区擦除、半块擦除、块擦除或芯片擦除命令将一个扇区中或整个存储器阵列中的所有位设置为1。先要将某一位擦除为1，然后才能将它编程为0。可以独立将每一位从1编程为0，但必须按一个扇区、半块、一块或整个阵列（芯片）的级别将位从0擦除为1。执行擦除命令前，必须执行写使能（WREN）命令。

8.1.8 安全区域、传统的块保护以及个别和区域保护

器件有一些命令用于读取和编程保存永久性保护数据（例如，序列号）的一次性保护（OTP）阵列。有一些命令用于控制防止进行编程和擦除操作的闪存存储器阵列扇区连续组（块）。还有一些命令用于控制防止对哪个个别闪存存储器阵列扇区进行编程和擦除操作。另外，有一种模式用于限制对安全区域3的读取访问，直到提供密码为止。

8.1.9 复位

在器件上电后，可以使用一些命令将器件复位为默认状态。然而，软件复位命令对SRP1或NVLOCK位没有产生影响。在所有其它方面，软件复位与硬件复位相同。

有一个命令用于复位（退出）连续读取模式。

8.1.10 保留

有些命令留给将来使用。在这一代FL-L系列产品中，有些命令指令可能未被使用，但并不会影响器件的操作；有些命令指令则会返回未定义结果。

这类命令中的一部分被保留，以确保可以执行传统或交替源器件命令，而不会产生影响。这样允许传统软件发送一些与当前一代FL-L系列无关的命令，并要保证这些命令不会导致意外行为。

有些命令将被保留，用于FL-L的特殊版本（在该文档中未提到）或下一代产品。这样，新的主机存储控制器设计可以活使用这些命令指令。如果在发布本文档的修订版本时已确定好，那么命令格式也会得到定义。

8.2 标识命令

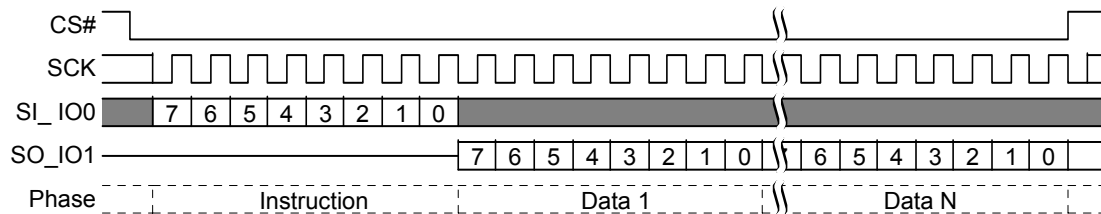
8.2.1 读取标识 (RDID 9Fh)

通过读取标识 (RDID) 命令, 可以对制造商标识和器件标识进行读取访问。制造商标识由JEDEC分配, 器件标识则由赛普拉斯分配。在执行编程、擦除或写入周期期间发送的任何RDID命令被忽略, 并且不会影响到这些操作的执行。

RDID指令通过SI/IO0传入到器件内。RDID指令的最后位传入到器件后, 通过SO/IO1信号连续输出一个制造商标识字节和两个器件标识字节。这全部信息被称为ID。有关ID内容的详细说明, 请参见第122页上的10.2 器件ID地址映射一节。

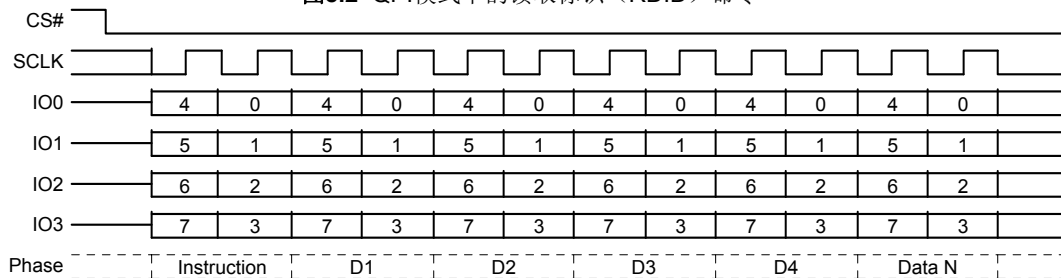
所定义的ID地址结束后, 继续移出的输出会提供未定义数据。输出数据期间可随时将CS#置为逻辑高电平状态, RDID命令序列立即被终止。传送RDID命令的频率最高可达108 MHz频率。

图8.1 读取标识 (RDID) 命令序列



在QPI模式下, 器件也支持该命令。在QPI模式下, 将通过IO0-IO3信号传入指令并输出返回数据。

图8.2 QPI模式下的读取标识 (RDID) 命令



8.2.2 读取四线标识 (RDQID AFh)

通过读取四线标识 (RDQID) 命令, 可以读取制造商标识和器件标识。该命令是RDID命令的交替选项, 用于在QPI模式下读取同样的信息。在所有其它方面, 该命令与RDID命令相同。

只有器件处于QPI模式 (CR2V[3] = 1) 或四线模式 (CR1V[1] = 1) 时, 才能识别该命令。在QPI模式下, 将通过IO0-IO3传入指令, 而在四线模式下会通过IO0传入指令。指令的最后一位传入到器件后, 通过IO0-IO3信号连续输出一个制造商标识字节和两个器件标识字节。这样全部信息被称为ID。有关ID内容的详细说明, 请参见第122页上的10.2 器件ID地址映射一节。

所定义的ID地址结束后, 继续移出的输出保存未定义数据。输出数据期间可以随时将CS#置为逻辑高电平状态, 该命令序列立即被终止。

图8.3 QPI模式下的读取四线标识（RDID）命令序列

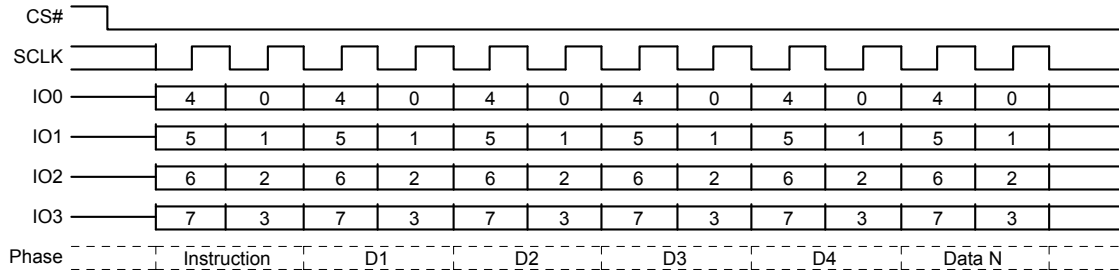
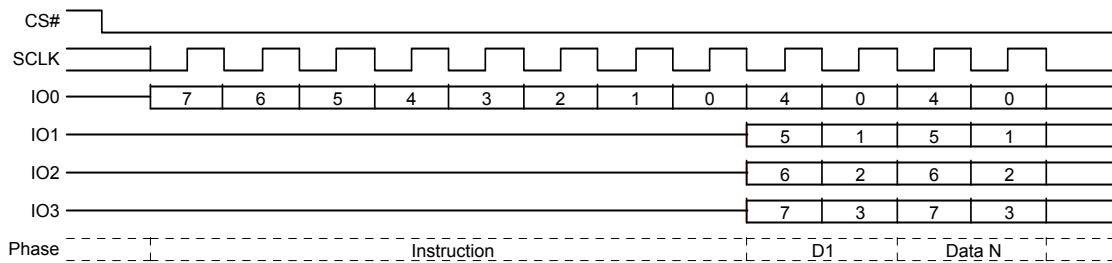


图8.4 四线模式下的读取四线标识（RDID）命令序列



8.2.3 读取串行闪存可发现参数（RSFDP 5Ah）

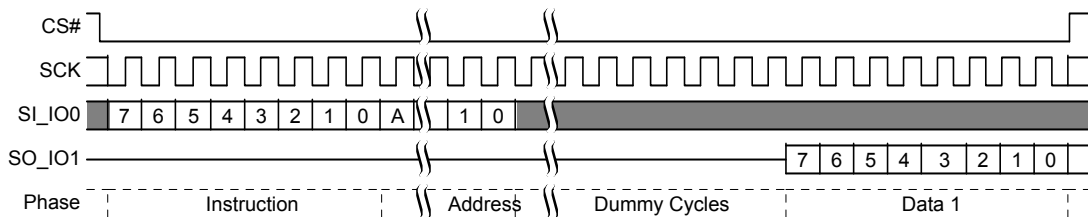
该命令开始于通过SI信号分别传入指令代码“5Ah”和一个24位（3字节）地址或32位（4字节）地址（取决于CR2V[0]地址长度的当前配置），随后是CR3V[3:0]字段中可变读取延迟配置所指定的读取延迟周期数（虚拟周期）。

虚拟周期结束后，通过SO/IO1从SCK下降沿开始传出SFDP字节。SFDP字节始终按最高有效位优先的方式传出。如果24位（3字节）地址或32位（4字节）地址被设为非零值，则SFDP空间中的选定位置作为读取数据的起始点。这样可以随机访问SFDP空间中的任意参数。在SPI模式下，RSFDP命令的时钟频率最高可达133 MHz。

为了符合JEDECJESD216SFDP标准，应将可变读取延迟设置为8个周期。器件从赛普拉斯出厂时，CR3NV中的非易失性可变读取延迟的默认值为8个虚拟周期。然而，由于RSFDP命令使用的实现与其它可变地址长度和延迟读取命令相同，所以如果用户需要，可以自由修改命令的地址长度和延迟。

读取SFDP命令支持连续读取。

图8.5 RSFDP命令序列

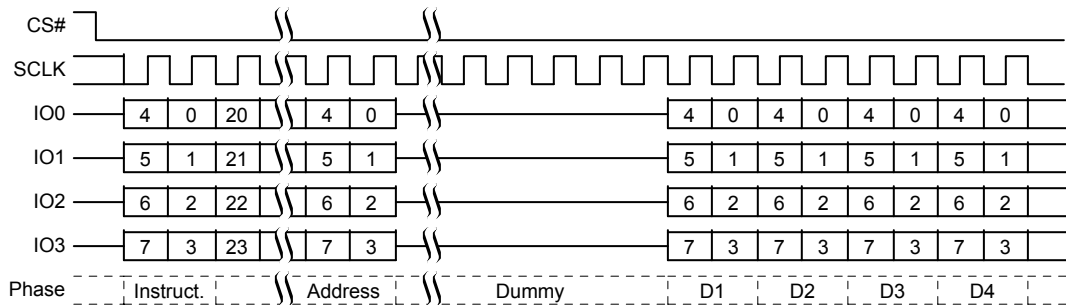


注意:

A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令13h)。

QPI模式也支持该命令。在该模式下，将通过IO0-IO3信号传入指令并输出返回数据。

图8.6 QPI模式下的RSFDP命令序列



8.2.4 读取唯一ID (RUID 4Bh)

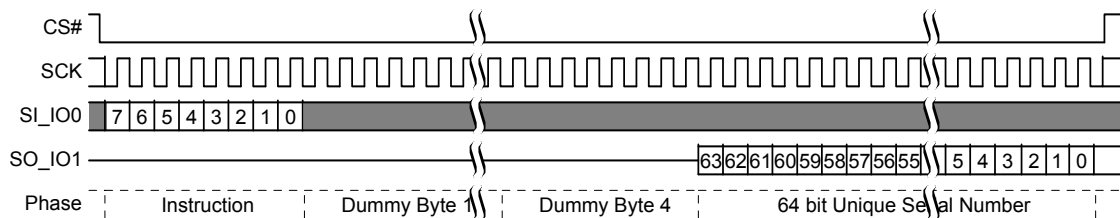
读取标识 (RUID) 命令允许读取访问工厂设置的只读64位编号，该编号是每个器件的唯一值。

通过SI发送RUID指令，然后发送四个虚拟字节或16个虚拟字节QPI (32个时钟周期)。延迟周期 (虚拟字节) 为器件的内部电路提供需要的时间来访问初始地址的数据。在这些延迟周期内，IO0-IO3上的数据值为“无需关注项”，该值可能为高阻抗。

然后，唯一ID的8个字节通过SO/IO1被连续输出。

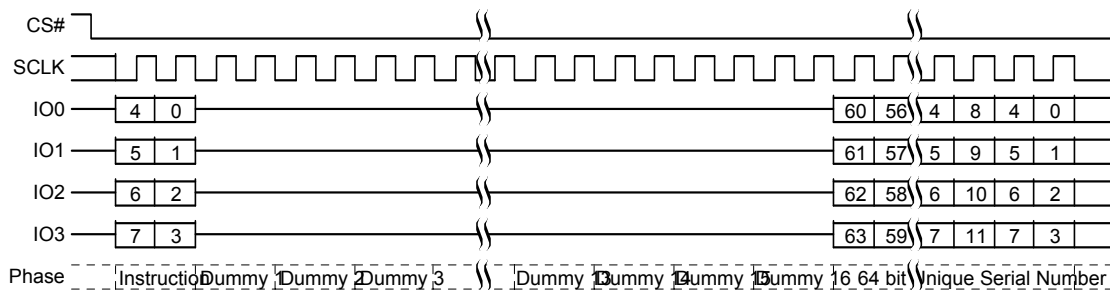
所定义的唯一ID地址结束后，继续移出的输出会提供未定义数据。输出数据期间可以随时将CS#置为逻辑高电平状态，RUID命令序列立即被终止。

图8.7 读取唯一ID (RUID) 命令序列



QPI模式也支持该命令。在该模式下，将通过IO0-IO3信号传入指令并输出返回数据。

图8.8 QPI模式下的读取唯一ID (RUID) 命令



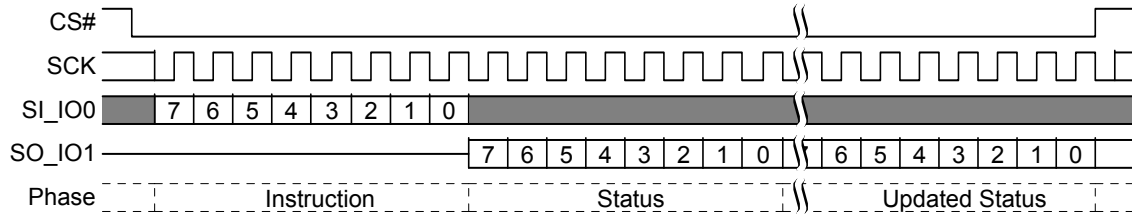
8.3 寄存器访问命令

8.3.1 读取状态寄存器1 (RDSR1 05h)

通过读取状态寄存器1 (RDSR1) 命令，可以从SO/IO1读取状态寄存器1的内容。

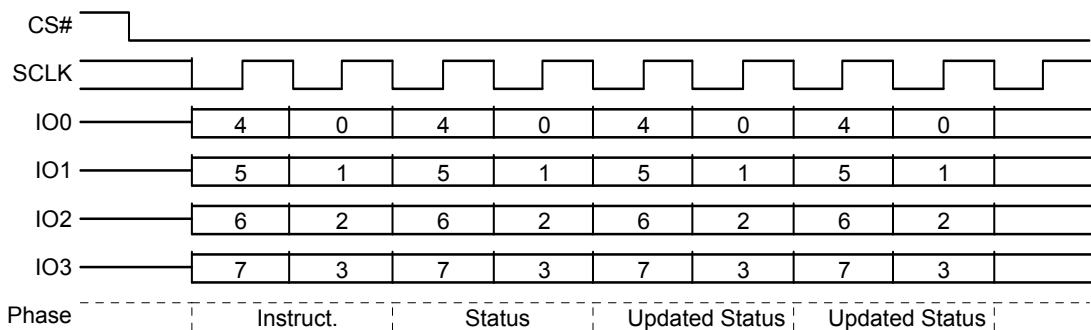
可以随时读取易失性状态寄存器1 (SR1V) 的内容，就算在执行编程、擦除或写入操作期间也能进行。通过提供8个时钟周期的倍数，可以连续读取状态寄存器1。每8个读取周期更新一次状态。

图8.9 读取状态寄存器1 (RDSR1) 命令序列



QPI模式也支持该命令。在该模式下，将通过IO0-IO3信号传入指令并输出返回数据。此外，还支持最高频率为108 MHz的读取状态寄存器。要想读取频率高于108 MHz的状态寄存器1，需要使用读取所有寄存器命令，请参考第75页上的8.3.14 读取任意寄存器 (RDAR 65h) 一节。

图8.10 QPI模式下的读取状态寄存器1 (RDSR1) 命令

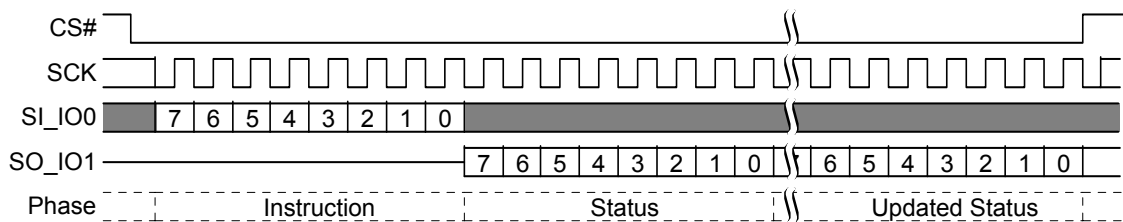


8.3.2 读取状态寄存器2 (RDSR2 07h)

通过读取状态寄存器2 (RDSR2) 命令，可以从SO/IO1读取状态寄存器2的内容。

可以随时读取易失性状态寄存器2 (SR2V) 的内容，就算在执行编程、擦除或写入操作期间也能进行。通过提供8个时钟周期的倍数，可以连续读取状态寄存器2。每8个读取周期更新一次状态。

图8.11 读取状态寄存器2 (RDSR2) 命令

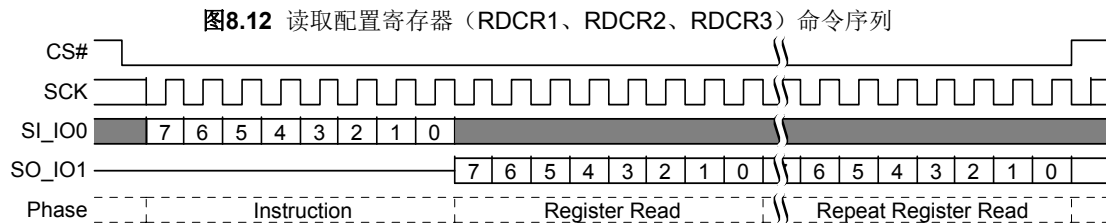


在QPI模式下，可以使用读取任意寄存器命令来读取状态寄存器2，请参考第75页上的8.3.14 读取任意寄存器 (RDAR 65h) 一节。

8.3.3 读取配置寄存器（RDCR1 35h、RDCR2 15h、RDCR3 33h）

通过读取配置寄存器（RDCR1、RDCR2、RDCR3）命令，可以从SO/IO1读取易失性配置寄存器（CR1V、CR2V、CR3V）的内容。

通过提供8个时钟周期的倍数，可以连续读取CR1V、CR2V和CR3V。可以随时读取配置寄存器的内容，就算在执行编程、擦除或写入操作期间也能进行。要想以更高频率读取配置寄存器1、2和3，需要使用读取任意寄存器命令，请参考第75页上的8.3.14 读取任意寄存器（RDAR 65h）一节。



在QPI模式下，可以使用读取任意寄存器命令来读取配置寄存器1、2和3，请参考第75页上的8.3.14 读取任意寄存器（RDAR 65h）一节。

8.3.4 写入寄存器（WRR 01h）

写入寄存器（WRR）命令允许将新值写入到状态寄存器1、配置寄存器1、配置寄存器2和配置寄存器3。器件接受写入寄存器（WRR）命令前，必须接收到写入使能（WREN）或易失性寄存器的写入使能（WRENV）命令。写入使能（WREN）命令被成功解码后，器件将设置状态寄存器中的写入使能锁存（WEL），从而使能非易失性写操作，并将后面的WRR命令中的值传入到非易失性SR1NV、CR1NV、CR2NV和CR3NV寄存器内。成功解码易失性寄存器的写使能（WRENV）命令后，器件会将下面WRR命令中的值传入到易失性SR1V、CR1V、CR2V和CRV3寄存器内。

通过在SI/IO0上传输指令和数据字节，可以输入写入寄存器（WRR）命令。状态寄存器的长度为一个数据字节。

针对在WREN命令后执行的非易失性寄存器的WRR操作，首先擦除非易失性寄存器，然后通过一次操作编程新值，最后将新的非易失性值复制到易失性寄存器内。对于在WRENV命令后执行的易失性寄存器的WRR操作，更新了易失性寄存器，而没有影响相关的非易失性寄存器值。如果WRR操作失败，则写入寄存器（WRR）命令将设置P_ERR或E_ERR位。有关错误位的说明，请参见第31页上的6.6.2 易失性状态寄存器2（SR2V）一节。器件将挂起繁忙状态，直到使用清除状态寄存器（CLSR）来清除错误和WIP（以使器件返回待机状态）为止。所有留作日后使用的状态或配置寄存器位都必须被写入为0。

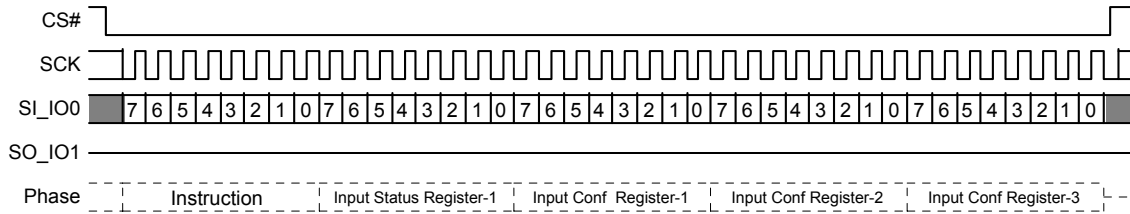
数据的第8、第16、第24或第32位被锁存后，必须将CS#置为逻辑高电平状态。否则，无法执行写入寄存器（WRR）命令。如果：

- 在第8个周期后将CS#置为高电平，那么只会对状态寄存器1进行写操作；
- 在第16个周期后将CS#置为高电平，则对状态寄存器 1和配置寄存器 1进行写入操作；
- 在第24个周期后将CS#置为高电平，则对状态寄存器1和配置寄存器1、2进行写入操作；
- 在第32个周期后将CS#置为高电平，则对状态寄存器 1和配置寄存器 1、2、3进行写入操作；

CS#被驱动为逻辑高电平状态后，会立即启动自定时写入寄存器（WRR）操作。在执行写入寄存器（WRR）操作的过程中，还可以读取状态寄存器，以检查正在写入（WIP）位的值。执行自定时写入寄存器（WRR）操作期间，WIP位为1；完成该操作时，WIP位为0。完成写入寄存器（WRR）操作时，写入使能锁存（WEL）位将被设置为0。

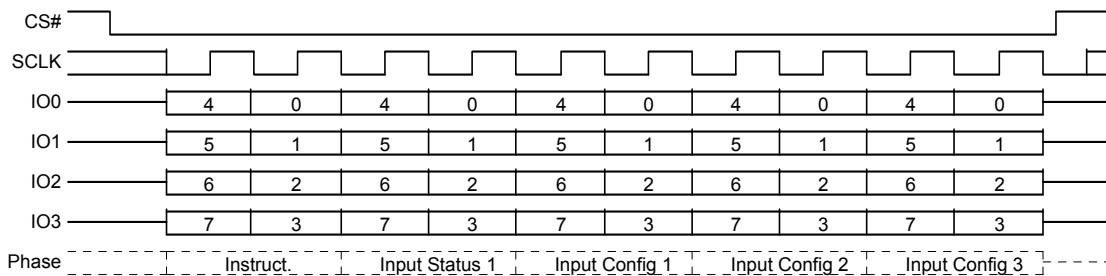
WRR命令是受保护的，以免受硬件或软件复位的影响，硬件复位和软件复位命令被忽略，并且不会对WRWRR命令的执行产生影响。

图8.13 写入寄存器（WRR）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令和数据传入到器件内。

图8.14 QPI模式下的写入寄存器（WRR）命令序列



通过写入寄存器（WRR）命令，用户可以修改非易失性状态寄存器1或易失性状态寄存器1中的传统块保护位值，用于定义只读区域的大小。

用户也可以通过写入寄存器（WRR）命令将状态寄存器保护0（SRP0）位设为1或0。状态寄存器保护0（SRP0）位和写入保护（WP#）信号允许对BP位进行硬件保护。

状态寄存器保护0（SRP0 SR1V[7]）位为0时，无论写入保护（WP#）信号被驱动为逻辑高或逻辑低状态，都可以对状态寄存器进行写操作，前提是已经发送了WREN或WRENV命令。

状态寄存器保护0（SRP0）位被设置为1时，需要考虑写入保护（WP#）信号的状态：

- 如果写入保护（WP#）信号被驱动为逻辑高状态，可以对状态和配置寄存器进行写操作，假定在WRR命令前已经发送了WREN或WRENV命令。
- 如果写入保护（WP#）信号被驱动为逻辑低状态，即使在WRR命令前已经发送了WREN或WRENV命令，也不能对状态和配置寄存器进行写操作。尝试对状态和配置寄存器进行的写操作被拒绝，不可执行，并且不提供错误指示。因此，保护存储器区域中的所有数据字节（由状态寄存器中的传统块保护位保护）也受WP#的硬件保护。

注意：在命令周期中推荐不要修改写保护WP#信号的状态。可以通过以下方式提供WP#硬件保护：

- 将写入保护（WP#）信号驱动为逻辑低状态后设置状态寄存器保护0（SRP0）；
- 或者，设置状态寄存器保护0（SRP0）位为1后将写入保护（WP#）信号驱动为逻辑低状态。

释放硬件保护的唯一方法是将写入保护（WP#）信号拉到逻辑高状态。如果WP#被永久性置为高电平，则不能激活BP位的硬件保护。

硬件保护将在使能四线模式（CR1V[1] = 1）或QPI模式（CR2V[3] = 1）下被禁用，因为这时WP#将成为IO2，因此不能使用它。

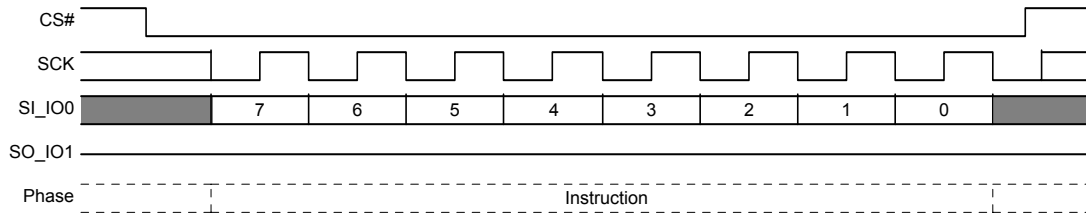
请参考第45页上的7.5 状态寄存器保护（SRP1、SRP0）一节，查看SRP和WP#对状态和配置寄存器保护的控制。

8.3.5 写入使能（WREN 06h）

写入使能（WREN）命令会将状态寄存器1（SR1V[1]）的写入使能锁存（WEL）位设置为1。需要发送写入使能（WREN）命令以使能写入、编程和擦除操作，这样才能将写入使能锁存（WEL）位设置为1。

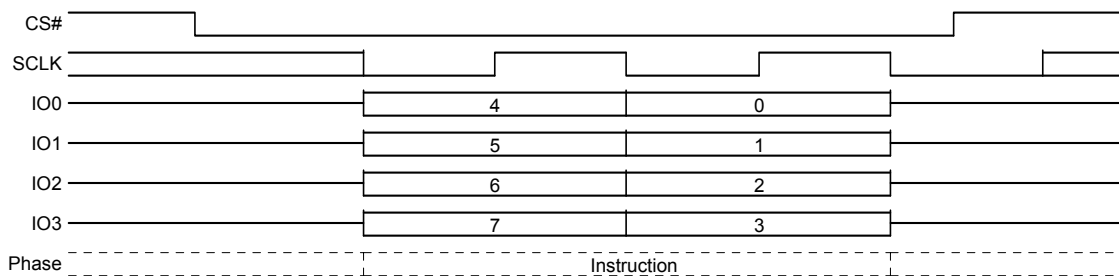
指令字节的第8位被锁存到SI/IO0上之后，必须将CS#信号驱动为逻辑高电平状态。否则，将不会执行写入使能操作。

图8.15 写入使能（WREN）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.16 QPI模式下的写入使能（WREN）命令序列



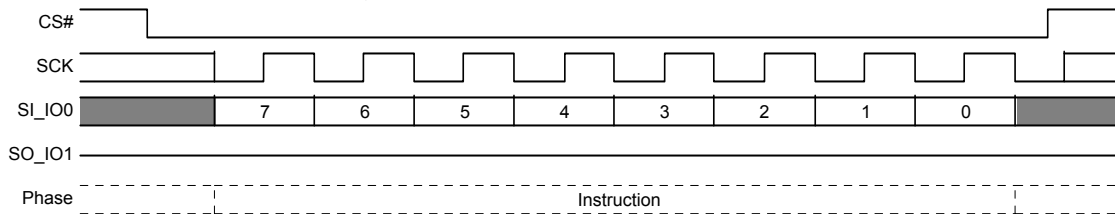
8.3.6 写入禁用（WRDI 04h）

写入禁用（WRDI）命令会将状态寄存器1的写入使能锁存（WEL）位（SR1V[1]）清除为0。

要想将写入使能锁存（WEL）位清除为0，需要发送写入禁用（WRDI）命令，以便禁止页编程（PP、4PP、QPP、4QPP）、扇区擦除（SE）、半块擦除（HBE）、块擦除（BE）、芯片擦除（CE）、写入寄存器（WRR或WRAR）、安全区域擦除（SECRE）、安全区域编程（SECRP）和其它命令。需要将WEL设为1才可执行这些命令。用户可以使用WRDI命令来保护存储区域，防止可能使存储器内容受损害的意外写操作。WIP位 = 1时，WRDI命令在执行嵌入式操作期间被忽略。

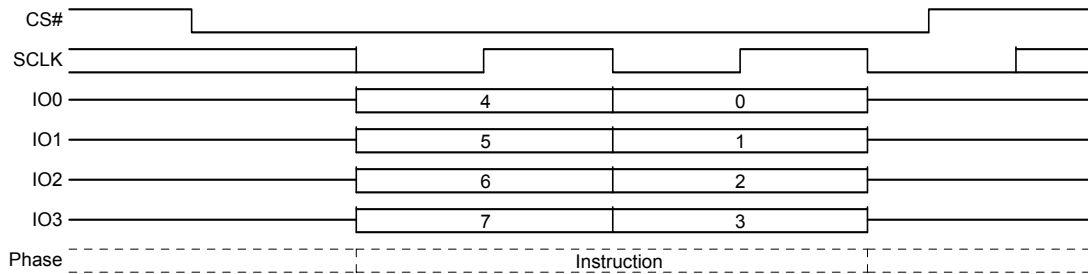
指令字节的第8位被锁存到SI/IO0上之后，必须将CS#信号驱动为逻辑高电平状态。否则将不会执行写入禁用操作。

图8.17 写入禁用（WRDI）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

图8.18 QPI模式下的写入禁用（WRDI）命令序列

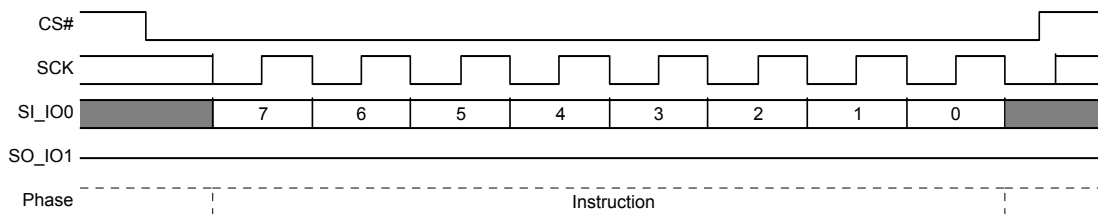


8.3.7 易失性寄存器的写入使能（WRENV 50h）

分别发送WRENV和WRR命令后，便能够对第27页上的6.6 寄存器一节中所述的易失性SR1V、CR1V、CR2V和CR3V寄存器进行写操作。这样可以灵活、快速地修改系统配置和存储器保护方案，不用等待典型的非易失性位的写周期，也不会影响非易失性状态或配置寄存器位的耐久性。WRENV命令将不会设置写入使能锁存（WEL）位，WRENV仅用于指导WRR命令修改易失性状态和配置寄存器位值。

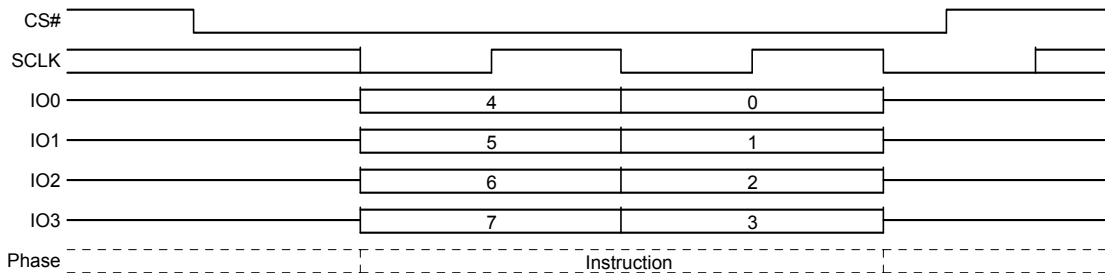
指令字节的第8位被锁存到SI/IO0上之后，需要将CS#信号驱动为逻辑高电平状态。否则，将不会执行写入使能操作。

图8.19 易失性寄存器的写入使能（WRENV）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

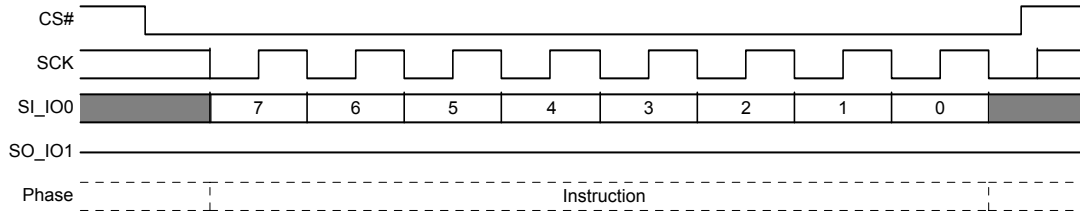
图8.20 QPI模式下的易失性寄存器的写入使能（WRENV）命令序列



8.3.8 清除状态寄存器 (CLSR 30h)

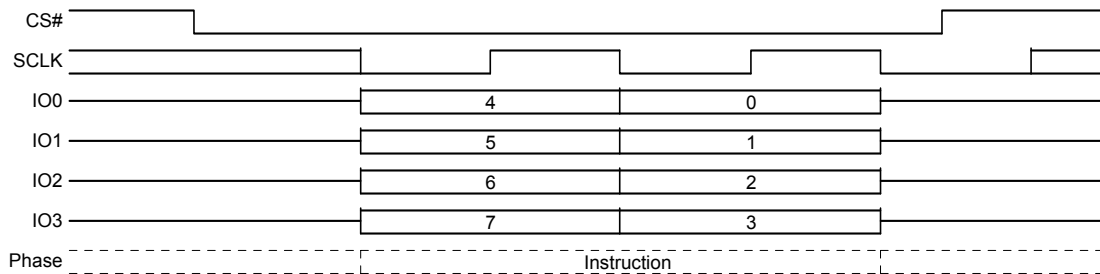
通过清除状态寄存器命令，可以将WIP (SR1V[0])、WEL (SR1V[1])、P_ERR (SR2V[5])和E_ERR (SR2V[6])位清除为0。执行清除状态寄存器命令前，不需要设置WEL位。即使器件由于WIP被设为1而保持繁忙状态，它也会接受清除状态寄存器命令，这是因为当两个错误位中的一个被设置时，器件也会保持繁忙状态。

图8.21 清除状态寄存器 (CLSR) 命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

图8.22 QPI模式下的清除状态寄存器 (CLSR)



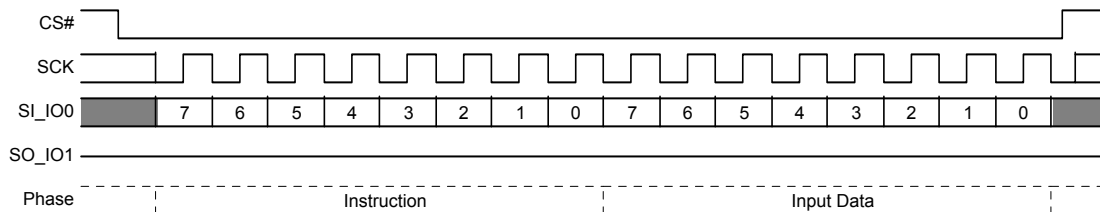
8.3.9 编程DLRNV (PDLRNV 43h)

接受编程DLRNV (PDLRNV) 命令前，器件必须发送写入使能 (WREN) 命令并对其进行解码。成功解码写入使能 (WREN) 命令后，器件将设置写入使能锁存 (WEL) 位以使能PDLRNV操作。

通过在SI/IO0上传指令和数据字节，可以输入PDLRNV命令。

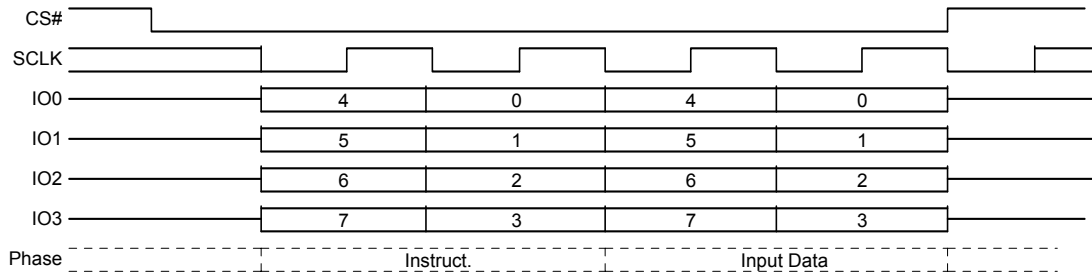
如果数据字节的第8位被锁存，则需要将CS#驱动为逻辑高电平。否则，将不能执行PDLRNV命令。CS#被驱动为逻辑高电平状态后，会立即启动自定时PDLRNV操作。在执行PDLRNV操作的过程中，可以读取状态寄存器以检查正在写入 (WIP) 位的值。执行自定时PDLRNV操作期间，WIP位为1；完成该操作时，WIP位为0。PDLRNV命令可以报告状态寄存器中P_ERR位的编程错误。完成PDLRNV操作时，写入使能锁存 (WEL) 位将被设置为0。PDLRNV命令的最大时钟频率为133 MHz。

图8.23 编程DLRNV (PDLRNV) 命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令和数据传入到器件内。

图8.24 QPI模式下的编程DLRNV (PDLRNV) 命令序列



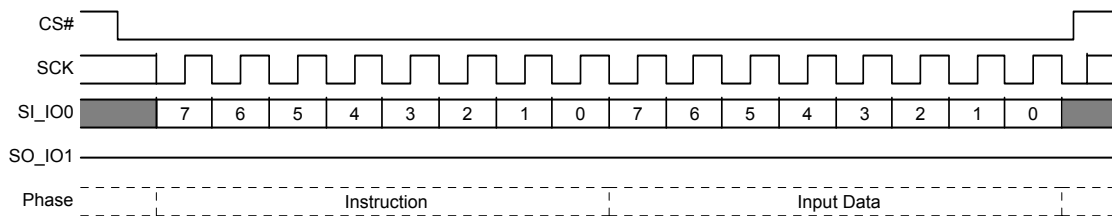
8.3.10 写入DLRV (WDLRV 4Ah)

接受写入DLRV (WDLRV) 命令前, 器件必须发送写入使能 (WREN) 命令并对其进行解码。成功解码写入使能 (WREN) 命令后, 器件将设置写入使能锁存 (WEL) 位以使能WDLRV操作。

通过将指令和数据字节移至SI/IO0, 可以输入WDLRV命令。

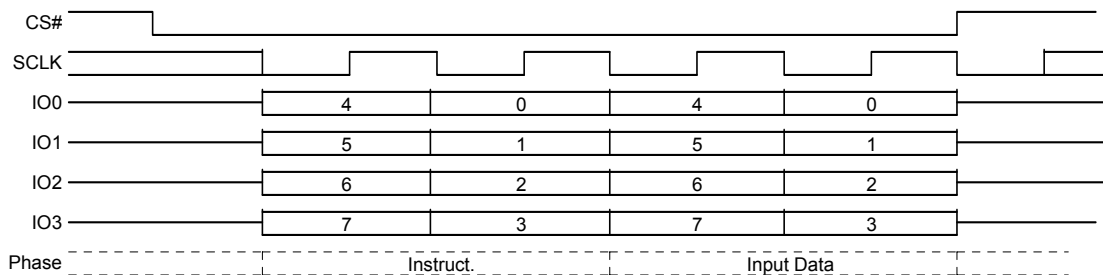
数据字节的第8位被锁存后, 必须将CS#驱动为逻辑高电平状态。否则, 不能执行WDLRV命令。CS#被驱动为逻辑高电平状态后, 会立即启动WDLRV操作。WDLRV命令的最大时钟频率为133 MHz。

图8.25 写入DLRV (WDLRV) 命令序列



QPI模式也支持该命令。在该模式下, 通过IO0-IO3信号可将指令和数据传入到器件内。

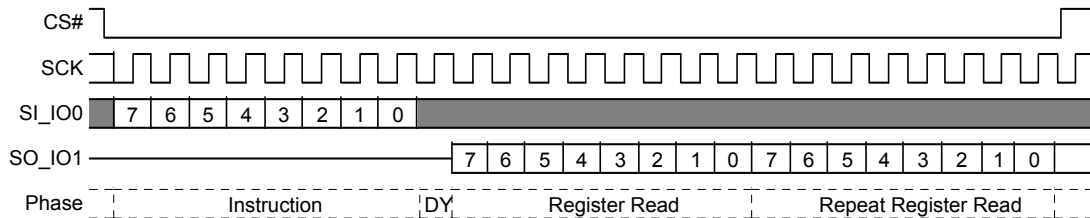
图8.26 QPI模式下的写入DLRV (WDLRV) 命令序列



8.3.11 数据学习模式读取 (DLPRD 41h)

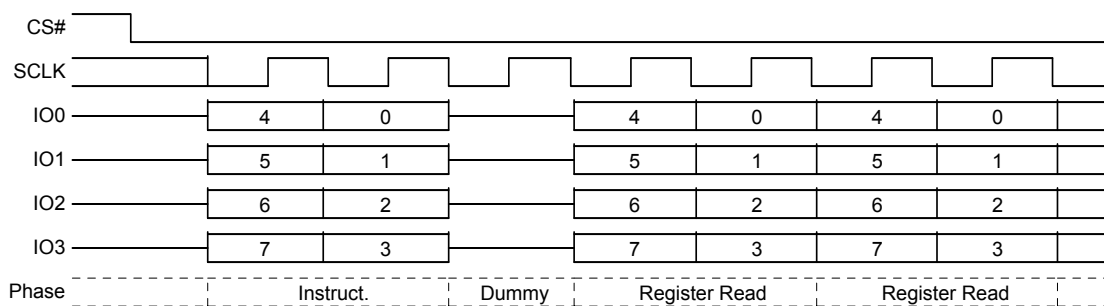
在SCK信号的上升沿上，将指令41h传送到SI/IO0，传入完成后存在一个虚拟周期。延迟周期为器件的内部电路提供需要的时间用于访问初始地址的数据。在这些延迟周期内，IO0-IO3上的数据值为“无需关注”项，该值可能为高阻状态。然后8位DLP通过SO/IO1信号被传出。通过提供8个时钟周期的倍数，可以连续读取DLP。DLPRD命令的最大工作时钟频率为133 MHz。

图8.27 DLP读取 (DLPRD) 命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令并传出返回的数据。

图8.28 QPI模式下的DLP读取 (DLPRD) 命令序列

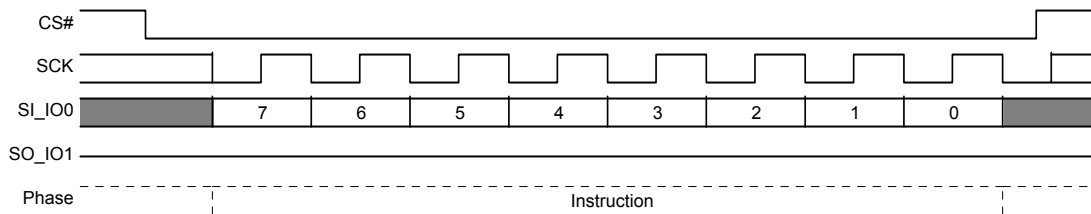


8.3.12 进入4字节地址模式 (4BEN B7h)

通过进入4字节地址模式 (4BEN) 命令，可以将易失性地址长度状态 (ADS) 位 (CR2V[0]) 设为1，使所有3字节地址命令都要求4字节地址。该命令对仅支持4字节的命令不会产生影响，这些命令仍要求4字节地址。

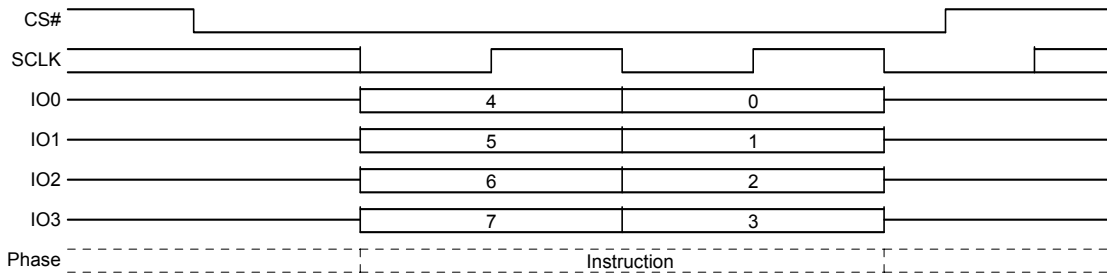
要想返回到3字节地址模式，可以通过4BEX或WRAR命令将易失性地址长度位CR2V[0]清除为0实现。此外，如果非易失性地址长度位CR2NV[1]为0，可以使用硬件复位或软件复位来返回3字节地址模式。

图8.29 进入4字节地址模式 (4BEN B7h) 命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

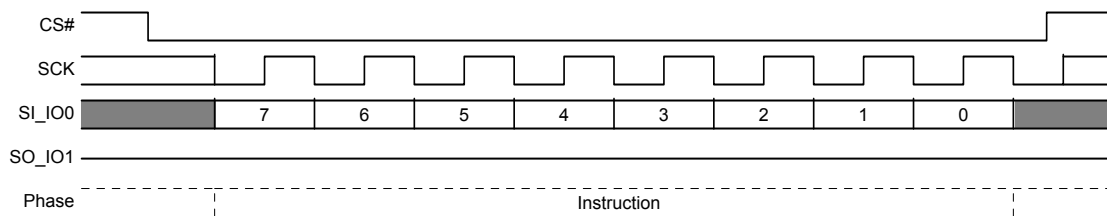
图8.30 进入4字节地址模式



8.3.13 退出4字节地址模式（4BEX E9h）

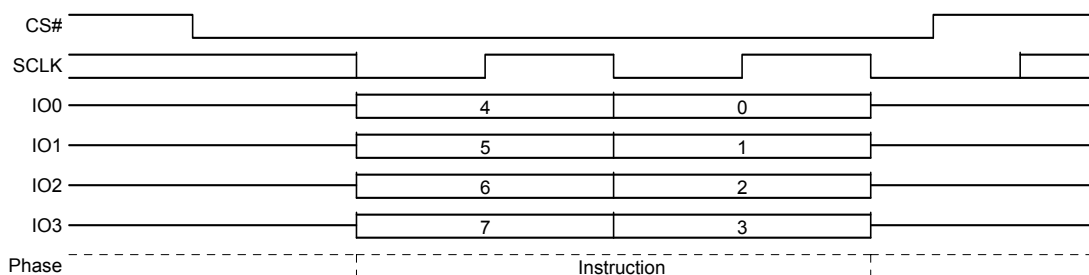
通过退出4字节地址模式（4BEX）命令，可以将易失性地址长度状态（ADS）位（CR2V[0]）设置为0，使几乎所有4字节地址命令都要求3字节的地址。该命令不会对支持4字节的命令产生影响，这些命令仍要求大小为4字节的地址。

图8.31 退出4字节地址模式（4BEX E9h）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

图8.32 QPI模式下的退出4字节地址模式命令



8.3.14 读取任意寄存器（RDAR 65h）

通过读取任意寄存器（RDAR）命令，可以读取器件寄存器。该指令后面是3或4字节地址（取决于地址长度配置CR2V[0]），最后是由CR3V[3:0]指定的延迟（虚拟）周期数量。然后将返回选定的寄存器的内容。如果继续执行读取访问，将返回同样的寄存器内容，直到命令终止为止。每个RDAR命令只能读取一个寄存器。

读取未定义地址会返回未定义数据。

在执行嵌入式操作期间，可以使用RDAR命令来读取状态寄存器1（SR1V）。

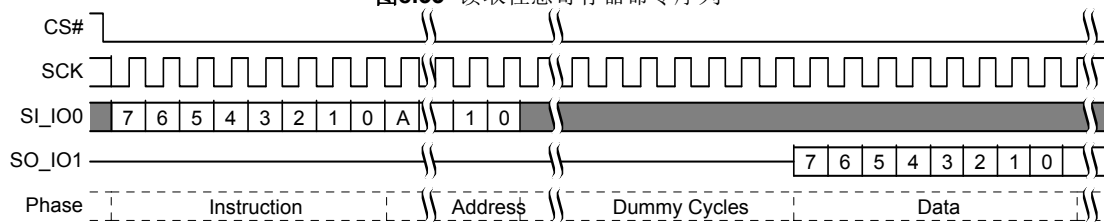
RDAR命令不用于读取反映更大阵列的寄存器，例如：IBLAR。需要使用一些单独命令来选择并读取被访问阵列中的位置。

如果通过将IRP[2]编程为0来选择IRP密码保护模式，则RDAR命令会从PASS寄存器中读取无效数据。

表 8.4 寄存器地址映射

字节地址（十六进制）	寄存器名称	说明
000000	SR1NV	非易失性状态和配置寄存器 读取非易失性状态和配置寄存器实际上是读取易失性寄存器
000001	N/A	
000002	CR1NV	
000003	CR2NV	
000004	CR3NV	
000005	NVDLP	
...	N/A	
000020	PASS[7:0]	非易失性密码寄存器
000021	PASS[15:8]	
000022	PASS[23:16]	
000023	PASS[31:24]	
000024	PASS[39:32]	
000025	PASS[47:40]	
000026	PASS[55:48]	
000027	PASS[63:56]	非易失性
...	N/A	
000030	IRP[7:0]	
000031	IRP[15:8]	指针区域保护寄存器A15:A8
...	N/A	
000039	PRPR[A15:A8]	
00003A	PRPR[A23:A16]	指针区域保护寄存器A23:A16
00003B	PRPR[A31:A24]	指针区域保护寄存器A31:A24
...	N/A	易失性状态和配置寄存器
800000	SR1V	
800001	SR2V	
800002	CR1V	
800003	CR2V	
800004	CR3V	
800005	VDLP	
...	N/A	易失性保护寄存器
800040	PR	
...	不可用	

图8.33 读取任意寄存器命令序列

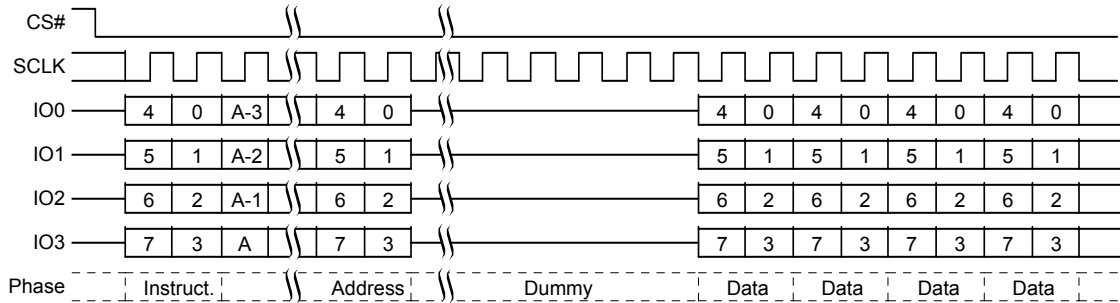


注意:

1. A = 地址的MSB为23 (CR2V[0] = 0) 或31 (CR2V[0] = 1)。

QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令和地址并传出返回数据。

图8.34 QPI模式下的读取任意寄存器命令序列



注意:

1. A = 地址的MSB为23 (CR2V[0] = 0) 或31 (CR2V[0] = 1)。

8.3.15 写入任意寄存器 (WRAR 71h)

通过写入任意寄存器 (WRAR) 命令，可以对任意的非易失性或易失性器件寄存器进行写操作。该指令后面是一个3或4字节的地址（取决于地址长度配置CR2V[0]），然后将将被写入到选定寄存器内的1字节的数据。

S25FL256L器件必须使能4字节地址模式 (CR2V[0] = 1)，以便设置指针区域保护寄存器PRPR（请参考第41页上的6.6.10节）。

接受WRAR命令前，器件必须发送写入使能 (WREN) 命令并对其进行解码。这样将设置状态寄存器中的写入使能锁存 (WEL) 位，用于使能所有写入操作。通过检查SR1V中的WIP位可以确定该操作是否已经完成。同样，通过检查SR2V中的P_ERR和E_ERR位可以确定在操作过程中是否发生了错误。

部分寄存器结合使用了各个位类型和规则，从而能够控制可修改位。有的是只读位，也有的是OTP位。

只读位永远不能被修改，另外WRAR命令中数据字节的相应位被忽略，并没有设置编程或擦除错误指示 (SR2V中的P_ERR或E_ERR)。因此，WRAR数据字节中这些位的值并不重要。

OTP位只能被编程为其默认状态的相反状态。将OTP位返回其默认状态的操作被忽略，并且不会设置错误。

WRAR命令需要非易失性寄存器的写入时间 (t_W) 来更新非易失性位的值。更新过程包括对非易失性寄存器位进行的擦除和编程操作。如果更新过程的擦除或编程操作失败，则SR2V中的相关错误位和SR1V中的WIP位将被设置为1。

WRAR命令需要易失性寄存器的写入时间 (t_{CS}) 来更新易失性位的值。

状态寄存器1被重复读取 (轮询)，用以监控正在写入 (WIP) 位 (SR1V[0])，从而确定是否完成了寄存器写入操作；并通过重复读取状态寄存器2的错误位 (SR2V[6、5]) 可确定是否发生写入故障。如果发生写入故障，需要使用清除状态命令来清除错误状态，使器件返回到待机状态。完成WRAR操作时，写入使能锁存 (WEL) 位将被设置为0。

但不能通过WRAR命令对PR寄存器进行写操作。PR寄存器的内容是只读的。只有使用NVLOCK位写入 (PRL) 命令才可写入PR寄存器。

如果使用WRAR命令，对SR1NV、CR1NV、CR2NV和CR3NV的写操作不受硬件和软件复位的影响，但是对其它所有寄存器的写操作由于这两种复位而被复位。

WRAR命令的序列和行为与PP或4PP命令相同，只是该命令能提供一个数据字节。请参考第89页上的8.5.2 页编程 (PP 02h或4PP 12H) 一节。

寄存器的地址映射如第76页上的表8.4. 寄存器地址映射所示的内容一样。

8.3.16 设置连续长度 (SBL 77h)

设置连续长度 (SBL) 命令用于配置连续回卷特性。在QIO或QPI模式下，可以将回卷突发特性同四线I/O读取和DDR四线I/O读取命令结合使用，用于访问长度固定并对齐的数据。某些应用可以利用该特性来提高整个系统代码执行的能力。通过回卷突发特性，使用缓存器的应用能够先将关键地址的指令或数据填充到第一个缓存行内，然后将长度固定的数据 (8/16/32/64字节) 填充到剩余的缓存行，而不需要发出多个读取命令。

通过将CS#引脚驱动为低电平，可以初始化设置连续长度命令，然后依次发送指令代码“77h”、24个虚拟位和8个回卷长度位 (WL[7]-WL[0])。该命令序列如第79页上的图8.35. 四线I/O模式下的设置连续长度命令序列和第79页上的图8.36. QPI模式下的设置连续长度命令序列中所示。回卷长度位WL[7]和低位半字节WL[3:0]均不被使用。请参考第36页上的6.6.5 配置寄存器3一节中的配置寄存器3 (CR3V[6:4])，了解有关WL[6]-WL[4]的编码。

一旦WL[6:4]位由设置连续长度命令设置时，则所有“四线I/O读取”命令都会使用WL[6:4]的设置来访问8/16/32/64字节的数据部分。请注意，必须将配置寄存器1的Quad位CR1V[1]或配置寄存器2的QPI位CR2V[3]置1，以便能够使用四线I/O读取和设置突发长度命令。要想退出“回卷”模式并返回到正常的读取操作，需要发出另一个设置连续回卷命令，将WL4设为1。在发生上电、硬件或软件复位事件时，WL[6:4]的值和CR2NV[6:5]的值相同。请使用WRR或WRAR命令设置CR2NV[6:2]的默认回卷长度。

设置连续长度 (SBL) 命令仅通过写入CR3V[6:4]位来使能或禁用回卷读取功能，并设置回卷边界。不能使用SBL命令来设置CR3V[3:0]中的读延迟。必须使用WRAR命令来设置CR3V或CR3NV中的读延迟。

请参考第78页上的表8.5. 连续回卷序列示例一节中的CR3V[6:5]的值，以了解它们的回卷边界和起始地址。使能回卷读取功能时，会将相关读取命令从连续读取改变为对一组字节进行连续回卷读取，直到该命令停止为止。

未使能回卷模式时 (请参考表 6.15和表 6.18)，可以连续读取无限长度的数据。

使能回卷模式时 (请参考表 6.15和表 6.18)，只能从读取命令提供的字节地址开始读取长度固定且对齐的8、16、32或64字节组，并且在该字节组边界中回卷。

字节组的长度和对齐情况都按8、16、32或64字节的边界实现。CR3V[6:5]用于选择该边界。请参见第38页上的6.6.5.2 易失性配置寄存器3 (CR3V) 一节。

读取命令的起始地址选择了字节组，并且所返回的第一个数据是寻址到的字节。然后，会连续读取各个字节，直到字节组边界结束为止。如果继续执行读取，则会返回到该组的起始地址，然后再次连续读取数据。该回卷读取序列会继续进行，直到将CS#返回高电平来停止该命令为止。

表 8.5 连续回卷序列示例

CR3V值 (十六进制)	回卷边界 (字节)	起始地址 (十六进制)	地址序列 (十六进制)
1X	连续	XXXXXX03	03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, ...
00	8	XXXXXX00	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, ...
00	8	XXXXXX07	07, 00, 01, 02, 03, 04, 05, 06, 07, 00, 01, ...
01	16	XXXXXX02	02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 00, 01, 02, 03, ...
01	16	XXXXXX0C	0C, 0D, 0E, 0F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, ...
02	32	XXXXXX0A	0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, ...
02	32	XXXXXX1E	1E, 1F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 00, ...
03	64	XXXXXX03	03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02, ...
03	64	XXXXXX2E	2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, ...

通过使用WRAR命令将CR3NV编程为所需的值，可以改变上电复位、硬件复位或软件复位的默认连续长度。

图8.35 四线I/O模式下的设置连续长度命令序列

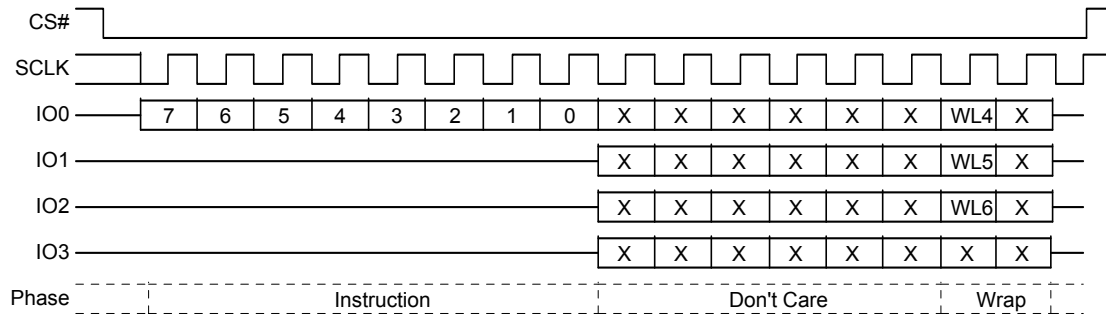
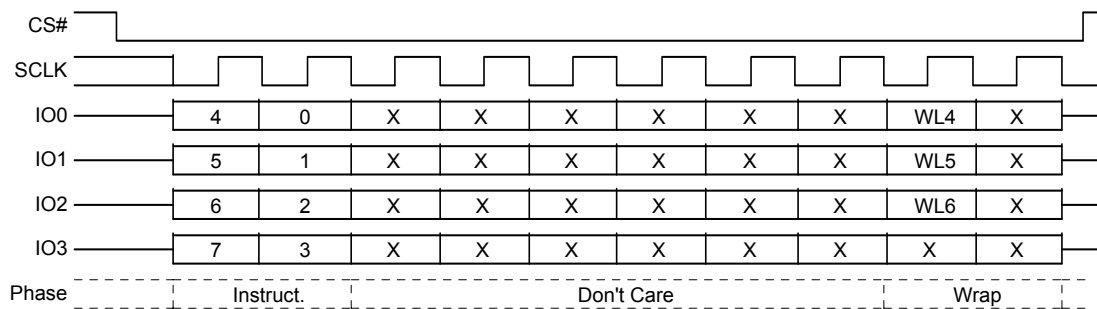


图8.36 QPI模式下的设置连续长度命令序列

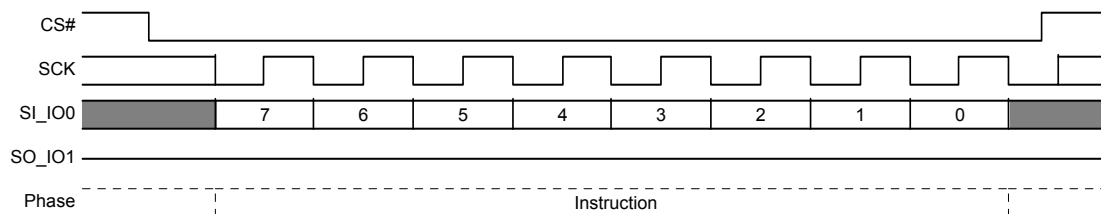


8.3.17 进入QPI模式 (QPIEN 38h)

进入QPI模式 (QPIEN) 命令可通过设置易失性QPI位 (CR2V[3] = 1) 来使能QPI模式。请参见第35页上的表6.13. 易失性配置寄存器2 (CR2V)。器件进入QPI模式所需要的时间为 t_{QEN} (请参考第134页上的表12.4. SDR交流特性)，在这段时间内，不允许执行任何其他命令。

要想返回SPI模式，需要发出QPIEX命令或对寄存器进行写操作 (CR2V[3] = 0)。清除非易失性QPI位 (CR2NV[3] = 0)，如果发生上电复位、硬件复位或者软件复位事件也会使器件返回到SPI模式。请参见第34页上的表6.11. 非易失性配置寄存器2 (CR2NV)。

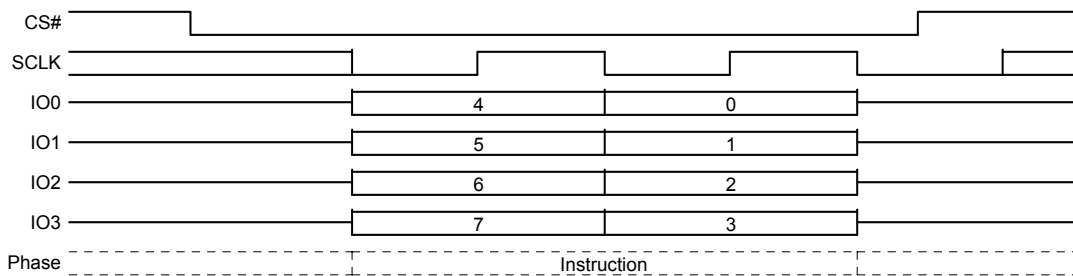
图8.37 进入QPI模式 (QPIEN 38h) 命令序列



8.3.18 退出QPI模式 (QPIEX F5h)

退出QPI模式 (QPIEX) 命令通过设置易失性QPI位 (CR2V[3] = 0) 可以禁用QPI模式，并使器件返回到SPI模式。请参见第35页上的表6.13. 易失性配置寄存器2 (CR2V)。器件退出QPI模式所需要的时间为 t_{QEX} (请参考第134页上的表12.4. SDR交流特性) 在这段时间内，不允许执行任意其他命令。

图8.38 退出QPI (QPIEX F5h) 命令序列



8.4 读取闪存阵列命令

闪存主阵列的读取命令允许灵活地与前一代SPI相兼容或增大SPI性能：

- 某些命令可在每个SCK的上升沿上传输地址或数据。它们被称为单倍数据速率（SDR）命令。
- 某些SDR命令可在每个SCK的下降沿上发送一位地址，并在SCK的上升沿上返回一位数据。它们被称为单比特宽命令。
- 某些SDR命令会在每个SCK的上升沿上发送双比特或四比特的地址和数据它们分别被称为双线I/O（针对双比特传输）和四线I/O和QPI（针对四比特传输）。QPI也会在每个SCK的上升沿上发送四比特的指令。
- 某些命令可在SCK的双边沿上传输地址和数据。它们被称为双倍数据速率（DDR）命令。
- 一些DDR命令可在每个SCK边沿上发送四位的地址或数据。它们被称为四线I/O DDR和QPI DDR。

所有这些命令（QPI读取命令除外）都以指令代码开始，并且将在每个SCK上升沿上传输一位。QPI读取命令在每个SCK的上升沿上发送四比特的指令，然后SDR或DDR会发送3或4字节的地址。在每个时钟沿上传输双比特或四比特地址/数据的命令被称为多路I/O（MIO）命令。对于256 Mb或容量更大的FL-L系列器件，传统的SPI 3字节地址不能直接对存储器阵列中所有位置进行寻址。要想访问整个地址空间，可以使用一些单独的4字节地址读命令。可以配置这些器件，使之能够通过传统的3字节地址命令从主机系统接收4字节的地址。通过将配置寄存器2中的地址长度位设置为1，可以为传统命令激活4字节的地址模式。在S25FL128L中，4字节地址命令中超过A23的高端地址位和使用4字节地址模式的命令互不相关，并且都被忽略。这是因为闪存阵列大小只有128 Mb。

双线I/O、四线I/O和QPI命令通过模式位（该位在发送地址位后被发送）可以提高器件的性能。模式位指出下一个读取命令的类型是否与当前读取命令相同，因此在下个读取命令的开始无需包含指令。通过这些模式位，进行一系列的双线或四线读访问时可减少指令周期。

某些命令要求移入地址或模式位后需要添加延迟周期（读取延迟），以便能够访问存储器阵列。读取延迟周期通常被称为虚拟周期。存储器忽略虚拟周期，因此主机在这些周期内发送的所有数据也被忽略（“无需关注”），此时主机可以将SI信号保持为高阻抗状态。使用MIO命令时，主机必须在最后虚拟周期结束前停止驱动I/O信号（各输出均为高阻抗状态）。使用DDR命令时，主机在任意虚拟周期内都不能驱动I/O信号。因为由配置寄存器2（CR3V[3:0]）延迟代码选定的SCK频率或性能不同，虚拟周期数量也不一样。虚拟周期指的是从SCK的下降沿到下一个SCK的下降沿的时间。通常在每个SCK的下降沿上将SPI输出驱动到新的数值。这时需要零个虚拟周期，因为在同一个SCK下降沿上，存储器将数据返回给主机的同时，主机也停止发送地址或模式位。

DDR命令可以在数据的开始前添加一个8边沿数据学习模式（DLP），该DLP由存储器通过所有数据输出在虚拟周期内驱动的。通过DLP，主机存储器控制器能够确定SCK与数据边沿之间的相移，从而存储器控制器可以在数据边沿的中心捕获数据。

SCK时钟频率较高（> 50 MHz）时使用SDR I/O命令，应选择提供了一个或多个虚拟周期的LC，这样可以为主机提供额外的时间，使它在存储器开始发送数据前停止执行发送操作，从而避免在I/O上发生冲突。如果使用的是使能了DLP的SDR I/O命令，应选择能够提供5个或多个虚拟周期的LC，这样可以为主机提供一个周期时长的额外时间，使它在存储器开始发送4个周期DLP前停止执行发送操作。

在数据返回期间，CS#从任意状态返回到高电平时，将结束读命令。在连续读取模式下或在虚拟周期中，CS#不能在返回数据前变为高电平，否则会使模式位错误地被捕获，从而不能确定器件是否仍处于连续读取模式。

8.4.1 读取（读取03h或4READ 13h）

指令

- 先发送03h (CR2V[0] = 0)，然后发送3字节地址 (A23-A0)，或
- 先发送03h (CR2V[0]=1)，然后发送4字节地址 (A31-A0)，或
- 先发送13h (CR2V[0]=1)，然后发送4字节地址 (A31-A0)

此时，存储器通过SO/IO1信号发送指定地址的内容。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后，会按顺序自动递增到下一个更高的地址。因此，只要使用一个读取命令和已给的地址000000h，便可以读取整个存储器空间。当到达最高地址时，地址计数器翻转到起始地址000000h，这样允许读取序列继续下去。

图8.39 读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0)，或 = 31 (若CR2V[0] = 1或命令13h)。

8.4.2 快速读取（FAST_READ 0Bh或4FAST_READ 0Ch）

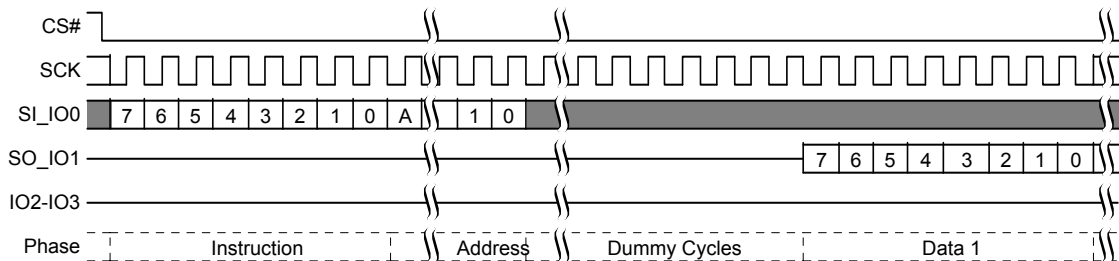
指令

- 先发送0Bh (CR2V[0] = 0)，然后发送3字节地址 (A23-A0)，或
- 先发送0Bh (CR2V[0] = 1)，然后发送4字节地址 (A31-A0)，或
- 先发送0Ch，然后发送4字节地址 (A31-A0)

发送地址后会提供虚拟周期，该周期数量取决于配置寄存器CR3V[3:0]中所设置的延迟值。虚拟周期为器件的内部电路提供额外的时间，使之能够访问重要的地址位置。在这些延迟周期内，SO/IO1上的所有数据均被忽略，并且该信号会保持为高阻抗状态。此时，存储器通过SO/IO1信号发送指定地址中的内容。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后，会按顺序自动递增到下一个更高的地址。因此，只要使用一个读取指令和给定的地址000000h，便可以读取整个存储器的内容。当到达最高地址时，地址计数器翻转到起始地址000000h，这样允许读取序列继续下去。

图8.40 快速读取（FAST_READ）命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0)，或 = 31 (若CR2V[0] = 1或命令0Ch)。

8.4.3 双线输出读取 (DOR 3Bh或4DOR 3Ch)

指令

- 先发送3Bh (CR2V[0] = 0), 然后发送3字节地址 (A23-A0), 或
- 先发送3Bh (CR2V[0] = 1), 然后发送4字节地址 (A31-A0), 或
- 先发送3Ch, 然后发送4字节地址 (A31-A0)

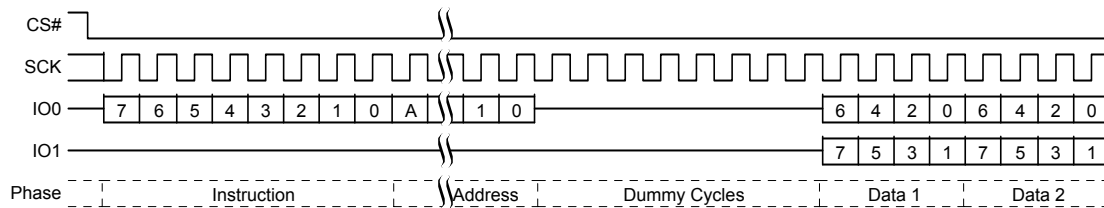
发送地址后会提供虚拟周期, 该周期数量取决于配置寄存器CR3V[3:0]中所设置的延迟值。虚拟周期为器件的内部电路提供额外的时间, 使之能够访问重要的地址位置。在这些虚拟周期内, IO0 (SI) 和IO1 (SO) 上的所有数据均被忽略, 并会保持为高阻抗状态。

此时, 存储器通过IO0 (SI) 和IO1 (SO) 信号每次发送给定地址中的双比特数据。在SCK的下降沿上以SCK频率输出双比特数据。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后, 会按顺序自动递增到下一个更高的地址。因此, 只要使用一个读取指令和给定的地址000000h, 便可以读取整个存储器的内容。当到达最高地址时, 地址计数器翻转到起始地址000000h, 这样可使读取序列继续下去。

对于双线输出读取命令, 从最后地址位移入IO0 (SI) 后到数据从IO0和IO1移出前, 需要几个虚拟周期。

图8.41 双线输出读命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令3Ch)。

8.4.4 双线输出读取 (QOR 6Bh或4QOR 6Ch)

指令

- 先发送6Bh (CR2V[0] = 0), 然后发送3字节地址 (A23-A0), 或
- 先发送6Bh (CR2V[0] = 1), 然后发送4字节地址 (A31-A0), 或
- 先发送6Ch, 然后发送4字节地址 (A31-A0)

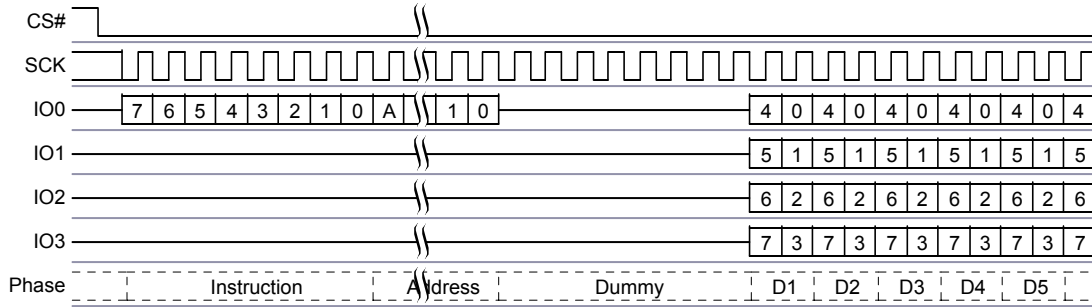
发送地址后会提供虚拟周期, 该周期数量取决于配置寄存器CR3V[3:0]中所设置的延迟值。虚拟周期为器件的内部电路提供额外的时间, 使之能够访问重要的地址位置。在这些虚拟周期内, IO0 - IO3上的所有数据均被忽略, 并且这些信号会保持为高阻抗状态。

此时, 存储器通过IO0 - IO3信号每次发送指定地址中的四位数据。在SCK的下降沿上以SCK频率输出半字节 (4位)。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后, 会按顺序自动递增到下一个更高的地址。因此, 只要使用一个读取指令和给定的地址000000h, 便可以读取整个存储器的内容。当到达最高地址时, 地址计数器翻转到起始地址000000h, 这样允许读取序列继续下去。

对于四线输出读取命令, 从最后地址位移入IO0后到数据从IO0 - IO3移出前, 需要几个虚拟周期。

图8.42 四线输出读命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0]=0), 或=31 (若CR2V[0]=1或命令6Ch)。

8.4.5 双线I/O读取 (DIOR BBh或4DIOR BCh)

指令

- 先发送BBh (CR2V[0]=0), 然后发送3字节地址 (A23-A0), 或
- 先发送BBh (CR2V[0]=1), 然后发送4字节地址 (A31-A0), 或
- 先发送BCh, 然后发送4字节地址 (A31-A0)

双线I/O读取命令通过两个I/O信号, 即IO0 (SI) 和IO1 (SO) 提高了吞吐量。该命令在每个SCK的上升沿上接收地址的输入, 并返回两位读取数据。在某些应用中, 减少地址输入和数据输出的时间会允许立即执行代码 (XIP), 即直接从存储器件执行代码。

双线I/O读取命令中的连续读取模式位也位于地址后面; 第一个双线读取命令发出模式位格式Axh来表示下一个命令也是双线I/O读取命令之后, 一系列双线I/O读取命令可会清除8位指令。一系列双线I/O读取命令中的第一个命令的顺序为: 8位指令、地址、4个模式位周期, 最后是可选的延迟周期。如果模式位格式为Axh, 假设下个命令是不提供指令位的其他双线I/O读取命令。该命令的组成部分分别为地址、模式位和可选的延迟。

从模式位移入SI和SO后到数据从IO0和IO1移出前, 需要添加可变的延迟。这样, 器件的内部电路会有足够的时间来访问初始地址中的数据。在这些延迟周期内, SI和SO上的所有数据均被忽略, 并且这些信号会保持为高阻抗状态。虚拟周期的数量由SCK的频率确定。该延迟在CR3V[3:0]中进行配置。

凭借连续读取特性, 在读访问的序列中无需读取指令位, 这样能够明显提高代码执行 (XIP) 的性能。模式位的高位半字节 (位7-4) 通过添加或清除第一个字节指令代码来控制下一个双线I/O读取命令的长度。模式位的低位半字节 (位3-0) 则被忽略 (使用“x”表示), 并保持为高阻抗状态。如果模式位为Axh, 则器件仍处于双线I/O连续读取模式, 并且在CS#被驱动为高电平, 再被确认为低电平后能够移入下一个地址 (而不需要读取BBh或BCh指令 (如图8.44所示)); 这样可以消除命令序列中的8个周期。发生以下情况时会使器件退出双线I/O连续读取模式, 然后器件可以接收标准的SPI命令:

1. 在执行双线I/O连续读取命令序列过程中, 如果模式位为非Axh值, 则下次将CS#驱动为高电平时会使器件退出双线I/O连续读取模式。
2. 发送模式复位命令。

请注意, 最后的读取地址周期被定时在IO0 (S) 和IO1 (SO) 后, 器件的内部电路需要4个模式位周期的延迟时间来访问初始地址。

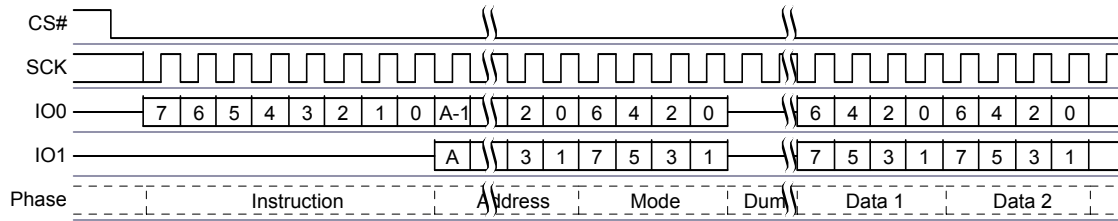
在第一个数据移出的时钟的下降沿上 (或之前), 必须将I/O信号设为高阻抗状态。在更高的时钟速度下, 存储器件开始发送数据前需要停止主机传输数据的时间 (即总线周转时间) 被减少。这样, 主机系统可以在最后两个“无需关注”的模式周期中或在任意虚拟周期内停止通过I/O信号发送数据 (将它们保持为高阻抗状态), 从而避免在I/O信号上发生冲突。

该延迟周期结束后, 存储器便通过IO0 (SI) 和IO1 (SO) 信号每次发送指定地址中的双比特数据。在SCK的下降沿上以SCK频率将输出双比特数据。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后, 会按顺序自动递增到下一个更高地址。因此, 只要使用一个读取指令和给定的地址000000h, 便可以读取整个存储器的内容。当到达最高地址时, 地址计数器翻转到起始地址000000h, 这样允许读取序列继续下去。

在模式/虚拟位周期内不应该将CS#驱动为高电平, 否则会使这些模式位的状态不确定。

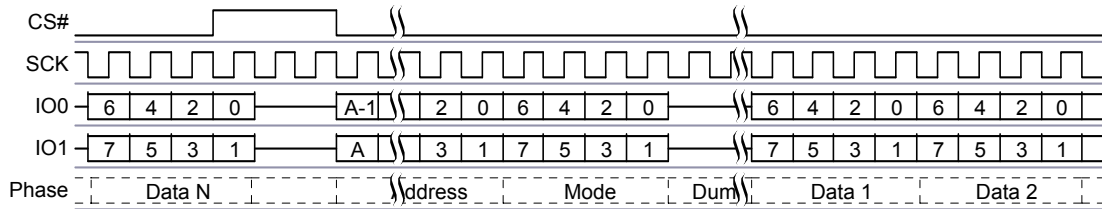
图8.43 双线I/O读命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令BCh)。
2. 4个最低有效模式位均被忽略, 主机可以选择发送这些位。主机在这些周期内会停止发送数据, 以便延长从主机发送模式位到存储器返回数据之间总线周转的时间。

图8.44 双线I/O连续读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令BCh)。

8.4.6 四线I/O读取（QIOR EBh或4QIOR ECh）

指令

- 先发送EBh（CR2V[0] = 0），然后发送3字节地址（A23-A0），或
- 先发送EBh（CR2V[0] = 1），然后发送4字节地址（A31-A0），或
- 先发送ECh先发送，然后发送4字节地址（A31-A0）

由于四线I/O读取命令使用了四个I/O信号（IO0-IO3），所以吞吐量得到提高。这样，可以在每个串行时钟SCK的边沿上移入4比特地址。在某些应用中，通过降低指令的开销，可以直接从FL-L系列器件执行代码（XIP）。必须设置配置寄存器1的QUAD位（CR1V[1] = 1）或配置寄存器2的QPI位（CR2V[1] = 1），以使能FL-L系列器件的四线模式。

执行四线I/O读取命令时，移入模式位后（如下所述）到通过IO0-IO3开始移出数据前需要添加一个延迟周期。该延迟周期（虚拟周期）为器件的内部电路提供了所需时间，用于访问初始地址的数据。在这些延迟周期内，IO0-IO3上的数据值均被忽略，并且这些信号会保持为高阻抗状态。虚拟周期的数量由SCK的频率确定。该延迟在CR3V[3:0]中进行配置。

该延迟周期结束后，存储器便通过IO0-IO3信号每次发送给定地址中的四比特数据。在SCK信号的下降沿上以SCK频率移出半字节（4位）。

该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后，会按顺序自动递增到下一个更高的地址。因此，只要使用一个读取指令和给定的地址000000h，便可以读取整个存储器的内容。当到达最高地址时，地址计数器翻转到起始地址000000h，这样可使读取序列继续下去。

不需要添加四线I/O读指令仍能跳到其他地址。通过设置地址序列后的模式位（如第85页上的图8.45所示），可以控制该操作。凭借这个附加的特性，无需读取指令序列，这样可明显提高代码执行（XIP）的能力。模式位的高位半字节（位7-4）通过添加或清除第一个字节指令代码来控制下一个四路I/O指令的长度。模式位的低位半字节（位3-0）被忽略（使用“x”表示）。如果模式位为A_{xh}，则器件仍处于四线I/O高性能读取模式，并且在CS#被驱动为高电平，然后再被确认为低电平后，能够移入下一个地址（不需要读取EBh或ECh指令（如第86页上的图8.47所示））。这样可以消除命令序列中的8个周期。发生以下情况时会使器件退出四线I/O高性能读取模式；然后器件可以接收标准的SPI命令：

1. 在执行四路I/O读取命令序列过程中，如果模式位为非A_{xh}的值，则下次将CS#驱动为高电平时会使器件退出四路I/O高性能读取模式。
2. 发送模式复位命令。

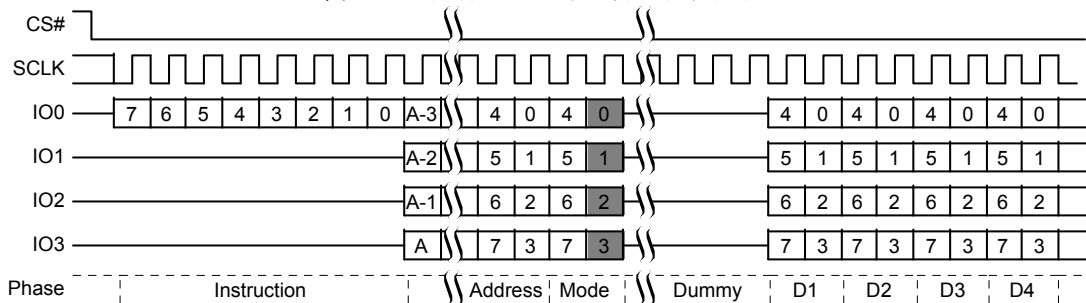
请注意，最后的读取地址周期被定时在IO0-IO3后，器件的内部电路需要两个模式位时钟周期和额外的等待时间（即虚拟周期）来访问初始地址。

应注意，在第一个数据移出的时钟的下降沿上（或之前），必须将IO0-IO3信号设为高阻抗状态。在更高的时钟速度下，存储器件开始发送数据前需要停止主机传输数据的时间（即总线周转时间）被减少。这样，主机系统可以在最后“无需关注”的模式周期内或在任意虚拟周期内停止通过IO0-IO3信号发送数据（将它们保持为高阻抗），从而避免在IO0-IO3信号上发生冲突。

在模式/虚拟位周期内不应该将CS#驱动为高电平，否则会使这些模式位的状态不确定。

在QPI模式下（CR2V[3] = 1），四线I/O指令都会在每个SCK的上升沿上发送4比特信息。该命令协议的其余部分和四线I/O命令完全相同。

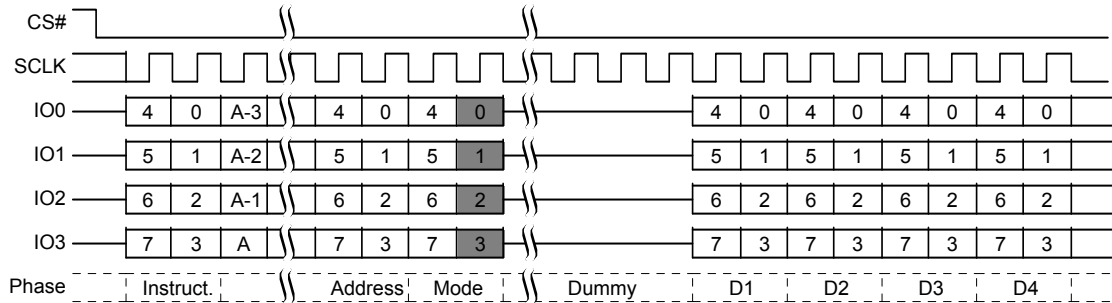
图8.45 初始访问的四线I/O读取命令序列



注意：

1. A = 地址的MSB = 23（若CR2V[0] = 0），或 = 31（若CR2V[0] = 1或命令ECh）。

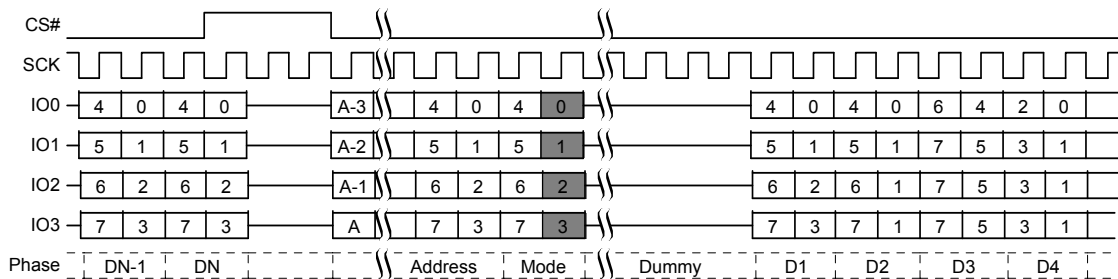
图8.46 在QPI模式下初始访问的四线I/O读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令ECh)。

图8.47 四线I/O连续读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令ECh)。
2. 在QPI模式下, 命令序列也相同。

8.4.7 DDR四线I/O读取（EDh、EEh）

DDR四线I/O读取命令因为使用了四个I/O信号（IO0-IO3），从而提高了吞吐量。它和四线I/O读取命令相同，但仍允许在每个时钟沿上移入4比特地址。在某些应用中，降低指令的开销，可以直接从FL-L系列器件执行代码（XIP）。必须设置配置寄存器1的QUAD位（CR1V[1] = 1）或配置寄存器2的QPI位（CR2V[1] = 1），以使能FL-L系列器件的四线模式。

指令

- 先发送EDh（CR2V[0] = 0），然后发送3字节地址（A23-A0），或
- 先发送EDh（CR2V[0] = 1），然后发送4字节地址（A31-A0），或
- 先发送EEh，然后发送4字节地址（A31-A0）

先发送地址，然后发送模式位。在DDR模式下，存储器通过IO0 - IO3信号每次在一个时钟沿上发送特定地址中的四比特数据。

DDR四线I/O读取命令的最大工作时钟频率为66 MHz。

执行DDR四线I/O读命令时，通过IO0-IO3移入最后地址和模式位后到数据从IO0-IO3开始移出前，需要添加一个延迟周期。这样，器件的内部电路会有足够的时间来访问初始地址的数据。在这些延迟周期内，IO0-IO3上的数据值均被忽略，并且这些信号会保持为高阻抗状态。使能数据学习模式（DLP）时，主机不能在虚拟周期内驱动IO信号，而必须将它保持为高阻抗状态，以便存储器能够驱动DLP。

虚拟周期的数量由SCK的频率确定。该延迟在CR3V[3:0]中进行配置。

通过模式位，一系列I/O DDR命令可以在第一条命令发送一个互补模式位图案后消除8位指令。由于具备该特性，因此不需要8位SDR指令，并且能够大量减少最初的访问时间（提高XIP性能）。模式位通过添加或清除第一个字节指令代码来控制下一个DDR四线I/O读取操作的长度。如果模式位的高位半字节（IO[7:4]）和低位半字节（IO[3:0]）是互补的（即5h和Ah），那么器件会切换为DDR四线I/O连续读取模式，并且能够在CS#被驱动为高电平再被确认为低电平后读取下一个地址（不需要读取EDh或EEh指令），从而可以清除命令序列的8个周期。发生以下情况时会使器件退出DDR四线I/O连续读取模式；然后器件可以接收标准的SPI命令：

1. 在执行DDR四线I/O读取命令序列过程中，如果模式位不是互补的，则下次CS#被驱动为高电平然后再被确认为低电平时会使器件退出DDR四线I/O读取模式。
2. 发送模式复位命令。

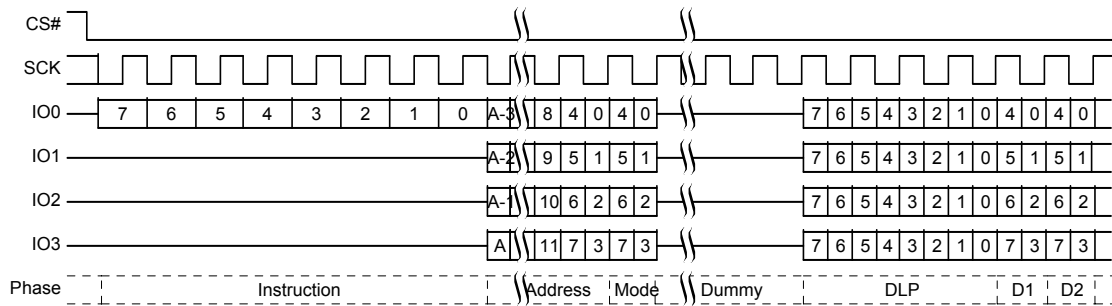
该地址可以从存储器阵列中的任意字节位置开始。输出每一个数据字节后，会按顺序自动递增到下一个更高的地址。因此，只要使用一个读取指令和给定的地址000000h，便可以读取整个存储器的内容。当到达最高地址时，地址计数器翻转到起始地址000000h，这样允许读取序列继续下去。

在模式/虚拟位周期内不应该将CS#驱动为高电平，否则会使这些模式位的状态不确定。请注意，存储器在发送第一个数据值前先在IO信号上发送一个前导码。前导码是一个数据学习模式（DLP），主机控制器通过它使数据捕获在更高的频率下得到优化。数据被输出前，前导码会立即驱动IO总线上四个时钟周期。请确保主机在存储器开始发送前导码前已经停止驱动IO总线。

前导码用于向主机控制器表示从主机驱动一个时钟沿到存储器返回相应数据的完整时间。主机控制器会在前导码期间使数据捕获点偏移，以便优化时序容限，然后使用同样的偏移时间捕获读取操作其余部分的数据。在所有读取操作的前导码期间，都可确定得到优化的数据捕获点。该优化特性不但能补偿存储器和主机控制器的PVT（处理过程、电压、温度），还可以补偿PCB板上数据传送时间所导致的系统级延迟。

虽然数据学习模式（DLP）是可编程的，但以下显示的是DLP为34h的示例。DLP 34h（或00110100）是在每个活跃的输出（即所有四个IO）上被发送的。选中该模式以包含“DC”和“AC”的数据切换情况。两个DC切换情况包括：数据在较长时间（2.5个时钟周期）内为低电平，然后切换为高电平（001）；互补低电平切换（110）。两个AC切换情况包括：数据在较短时间（1.5个时钟周期）内为低电平，然后切换为高电平（101）；互补低电平切换（010）。与AC切换相比，直流切换通常在更接近于电源轨的起始点上发生。在该起始点上，AC切换的电源未达到稳定状态。在多种情况下，直流切换决定了数据有效期间的开始，交流切换则决定了数据有效期间的结束。通过这些切换，主机控制器可以确定数据有效眼图的开始和结束。一旦确认了数据眼图，便可以选中最佳的数据捕获点。在QPI模式下（CR2V[3] = 1），DDR四线I/O指令都会在每个SCK的上升沿上发送4比特的信息。该命令协议的其余部分和DDR四线I/O命令完全相同。

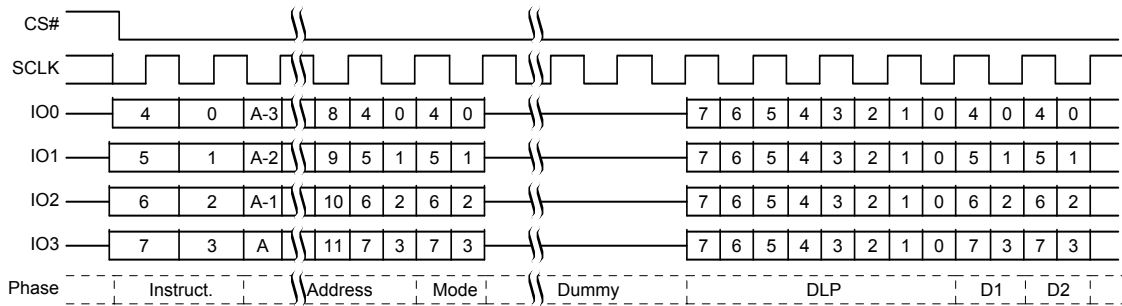
图8.48 DDR初始访问的四线I/O读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令EEh)。
2. DLP为34h (或00110100) 的示例

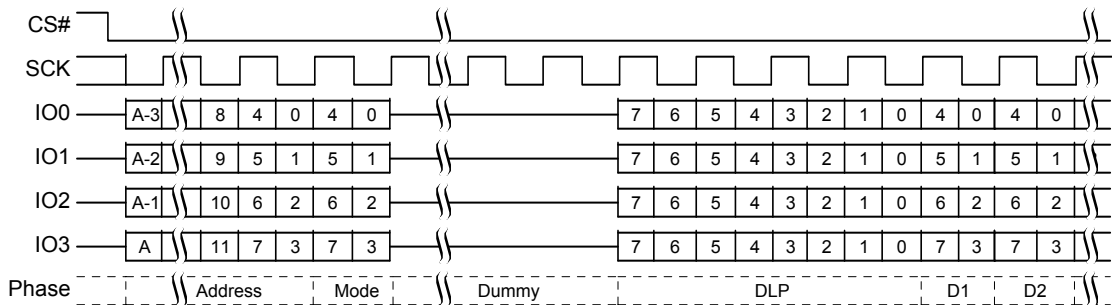
图8.49 在QPI模式下初始访问的DDR四路I/O读取命令序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令EEh)。
2. DLP为34h (或00110100) 的示例。

图8.50 后续访问的DDR四线I/O连续读取序列



注意:

1. A = 地址的MSB = 23 (若CR2V[0] = 0), 或 = 31 (若CR2V[0] = 1或命令EEh)。
2. 在QPI模式下, 命令序列也相同。
3. DLP为34h (或00110100) 的示例。

8.5 编程闪存阵列命令

8.5.1 编程粒度

8.5.1.1 页编程

通过将需要编程的数据加载到页缓冲区，并发送编程命令将缓冲区中的数据传送到存储器阵列内，这样可以实现页编程操作。该操作设置了通过单个编程命令可以编程的数据量的上限。每次最多可编程全页（256字节）。该页在页大小地址边界上对齐。每次进行页编程操作，可以编程一位到一页的空间。要想获得最佳性能，应在256字节边界上一次性完成整页256字节对齐的编程操作，并且每一页仅编程一次。

8.5.1.2 单字节编程

单字节编程通过允许在存储器阵列中的任何位置内编程单个字节，因此该操作与传统标准SPI页编程（PP）命令完全向后兼容。

8.5.2 页编程（PP 02h或4PP 12H）

通过页编程（PP）命令，可在存储器中进行字节编程（将各个位值从1改成0）。接收页编程（PP）命令前，器件必须发送写入使能（WREN）命令，并对其进行解码。成功解码写入使能（WREN）命令后，器件将设置状态寄存器中的写入使能锁存（WEL）位，以使能任意写入操作。

指令

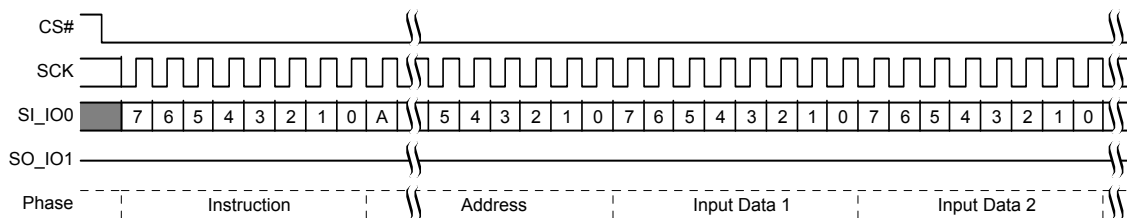
- 先发送02h（CR2V[0] = 0），然后发送3字节地址（A23-A0），或
- 先发送02h（CR2V[0] = 1），然后发送4字节地址（A31-A0），或
- 先发送12h，然后发送4字节地址（A31-A0）

并且SI/IO0信号上至少有一个数据字节。通过传送指令02h，然后传送3字节地址或通过传送指令12h，然后传送4字节地址后，可在SI/IO0发送多达一页的空间。对于写入和擦除命令，锁存最后字节的第8位后，必须将CS#引脚驱动为高电平。否则，不能执行页编程命令。将CS#信号驱动为高电平后，可在t_{pp}时间内开始执行自定时间的页编程命令。

同将部分页加载到编程缓冲区内相比，通过页编程（PP）命令加载页边界内整页的内容，可缩短个编程时间。

编程过程由闪存存储器器件的内部控制逻辑管理。发送编程命令后，可使用读取状态寄存器1命令检查编程操作状态。WIP位（SR1V[0]）指出了编程操作是否完成。P_ERR位（SR2V[5]）指出该编程操作是否存在错误（如果发生该错误，编程操作无法成功完成）。它包括保护区的尝试编程。

图8.51 页编程（PP 02h或4PP 12H）命令序列

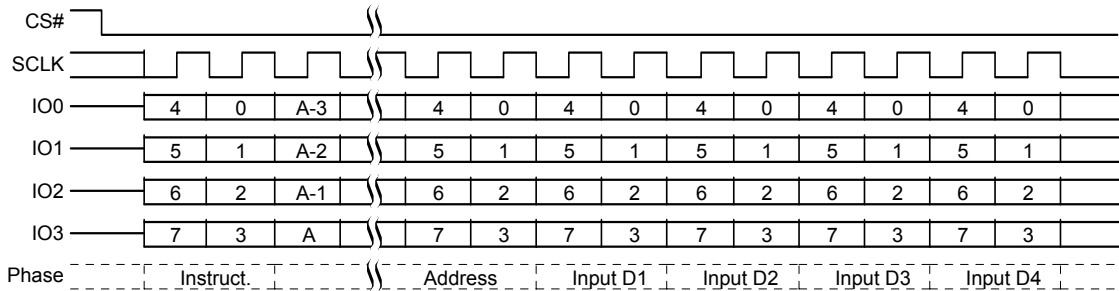


注意：

1. A = 地址的MSB = A23（在PP 02h和CR2V[0] = 0条件下），或 = A31（在PP 02h和CR2V[0] = 1，或4PP 12h条件下）。

QPI模式也支持该命令。在该模式下，将通过IO0-IO3输入指令、地址和数据。

图8.52 QPI模式下的页编程（PP 02h或4PP 12H）命令序列



注意:

1. A = 地址的MSB = A23 (在PP 02h和CR2V[0] = 0条件下), 或 = A31 (在PP 02h和CR2V[0] = 1, 或4PP 12h条件下)。

8.5.3 四线页编程（QPP 32h或4QPP 34h）

通过四路输入页编程（QPP）命令，可在存储器中进行字节编程（将位值从1改为0）。四线输入页编程（QPP）命令允许通过使用四个信号（IO0-IO3）将一个数据页加载到页缓冲区内。通过在每个时钟周期内加载4位数据，QPP可以提高具有更慢时钟速度（< 12 MHz）的PROM编程器和各个应用的性能。因为内在的页编程时间大于时钟数据所需的时间，所以具有更高时钟频率的系统体现不了QPP命令所带来的好处。QPP命令的最大频率为133 MHz。

要想使用四线页编程命令，必须设置配置寄存器中的四线使能位（QUAD=1）。接受QPP命令之前，器件需要执行写入使能命令（状态寄存器1的WEL = 1）。

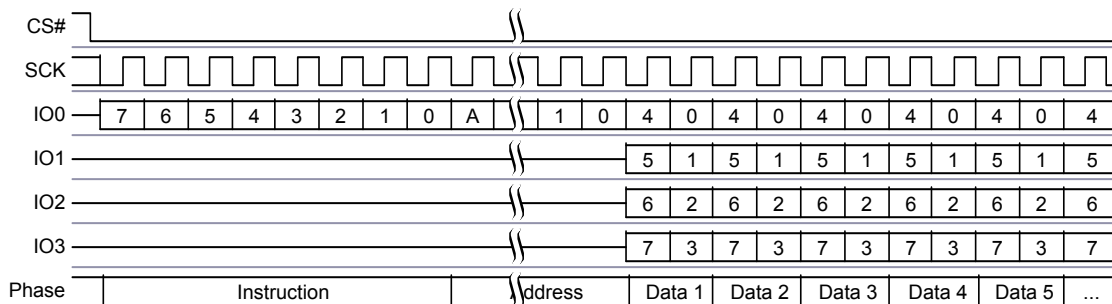
指令

- 先发送32h（CR2V[0] = 0），然后发送3字节地址（A23-A0），或
- 先发送32h（CR2V[0] = 1），然后发送4字节地址（A31-A0），或
- 先发送34h，然后发送4字节地址（A31-A0）

并且在IO信号上，至少有一个数据字节。必须在先前被擦除的（FFh）存储器位置中编程数据。

QPP的所有其他功能与页编程的相同。QPP命令序列如下面图所示。

图8.53 四线页编程命令序列



注意:

1. A = 地址的MSB = A23 (在QPP 32h和CR2V[0] = 0条件下) 或 = A31 (在QPP 32h和CR2V[0] = 1, 或4QPP 34h条件下)

8.6 擦除闪存阵列命令

8.6.1 扇区擦除 (SE 20h或4SE 21h)

扇区擦除 (SE) 命令将地址扇区中的所有位设置为1 (这样所有字节均为FFh)。接收扇区擦除 (SE) 命令前, 器件必须发送写入使能 (WREN) 命令并对其进行解码。这样将设置状态寄存器中的写入使能锁存 (WEL) 位, 以使能所有写入操作。

指令

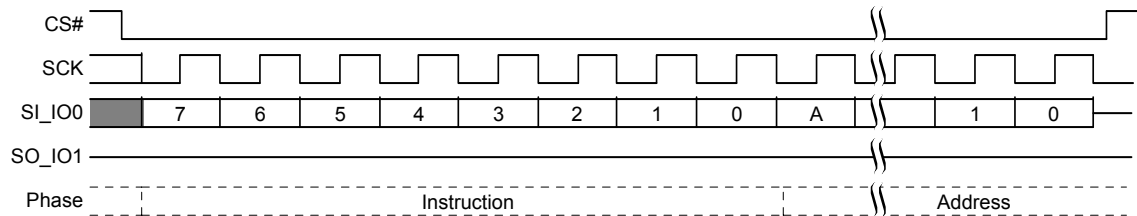
- 先发送20h[CR2V[0] = 0], 然后发送3字节地址 (A23-A0), 或
- 先发送20h[CR2V[0] = 1], 然后发送4字节地址 (A31-A0), 或
- 先发送21h, 然后发送4字节地址 (A31-A0)

地址的第24或第32位通过SI/IO0被锁存在芯片内后, 必须将CS#信号驱动为逻辑高电平状态。该操作将启动内部擦除周期, 包括预编程和擦除闪存存储器阵列的已选扇区。如果锁存地址的最后一位之后未将CS#驱动为高电平, 则不会执行扇区擦除操作。

将CS#信号驱动为高电平后, 将启动内部擦除周期。在内部擦除周期中, 用户可以读取正在进行写入操作 (WIP) 位的值, 从而确定该操作是否完成。在擦除周期期间, WIP位将为1; 在擦除周期完成后, 该位显示为0。

用于一个扇区的SE或4SE命令不被执行, 而用于设置E_ERR状态 (该扇区通过传统的块保护、单独块锁定或指针区域保护得到写入保护)。

图8.54 扇区擦除 (SE 20h或4SE 21h) 命令序列

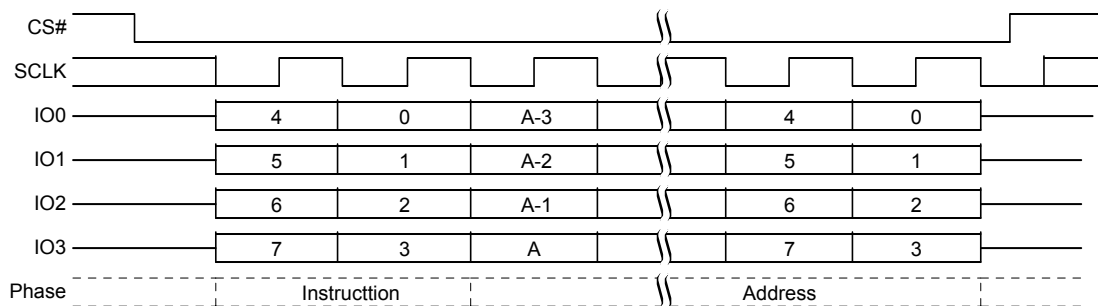


注意:

1. A = 地址的MSB = A23 (在SE 20h和CR2V[0] = 0条件下), 或 = A31 (在SE 20h和CR2V[0] = 1或4SE 21h条件下)。

QPI模式也支持该命令。在该模式下, 指令和地址将通过IO0-IO3信号传入到芯片内。

图8.55 扇区擦除 (SE 20h或4SE 21h) 的QPI模式命令序列



注意:

1. A = 地址的MSB = A23 (在SE 20h和CR2V[0] = 0条件下), 或 = A31 (在SE 20h和CR2V[0] = 1或4SE 21h条件下)。

8.6.2 半块擦除 (HBE 52h或4HBE 53h)

半块擦除 (HBE) 命令将地址半块中的所有位设置为1 (这样所有字节均为FFh)。接收半块擦除 (HBE) 命令前, 器件必须发送写入使能 (WREN) 命令并对其进行解码。这样将设置状态寄存器中的写入使能锁存 (WEL) 位, 以使能所有写入操作。

指令

- 先发送52h[CR2V[0] = 0], 然后发送3字节地址 (A23-A0), 或
- 先发送52h[CR2V[0] = 1], 然后发送4字节地址 (A31-A0), 或
- 先发送53h, 然后发送4字节地址 (A31-A0)

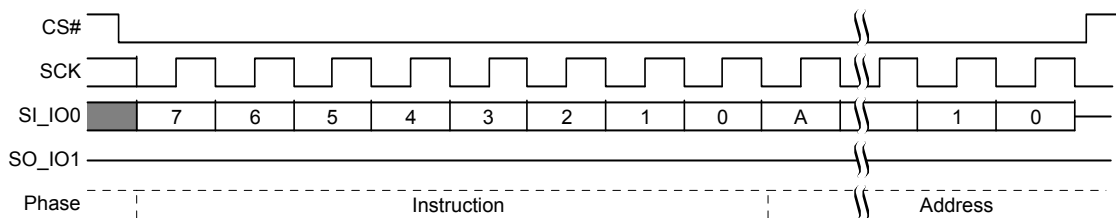
地址的第24或第32位被锁存到SI/IO0上之后, 必须将CS#信号驱动为逻辑高电平状态。该操作将启动擦除周期, 包括预编程和擦除已选模块的每个扇区。如果锁存地址的最后位后未将CS#驱动为高电平, 则半块擦除操作将不被执行。

将CS#信号驱动为逻辑高电平状态后, 将启动内部擦除周期。在内部擦除周期中, 用户可以读取正在写入 (WIP) 位的值, 从而检查该操作是否完成。在擦除周期期间, WIP位将为1; 在擦除周期完成后, 该位显示为0。

半块擦除 (HBE) 命令不被执行于整块 (该数据块通过传统的块保护、单独块锁定或指针区域保护得到写入保护), 而用于设置E_ERR状态。

如果使用了半块擦除命令并且半块擦除区中的任意区域、扇区或块得到保护, 那么将不会在32 KB范围内执行擦除操作, 但会设置E_ERR状态。

图8.56 半块擦除 (HBE 52h或4HBE 53h) 命令序列

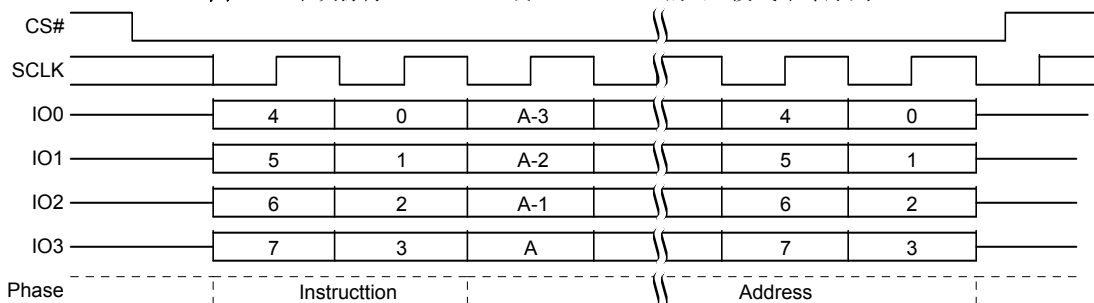


注意:

1. A = 地址的MSB = A23 (在HBE 52h和CR2V[0] = 0条件下), 或 = A31 (在HBE 52h和CR2V[0] = 1或4HBE 53h条件下)。
2. A[15] = 0时, 将擦除块的第0-7扇区; A[15] = 1时, 则擦除块的第8-15扇区。

QPI模式也支持该命令。在该模式下, 指令和地址将通过IO0-IO3信号传入到芯片内。

图8.57 半块擦除 (HBE 52h或4HBE 53h) 的QPI模式命令序列



注意:

1. A = 地址的MSB = A23 (在HBE 52h和CR2V[0] = 0条件下), 或 = A31 (在HBE 52h和CR2V[0] = 1或4HBE 53h条件下)。
2. A[15] = 0时, 将擦除块的第0-7扇区; A[15] = 1时, 则擦除块的第8-15扇区。

8.6.3 块擦除 (BE D8h或4BE DCh)

块擦除 (BE) 命令将地址块中的所有位设置为1 (所有字节均为FFh)。接收块擦除 (BE) 命令前, 器件必须发送写入使能 (WREN) 命令并对其进行解码。这样将设置状态寄存器中的写入使能锁存 (WEL) 位, 以使能所有写入操作。

指令

- 先发送D8h[CR2V[0] = 0], 然后发送3字节地址 (A23-A0), 或
- 先发送D8h[CR2V[0] = 1], 然后发送4字节地址 (A31-A0), 或
- 先发送DCh, 然后发送4字节地址 (A31-A0)

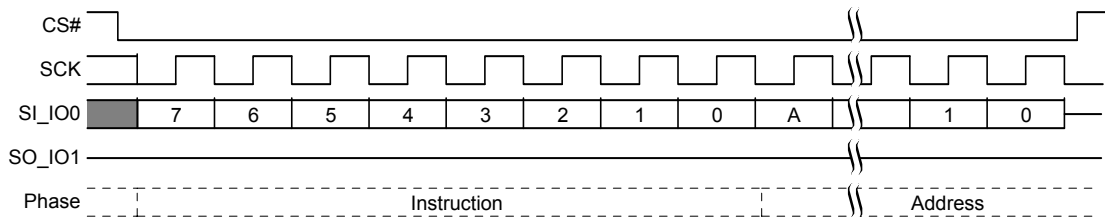
地址的第24或第32位通过SI/IO0锁存在芯片内后, 必须将CS#信号驱动为逻辑高电平状态。该操作将启动擦除周期, 包括预编程和擦除已选模块的每个扇区。如果锁存地址的最后位后未将CS#驱动为高电平, 将不执行块擦除操作。

将CS#信号驱动为逻辑高电平状态后, 将启动内部擦除周期。在内部擦除周期中, 用户可以读取正在写入 (WIP) 位的值, 从而检查该操作是否完成。在擦除周期期间, WIP位将为1; 在擦除周期完成后, 该位显示为0。

应用于整块的块擦除 (BE) 命令不被执行 (该模块通过传统的模块保护、单独模块锁定或指针区域保护获得写保护), 而且该命令将设置E_ERR状态。

如果使用块擦除命令, 但任意区域或扇区得到保护, 那么将不在64 KB范围内进行擦除操作, 并会设置E_ERR状态。

图8.58 块擦除 (BE D8h或4BE DCh) 命令序列

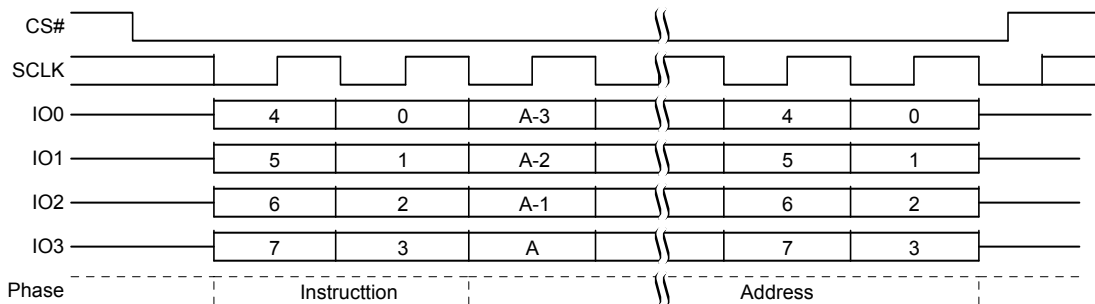


注意:

1. A = 地址的MSB = A23 (在BE D8h和CR2V[0] = 0条件下), 或 = A31 (在BE D8h和CR2V[0] = 1, 或4BE DCh条件下)。

QPI模式也支持该命令。在该模式下, 指令、地址和数据将通过IO0-IO3信号传入到芯片内。

图8.59 块擦除 (BE D8h或4BE DCh) 的QPI模式命令序列



注意:

1. A = 地址的MSB = A23 (在BE D8h和CR2V[0] = 0条件下), 或 = A31 (在BE D8h和CR2V[0] = 1, 或4BE DCh条件下)。

8.6.4 芯片擦除（CE 60h或C7h）

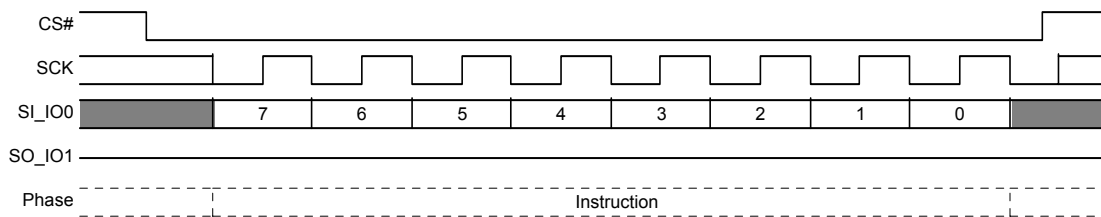
在整个闪存寄存器阵列中，芯片擦除（CE）命令将所有位设置为1（所有字节均为FFh）。接受CE命令之前，器件必须发送写入使能（WREN）命令并对其进行解码。该器件也设置状态寄存器中的写入使能锁存（WEL）命令，以使能所有操作。

指令字节的第8位被锁存到SI/IO0上之后，必须将CS#信号驱动为逻辑高电平状态。该操作将导致擦除周期发生，该周期包含对整个闪存存储器阵列的预编程和擦除。如果锁存指令的最后位之后未将CS#驱动为高电平，则不会执行CE操作。

一旦将CS#信号驱动为逻辑高电平状态，就发生擦除周期。在擦除周期过程中，用户可以读取正在写入（WIP）位的值，从而确定该操作是否完成。在擦除周期期间，WIP位将为1；在擦除周期完成后，该位显示为0。

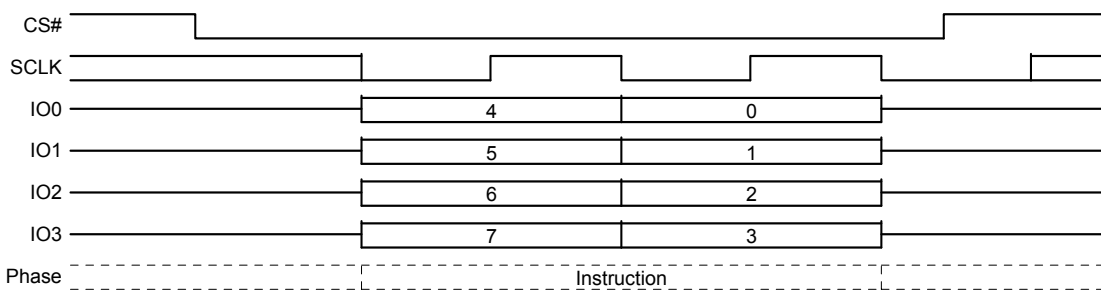
当设置传统块保护、单独块锁定或指针区域保护来保护任何扇区或块时，CE命令将不被执行，但会设置E_ERR状态位。

图8.60 芯片擦除命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.61 QPI模式下的芯片擦除命令序列



8.6.5 编程或擦除挂起（PES 75h）

通过PES命令，系统可中断编程或擦除操作，然后对任何其他非擦除挂起扇区或非编程挂起页进行读取。仅在编程或扇区擦除、半块擦除或块擦除操作过程中，编程或擦除挂起有效。但不能挂起芯片擦除操作。

必须检查状态寄存器1中的正在写入（WIP）位（SR1V[0]），以了解编程或擦除操作停止的时间。可以使用状态寄存器2（SR2[0]）中的编程挂起状态位来指出编程操作在WIP位改成0时被挂起还是完成。同样，可以使用状态寄存器2（SR2[1]）中的擦除挂起状态位来确定擦除操作在WIP位改成‘0’时已被挂起还是完成。挂起操作完成所需的时间为 t_{SL} ，请参见第139页上的表12.7. 编程或擦除挂起交流参数。

可挂起擦除操作以执行编程或读取操作。在擦除挂起过程中，可通过对IBL阵列的读取来检查扇区的保护，并可通过对其的写入来对被编程的扇区进行删除或恢复保护。恢复操作时，保护位不被重新检查，所以所进行的任何修改将不影响操作过程中的当前值。

可挂起编程操作来执行读取操作。

在已挂起擦除或编程操作中不允许执行新的挂起操作。在此情况下，挂起命令将被忽略。

表 8.6 在编程或擦除挂起期间可执行的命令

指令名称	指令代码 (十六进制)	在擦除挂起 期间可用	在编程挂起 期间可用	注释
READ	03	X	X	在挂起过程中可执行所有阵列读取操作
RDSR1	05	X	X	读取WIP以确定挂起过程是否结束的方法
RDAR	65	X	X	读取WIP以确定挂起过程是否结束的其他方法
RDSR2	07	X	X	用于读取挂起状态，以确定操作仍被挂起还是已经完成。
RDCR1	35	X	X	用于读取配置寄存器1
RDCR2	15	X	X	用于读取配置寄存器2
RDCR3	33	X	X	用于读取配置寄存器3
RUID	4B	X	X	用于读取唯一ID
RDID	9F	X	X	用于读取器件ID
RDQID	AF	X	X	用于四线读取器件ID
RSFDP	5A	X	X	用于读取SFDP
SBL	77	X	X	用于设置连续长度
WREN	06	X	X	用于执行擦除挂起期间所发出的编程命令
WRDI	04	X	X	用于执行擦除挂起期间所发出的编程命令
PP	02	X		用于在擦除挂起期间执行阵列编程操作。仅在无编程挂起操作时有效 (SR2V[0]=0)。存在某个挂起编程操作时，编程命令均被忽略。如果将编程命令发送到擦除挂起扇区中某个位置，则编程操作失败，并P_ERR位被置为1。
4PP	12	X		用于在擦除挂起期间执行阵列编程操作。仅在无编程挂起操作时有效 (SR2V[0]=0)。存在某个挂起编程操作时，编程命令均被忽略。如果将编程命令发送到擦除挂起扇区中某个位置，则编程操作失败，并P_ERR位被置为1。
QPP	32	X		用于在擦除挂起期间执行阵列编程操作。仅在无编程挂起操作时有效 (SR2V[0]=0)。存在某个挂起编程操作时，编程命令均被忽略。如果将编程命令发送到擦除挂起扇区中某个位置，则编程操作失败，并P_ERR位被置为1。
4QPP	34	X		用于在擦除挂起期间执行阵列编程操作。仅在无编程挂起操作时有效 (SR2V[0]=0)。存在某个挂起编程操作时，编程命令均被忽略。如果将编程命令发送到擦除挂起扇区中某个位置，则编程操作失败，并P_ERR位被置为1。
4READ	13	X	X	在挂起过程中可执行所有阵列读取操作
CLSR	30	X	X	擦除挂起期间，如果编程操作失败，则允许使用擦除状态寄存器命令。
EPR	7A	X	X	用于从擦除或编程挂起恢复。
RSTEN	66	X	X	复位可随时发生
RST	99	X	X	复位可随时发生
FAST_READ	0B	X	X	在挂起过程中可执行所有阵列读取操作
4FAST_READ	0C	X	X	在挂起过程中可执行所有阵列读取操作
DOR	3B	X	X	在挂起过程中可执行所有阵列读取操作
4DOR	3C	X	X	在挂起过程中可执行所有阵列读取操作
DIOR	BB	X	X	在挂起过程中可执行所有阵列读取操作
4DIOR	BC	X	X	在挂起过程中可执行所有阵列读取操作
IBLRD	3D	X	X	擦除挂起期间，可能需要移除和恢复单独块锁定，以便执行编程操作。
4IBLRD	E0	X	X	擦除挂起期间，可能需要移除和恢复单独块锁定，以便执行编程操作。
IBL	36	X	X	擦除挂起期间，可能需要恢复单独块锁定，以便执行编程操作。
4IBL	E1	X	X	擦除挂起期间，可能需要恢复单独块锁定，以便执行编程操作。
IBUL	39	X	X	擦除挂起期间，可能需要移除单独块锁定，以便执行编程操作。
4IBUL	E2	X	X	擦除挂起期间，可能需要移除单独块锁定，以便执行编程操作。
QOR	6B	X	X	四线读取输出 (3或4字节地址) (1)
4QOR	6C	X	X	四线读取输出 (4字节地址) (1)
QIOR	EB	X	X	在挂起过程中可执行所有阵列读取操作 (1)
4QIOR	EC	X	X	在挂起过程中可执行所有阵列读取操作 (1)

表 8.6 在编程或擦除挂起期间可执行的命令（续）

指令名称	指令代码 (十六进制)	在擦除挂起 期间可用	在编程挂起 期间可用	注释
DDRQIOR	ED	X	X	在挂起过程中可执行所有阵列读取操作 (1)
DDR4QIOR	ED	X	X	在挂起过程中可执行所有阵列读取操作 (1)
MBR	FF	X	X	在挂起过程中可用于复位读取操作
SECRP	42	X		在擦除挂起过程中可执行所有安全区域编程操作
SECRR	48	X	X	在挂起过程中可执行所有安全区域读取操作

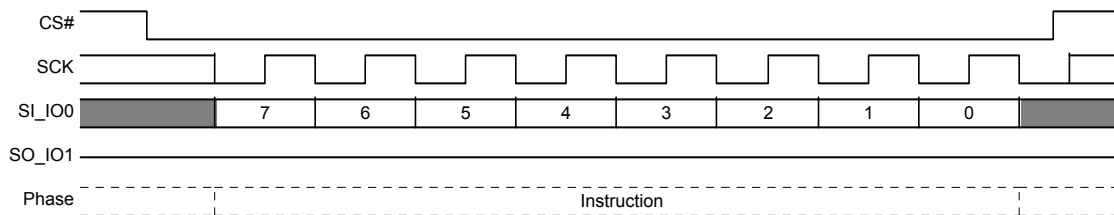
注意:

1. 对于所有四线命令，由于挂起状态中不允许执行WRR/WRAR命令，所以初始化编程或擦除操作之前，需要将四线使能CR1V[1]位（请参见第33页上的表6.10）设置为1。在擦除或编程挂起过程中，不允许执行第95页上的表8.6. 在编程或擦除挂起期间可执行的命令中不包含的所有命令。在擦除或编程挂起过程中，不允许执行WRR、WRAR或SPRP命令，因此，在擦除挂起时不能改变传统块保护位或指针区域保护位。

读取擦除挂起扇区或编程挂起页中的任何地址都会返回不确定的数据。

擦除挂起模式下的编程操作完成后，器件返回擦除挂起模式。系统可以通过读取状态寄存器的WIP位来确定编程操作的状态，同在执行标准编程操作一样。

图8.62 编程或擦除挂起命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

图8.63 QPI模式下编程或擦除挂起的命令序列

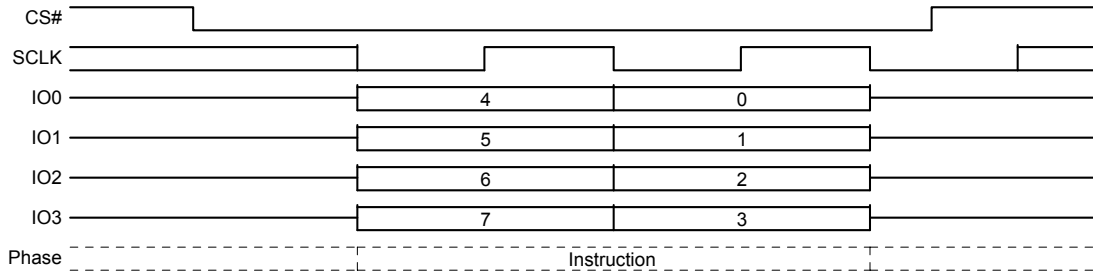
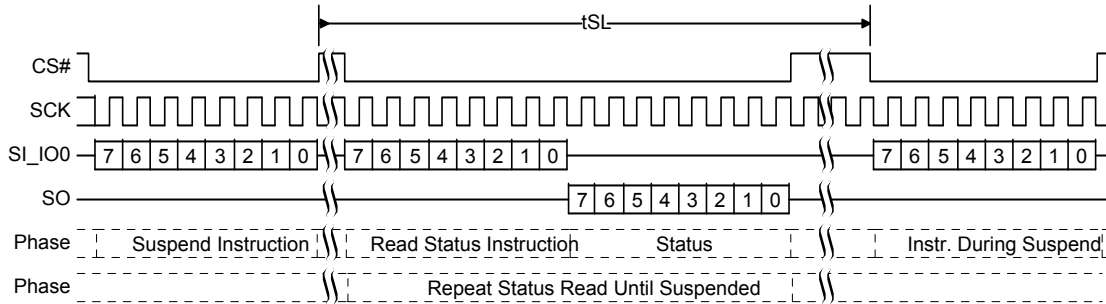


图8.64 带有连续命令序列的编程或擦除挂起命令



8.6.6 擦除或编程恢复 (EPR 7Ah)

编程或读取操作完成后，在编程或擦除挂起过程中，器件将发送擦除或编程恢复命令，以便继续执行被挂起的操作。

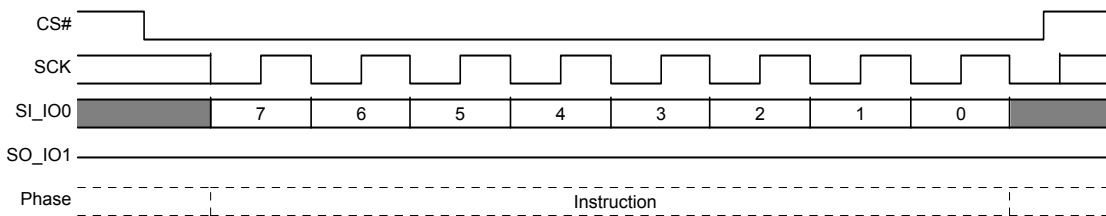
发送擦除或编程恢复命令后，状态寄存器中的WIP位将被设置为1，并且恢复被挂起的操作。如果没有任何被挂起的编程或擦除操作，将忽略恢复命令。

编程和擦除操作可以根据需要被中断。例如，执行编程挂起命令后可立即执行编程恢复命令。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间不能小于 t_{RNS} 。请参见第139页上的表12.7. *编程或擦除挂起交流参数*。

可以使用状态寄存器2 (SR2[0]) 中的编程挂起状态位来确定编程操作已被挂起还是已在WIP位改成0时完成。同样，可以使用状态寄存器2 (SR2[1]) 中的擦除挂起状态位来确定擦除操作被挂起还是已在WIP位改成0时完成。请参见第31页上的6.6.2 *易失性状态寄存器2 (SR2V)* 一节。

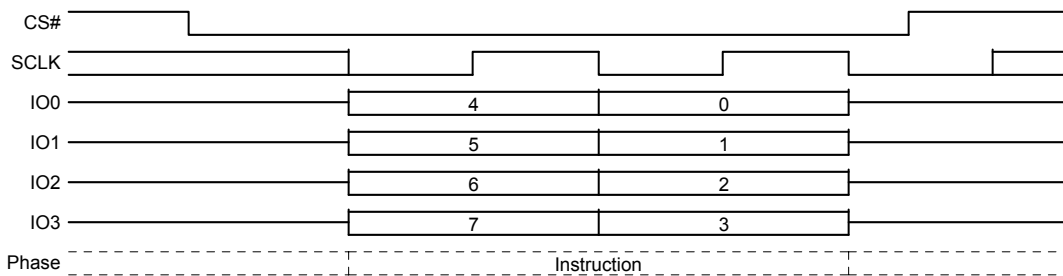
器件必须写入擦除或编程恢复命令来恢复被挂起的操作。

图8.65 擦除或编程恢复的命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号可将指令传入到器件内。

图8.66 QPI模式下擦除或编程恢复的命令序列



8.7 安全区域阵列命令

通过安全区域命令可以选择地址A15到A8将使用的区域，具体如下所示。

- 安全区域0: A23-16 = 00h; A15-8 = 00h; A7-0 = 字节地址
- 安全区域1: A23-16 = 00h; A15-8 = 01h; A7-0 = 字节地址
- 安全区域2: A23-16 = 00h; A15-8 = 02h; A7-0 = 字节地址
- 安全区域3: A23-16 = 00h; A15-8 = 03h; A7-0 = 字节地址

8.7.1 安全区域擦除 (SECRE 44h)

通过使用安全区域擦除命令来擦除安全区域中的数据，该安全区域和主阵列数据的地址空间不一样。安全区域大小为1024个字节，因此，使用该命令时需要将地址位A23到A10 (S25FL128L) 以及A24到A10 (S25FL256L) 全部设置为0。可以单独擦除每个区域。欲了解安全区域的详细信息，请参考第26页上的6.5 安全区域地址空间一节。

接收安全区域擦除命令前，器件必须发送写入使能 (WREN) 命令，并对其进行解码。这样状态寄存器中的写入使能锁存 (WEL) 位得到设置，从而使能所有写入操作。通过检查SR1V中的WIP位，可以确定该操作是否完成。同样，通过检查SR2V中的E_ERR位，可以确定在操作过程中是否发生了任何错误。

通过使用配置寄存器1中的安全区域锁定位 (CR1NV[2-5])，可以防止擦除安全区域。一旦锁定位被设置为1，将永久锁定相应的安全区域。尝试擦除被锁定的区域时，会引起结果失败，而且SR2V[6]中的E_ERR位被设置为1。

当保护寄存器的NVLOCK位为0时，将不能对安全区域2和3进行编程或擦除操作。尝试擦除被锁定的区域时，将导致结果失败，而且SR2V[6]中的E_ERR位被设置为1。请参见第56页上的7.7.2.1 NVLOCK位 (PR[0]) 一节。

密码保护模式的锁定位 (IRP[2]) 防止对区域2和区域3进行擦除操作，直到提供一个正确的密码来擦除这些安全区域为止。尝试擦除用密码锁定的区域时，将导致失败结果，而且SR2V[6]中的E_ERR位被设置为1。请参见第57页上的安全区域读取密码保护。

安全区域擦除命令和扇区擦除命令使用相同的协议。欲了解命令序列，请参见第91页上的8.6.1 扇区擦除 (SE 20h或4SE 21h) 一节。支持QPI模式。

8.7.2 安全区域编程 (SECRP 42h)

通过使用安全区域编程命令来编程安全区域中的数据，该安全区域和主阵列数据的地址空间不一样。安全区域大小为1024个字节，因此，使用该命令时需要将地址位A23到A10 (S25FL128L) 以及A24到A10 (S25FL256L) 设置为0。欲了解安全区域的详细信息，请参考第26页上的6.5 安全区域地址空间一节。

接收安全区域编程命令前，器件必须发送写入使能 (WREN) 命令并对其进行解码。这样将设置状态寄存器中的写入使能锁存 (WEL) 位，以使能所有写入操作。通过检查SR1V中的WIP位可以确定该操作是否完成。同样，通过检查SR2V中的P_ERR位可以确定在操作过程中是否发生了任何错误。

为了以位粒度程度仔细编程安全区域阵列，可将数据字节中的剩下位设置为1。

如果存储器中的安全区域不被锁定，则可对它进行一次或多次编程。然而，要想获得最佳的数据完整性，建议将一个或多个16字节长度和字节的对齐组一起编程，并且在每一区域的擦除操作之间只编程一次。

通过使用配置寄存器1中的安全区域锁定位 (CR1NV[2-5])，可以防止对安全区域进行编程。一旦将锁定位设置为1，就永久性锁定相应的安全区域。尝试将0或1编程到被锁定 (受保护) 的区域时，将导致结果失败，而且SR2V[5]中的P_ERR位被设置为1。将1编程到未受保护的区域时，不会导致错误，也不会设置P_ERR位。(请参考第32页上的配置寄存器1，以了解详细的说明)。

当保护寄存器的NVLOCK位为0时，将不能对安全区域2和3进行编程或擦除操作。尝试编程被锁定的区域时，将导致失败结果，而且SR2V[5]中的P_ERR位被设置为1。请参见第56页上的7.7.2.1 NVLOCK位 (PR[0]) 一节。

密码保护模式的锁定位 (IRP[2]) 防止对区域2和区域3进行编程操作，直到提供一个正确的密码来编程安全区域为止。尝试编程用密码锁定的区域时，将导致失败结果，而且SR2V[5]中的P_ERR位被设置为1。请参见第56页上的密码保护模式。

安全区域编程命令和页编程命令使用相同的协议。欲了解指令序列，请参见第89页上的8.5.1.1 页编程一节。支持QPI模式。

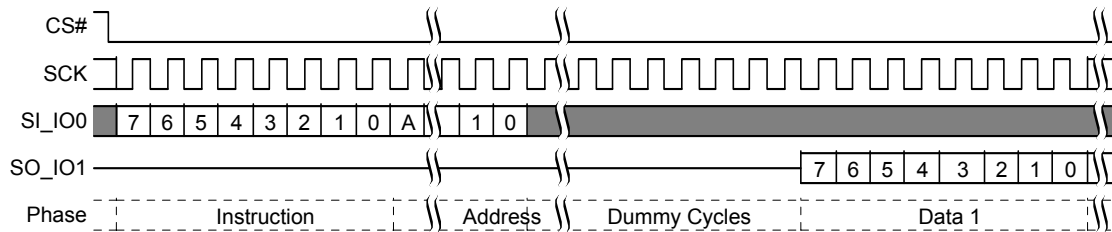
8.7.3 安全区域读取 (SECRR 48h)

安全区域读取 (SECRR) 命令提供了从安全区域读取数据的方法。安全区域大小为1024个字节，因此，使用该命令时需要将地址位A23到A10 (S25FL128L) 以及A24到A10 (S25FL256L) 设置为0。欲了解安全区域的详细信息，请参考第26页上的6.5 安全区域地址空间一节。

该指令后面是3或4字节地址 (取决于地址长度配置CR2V[0])，最后是由CR3V[3:0]指定的延迟 (虚拟) 周期数量。这样将返回选定的寄存器数据。当读取完安全区域地址的最大值后，安全区域读取命令的协议将不会返回到起始地址；相反，超出最大地址的数据将是不确定的。安全区域读取命令的读取延迟由CR3V[3:0]决定的延迟值设置。

安全区域读取的密码模式的使能位 (IRP[6]) 防止对区域3进行读取操作，直到提供一个正确的密码来使能这些安全区域的读取操作为止。尝试读取密码被锁定的区域时，将返回无效和未定义的数据。请参见第57页上的安全区域读取密码保护。

图8.67 安全区域的读取命令序列

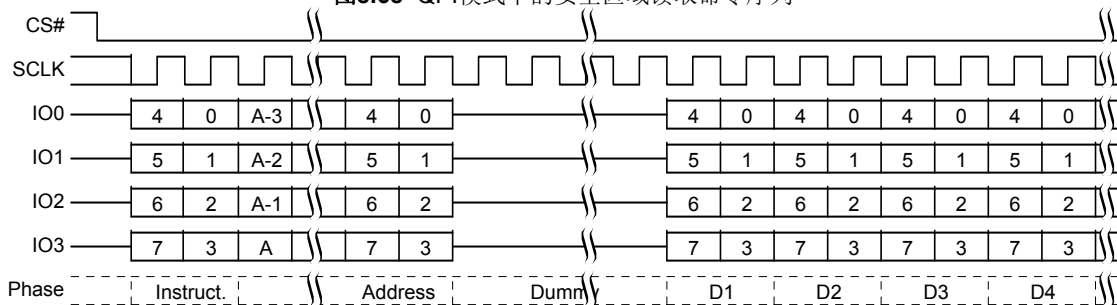


注意:

1. A = 地址的MSB = 23 (在地址长度 (CR2V[0] = 0条件下), 或 = 31 (在CR2V[0] = 1条件下)。

QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令和地址并传出返回数据。

图8.68 QPI模式下的安全区域读取命令序列



注意:

1. A = 地址的MSB = 23 (在CR2V[0] = 0条件下), 或 = 31 (在CR2V[0] = 1条件下)。

8.8 单块锁定命令

要想使用单块锁定命令，必须将配置寄存器2中的WPS位（CR2V[2]）置1，以选定IBL保护方案。如果不选择IBL保护方案（CR2V[2] = 0），则忽略IBL命令。

单块锁定位（IBL）均是易失性的，其中每一位用于一个扇区/块，并且可被单独修改。通过发送IBL或GBL命令，可以将IBL位设置为0，从而保护每个相关的扇区/块。通过发送IBUL或GUL命令，可以将IBL位清除为1，从而解除每个相关的扇区/块的保护。通过发送IBLRD命令，可以读取每个IBL位保护的状态。

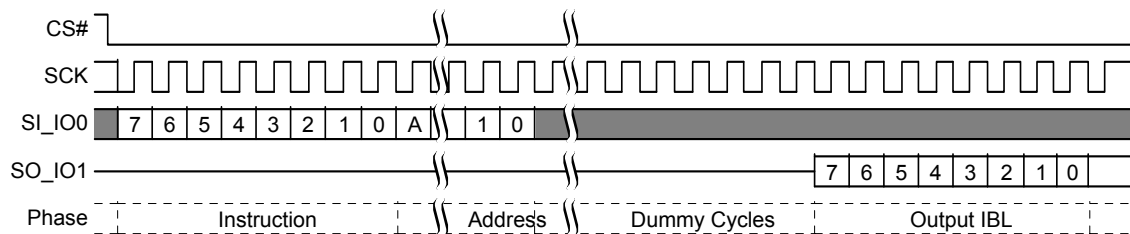
8.8.1 IBL读取（IBLRD 3Dh或4IBLRD E0h）

通过发送IBLRD/4IBLRD命令，可以读取每个IBL位保护的状态。

指令在SCK信号的上升沿上被锁存在SI中。先读取命令，然后锁存24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值），在所需扇区中选择位置0。

然后，8位IBL访问寄存器内容通过串行输出SO/IO1被移出。在SCK的下降沿上以SCK频率输出每一位数据。通过提供8个时钟周期的倍数，可以连续读取相同的IBL访问寄存器。IBL寄存器的地址不增加，所以这不是读取整个IBL阵列的方法。必须通过一个单独IBL读取命令读取每个位置。

图8.69 IRPRD命令序列

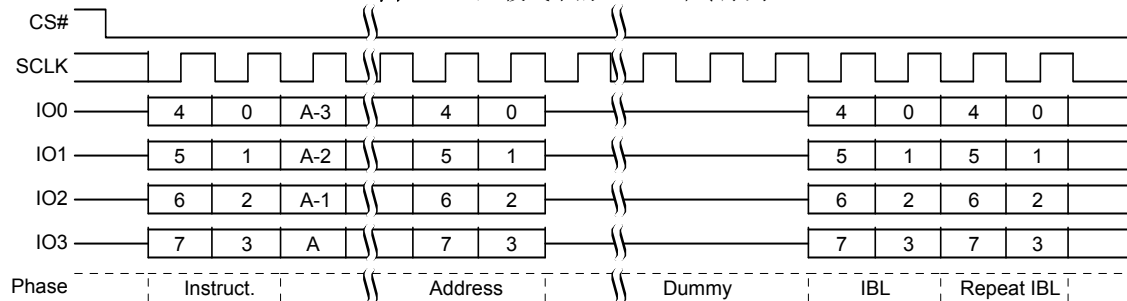


注意：

1. A = 地址的MSB = 23（在地址长度（CR2V[0] = 0条件下），或 = 31（在CR2V[0] = 1和命令3Dh条件下）。
2. A = 地址的MSB = 31（在命令E0h的条件下）。

QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令和地址并传出返回数据。

图8.70 QPI模式下的IRPRD命令序列



注意：

1. A = 地址的MSB = 23（地址长度CR2V[0] = 0），或 = 31（CR2V[0] = 1，命令3Dh）。
2. A = 地址的MSB = 31（在命令E0h的条件下）。

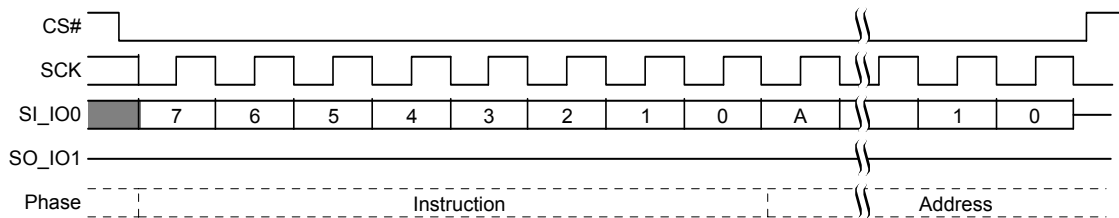
8.8.2 IBL 锁定 (IBL 36h或4IBL E1h)

IBL/4IBL命令将IBL位设置为0，以保护相关的每个扇区/块。

通过将CS#信号驱动为逻辑低电平状态，然后传送指令，最后传送24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值），可以输入IBL命令。与所有其他编程操作相同，IBL命令也会影响状态和配置寄存器的WIP位。

锁存24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值）后，必须将CS#信号驱动为逻辑高电平状态。CS#被驱动为逻辑高电平状态时，会立即启动自定时IBL操作。在执行IBL操作的过程中，可以读取状态寄存器以检查正在进行写入操作（WIP）位的值。执行自定时IBL操作期间，WIP位为1；完成该操作时，WIP位为0。

图8.71 IBL命令序列

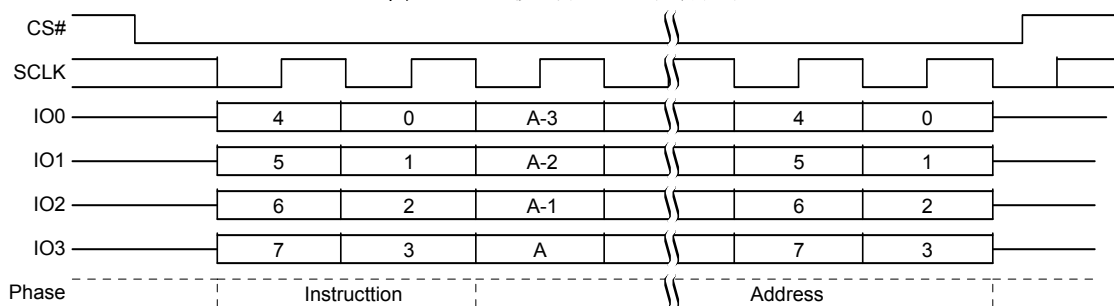


注意:

1. A = 地址的MSB = 23（在地址长度（CR2V[0] = 0条件下），或 = 31（在CR2V[0] = 1和命令36h条件下）。
2. A = 地址的MSB = 31（在命令E1h的条件下）。

QPI模式也支持该命令。在该模式下，指令、地址和数据通过IO0-IO3信号被传入到芯片内。

图8.72 QPI模式下的IBL命令序列



注意:

1. A = 地址的MSB = 23（在地址长度（CR2V[0] = 0条件下），或 = 31（在CR2V[0] = 1和命令36h条件下）。
2. A = 地址的MSB = 31（在命令E1h的条件下）。

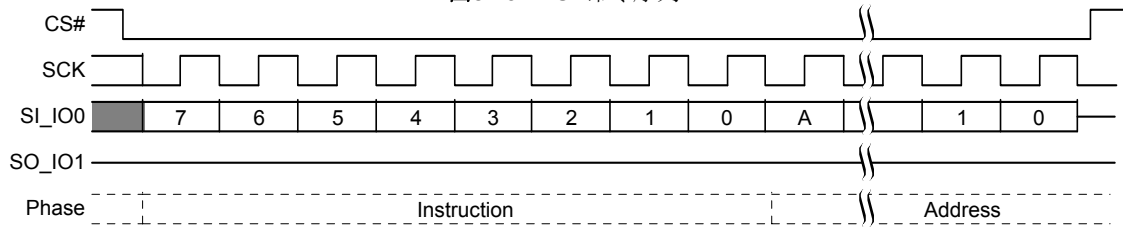
8.8.3 IBL 解锁 (IBUL 39h或4IBUL E2h)

IBUL/4IBUL命令将已选定的IBL位清除成1，以取消每个相关扇区/块的保护。

通过将CS#信号驱动为逻辑低电平状态，然后锁存指令，最后锁存24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值），可以输入IBUL命令。与所有其他编程操作相同，IBUL命令也会影响状态和配置寄存器的WIP位。

锁存24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值）后，必须将CS#信号驱动为逻辑高电平状态。将CS#驱动为逻辑高电平状态时，会立即启动自定时IBL操作。在执行IBUL操作的过程中，可以读取状态寄存器以检查正在进行写入操作（WIP）位的值。执行自定时IBUL操作期间，WIP位为1。完成该操作时，WIP位为0。

图8.73 IBUL命令序列

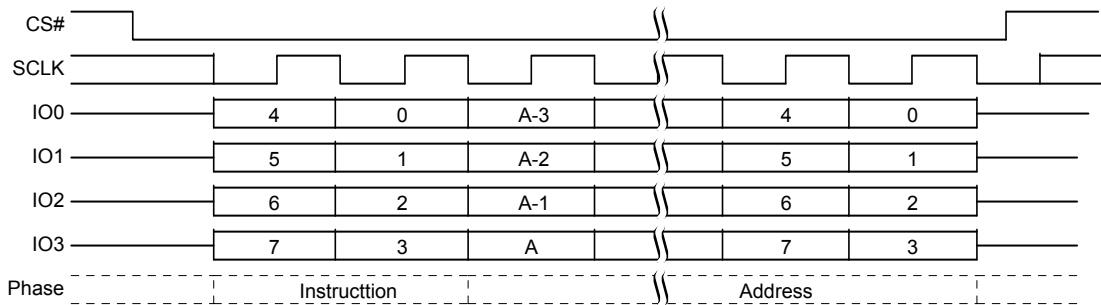


注意:

1. A = 地址的MSB = 23 (在地址长度 (CR2V[0] = 0条件下), 或 = 31 (在CR2V[0] = 1和命令39h条件下)。
2. A = 地址的MSB = 31 (在命令E2h的条件下)。

QPI模式也支持该命令。在该模式下, 指令、地址和数据将通过IO0-IO3信号传入到芯片内。

图8.74 QPI模式下的IBUL命令序列



注意:

1. A = 地址的MSB = 23 (在地址长度 (CR2V[0] = 0条件下), 或 = 31 (在CR2V[0] = 1和命令39h条件下)。
2. A = 地址的MSB = 31 (在命令E2h的条件下)。

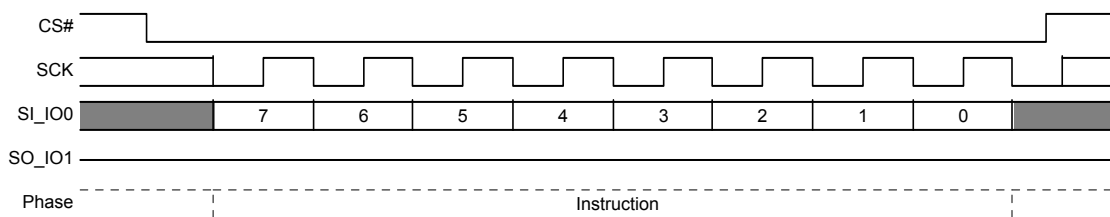
8.8.4 全局IBL锁定 (GBL 7Eh)

GBL命令将所有IBL位清除为0, 以保护所有扇区/块。

指令字节的第8位被锁存到SI上之后, 必须将CS#信号驱动为逻辑高电平状态。该操作将启动GBL命令。如果锁存指令的最后位之后未将CS#驱动为高电平, 则将不执行GBL操作。

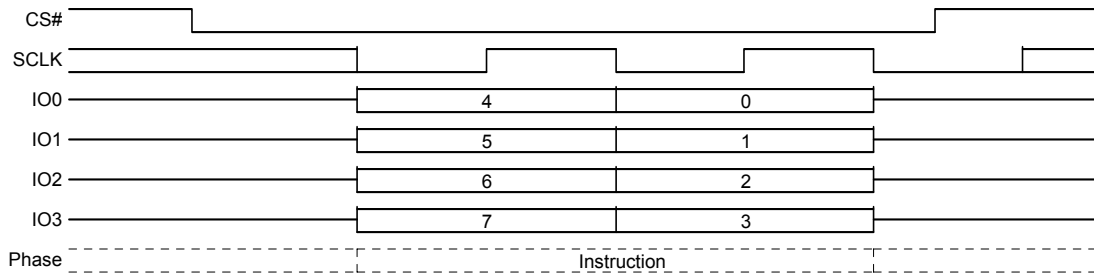
将CS#驱动为逻辑高电平状态时, 会立即启动GBL操作。在执行GBL操作过程中, 用户可以读取WIP位的值, 以确定该操作是否完成。在执行GBL操作期间, WIP位将显示为1; 该操作结束时, WIP位则显示为0。

图8.75 全局IBL锁定 (GBL) 命令序列



QPI模式也支持该命令。在该模式下, 通过IO0-IO3信号将指令传入到器件内。

图8.76 QPI模式下的全局IBL锁定（GBL）命令序列



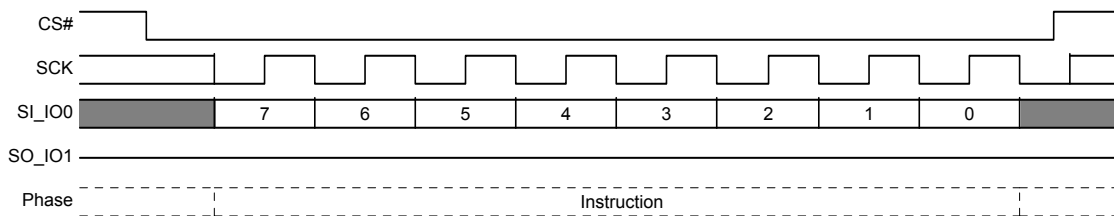
8.8.5 全局IBL解锁（GBUL 98h）

GBUL命令将所有IBL位清除为1，从而取消所有扇区/块的保护。

指令字节的第8位被锁存到SI上之后，必须将CS#信号驱动为逻辑高电平状态。该操作将启动GBUL命令。如果锁存指令最后位结束后未将CS#驱动为高电平，便不会执行GBL操作。

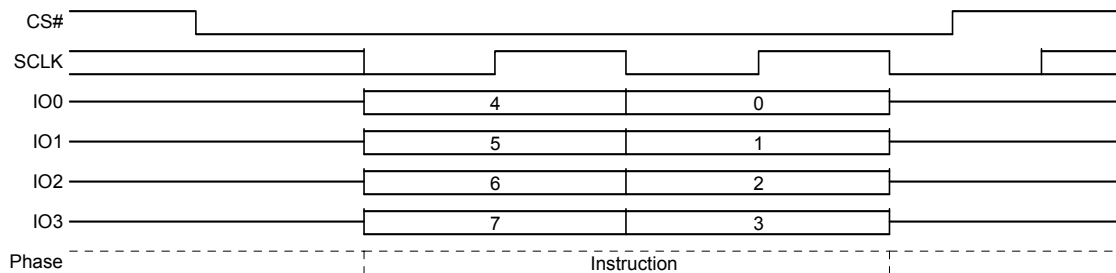
将CS#驱动为逻辑高电平状态时，会立即启动GBL操作。在执行GBL操作过程中，用户可以读取WIP位的值，以确定该操作是否完成。在执行GBUL操作期间，WIP位将显示为1；该操作结束时，WIP位则显示为0。

图8.77 全局IBL解锁（GBUL）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.78 QPI模式下的全局IBL解锁（GBUL）命令序列



8.9 指针区域命令

8.9.1 设置指针区域保护（SPRP FBh或4SPRP E3h）

在挂起状态期间，将忽略SPRP或4SPRP命令。这是因为在挂起状态下不可擦除或重新编程指针值。

如果默认电源的锁定保护NVLOCK PR[0] = 0或电源锁定保护使能IRP[1] = 0或密码保护使能IRP[2] = 0和NVLOCK PR[0] = 0，则忽略SPRP或4SPRP命令。

S25FL256L 器件必须使能4字节地址命令（CR2V[0] = 1），以设置指针区域保护寄存器PRPR（请参考第41页上的6.6.10节），这样才能确保A24和A25被正确设置。S25FL128L 器件能够激活4字节寻址命令（CR2V[0] = 1）或3字节寻址命令（CR2V[0] = 0）。

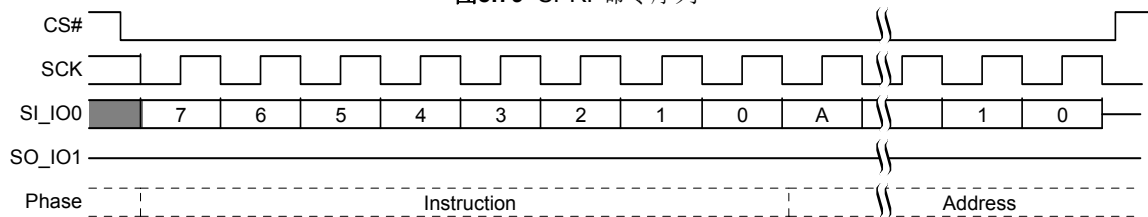
接收SPRP或4SPRP命令之前，器件必须发送写入使能（WREN）命令。成功解码写入使能（WREN）命令后，器件将设置状态寄存器中的写入使能锁存（WEL）位，以使能任意写入操作。

通过将CS#信号驱动为逻辑低电平状态，然后传送指令，最后再传送24位或32位地址（地址的长度取决于配置寄存器2中的CR2V[0]值），这样可以输入SPRP或4SPRP命令。请参见第52页上的指针区域保护（PRP）以了解地址值的详细信息，从而选择保护选项。

锁存地址最后位结束后，必须将CS#驱动为逻辑高电平状态。否则，不能执行SPRP命令。将CS#驱动为逻辑高电平状态时，会立即启动自定时SPRP操作。在执行SPRP操作的过程中，可以读取状态寄存器，以检查正在进行写入操作（WIP）位上的值。执行自定时SPRP操作期间，WIP位为1；完成该操作时，WIP位为0。完成SPRP操作时，写入使能锁存（WEL）位将被设置为0。如果设置指针区域保护操作失败，SPRP或4SPRP命令将设置P_ERR或E_ERR位。

欲了解用于定义存储器中受保护和不受保护区域间扇区边界的地址指针的详细信息，请参见第52页上的指针区域保护（PRP）。

图8.79 SPRP命令序列

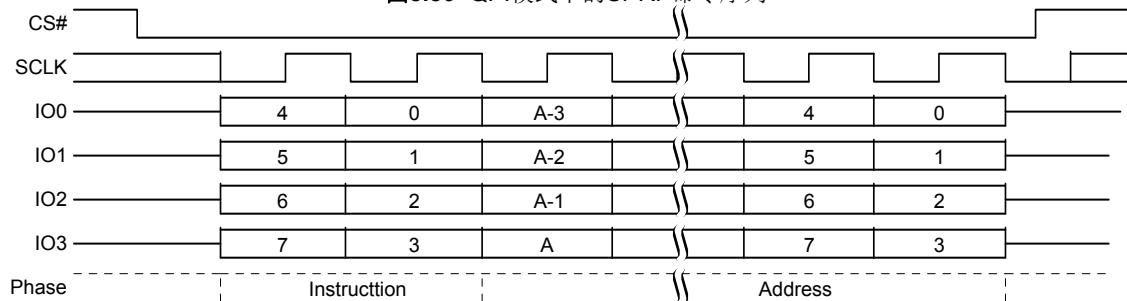


注意:

1. A = 地址的MSB = 23（在地址长度（CR2V[0] = 0条件下），或31（在CR2V[0] = 1和命令FDh条件下）。
2. A = 地址的MSB = 31（在命令E3h的条件下）。

QPI模式也支持该命令。在该模式下，将通过IO0-IO3输入指令和地址。

图8.80 QPI模式下的SPRP命令序列



注意:

1. A = 地址的MSB = 23（在地址长度（CR2V[0] = 0条件下），或31（在CR2V[0] = 1和命令FDh条件下）。
2. A = 地址的MSB = 31（在命令E3h的条件下）。

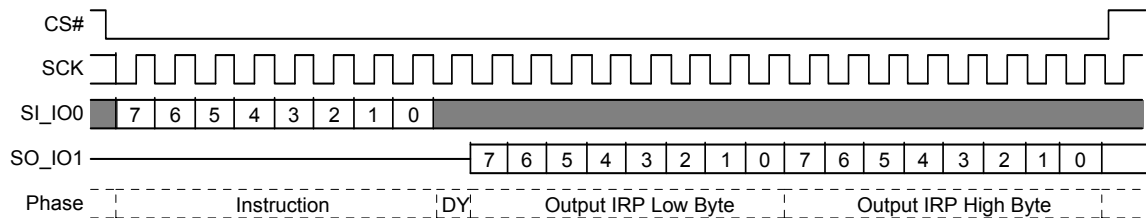
8.10 单独和区域保护（IRP）命令

8.10.1 IRP寄存器读取（IRPRD 2Bh）

在SCK信号的上升沿上，将IRP寄存器读取命令2Bh转移给SI/IO0，转移完成后存在一个虚拟周期。器件的内部电路需要该延迟周期以进行访问初始地址的数据。在这些延迟周期内，IO0-IO3上的数据值为“无需关注项”，该值可能为高阻抗。

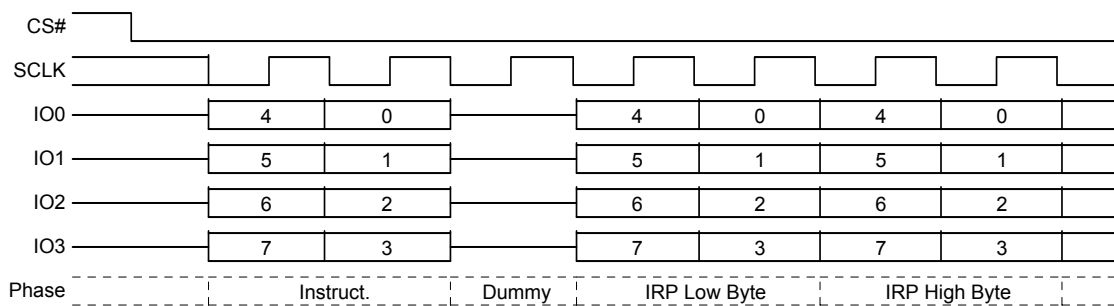
然后，在串行输出SO/IO1上移出16位IRP寄存器的内容，最低有效字节优先。在SCK的下降沿上以SCK频率输出每一位数据。通过提供16时钟周期的倍数，可以连续读取IRP寄存器。

图8.81 IRPRD命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令并传出返回的数据。

图8.82 IRPRD命令序列 — QPI模式



8.10.2 IRP编程（IRPP 2Fh）命令

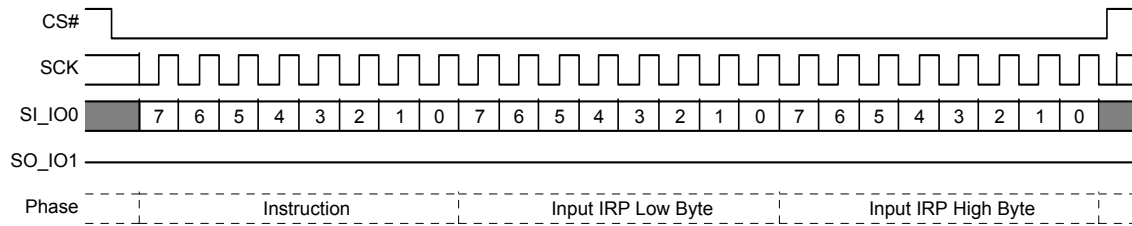
接受IRP编程（IRPP）命令前，器件必须发送写入使能（WREN）命令。对写入使能（WREN）命令进行解码后，器件将设置状态寄存器中的写入使能锁存（WEL）位以启用所有写入操作。

通过将CS#驱动为逻辑低电平状态，可以输入IRPP命令，然后输入SI上的指令和两个数据字节，最低有效字节优先。IRP寄存器的长度为两个数据字节。

与所有其他编程操作相同，IRPP命令也会影响状态和配置寄存器的P_ERR和WIP位。

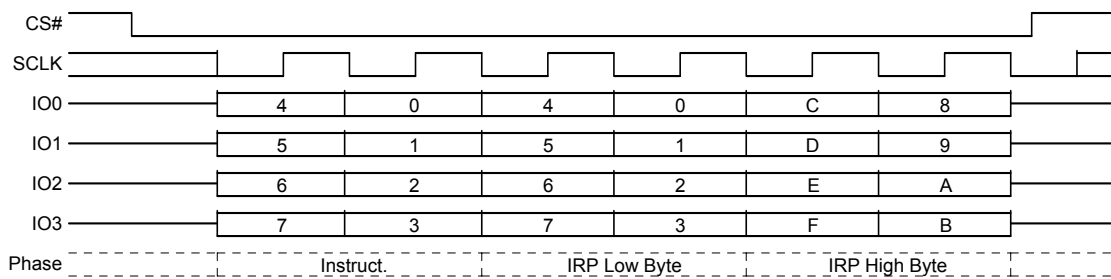
数据字节的第十六位被锁存后，必须将CS#驱动为逻辑高电平状态。否则，不能执行IRPP命令。CS#被驱动为逻辑高电平状态时，会立即启动自定时IRPP操作。在执行IRPP操作的过程中，可以读取状态寄存器以检查WIP位的值。执行自定时IRPP操作期间，WIP位为1；完成该操作时，WIP位为0。完成IRPP操作时，写入使能锁存（WEL）位将被设为0。

图8.83 IRP编程 (IRPP) 命令



该命令也被支持于QPI模式。在该模式下，通过IO0-IO3信号将指令和数据传入到器件内。

图8.84 IRP编程 (IRPP) 命令QPI



8.10.3 保护寄存器读取 (PRRD A7h) 命令

使用保护寄存器读取 (PRRD) 命令可以通过SO/IO1读取保护寄存器的内容。在SCK信号的上升沿上，将读取指令A7h传入到SI内，传入完成后存在一个虚拟周期。器件的内部电路需要该延迟周期以进行访问初始地址中的数据。在这些延迟周期内，IO0-IO3上的数值为“无需关注项”，该值可能为高阻抗。

然后，通过串行输出SO/IO1移出8位保护寄存器的内容。在SCK的下降沿上以SCK频率输出两位数据。通过提供8时钟周期的倍数，可以连续读取保护寄存器。

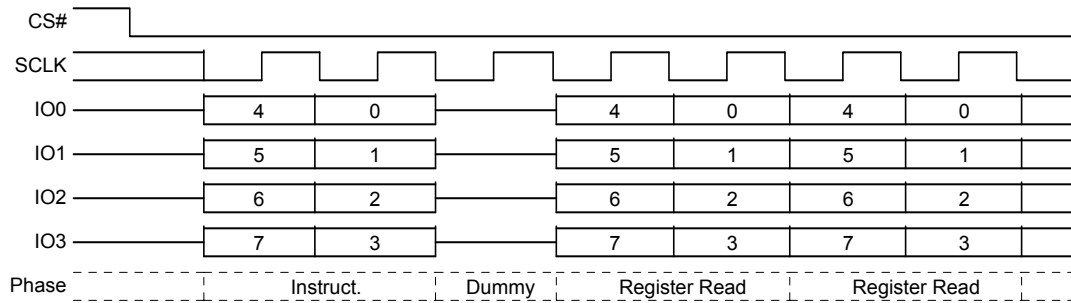
如果器件处于待机状态并不在执行任何其他操作，则仅能读取保护寄存器中的内容。

图8.85 保护寄存器读取 (PRRD) 命令序列



QPI模式也支持该命令。在QPI模式下，通过IO0-IO3信号将指令传入到芯片内，同时传出返回的数据。

图8.86 保护寄存器读取（PRRD）命令序列 — QPI模式



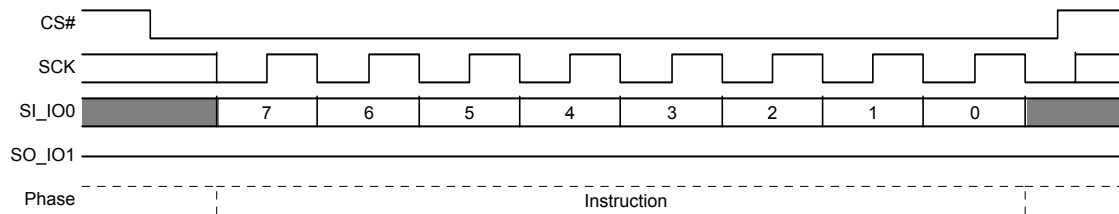
8.10.4 保护寄存器锁定（PRL A6h）命令

保护寄存器锁定（PRL）命令会将NVLOCK位（PR[0]）清除为0并将IRP[6]值载入到SECRRP（PR[6]）内。请参见第40页上的6.6.8 [保护寄存器（PR）](#) 一节。接收PRL命令前，器件必须发送写入使能（WREN）命令并对其进行解码。器件也设置状态寄存器中的写入使能锁存（WEL）位以使能所有写入操作。

通过将CS#驱动为逻辑低电平状态，可以输入PRL命令，然后输入指令。

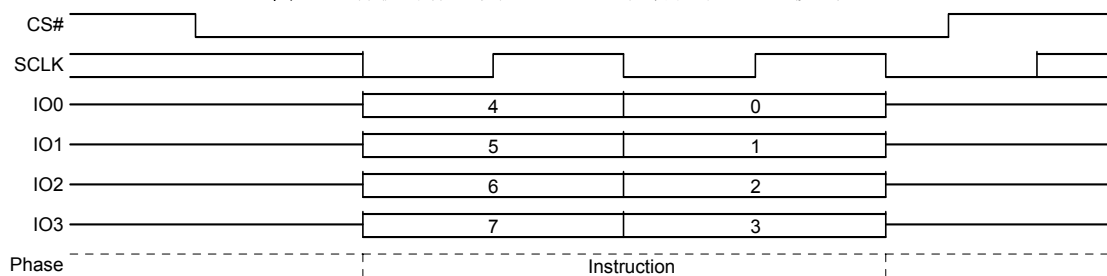
指令的第8位被锁存后，必须将CS#信号驱动为逻辑高电平状态。否则，不能执行PRL命令。CS#被驱动为逻辑高电平状态时，会立即启动自定时PRL操作。在执行PRL操作的过程中，还可以读取状态寄存器，以检查WIP位上的值。执行自定时PRL操作期间，WIP位为1；完成该操作时，WIP位为0。完成PRL操作时，写使能锁存（WEL）位将被设置为0。

图8.87 保护寄存器锁定（PRL）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.88 保护寄存器锁定（PRL）命令序列 — QPI模式

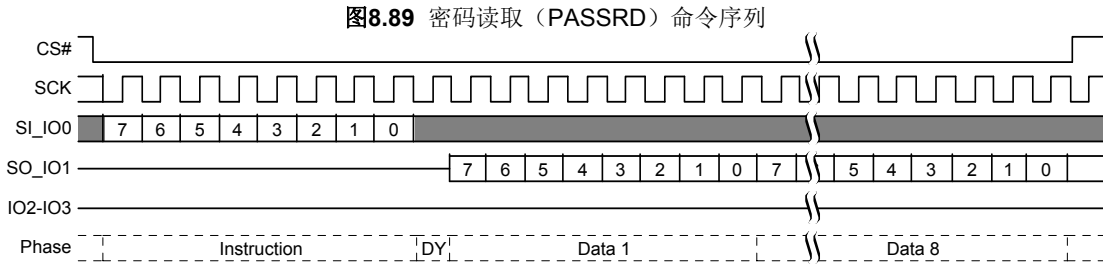


8.10.5 密码读取 (PASSRD E7h) 命令

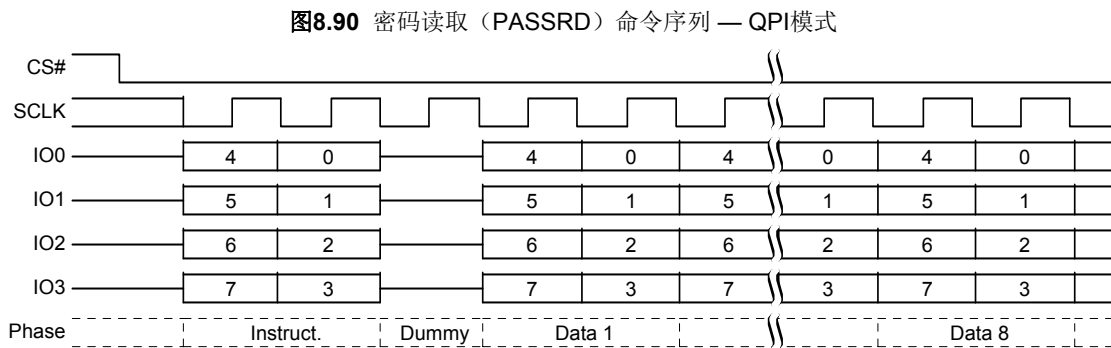
仅在编程正确密码值后且通过在IRP寄存器 (IRP[2]) 中将密码保护模式位编程为0来选定密码模式前, 才能读取正确密码值。选定密码保护模式后, 密码不再可读, 此时PASSRD命令将输出未定义的数据。

PASSRD命令被传入SI, 传入完成后存在一个虚拟周期。器件的内部电路需要该延迟周期用于进行访问初始地址的数据。在这些延迟周期内, IO0-IO3上的数据值为“无需关注”项, 该值可能为高阻状态。

然后, 在串行输出上移出64位密码, 并且最低有效字节优先, 每个字节的最高有效位优先。在SCK的下降沿上以SCK频率输出一位数据。通过提供64时钟周期的倍数, 可以连续读取密码。



在QPI模式下, 器件也支持该命令。在QPI模式下, 通过IO0-IO3信号将指令传入到芯片内, 同时传出返回的数据。



8.10.6 密码编程 (PASSP E8h)

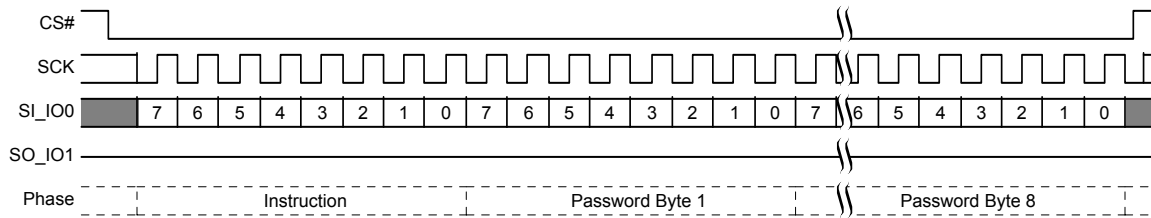
接受密码编程 (PASSP) 命令前, 器件必须发送写入使能 (WREN) 命令并对其进行解码。对写入使能 (WREN) 命令进行解码后, 器件将设置状态寄存器中的写入使能锁存 (WEL) 位以使能PASSP操作。

仅当通过在IRP寄存器 (IRP[2]) 中将密码保护模式位编程为‘0’来选定密码模式前, 才能编程密码。选定密码保护模式后, PASSP命令将被忽略。

通过将CS#驱动为逻辑低电平状态, 可以输入PASSP命令, 然后输入SI/IO0上的指令和密码数据字节, 最低有效字节优先, 每个字节的最高有效位优先。密码的长度为六十四 (64) 位。

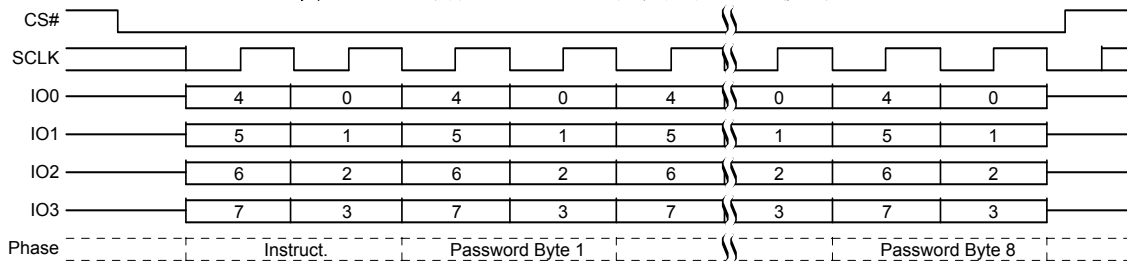
数据字节的第六十四 (64) 位被锁存后, 必须将CS#驱动为逻辑高电平状态。否则, 不能执行PASSP命令。CS#被驱动为逻辑高电平状态时, 会立即启动自定时PASSP操作。在执行PASSP操作的过程中, 可以读取状态寄存器以检查WIP位的值。执行自定时PASSP操作期间, WIP位为1; 完成该操作时, WIP位为0。PASSP命令可以报告状态寄存器中P_ERR位的编程错误。完成PASSP操作时, 写使能锁存 (WEL) 位将被设置为0。

图8.91 密码编程 (PASSP) 命令序列



QPI模式也支持该模式。在该模式下，通过IO0-IO3信号将指令和数据传入到芯片内。

图8.92 密码编程 (PASSP) 命令序列 — QPI模式



8.10.7 密码解锁 (PASSU EAh) 命令

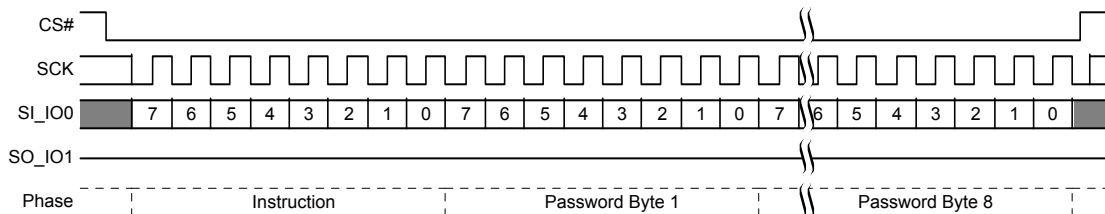
通过将CS#驱动为逻辑低电平状态，可以输入PASSU命令，然后输入SI上的指令和密码数据字节，最低有效字节优先，每个字节的最高有效位优先。密码的长度为六十四（64）位。

数据的第六十四（64）位被锁存后，必须将CS#驱动为逻辑高电平状态。否则，不能执行PASSU命令。CS#被驱动为逻辑高电平状态时，会立即启动自定时PASSU操作。在执行PASSU操作的过程中，可以读取状态寄存器以检查WIP位的值。执行自定时PASSU操作期间，WIP位为1；完成该操作时，WIP位为0。

如果PASSU命令提供的密码与密码寄存器中隐藏密码不匹配，将P_ERR位设为1便可以报告错误。状态寄存器的WIP位仍保持为1。另外需要使用CLSR命令来清除状态寄存器中的内容，使用软件复位命令（先发送RSTEN66h，然后是RST99h）来复位器件，或驱动RESET#和IO3 / RESET#以启动硬件复位，从而使P_ERR和WIP位返回0。这样会使器件进入待机状态，以准备好执行新命令（如重新执行PASSU命令）。

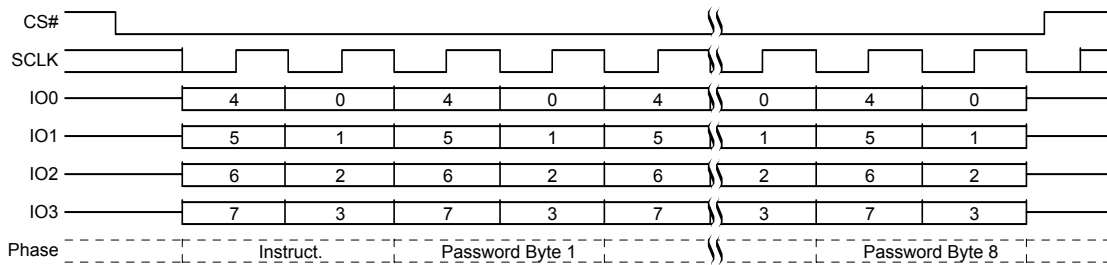
如果密码不匹配，NVLOCK位将被设置为1。

图8.93 密码解锁 (PASSU) 命令序列



在QPI模式下，器件也支持该命令。在该模式下，通过IO0-IO3信号将指令和数据传入到芯片内。

图8.94 密码解锁 (PASSU) 命令序列 — QPI模式



8.11 复位命令

通过将非易失性默认值重新载入易失性存储器内，软件控制的复位命令将器件恢复为其初始的加电状态。如果在执行对寄存器的擦除、编程或写入操作期间启动软件复位且扇区、页面或寄存器内的数据不稳定，那么需要重新启动被中断的操作。

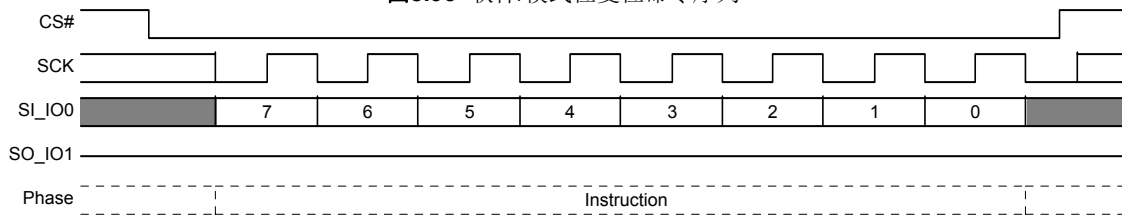
然而，一个软件复位不能修改配置寄存器CR1V[0]中的SRP1易失性位和保护寄存器中的NVLOCK易失性位。不能使用软件复位来将其其他安全配置位代替SRP1或NVLOCK位的保护机制。

进行软件复位前，SRP1位和NVLOCK位会保持其设置为最后的值。要想清除SRP1位并将NVLOCK位设置为其保护模式选定的上电状态，需要完成完整的上电复位序列或硬件复位。

如果在指令结束时CS#被置为高电平，将会执行软件复位命令（由RST 99h随后的RSTEN 66h），该过程需要 t_{RPH} 时间。

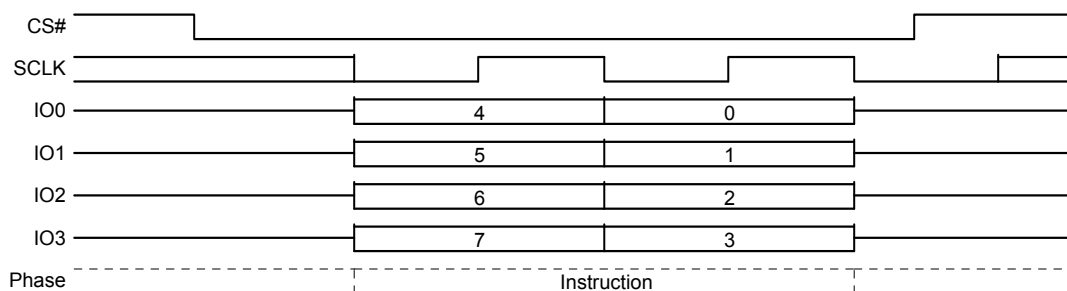
如果无法完成先前的加电复位（POR），那么复位命令会触发一个完整的加电序列，需要 t_{PU} 来完成该序列。

图8.95 软件/模式位复位命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.96 软件复位/模式位命令序列 — QPI模式



8.11.1 软件复位使能 (RSTEN 66h)

在软件复位命令 (RST 99h) 前立即需要复位使能 (RSTEN) 命令, 这样的软件复位是一个双命令的序列。随后RSTEN指令的任何命令 (RST除外) 将清除复位使能条件并防止识别随后的RST命令。

8.11.2 软件复位 (RST 99h) 命令

随后RSTEN命令的复位 (RST) 命令会启动软件复位程序。随后RSTEN指令的任何命令 (RST除外) 将清除复位使能条件并防止识别随后的RST命令。

8.11.3 模式位复位 (MBR FFh) 命令

模式位复位 (MBR) 命令用于使器件从高性能读取模式返回到正常待机模式, 此时器件会等待所有新命令。因为硬件RESET#输入可能被禁用, 并且处于高性能连续读取模式的器件可能不会识别出任何正常的SPI命令, 因此器件可能无法区分是系统硬件复位还是软件复位命令。在执行系统复位 (当RESET#信号不可用时) 后或在发送软件复位前, 建议使用MBR命令以确保器件已经退出高性能连续读取模式。

在八个SCK周期内, MBR命令将通过SI/IO0发送1。在这些周期期间, IO1-IO3为“无需关注内容”。

8.12 深度掉电命令

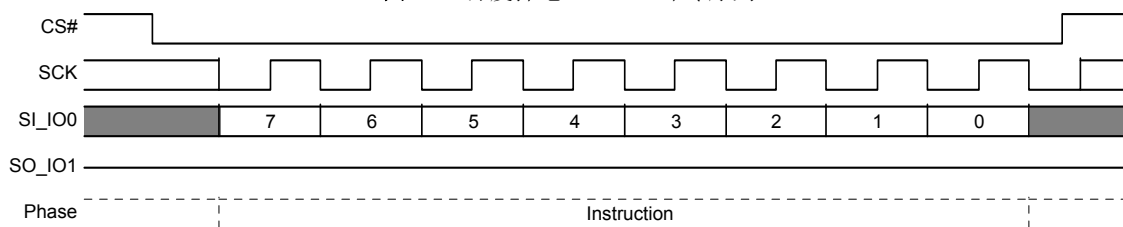
8.12.1 深度掉电 (DPD B9h) 命令

虽然在执行正常操作时待机电流相对低, 但使用深度掉电命令还可以更加降低待机电流。由于能够降低功耗, 深度掉电 (DPD) 命令对电池供电的应用非常有用 (请参考 (第127页上的11.6 直流特性一节) 中的 I_{CC1} 和 I_{CC2})。通过将CS#引脚驱动为低电平并发送指令代码“B9h”, 启动该命令。

锁存第八位后, 必须将CS#引脚驱动为高电平。否则, 不能执行深度掉电命令。CS#被驱动为高电平后, 在 t_{DP} (第134页上的表12.4) 期间, 器件将进入掉电状态。在掉电状态下, 器件只能识别出退出深度掉电模式/器件ID命令 (使器件恢复正常操作)。所有其他命令均被忽略, 包括读取状态寄存器命令, 在执行正常操作过程中, 该命令始终可用。这样, 在掉电模式下能够实现写保护的级别。

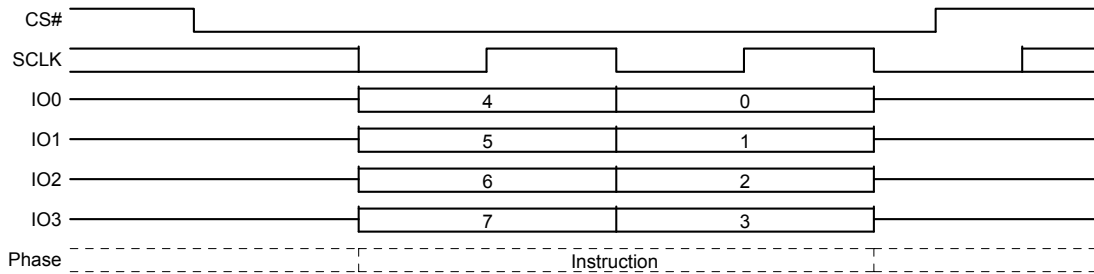
在深度掉电模式下, 器件只接受一个硬件复位, 该复位会启动上电复位 (用于使器件恢复正常操作)。在正常操作模式下, 器件始终被加电, 消耗了 I_{CC1} 大小的待机电流。

图8.97 深度掉电 (DPD) 命令序列



QPI模式也支持该命令。在该模式下, 通过IO0-IO3信号将指令传入到器件内。

图8.98 深度掉电（DPD）命令序列 — QPI模式



8.12.2 退出深度掉电模式/器件ID (RES ABh)

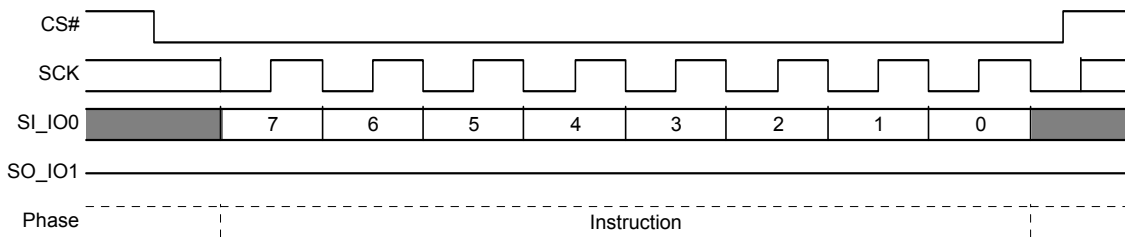
退出深度掉电模式/器件ID命令是一个多功能命令。该命令可用于使器件退出深度掉电状态或取得器件电子标识（ID）编号。

要想使器件退出深度掉电状态，首先要将CS#引脚驱动为低电平、发送指令代码“ABh”并将CS#信号驱动为高电平，然后发送该命令。器件恢复正常操作并接受其他命令前，执行退出深度掉电命令需要 t_{RES} （第134页上的表12.4）时长。在 t_{RES} 期间，CS#引脚必须保持为高电平状态。

如果不在深度掉电状态下使用该命令的唯一目的是为了获取器件ID，那么可以通过将CS#引脚驱动为低电平并发送指令代码“ABh”（3个虚拟字节随后）来启动该命令。然后，在CLK的下降沿上移出器件ID位，最高有效位（MSB）优先。S25FL-L系列的器件ID值将在第122页上的表10.5. 制造商器件类型中详细说明。所定义的ID地址结束后，持续移出的输出会提供未定义数据。通过将CS#驱动为高电平来完成该命令。

当用于使器件退出深度掉电状态和获取器件ID时，该命令与上面所述相同，并在图8.101和图8.102中详细显示，唯一不同的是，CS#被驱动为高电平后它必须保持高电平状态 t_{RES} 时间。经过这段时间后，器件将恢复正常操作，并接受其他命令。如果在执行擦除、编程或写周期过程（BUSY寄存器等于1时）中发送退出深度掉电模式/器件ID命令，该命令将被忽略，并且不会影响到当前周期。

图8.99 退出深度掉电模式（RES）命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号将指令传入到器件内。

图8.100 退出深度掉电模式（RES）命令序列 — QPI模式

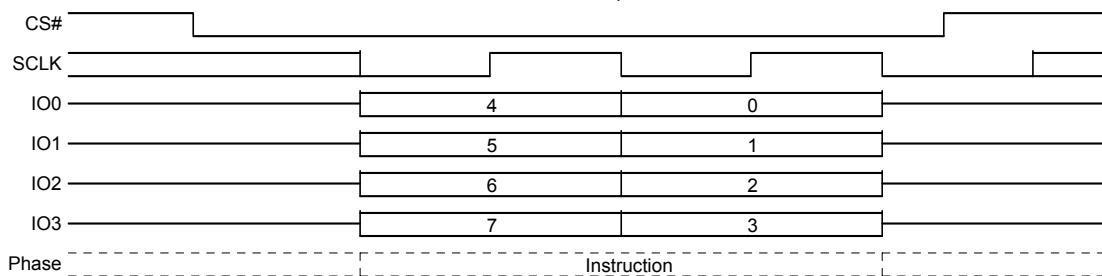
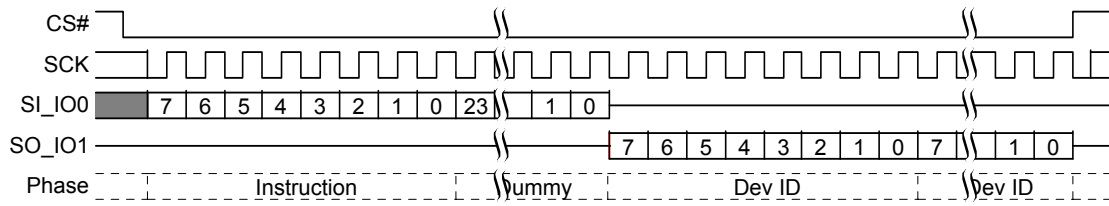
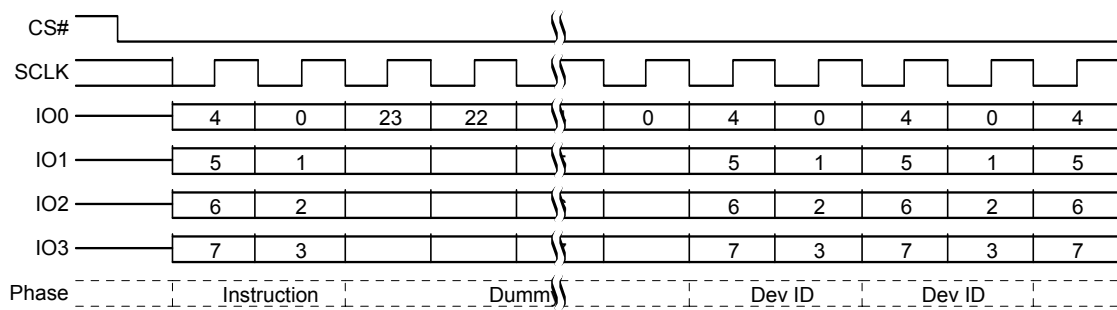


图8.101 读取标识 (RES) 命令序列



QPI模式也支持该命令。在该模式下，通过IO0-IO3信号传入指令并传出返回数据。

图8.102 QPI模式下的读取标识 (RES) 命令



9. 数据完整性

9.1 擦除次数

表 9.1 擦除次数

参数	最小值	单位
每个主闪存阵列扇区的编程/擦除次数	10万	编程/擦除次数
安全区域或非易失性寄存器阵列的编程/擦除次数 (1)	1千	编程/擦除次数

注意:

1. 每次将写命令发送到非易失性寄存器都会使整个非易失性寄存器阵列上的数据增加一个P/E周期。

9.2 数据保持时间

表 9.2 数据保持时间

参数	测试条件	最小时间	单位
数据保持时间	1万次编程/擦除	20	年
	10万次编程/擦除	2	年

更多有关数据完整性的详细信息，请联系赛普拉斯FAE或销售代表。

10. 软件接口参考

10.1 JEDEC JESD216B 串行闪存可发现参数

本文档定义了串行闪存可发现参数 (SFDP) 版本B的数据结构，在以下各赛普拉斯串行闪存器件中使用：

- S25FL-L 系列

这些数据结构值是旧版本SFDP数据结构（当前在上述器件存在）的更新值。

根据串行闪存可发现参数的JEDEC JESD216B标准，通过读取SFDP (RSFDP) 命令 (5Ah) 可以读取单独闪存存储器地址空间的内容，以获取器件标识、特性和配置信息。

SFDP数据结构包含一个头文件表，该表用于识别受支持的JESD216头文件格式的版本并提供一个版本编号以及每个SFDP参数表的指针。随后SFDP头文件是参数表。然而，可以将参数表放置在SFDP地址空间中任何物理位置和顺序。这些表格不需要相邻放置，也不需要按照其头文件表条目的顺序放置。

SFDP头文件指向下面各参数表：

- 基本闪存
 - 这是原始的SFDP表。在表格末端添加了一些已修改的字段和新的额外字段。
- 4字节地址指令
 - 这是原始的SFDP表。在表格末端添加了一些已修改的字段和新的额外字段。

SFDP地址空间中表格的物理顺序为：SFDP头文件、基本闪存扇区映射、4字节指令。

该SFDP地址空间由赛普拉斯编程，主机系统只能对其进行读取操作。

10.1.1 串行闪存可发现参数 (SFDP) 地址映射

SFDP地址空间具有起始地址为0的头文件，用于识别SFDP数据结构并向每个参数提供一个指针。一个基本闪存参数由JEDEC JESD216B标准规定。后面的基本闪存表是4字节地址指令的可选参数表。

表 10.1 SFDP 映射概述

字节地址	说明
0000h	JEDEC JESD216B SFDP空间中的地址零 — SFDP头文件的起点
...	SFDP头文件其余部分，后面是未定义空间
0300h	SFDP参数的开始
...	SFDP JEDEC参数其余部分，后面是未定义空间

10.1.2 SFDP头文件字段定义

表 10.2 SFDP 头文件

SFDP字节地址	SFDP Dword名称	数据	说明
00h	SFDP头文件 第一个DWORD	53H	读取SFDP (5Ah) 的入口点, 即为SFDP空间中的地址零 ASCII “S”
01h		46H	ASCII “F”
02h		44h	ASCII “D”
03h		50h	ASCII “P”
04h	SFDP头文件 第二个DWORD	06h	SFDP次要版本 (06h = JEDEC JESD216版本B) —该版本与以前所有次要版本向后兼容。SFDP读取和分析软件能够用于比设置所指定的次要版本编号更高的版本。针对较高版本设计的软件要能够处理较低版本的操作。例如: 针对次要版本0设计的SFDP读取和分析软件能够用于次要版本6, 而针对次要版本6设计的SFDP读取和分析软件则要确保能够读取次要版本0或5。因此, 用户不应针对次要版本编号进行简单的比较, 也不应该仅关注软件设计指定的版本编号。完全可以使用更高的次要版本。
05h		01h	SFDP主要版本。 这是原始的主要版本。该主要版本与所有SFDP读取和分析软件相兼容。
06h		01h	参数头文件的数量 (以零为基准, 01h = 2个参数)
07h		FFh	未使用
08h	参数头文件 0 第一个DWORD	00h	参数ID LSB (00h = JEDEC SFDP基本SPI闪存参数)
09h		06h	参数次要版本 (06h = JESD216版本B)
0Ah		01h	参数主要版本 (01h = 原始主要版本 — 所有SFDP软件均与该版本兼容)。
0Bh		10h	参数表长度 (单位为双字 = Dwords = 4字节) 10h = 16 Dwords
0Ch	参数头文件 0 第二个DWORD	00h	参数表指针字节0 (Dword = 4字节对齐) JEDEC基本SPI闪存参数字节偏移 = 0300h地址
0Dh		03h	参数表指针字节1
0Eh		00h	参数表指针字节2
0Fh		FFh	参数ID MSB (FFh = JEDEC定义的参数)
10h	参数头文件 1 第一个DWORD	84h	参数ID LSB (84h = SFDP 4字节地址指令参数)
11h		00h	参数次要版本 (00h = 初始版本, 如JESD216版本B的定义)
12h		01h	参数主要版本 (01h = 原始主要版本 — 识别该参数ID的所有SFDP软件均与该主要版本兼容)。
13h		02h	参数表长度 (单位为双字 = Dwords = 4字节) (2h = 2 Dwords)
14h	参数头文件 1 第二个DWORD	40h	参数表指针字节0 (Dword = 4字节对齐) JEDEC参数字节偏移 = 0340h
15h		03h	参数表指针字节1
16h		00h	参数表指针字节2
17h		FFh	参数ID MSB (FFh = JEDEC定义参数)

10.1.3 JEDEC SFDP基本SPI闪存参数

表 10.3 基本 SPI 闪存参数， JEDEC SFDP 版本 B

SFDP参数的相对字节地址	SFDP Dword名称	数据	说明
00h	JEDEC基本闪存参数 Dword-1	E5h	SFDP JEDEC参数开始 位7:5 = 未使用 = 111b 位4:3 = 05h是易失性状态寄存器写入指令和状态寄存器的非易失性默认值 = 00b 位2 = 编程缓冲区 > 64个字节 = 1 位1:0 = 器件支持统一的4 KB擦除 = 01b
01h		20h	位15:8 = 统一的4 KB擦除指令 = 20h
02h		FBh	位23 = 未使用 = 1b 位22 = 指出器件是否支持DOR读取。若支持，该值为1b 位21 = 指出器件是否支持QIO读取。若支持，该值为1b 位20 = 指出器件是否支持DIO读取。若支持，该值为1b 位19 = 指出器件是否支持DDR。若支持，该值为1b 位18:17 = 指出地址字节的数量。若数量为3或4，则该值为01b 位16 = 指出器件是否支持快速读取SIO和DIO。若支持，该值为1b
03h		FFh	位31:24 = 未使用 = FFh
04h	JEDEC基本闪存参数 Dword-2	FFh	容量（单位为位，基于零）， 128 Mb = 07FFFFFFh 256 Mb = 0FFFFFFFh 512 Mb = 1FFFFFFFh
05h		FFh	
06h		FFh	
07h		07h 128 Mb 0Fh 256 Mb 1Fh 512 Mb	
08h	JEDEC基本闪存参数 Dword-3	48h	位7:5 = QIO模式周期的数量 = 010b 位4:0 = 快速读取QIO虚拟周期的数量 = 01000b（对于默认延迟代码）
09h		EBh	快速读取QIO指令代码
0Ah		08h	位23:21 = 四线输出模式周期的数量 = 000b 位20:16 = 四线输出虚拟周期的数量 = 01000b（对于默认延迟代码）
0Bh		6Bh	四线输出指令代码
0Ch	JEDEC基本闪存参数 Dword-4	08h	位7:5 = 双线输出模式周期的数量 = 000b 位4:0 = 双线输出虚拟周期的数量 = 01000b（对于默认延迟代码）
0Dh		3Bh	双线输出指令代码
0Eh		88 h	位23:21 = 双线I/O模式周期的数量 = 100b 位20:16 = 双线I/O虚拟周期的数量 = 01000b（对于默认延迟代码）
0Fh		BBh	双线I/O指令代码
10h	JEDEC基本闪存参数 Dword-5	FEh	位7:5 RFU = 111b 位4 = 支持QPI = 1b 位3:1 RFU = 111b 位0 = 不支持所有双线指令 = 0b
11h		FFh	位15:8 = RFU = FFh
12h		FFh	位23:16 = RFU = FFh
13h		FFh	位31:24 = RFU = FFh
14h	JEDEC基本闪存参数 Dword-6	FFh	位7:0 = RFU = FFh
15h		FFh	位15:8 = RFU = FFh
16h		FFh	位23:21 = 所有双线指令的模式周期 = 111b 位20:16 = 所有双线指令的虚拟周期 = 11111b
17h		FFh	所有双线指令代码
18h	JEDEC基本闪存参数 Dword-7	FFh	位7:0 = RFU = FFh
19h		FFh	位15:8 = RFU = FFh
1Ah		48h	位23:21 = QPI模式周期的数量 = 010b 位20:16 = QPI虚拟周期的数量 = 01000b（对于默认延迟代码）
1Bh		EBh	QPI快速读取指令代码（QPI使能时，与QIO相同）

表 10.3 基本 SPI 闪存参数， JEDEC SFDP 版本 B（续）

SFDP 参数的相对字节地址	SFDP Dword 名称	数据	说明
1Ch	JEDEC 基本闪存参数 Dword-8	0Ch	扇区1类大小 = 2^N 字节 = 4 KB = 0Ch (统一为4 KB)
1Dh		20h	扇区1类指令
1Eh		0Fh	扇区2类大小 = 2^N 字节 = 32 KB = 0Fh (统一为32 KB)
1Fh		52h	扇区2类指令
20h	JEDEC 基本闪存参数 Dword-9	10h	扇区3类大小 = 2^N 字节 = 64 KB = 10h (统一为64 KB)
21h		D8h	扇区3类指令
22h		00h	扇区4类大小 = 2^N 字节 = 不支持 = 00h
23h		FFh	扇区4类指令 = 不支持 = FFh
24h	JEDEC 基本闪存参数 Dword-10	21h	位31:30 = 扇区4类擦除，典型时间单位 (00b: 1 ms、01b: 16 ms、10b: 128 ms、11b: 1 s) = RFU = 11b
25h		5Ah	位29:25 = 扇区4类擦除，典型时间计数 = RFU = 1_1111b (典型擦除时间 = 计数值 + 1 * 单位 = RFU = 11111)
26h		C1h	位24:23 = 扇区3类擦除，典型时间单位 (00b: 1 ms、01b: 16 ms、10b: 128 ms、11b: 1 s) = 16 ms = 01b
27h		FEh	位22:18 = 扇区3类擦除，典型时间计数 = 1_0000b (典型擦除时间 = 计数值 + 1 * 单位 = 17×16 ms = 272 ms) 位17:16 = 扇区2类擦除，典型时间单位 (00b: 1 ms、01b: 16 ms、10b: 128 ms、11b: 1 s) = 16 ms = 01b 位15:11 = 扇区2类擦除，典型时间计数 = 0_1011b (典型擦除时间 = 计数值 + 1 * 单位 = 12×16 ms = 192 ms) 位10:9 = 扇区1类擦除，典型时间单位 (00b: 1 ms、01b: 16 ms、10b: 128 ms、11b: 1 s) = 16 ms = 01b 位8:4 = 扇区1类擦除，典型时间计数 = 0_0010b (典型擦除时间 = 计数值 + 1 * 单位 = 3×16 ms = 48 ms) 位3:0 = 计数值 = (最长的擦除时间 / (2 * 典型擦除时间)) - 1 = 0001b 典型的擦除时间与最长擦除时间之间的倍数 = 4x 最长擦除时间 = 2 * (计数值 + 1) * 典型擦除时间 二进制字段: 11-11111-01-10000-01-01011-01-00010-0001 半字节格式: 1111_1110_1100_0001_0101_1010_0010_0001 十六进制格式: FE_C1_5A_21
28h	JEDEC 基本闪存参数 Dword-11	81h	位23 = 字节编程典型时间，附加字节单位 (0b: 1 us、1b: 8 us) = 1 us = 0b
29h		E4h	位22:19 = 字节编程典型时间，附加字节计数，(计数值+1) * 单位，计数值 = 0101b，(典型编程时间 = 计数值 + 1 * 单位 = 6×1 us = 6 us)
2Ah		29h	位18 = 字节编程典型时间，第一字节单位 (0b: 1 us、1b: 8 us) = 1 us = 0b
			位17:14 = 字节编程典型时间，第一字节计数，(计数值+1) * 单位，计数值 = 0111b，(典型编程时间 = 计数值 + 1 * 单位 = 8×1 us = 8 us)
2Bh	D1h 128Mb E2h 256 Mb	位13 = 页编程典型时间单位 (0b: 8 us、1b: 64 us) = 64 us = 1b	
		位12:8 = 页编程典型时间计数，(计数值+1) * 单位，计数值 = 00100b，(典型编程时间 = 计数值 + 1 * 单位 = 5×64 us = 320 us)	
		位7:4 = N = 1000b，页大小 = 2^N = 256B 页	
		位3:0 = 计数值 = 0001b = (最大页编程时间 / (2 * 典型页编程时间)) - 1	
		典型页编程时间与最大页编程时间之间的倍数 = 4x 最大页编程时间 = 2 * (计数值 + 1) * 典型页编程时间 二进制字段: 0-0101-0-0111-1-00100-1000-0001 半字节格式: 0010_1001_1110_0100_1000_0001 十六进制格式: 29_74_81	
		128 Mb = 1101_0001b = D1h	
		位31保留 = 1b	
		位30:29 = 芯片擦除，典型时间单位 (00b: 16 ms、01b: 256 ms、10b: 4 s、11b: 64 s) = 4 s = 10b	
		位28:24 = 芯片擦除，典型时间计数，(计数值+1) * 单位，计数值 = 10001b，(典型编程时间 = 计数值 + 1 * 单位 = 18×4 s = 72 s)	
		256 Mb = 1110_0010b = E2h	
		位31保留 = 1b	
		位30:29 = 芯片擦除，典型时间单位 (00b: 16 ms、01b: 256 ms、10b: 4 s、11b: 64 s) = 64 s = 11b	
		位28:24 = 芯片擦除，典型时间计数，(计数值+1) * 单位，计数值 = 00010b，(典型编程时间 = 计数值 + 1 * 单位 = 3×64 s = 192 s)	

表 10.3 基本 SPI 闪存参数， JEDEC SFDP 版本 B（续）

SFDP参数的相对字节地址	SFDP Dword名称	数据	说明
2Ch	JEDEC基本闪存参数 Dword-12	CCh	位31 = 支持挂起和恢复 = 0b
2Dh		83h	位30:29 = 挂起正在执行的擦除操作的最长延迟单位（00b: 128ns、01b: 1us、10b: 8us、11b: 64us） = 8us = 10b
2Eh		18h	位28:24 = 挂起正在执行的擦除操作的最长延迟计数值 = 00100b，最长擦除挂起延迟 = 计数值 + 1 * 单位 = 5*8us = 40us
2Fh		44h	位23:20 = 擦除恢复到挂起间隔计数值 = 0001b，间隔 = 计数值 + 1 * 64us = 2 * 64us = 128us 位19:18 = 挂起正在执行的编程操作的最长延迟单位（00b: 128ns、01b: 1us、10b: 8us、11b: 64us） = 8us = 10b 位17:13 = 挂起正在执行的编程操作的最长延迟计数值 = 00100b，最长擦除挂起延迟 = 计数值 + 1 * 单位 = 5*8us = 40us 位12:9 = 编程恢复到挂起间隔计数值 = 0001b，间隔 = 计数值 + 1 * 64us = 2 * 64us = 128us 位8 = RFU = 1b 位7:4 = 擦除挂起过程中禁用各种操作 = xxx0b: 可能不会启动任何新的擦除操作（不允许擦除嵌套） + xx0xb: 可能不会启动任何页编程操作 + x1xxb: 可能不会在擦除挂起的扇区空间内启动读操作 + 1xxxb: 位5:4中的擦除和编程限制充足 = 1100b 位3:0 = 编程挂起过程中禁用各种操作 = xxx0b: 可能不会启动任何新的擦除操作（不允许擦除嵌套） + xx0xb: 可能不会启动任何新的页编程操作（不允许编程嵌套） + x1xxb: 可能不会在编程挂起的页空间内启动读操作 + 1xxxb: 位1:0中的擦除和编程限制充足 = 1100b 二进制字段: 0-10-00100-0001-10-00100-0001-1-1100-1100 半字节格式: 0100_0100_0001_1000_1000_0011_1100_1100 十六进制格式: 44_18_83_CC
30h	JEDEC基本闪存参数 Dword-13	7Ah	位31:24 = 擦除挂起指令 = 75h 位23:16 = 擦除恢复指令 = 7Ah 位15:8 = 编程挂起指令 = 75h 位7:0 = 编程恢复指令 = 7Ah
31h		75h	
32h		7Ah	
33h		75h	
34h	JEDEC基本闪存参数 Dword-14	F7h	位31 = 指出器件是否支持深度下电模式。若支持，该值为0 位30:23 = 进入深度下电指令 = B9h = 1011_1001b 位22:15 = 退出深度下电指令 = ABh = 1010_1011b 位14:13 = 退出深度下电到执行下一个操作的延迟单位（00b: 128ns、01b: 1us、10b: 8us、11b: 64us）。若延迟单位为1us，该位为01b 位12:8 = 退出深度下电到执行下一个操作的延迟计数 = 00010b，退出深度下电直到下一个操作的延迟 = (计数值+1) * 单位 = 3*1us = 3us 位7:4 = RFU = Fh 位3:2 = 通过轮询状态寄存器指出器件是否忙碌 = 01b: 支持传统状态轮询 = 发出05h指令来读取状态寄存器并检查WIP位（0 = 就绪；1 = 忙碌），以进行传统轮询。 位1:0 = RFU = 11b 二进制字段: 0-10111001-10101011-01-00010-1111-01-11 半字节格式: 0101_1100_1101_0101_1010_0010_1111_0111 十六进制格式: 5C_D5_A2_F7
35h		A2h	
36h		D5h	
37h		5Ch	

表 10.3 基本 SPI 闪存参数， JEDEC SFDP 版本 B（续）

SFDP参数的相对字节地址	SFDP Dword名称	数据	说明
38h	JEDEC基本闪存参数 Dword-15	22h	位31:24 = RFU = FFh 位23 = 禁用Hold和WP = 不支持 = 0b 位22:20 = 四线使能要求 = 101b: QE是状态寄存器2的位1。使用读取状态指令05h便可读取状态寄存器1。通过指令35h读取状态寄存器2。通过带有两个数据字节（其中的第一个字节为“1”）的写入状态指令01h设置QE。通过带有两个数据字节（其中的第二个字节的位1为“0”）的写入状态指令来将其清零。 位19:16 0-4-4模式进入方法 = xxx1b: 模式位[7:0] = A5h 注意：使用该模式前，需要设置QE + x1xxb: 模式位[7:0] = Axh + 1xxb: RFU = 1101b 位15:10 0-4-4模式退出方法 = xx_xxx1b: 模式位[7:0] = 00h，那么在当前读取操作结束时，将终止该模式 +xx_1xxb: 8个周期内在DQ0-DQ3上输入Fh（模式位复位）。这样会使模式在执行下一个读操作前被终止。 + 11_x1xx: RFU = 111101 位9 = 支持0-4-4模式 = 1 位8:4 = 4-4-4模式使能序列 = 0_0010b: 发出指令38h 位3:0 = 4-4-4模式禁用序列 = 0010b: 4-4-4发出F5h指令 二进制字段: 11111111-0-101-1101-111101-1-00010-0010 半字节格式: 1111_1111_0101_1101_1111_0110_0010_0010 十六进制格式: FF_5D_F6_22
39h		F6h	
3Ah		5Dh	
3Bh		FFh	
3CH	JEDEC基本闪存参数 Dword-16	E8h	位31:24 = 进入4字节地址模式 = xxxx_xxx1b: 发出指令B7（无需事先使能写入） = xxxx_1xxb: 使用8位易失性组寄存器来定义A[30:24]位。MSB（位[7]）用于使能/禁用4字节地址模式。MSB为“1”时，4字节地址模式有效，A[30:24]位为无需关注项。使用指令16h进行读取。写指令为17h，数据长度为1字节。MSB被清零时，正确设置A[30:24]位，以选择有效的128 Mb段，并使用3字节地址。 + xx1x_xxxb: 支持专用的4字节地址指令集。请查看供应商数据手册以了解指令集或查看4字节地址参数表。 + 1xxx_xxxb: 保留 = 10100001b 位23:14 = 退出4字节地址模式 = xx_xxxx_xxx1b: 发出指令E9h以退出4字节地址模式（无需使用写入使能指令06h） = xx_xxxx_1xxb: 使用8位易失性组寄存器来定义A[30:24]位。MSB（位[7]）用于使能/禁用4字节地址模式。MSB被清零时，将进入3字节地址模式，并且A30:A24用于选择有效的128 Mb段。使用指令16h进行读取。写指令为17h，数据长度为1字节。 + xx_xx1x_xxxb: 硬件复位 + xx_x1xx_xxxb: 软件复位（请参见该DWORD中的位13:8） + xx_1xxx_xxxb: 电源循环 + x1_xxxx_xxxb: 保留 + 1x_xxxx_xxxb: 保留 = 1111100001b 位13:8 = 支持软复位和挽救序列 = x1_xxxb: 发出复位使能指令66h，然后发出复位指令99h。根据器件的操作模式，复位使能和复位序列可能在1、2或4条线上被发送。 = 010000b 位7 = RFU = 1 位6:0 = 易失性或非易失性寄存器和状态寄存器1的写使能指令 =xxx_1xxb: 非易失性/易失性状态寄存器1上电到向非易失性状态寄存器写入最后数值期间，使用指令06h使能对非易失性状态寄存器进行的写入操作。上电后，易失性状态寄存器可能被激活以替代非易失性状态寄存器，使用指令50h来使能写入操作和激活易失性状态寄存器。 + x1x_xxxb: 保留 + 1xx_xxxb: 保留 = 1101000b 二进制字段: 10100001-1111100001-010000-1-1101000 半字节格式: 1010_0001_1111_1000_0101_0000_1110_1000 十六进制格式: A1_F8_60_E8
3Dh		50h	
3Eh		F8h	
3Fh		A1h	

10.1.4 JEDEC SFDP 4字节地址指令表

表 10.4 4 字节地址指令， JEDEC SFDP 版本 B

SFDP参数的相对字节地址	SFDP Dword名称	数据	说明
40h	JEDEC 4字节地址指令 参数Dword-1h	FBh	支持 = 1, 不支持 = 0 位31:20 = RFU = FFFh 位19 = 支持非易失性单独扇区定位的写入命令, 指令 = E3h = 0 位18 = 支持非易失性单独扇区定位的读取命令, 指令 = E2h = 0 位17 = 支持易失性单独扇区定位的写入命令, 指令 = E1h = 1 位16 = 支持易失性单独扇区定位的读取命令, 指令 = E0h = 1 位15 = 支持(1-4-4) DTR_Read命令, 指令 = EEh = 1 位14 = 支持(1-2-2) DTR_Read命令, 指令 = BEh = 0 位13 = 支持(1-1-1) DTR_Read命令, 指令 = 0Eh = 0 位12 = 支持类型4的擦除命令 = 0 位11 = 支持类型3的擦除命令 = 1 位10 = 支持类型2的擦除命令 = 1 位9 = 支持类型1的擦除命令 = 1 位8 = 支持(1-4-4)页编程命令, 指令 = 3Eh = 0 位7 = 支持(1-1-4)页编程命令, 指令 = 34h = 1 位6 = 支持(1-1-1)页编程命令, 指令 = 12h = 1 位5 = 支持(1-4-4) FAST_READ命令, 指令 = ECh = 1 位4 = 支持(1-1-4) FAST_READ命令, 指令 = 6Ch = 1 位3 = 支持(1-2-2) FAST_READ命令, 指令 = BCh = 1 位2 = 支持(1-1-2) FAST_READ命令, 指令 = 3Ch = 0 位1 = 支持(1-1-1) FAST_READ命令, 指令 = 0Ch = 1 位0 = 支持(1-1-1) READ命令, 指令 = 13h = 1 半字节格式: 1111_1111_1111_0011_1000_1110_1111_1011 十六进制格式: FF_F3_8E_FB
41h		8Eh	
42h		F3h	
43h		FFh	
44h	JEDEC 4字节地址指令 参数Dword-2h	21h	位31:24 = FFh = 类型4的擦除指令: 保留 位23:16 = DCh = 类型3的擦除指令: 擦除块 位15:8 = 52h = 类型2的擦除指令: 擦除半块 位7:0 = 21h = 类型1的擦除指令: 擦除扇区
45h		52h	
46h		DCh	
47h		FFh	

10.2 器件ID地址映射

10.2.1 字段定义

表 10.5 制造商器件类型

字节地址	数据	说明
00h	01h	赛普拉斯的制造商ID
01h	60h	器件ID最高有效字节 — 存储器接口类型
02h	18h (128 Mb) 19h (256 Mb)	器件ID最低有效字节 — 容量和特性
03h	未定义	保留供将来使用

表 10.6 器件的唯一 ID

字节地址	数据	说明
00h到07	8字节大小的唯一器件ID	64位大小的唯一ID, 请参考第26页上的 6.3.1 器件的唯一ID 一节
08h到0F	附加8字节大小的唯一器件ID	128位大小的唯一ID的附加字节
10到1Fh	未定义	保留供将来使用
20h到37h	24字节OEM名称	OEM名称

10.3 初始供应状态

该器件由赛普拉斯生产，其非易失性位如下设置：

- 整个存储器阵列被擦除：例如，所有位均被设为1（因此每个字节都包含FFh）。
- 安全区域地址空间的全部字节都被擦除为FFh。
- SFDP地址空间包含如SFDP地址空间的说明部分中所定义的值。
- ID地址空间包含如ID地址空间的说明部分中所定义的值。
- 非易失性状态寄存器1包含00h（全部SR1NV位均被清除为0）。
- 非易失性配置寄存器1包含00h。
- 非易失性配置寄存器2包含60h。
- 非易失性配置寄存器3包含78h。
- 密码寄存器包含FFFFFFFF-FFFFFFFFh
- 对于标准器件，IRP寄存器位是FFFDh；对于高安全性器件，IRP寄存器位是FFFFh。
- PRPR寄存器位是FFFFFFh

11. 电气规范

11.1 绝对最大额定值

(注意3)

塑料封装存储温.....	-65°C ~ +150°C
通电时的环境温度.....	-65°C ~ +125°C
V _{DD}	-0.5 V ~ +4.0 V
对地 (V _{SS}) 输入电压 (注意1)	-0.5 V ~ V _{DD} + 0.5 V
输出短路电流 (注意2)	100 mA

注意:

1. 请参考第124页上的11.4.3 输入信号过冲一节以了解信号切换期间受允许的最大值。
2. 不能有多个输出同时对地短路。并且短路时间不能超过一秒。
3. 器件在高于“最大绝对额定值”部分中所列出的值工作可能会造成永久性的损害。它只是一个额定值，并不表示器件一定要采用这些值，或者采用本数据手册操作部分所显示的数值都能正常运行。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。

11.2 闭锁特性

表 11.1 闭锁规范

说明	最小值	最大值	单位
所有输入连接上相对于V _{SS} 的输入电压	-1.0	V _{IO} + 1.0	V
所有I/O连接上相对于V _{SS} 的输入电压	-1.0	V _{IO} + 1.0	V
V _{DD} 电流	-100	+100	mA

注意:

1. 不包括电源电压V_{DD}。测试条件: V_{DD} = 3.0 V。每次检查一个连接，V_{SS}下的连接不被检查。

11.3 热阻

表 11.2 热阻

参数	说明	SO316	SOC008	WND008	WNG008	FAB024	FAC024	单位
Theta JA	热阻 (结温)	38	53.27	32	18	39	39	°C/W

11.4 工作范围

工作范围定义了一些限值，在这些限值之间能够保证设备正常运行。

11.4.1 供电电压

V_{DD}	2.7 V ~ 3.6 V
----------	---------------

11.4.2 温度范围

参数	符号	器件	规范		单位
			最小值	最大值	
环境温度	T_A	工业级 (I)	-40	+85	°C
		扩展的工业级 (V)	-40	+105	
		扩展型 (N)	-40	+125	
		汽车级, AEC-Q100等级3 (A)	-40	+85	
		汽车级, AEC-Q100等级2 (B)	-40	+105	
		汽车级, AEC-Q100等级1 (M)	-40	+125	

11.4.3 输入信号过冲

在直流条件下，输入或I/O信号应处于从 V_{SS} 到 V_{DD} 之间的范围内。在电压转换期间，输入或I/O信号可能低于 V_{SS} ，降到-1.0 V或升到 $V_{DD}+1.0 V$ ，时间最长为20 ns。

图11.1 最大负过冲波形

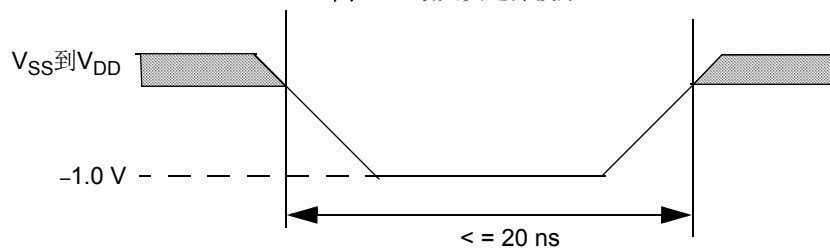
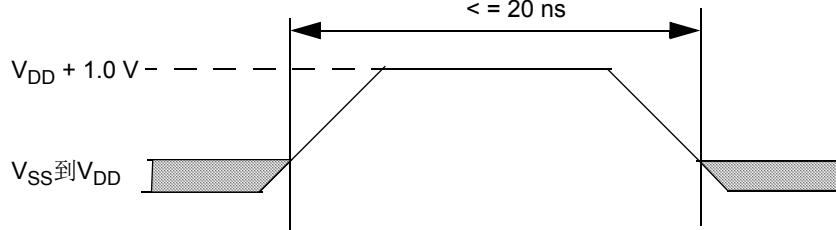


图11.2 最大正过冲波形
<= 20 ns



11.5 上电和下电

发生上电或下电（CS#的电压必须采用V_{DD}上的电压）时，不能选择器件，直到V_{DD}准确满足以下值：

- 上电时为V_{DD} (min)并在t_{PU}的更长延迟期间持续保持该值
- 下电时为V_{SS}

为保证安全、准确进行上电和下电操作，可以在片选（CS#）上安装一个简单的上拉电阻。

用户不能输入任何命令，直到V_{DD}上升超过V_{DD}最小阈值后再经过t_{PU}的有效延迟时间结束为止。请参见图11.3。但在t_{PU}期间，如果V_{DD}回到低于V_{DD} (min)的电平状态，将不能保证器件正常操作。在t_{PU}结束前，不应该发出任何命令。

在t_{PU}期间，器件消耗的电流为I_{POR}。上电（t_{PU}）后，器件将进入待机模式，并消耗CMOS待机电流（I_{SB}），WEL位被复位。

在下电期间或电压降至V_{DD} (cut-off)以下时，电压必须在t_{PD}的时间内保持小于V_{DD} (low)，这样，器件上电后能够正常初始化。请参见图11.4。如果在电压下降过程中，V_{DD}保持为高于V_{DD} (cut-off)的状态，那么当V_{DD}再次超过V_{DD} (min)时，器件将被初始化，并正常运行。如果上电后不能正常完成上电复位（POR）事件，那么当激活RESET#信号或接收到一个软件复位命令（先发送RSTEN66h，然后是RST 99h）时，器件将重启POR程序。

在执行嵌入式编程或擦除操作期间，V_{DD}下降到V_{DD} (Cut-off)以下，嵌入式操作可能被中止，存储器区域中的数据可能不正确。

为了稳定器件的V_{DD}电源，必须采取正常的预防措施进行电源退耦。系统中的每个器件都应该由封装连接附近大小合适的电容对V_{DD}轨进行退耦（该电容通常为0.1 μf）。

表 11.3 上电 / 下电电压和时序

符号	参数	最小值	最大值	单位
V _{DD} (min)	V _{CC} (最小工作电压)	2.7	-	V
V _{DD} (cut-off)	V _{DD} (关闭电压，在此需要重新进行初始化)	2.4	-	V
V _{DD} (low)	V _{DD} (确保发生初始化的低电压)	1.0	-	V
t _{PU}	从V _{DD} 为最小值到进行读取操作的时间	-	300	μs
t _{PD}	V _{DD} 为低电平的时间	10.0	-	μs

图11.3 上电

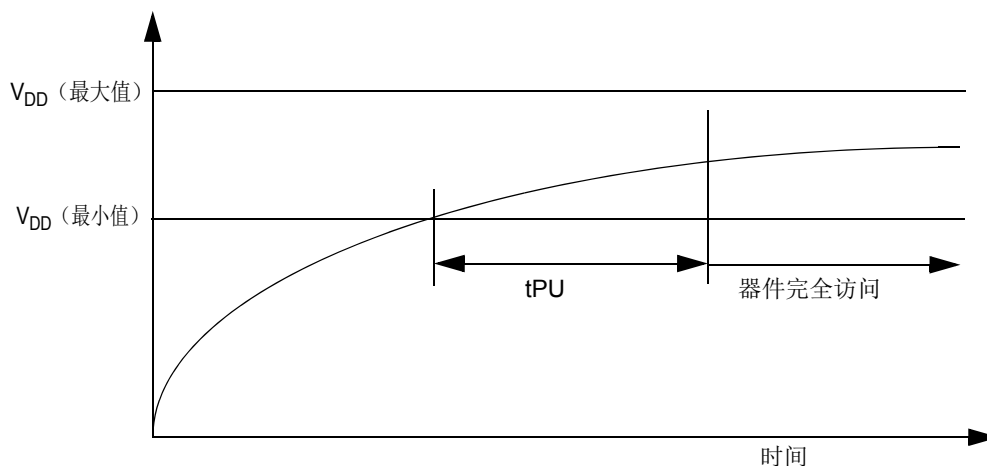
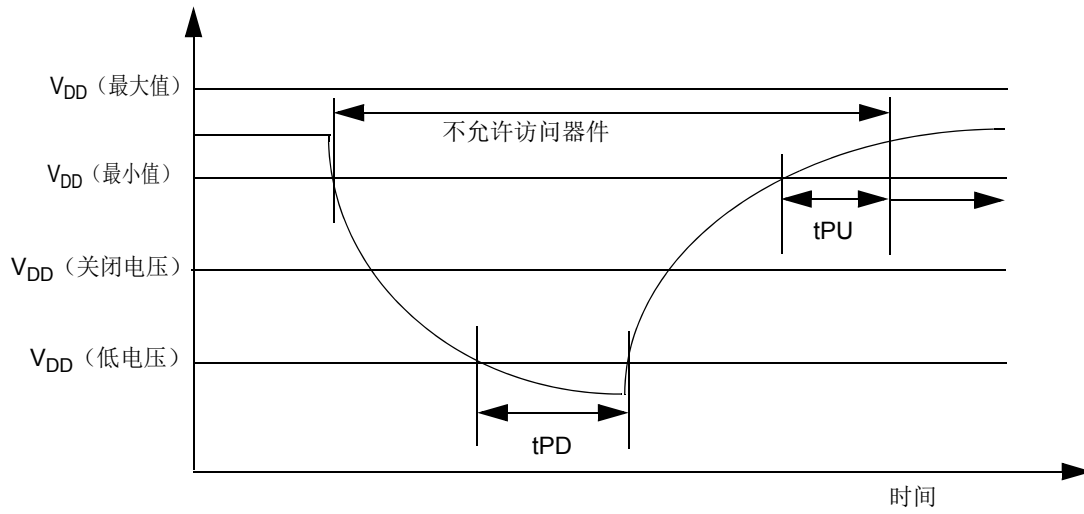


图11.4 下电或电压下降



11.6 直流特性

表 11.4 直流特性 — 工作温度范围：-40°C 到 +85°C

符号	参数	测试条件	最小值	典型值 (1)	最大值	单位
V_{IL}	输入低电压	-	-0.5	-	$0.3 \times V_{DD}$	V
V_{IH}	输入高电压	-	$0.7 \times V_{DD}$	-	$V_{DD}+0.4$	V
V_{OL}	输出低电压	$I_{OL} = 0.1 \text{ mA}$, $V_{DD} = V_{DD} \text{ min}$	-	-	0.2	V
V_{OH}	输出高电压	$I_{OH} = -0.1 \text{ mA}$	$V_{DD} - 0.2$	-	-	V
I_{LI}	输入漏电流	$V_{DD} = V_{DD} \text{ Max}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 2	μA
I_{LO}	输出漏电流	$V_{DD} = V_{DD} \text{ Max}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 2	μA
I_{CC1}	活动模式下的电流 (读取) (2)	串行SDR@5 MHz 串行SDR@10 MHz 串行SDR@20 MHz 串行SDR@50 MHz 串行SDR@108 MHz 串行SDR@133 MHz QIO/QPI SDR@108 MHz QIO/QPI SDR@133 MHz QIO/QPI DDR@30 MHz QIO/QPI DDR@66 MHz	-	10 10 10 15 25 30 25 30 30 15 30	15 15 20 30 35 30 35 20 35	mA
I_{CC2}	活动模式下的电流 (页编程)	$CS\# = V_{DD}$	-	40	50	mA
I_{CC3}	活动模式下的电流 (WRR或WRAR)	$CS\# = V_{DD}$	-	40	50	mA
I_{CC4}	活动模式下的电流 (SE)	$CS\# = V_{DD}$	-	40	50	mA
I_{CC5}	活动模式下的电流 (HBE、BE)	$CS\# = V_{DD}$	-	40	50	mA
I_{SB}	待机模式下的电流	RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS} ; SPI、 双线I/O和四线I/O模式	-	20	35	μA
		RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS} ; QPI模式	-	60	100	μA
I_{DPD}	深度下电模式下的电流	RESET#, $CS\# = V_{CC}$, $V_{IN} = \text{GND}$ 或 V_{CC}	-	2	20	μA
I_{POR}	上电复位电流	RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS}	-	15	30	mA

注意:

- 典型值的条件为 $T_{AI} = 25^\circ\text{C}$ 和 $V_{DD} = 3.0 \text{ V}$ 。
- 读取返回数据期间，输出处于未连接状态。不包含输出开关电流。

表 11.5 直流特性 — 工作温度范围：-40°C 到 +105°C

符号	参数	测试条件	最小值	典型值 (1)	最大值	单位
V_{IL}	输入低电压	-	-0.5	-	$0.3 \times V_{DD}$	V
V_{IH}	输入高电压	-	$0.7 \times V_{DD}$	-	$V_{DD} + 0.4$	V
V_{OL}	输出低电压	$I_{OL} = 0.1 \text{ mA}$, $V_{DD} = V_{DDmin}$	-	-	0.2	V
V_{OH}	输出高电压	$I_{OH} = -0.1 \text{ mA}$	$V_{DD} - 0.2$	-	-	V
I_{LI}	输入漏电流	$V_{DD} = V_{DDMax}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 4	μA
I_{LO}	输出漏电流	$V_{DD} = V_{DDMax}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 4	μA
I_{CC1}	活动模式下的电流 (读取) (2)	串行SDR@5 MHz 串行SDR@10 MHz 串行SDR@20 MHz 串行SDR@50 MHz 串行SDR@108 MHz 串行SDR@133 MHz QIO/QPI SDR@108 MHz QIO/QPI SDR@133 MHz QIO/QPI DDR@30 MHz QIO/QPI DDR@66 MHz	-	10 10 10 15 25 30 25 30 15 30	20 20 20 25 35 40 35 40 25 40	mA
I_{CC2}	活动模式下的电流 (页编程)	$CS\# = V_{DD}$	-	40	60	mA
I_{CC3}	活动模式下的电流 (WRR或WRAR)	$CS\# = V_{DD}$	-	40	60	mA
I_{CC4}	活动模式下的电流 (SE)	$CS\# = V_{DD}$	-	40	60	mA
I_{CC5}	活动模式下的电流 (HBE、BE)	$CS\# = V_{DD}$	-	40	60	mA
I_{SB}	待机模式下的电流	RESET#, $CS\# = V_{DD}$; SI, $SCK = V_{DD}$ 或 V_{SS} ; SPI、 双线I/O和四线I/O模式	-	20	45	μA
		RESET#, $CS\# = V_{DD}$; SI, $SCK = V_{DD}$ 或 V_{SS} ; QPI模式	-	60	110	μA
I_{DPD}	深度下电模式下的电流	RESET#, $CS\# = V_{CC}$, $V_{IN} = GND$ 或 V_{CC}	-	2	30	μA
I_{POR}	上电复位电流	RESET#, $CS\# = V_{DD}$; SI, $SCK = V_{DD}$ 或 V_{SS}	-	15	30	mA

注意:

1. 典型值的条件为 $T_{AI} = 25^\circ\text{C}$ 和 $V_{DD} = 3.0 \text{ V}$ 。
2. 读取返回数据期间, 输出处于未连接状态。不包含输出开关电流。

表 11.5 直流特性 — 工作温度范围：-40°C 到 +125°C

符号	参数	测试条件	最小值	典型值 (1)	最大值	单位
V_{IL}	输入低电压	-	-0.5	-	$0.3 \times V_{DD}$	V
V_{IH}	输入高电压	-	$0.7 \times V_{DD}$	-	$V_{DD}+0.4$	V
V_{OL}	输出低电压	$I_{OL} = 0.1 \text{ mA}$, $V_{DD} = V_{DDmin}$	-	-	0.2	V
V_{OH}	输出高电压	$I_{OH} = -0.1 \text{ mA}$	$V_{DD} - 0.2$	-	-	V
I_{LI}	输入漏电流	$V_{DD} = V_{DDMax}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 4	μA
I_{LO}	输出漏电流	$V_{DD} = V_{DDMax}$, $V_{IN} = V_{IH}$ 或 V_{SS} , $CS\# = V_{IH}$	-	-	± 4	μA
I_{CC1}	活动模式下的电流 (读取) (2)	串行SDR@5 MHz 串行SDR@10 MHz 串行SDR@20 MHz 串行SDR@50 MHz 串行SDR@108 MHz 串行SDR@133 MHz QIO/QPI SDR@108 MHz QIO/QPI SDR@133 MHz QIO/QPI DDR@30 MHz QIO/QPI DDR@66 MHz	-	10 10 10 15 25 30 25 30 15 30	30 30 30 35 45 50 45 50 35 50	mA
I_{CC2}	活动模式下的电流 (页编程)	$CS\# = V_{DD}$	-	40	70	mA
I_{CC3}	活动模式下的电流 (WRR或WRAR)	$CS\# = V_{DD}$	-	40	70	mA
I_{CC4}	活动模式下的电流 (SE)	$CS\# = V_{DD}$	-	40	70	mA
I_{CC5}	活动模式下的电流 (HBE、BE)	$CS\# = V_{DD}$	-	40	70	mA
I_{SB}	待机模式下的电流	RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS} ; SPI、 双线I/O和四线I/O模式	-	20	75	μA
		RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS} ; QPI模式	-	60	150	μA
I_{DPD}	深度下电模式下的电流	RESET#, $CS\# = V_{CC}$, $V_{IN} = \text{GND}$ 或 V_{CC}	-	2	50	μA
I_{POR}	上电复位电流	RESET#, $CS\# = V_{DD}$; SI, SCK = V_{DD} 或 V_{SS}	-	15	35	mA

注意:

1. 典型值的条件为 $T_{AI} = 25^\circ\text{C}$ 和 $V_{DD} = 1.8 \text{ V}$ 。
2. 读取返回数据期间，输出处于未连接状态。不包含输出开关电流。

11.6.1 活动模式和待机模式

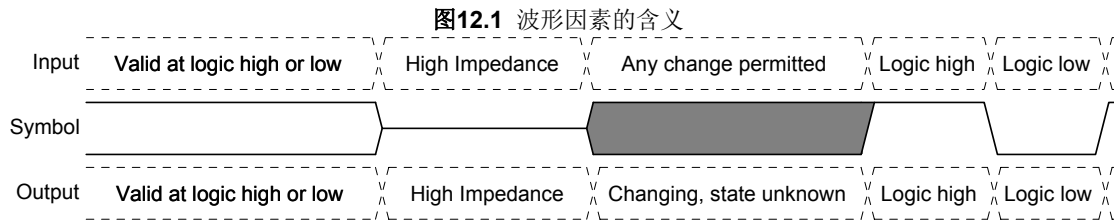
如果片选 (CS#) 为低电平，器件被使能且处于活动模式。如果CS#为高电平，器件将被禁用，但仍处于活动模式，直到完成所有编程、擦除以及写入操作为止。然后，器件会进入待机模式，功耗降低到 I_{SB} 。

11.6.2 深度下电模式 (DPD)

通过输入命令指令代码“B9h”，并将功耗降低到 I_{DPD} ，可使能深度下电模式。在DPD模式下，器件仅对从DPD模式恢复的命令 (RES ABh) 或硬件复位 (RESET#和IO3/RESET#) 发出响应。在DPD模式下，所有其他命令均被忽略。

12. 时序规范

12.1 切换波形的关键



12.2 交流测试条件

图12.2 测试设置

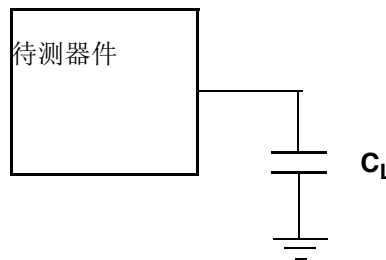


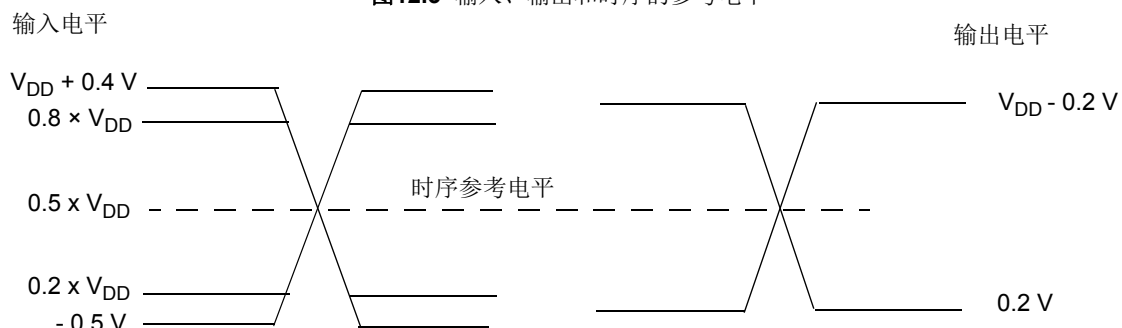
表 12.1 交流测量条件

符号	参数	最小值	最大值	单位
C_L	负载电容	-	15/30 (1)	pF
-	输入脉冲电压	$0.2 \times V_{DD}$	$0.8 V_{DD}$	V
-	输入时序参考电压		$0.5 V_{DD}$	V
-	输出时序参考电压		$0.5 V_{DD}$	V

注意:

1. 负载电容取决于操作频率或操作模式。
2. 交流特性表假定时钟和数据信号具有相同的转换速率（斜率）。请参见第134页上的SDR交流特性。注意6中的内容说明了在操作频率时的转换速率。

图12.3 输入、输出和时序的参考电平



12.2.1 电容特性

表 12.2 电容

	参数	测试条件	最小值	最大值	单位
C_{IN}	输入电容 (SCK、CS#、RESET#、IO3/RESET#上)	1 MHz	-	8	pF
C_{OUT}	输出电容 (在所有I/O上)	1 MHz	-	8	pF

12.3 复位

如果在执行对寄存器的擦除、编程或写入操作期间，启动了硬件复位，并且扇区、页面或寄存器内的数据不稳定，那么需要重新启动被中断的操作。在执行软件复位操作期间，如果启动硬件复位，该硬件复位可能被忽略。

12.3.1 上电（冷）复位

器件会执行上电复位 (POR) 程序，直到 V_{DD} 上升到超过 V_{DD} 最小阈值后经过 t_{PU} 延迟时间为止。请参考第125页上的图11.3、第125页上的表11.3。上电期间 (t_{PU})，不能选择器件（不允许CS#的电压上升到 V_{DD} ）。例如，不能将任何命令发送给器件，直到经过 t_{PU} 为止。

在POR期间，RESET#和IO3/RESET#复位功能被忽略。如果在POR期间，RESET#或IO3/RESET#为低电平，并且在 t_{PU} 期间以及之后仍然保持为低电平状态，那么CS#必须保持为高电平，直到RESET#和IO3/RESET#返回高电平状态并经过 t_{RH} 为止。在比 t_{RS} 更长的时间内，并且返回低电平状态前，RESET#和IO3_RESET#必须返回高电平状态，以便启动一次硬件复位。

在比 t_{CS} 更长的时间内，或者未使能四线模式或QPI模式 ($CR1V[1] = 0$ 或 $CR2V[3] = 0$) 时，IO3/RESET#输入可作为RESET#信号使用。

图12.4 POR结束时，复位信号为低电平

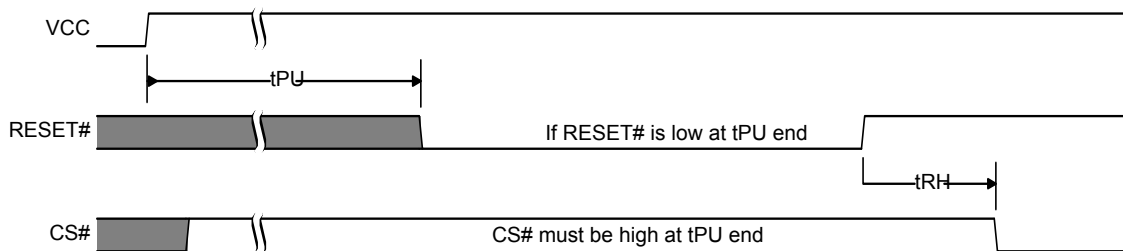


图12.5 POR结束时，复位信号为高电平

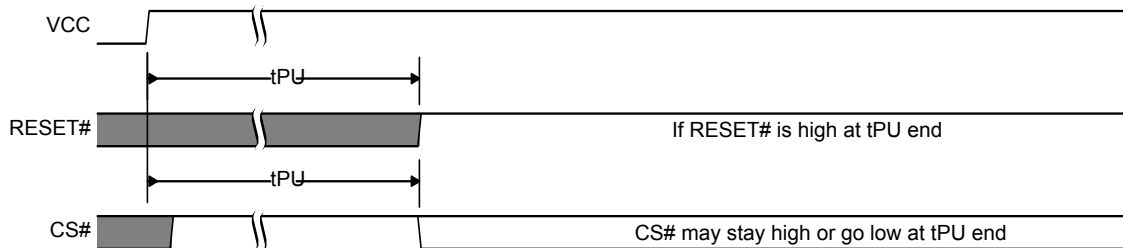
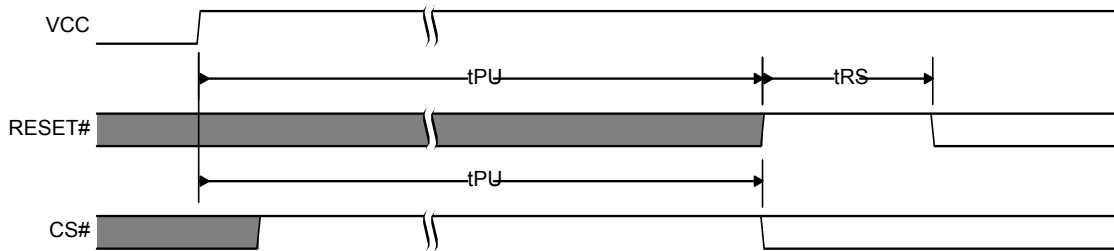


图12.6 上电复位后发生硬件复位



12.3.2 启动硬件（暖）复位的RESET #和IO3_RESET#输入

RESET#和IO3/RESET#输入可作为RESET#信号使用。根据条件，两个输入可以启动复位操作。

在比t_{RP}更长的时间内，当从V_{IH}转换为V_{IL}时，RESET#输入启动了复位操作，器件会使用与上电复位相同的方式复位寄存器状态，但没有经过POR期间所执行的完整复位程序。完成硬件复位程序需要t_{RPH}时长。RESET#输入仅在SOIC 16 含铅封装和BGA球型焊盘封装中有效。

如果超过t_{CS}时间，或未使能四线模式或QPI模式（CR1V[1] = 0或CR2V[3] = 0），CS#为高电平，那么IO3/RESET#输入将启动复位操作。IO3/RESET#输入具有一个连接至VDD的内部上拉电阻；如果不处于四线模式或QPI模式，该输入会保持为无连接状态。CS#为低电平时该输入用作四线或QPI模式I/O信号后，在CS#变为高电平后的t_{CS}延迟时间内，存储器或主机系统将IO3驱动为高电平。然后，将内部上拉电阻连接到VDD，可保持IO3/RESET#为高电平，直到主机系统开始驱动IO3/RESET#为止。在t_{CS}期间，CS#保持高电平状态时，IO3/RESET#输入将被忽略，从而可以避免发生意外的复位操作。如果CS#被驱动为低电平，便会启动一条新命令，这时，IO3/RESET#将作为IO3。

当器件并非处于四线模式或QPI模式时，或者CS#为高电平，并且IO3/RESET#在超过t_{RP}（位于t_{CS}后面）的时间内从V_{IH}转换为V_{IL}时，器件会使用与上电复位相同的方式复位寄存器状态，但并未执行完在POR期间所执行的完整复位程序。

完成硬件复位程序需要t_{RPH}的时间。如果由于任意原因不能在上电（t_{PU}）期间完成POR程序，RESET#变为低电平会启动完整的POR程序（而不是硬件复位程序），并需要占用t_{PU}长的时间才能完成POR程序。

软件复位命令（由RST 99h随后的RSTEN 66h）独立于RESET #和IO3_RESET#的状态。如果RESET#和IO3/RESET#为高电平，且已经发送了软件复位指令，则器件将执行软件复位。

其他注意事项：

- 如果RESET#和IO3/RESET#输入选项均可用，那么在您的系统中仅使用了一个复位选项。通过将CR2NV[7]设置为0（请参考第34页上的表6.11.非易失性配置寄存器2（CR2NV）），并设置IO3_RESET，使其仅作为IO3使用，可以禁用IO3/RESET#输入复位操作。不连接RESET#输入，或将其绑定到VIH上，可以禁用该输入。重新变为低电平以启动硬件复位前，经过t_{PU}或t_{RPH}时间后，在t_{RS}时间内RESET#和IO3/RESET#必须为高电平。
- 经过t_{CS}时长后，如果IO3/RESET#在长于最短时间（t_{RP}）内被驱动为低电平，则器件会立即终止正在执行的所有操作，使所有输出变为高阻抗，并在t_{RPH}时间内忽略所有读/写命令。器件将接口复位为待机状态。
- 如果使能了四线模式或QPI模式和IO3/RESET#性能，则在t_{CS}期间，主机系统不会将IO3驱动为低电平，以避免在IO3上发生冲突。执行以下各条命令（在四线模式或QPI模式下将数据传输到主机）（例如，四线I/O读取命令）后，在t_{CS}期间，存储器会立即将IO3/RESET#驱动为高电平，以避免意外发生复位操作。执行完以下各条命令（在四线模式下将数据传输到存储器）（例如，页编程命令）后，在t_{CS}时间内，主机系统会立即将IO3/RESET#驱动为高电平，以避免意外发生复位操作。
- 如果未使能四线模式或QPI模式，并且IO3/RESET#被置为低电平时CS#处于低电平状态，那么经过t_{RH}后CS#被重新置为低电平前，该信号必须在t_{RPH}时间内返回到高电平状态。

表 12.3 硬件复位参数

参数	说明	限制	时间	单位
t_{RS}	复位设置 — 从上次复位命令结束并且RESET#为高电平到RESET#为低电平的时间	最小值	50	ns
t_{RPH}	复位脉冲保持 — RESET#为低电平到CS#为低电平	最小值	100	μ s
t_{RP}	RESET#脉冲宽度	最小值	200	ns
t_{RH}	复位保持 — CS#为低电平前RESET#为高电平的时间	最小值	150	ns

注意:

1. 上电 (t_{PU}) 期间内, RESET#和IO3/RESET#低电平将被忽略。如果 t_{PU} 结束时Reset#被激活, 那么器件将保持为复位状态, t_{RH} 会决定CS#变为低电平的时间。
2. 如果使能了四线模式或QPI模式, 则在 t_{CS} 期间, IO3/RESET#低电平将被忽略。
3. $t_{RP} + t_{RH}$ 的和不能小于 t_{RPH} 。

图12.7 使用RESET#输入的硬件复位

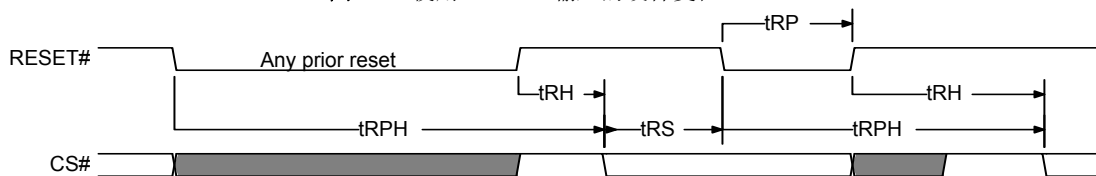


图12.8 未使能四线模式或QPI模式, 并且使能了IO3/RESET#时的硬件复位

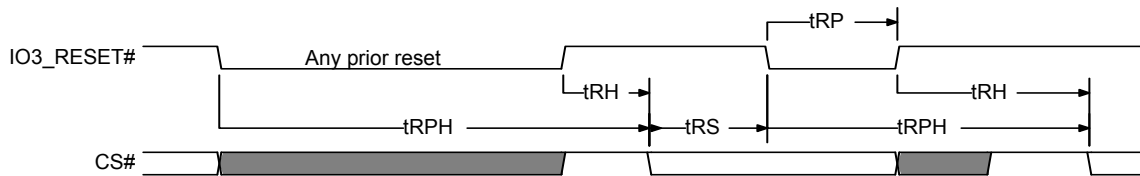
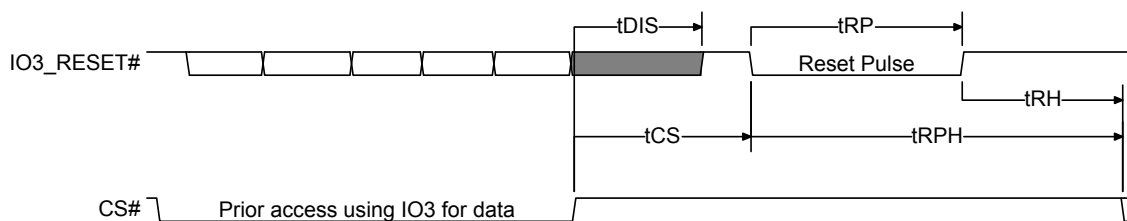


图12.9 使能了四线模式或QPI模式和IO3/RESET#时的硬件复位



12.4 SDR交流特性

表 12.4 SDR 交流特性

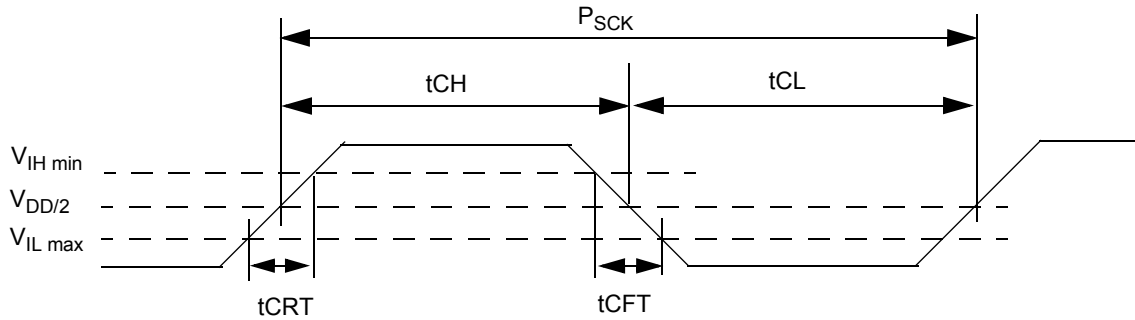
符号	参数	最小值	最大值	单位
$F_{SCK, R}$	使用于READ和4READ指令的SCK时钟频率	DC	50	MHz
$F_{SCK, C}$	使用于QOR、4QOR、DIOR、4DIOR、QIOR、4QIOR等双线命令和四线命令的SCK时钟频率	DC	133	MHz
P_{SCK}	SCK时钟周期	$1/F_{SCK}$	–	–
t_{WH}, t_{CH}	时钟为高电平的时间	$50\% P_{SCK} \pm 5\%$	–	ns
t_{WL}, t_{CL}	时钟为低电平的时间	$50\% P_{SCK} \pm 5\%$	–	ns
t_{CRT}, t_{CLCH}	时钟上升时间 (转换速率) (6)	0.1	–	V/ns
t_{CFT}, t_{CHCL}	时钟下降时间 (转换速率) (6)	0.1	–	V/ns
t_{CS}	CS#为高电平的时间 (执行任何读指令时)	20	–	ns
	CS#为高电平的时间 (执行所有的其他非读指令时)	50	–	ns
t_{CSS}	CS#有效的建立时间 (相对于SCK)	3	–	ns
t_{CSH}	CS#有效的保持时间 (相对于SCK)	5	–	ns
t_{SU}	数据输入的建立时间	3	–	ns
t_{HD}	数据输入的保持时间	2	–	ns
t_V	从时钟为低电平到输出有效的的时间	–	8 (2) 6 (3)	ns
t_{HO}	输出保持时间	1	–	ns
t_{DIS}	输出禁用时间 (4)	–	8	ns
	输出禁用时间 (当复位功能和四线模式均被使能时)	–	20 (5)	ns
t_{WPS}	WP#建立时间 (1)	20	–	ns
t_{WPH}	WP#保持时间 (1)	100	–	ns
T_{DP}	从CS#为高电平到进入深度下电模式的时间	–	3	us
T_{RES}	从CS#为高电平到退出深度下电模式的时间	–	5	us
t_{QEN}	进入QIO或QPI模式后执行下一条命令所需的时间	–	1.5	us
t_{QEXN}	退出QIO或QPI模式后执行下一条命令所需的时间	–	1	us

注意:

1. 仅在SRP0被设置为1时, 才能作为WRR或WRAR指令的限制。
2. 整个 V_{DD} 范围和 $CL = 30$ pF。
3. 整个 V_{DD} 范围和 $CL = 15$ pF。
4. 输出高阻态是数据不再被驱动的点。
5. 当使能复位功能和四线模式时 ($CR2V[7] = 1$ 和 $CR1V[1] = 1$), t_{DIS} 需要其他时间。
6. t_{CRT}, t_{CLCH} 时钟上升沿和下降沿转换速率, 使用快速时钟 (133 MHz) 时, 最小转换速率为1.5 V/ns, 使用慢速时钟 (50 MHz) 时, 该值为1.0 V/ns。

12.4.1 时钟时序

图12.10 时钟时序



12.4.2 输入/输出时序

图12.11 SPI单位输入时序

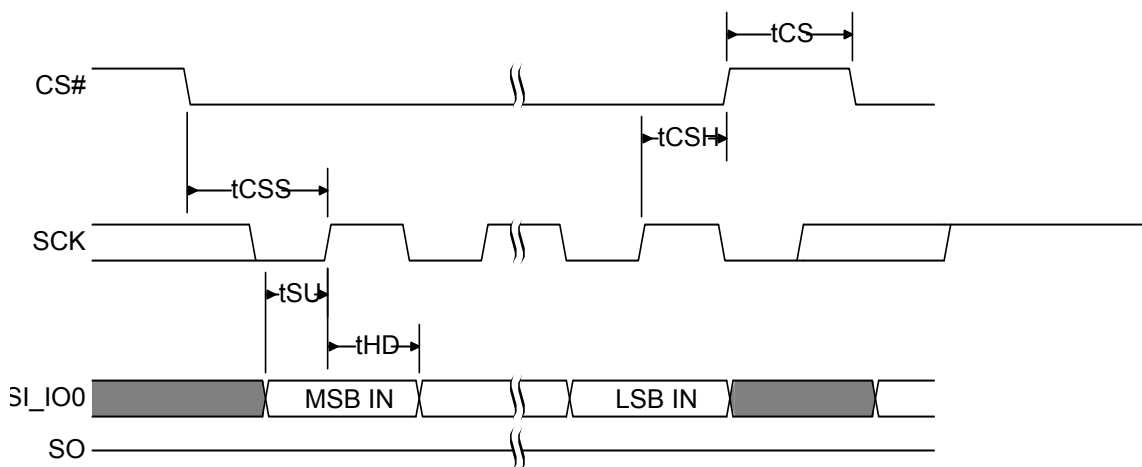


图12.12 SPI单位输出时序

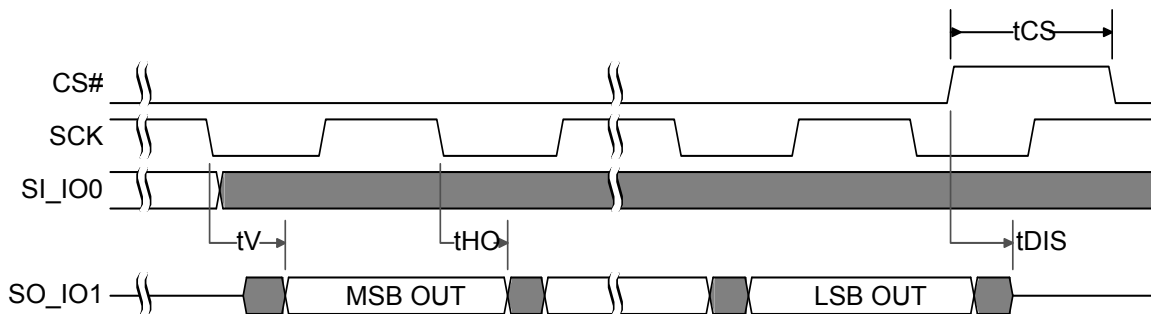


图12.13 SDR MIO时序

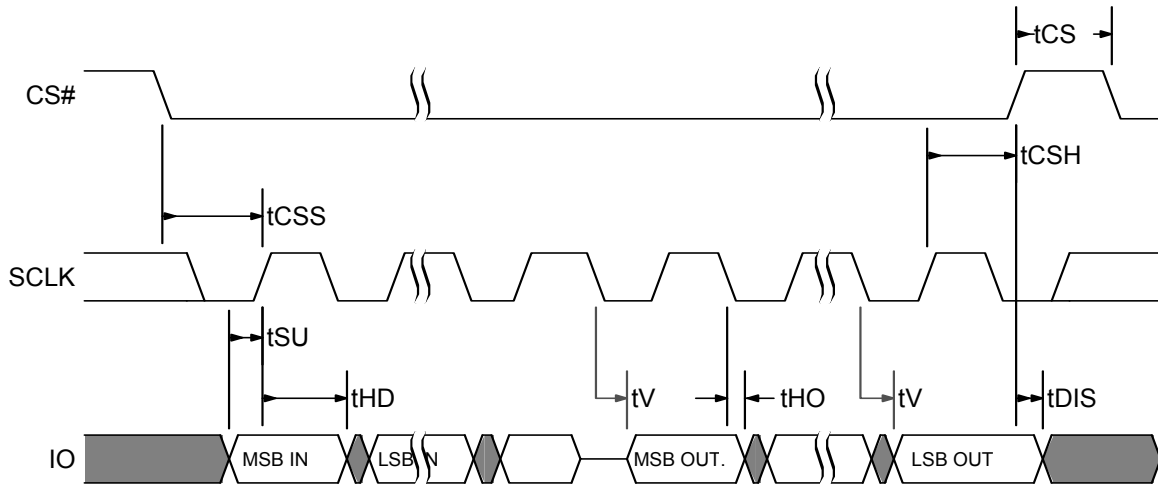
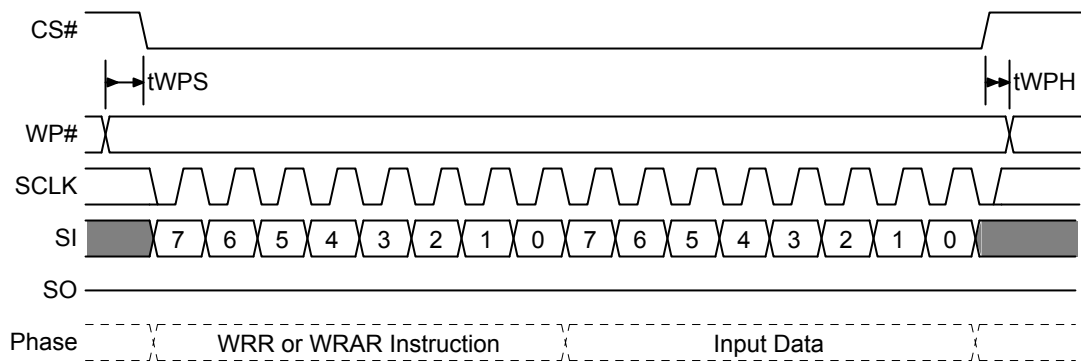


图12.14 WP#输入时序



12.5 DDR交流特性

表 12.5 工作频率为 66 MHz 时的 DDR 交流特性

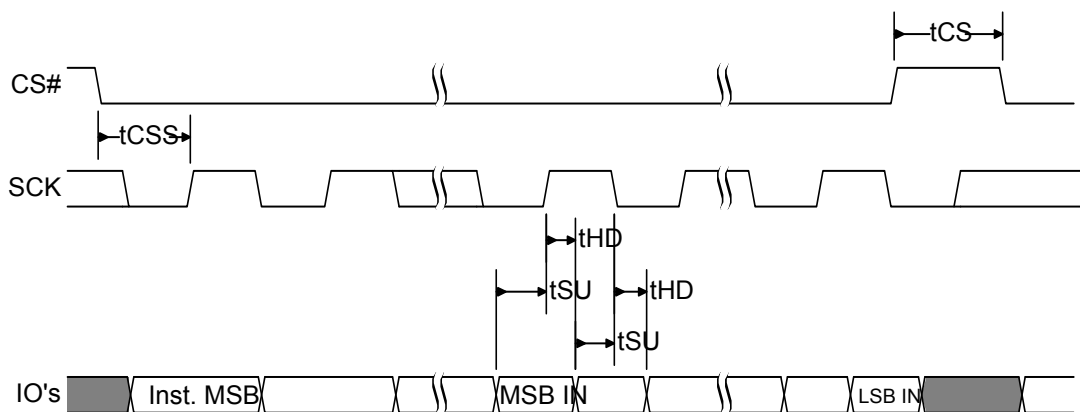
符号	参数	最小值	最大值	单位
$F_{SCK, R}$	使用于DDR READ指令的SCK时钟频率	DC	66	MHz
$P_{SCK, R}$	使用于DDR READ指令的SCK时钟周期	$1/F_{SCK}$	-	ns
t_{crt}	时钟上升时间 (转换速率)	1.5	-	V/ns
t_{cft}	时钟下降时间 (转换速率)	1.5	-	V/ns
t_{WH}, t_{CH}	时钟为高电平的时间	$50\% P_{SCK} - 5\%$	-	ns
t_{WL}, t_{CL}	时钟为低电平的时间	$50\% P_{SCK} - 5\%$	-	ns
t_{CS}	CS#为高电平的时间 (执行读取指令时) CS#为高电平的时间 (使能复位功能并执行读取指令时)	20 50	-	ns
t_{CSS}	CS#有效的建立时间 (相对于SCK)	3	-	ns
t_{SU}	IO输入的建立时间	3	-	ns
t_{HD}	IO输入的保持时间	2	-	ns
t_v	从时钟为低电平到输出有效的时间	-	8 (1) 6 (2)	ns
t_{HO}	输出保持时间	1	-	ns
t_{Dis}	输出禁用时间 输出禁用时间 (使能复位功能时)	-	8 20	ns
t_{O_skew}	从第一个IO数据有效到最后IO数据有效的的时间	-	600 (3)	ps

注意:

1. 整个 V_{DD} 范围和 $CL = 30\text{ pF}$ 。
2. 整个 V_{DD} 范围和 $CL = 15\text{ pF}$ 。
3. 未经测试。

12.5.1 DDR输入时序

图12.15 SPI DDR输入时序



12.5.2 DDR输出时序

图12.16 SPI DDR输出时序

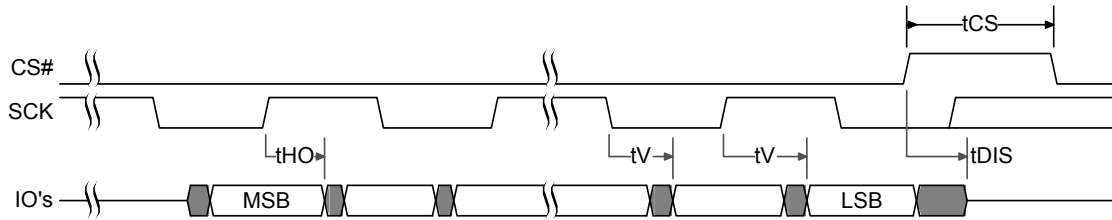
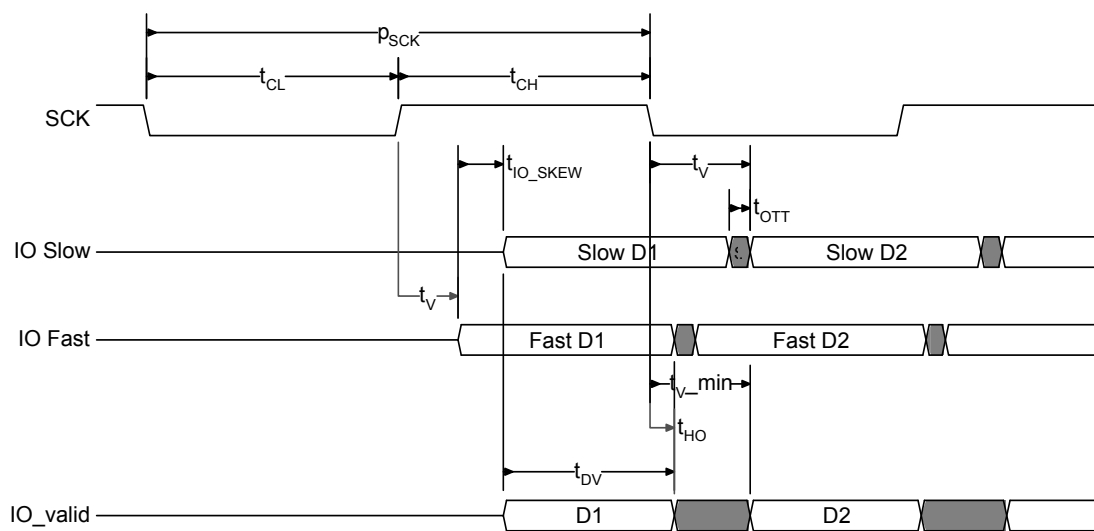


图12.17 SPI DDR数据有效窗口



注意:

1. t_{CLH} 是 t_{CL} 或 t_{CH} 中较小的值。
2. t_{O_SKEW} 是所有IO信号中最小值同最大 t_V (输出有效) 间的最大差值 (偏差)。
3. t_{OTT} 是每个IO上从一个有效数据值到下一个有效数据值的最大输出转换时间。
4. t_{OTT} 由系统级注意事项决定, 这些注意事项包括:
 - a. 存储器器件输出阻抗 (强驱动)。
 - b. IO上的系统级寄生电容 (通常为总线电容)。
 - c. 主存储控制器输入 V_{IH} 和 V_{IL} 电平, 在该电平下可以识别0到1和1到0的转换。
 - d. 例如, 假设上述注意事项导致存储器输出转换速率为 2 V/ns 而且主机需要 3 V 转换 (从1到0或从0到1), 那么 t_{OTT} 将为: $t_{OTT} = 3 \text{ V} / (2 \text{ V/ns}) = 1.5 \text{ ns}$ 。
 - e. t_{OTT} 不是赛普拉斯的测试要求。它的大小由系统决定, 并且系统设计师必须根据上述注意事项推导出该值。

12.5.3 DDR最小数据有效窗口

最小数据有效窗口 (t_{DV}) 如下计算:

例如, 假设: 66 MHz时钟频率 = 15 ns时钟周期, 而且DDR操作的占空比被指定为45%以上。

- $t_{CLH} = 0.45 * PSCK = 0.45 * 15 \text{ ns} = 6.75 \text{ ns}$
- $t_{O_SKEW} = 600 \text{ ps}$
- $t_{OTT} = 1.5 \text{ ns}$
- $t_{DV} = t_{CLH} - t_{O_SKEW} - t_{OTT}$
– $t_{DV} = 6.75 \text{ ns} - 600 \text{ ps} - 1.5 \text{ ns} = 4.65 \text{ ns}$
- $t_{V_min} = t_{HO} + t_{O_SKEW} + t_{OTT}$
– $t_{V_min} = 1.0 \text{ ns} + 600 \text{ ps} + 1.5 \text{ ns} = 3.1 \text{ ns}$

12.6 嵌入式算法性能表

表 12.6 双四线编程和擦除性能

符号	参数	最小值	典型值 (1)	最大值	单位
t_W	非易失性寄存器写入时间	–	145	750	ms
t_{PP}	页编程时间 (256字节)	–	300	1,200	μs
t_{BP1}	字节编程时间 (第一个字节) (3)	–	50	60	μs
t_{BP2}	其他字节编程时间 (第一个字节后) (3)	–	6	20	μs
t_{SE}	扇区擦除时间 (4 KB物理扇区)	–	50	200	ms
t_{HBE}	半块擦除时间 (32 KB物理扇区)	–	190	363	ms
t_{BE}	块擦除时间 (64 KB物理扇区)	–	270	725	ms
t_{CE}	芯片擦除时间 (S25FL128L)	–	70	180	s
t_{CE}	芯片擦除时间 (S25FL256L)	–	140	360	s

注意:

- 在以下条件下得到典型的编程和擦除时间: 温度 = 25C、 $V_{DD} = 3.0 \text{ V}$ 、一万个周期、一个棋盘式数据组合。
- 任何OTP编程命令的编程时间都与 t_{PP} 相同。这些命令包括IRPP 2Fh、PASSP E8h和PDLRNV 43h。
- 对于某页内第一个字节后的多个字节, $t_{BPN} = t_{BP1} + t_{BP2} * N$ (typical)和 $t_{BPN} = t_{BP1} + t_{BP2} * N$ (max), 其中N = 被编程的字节数量。

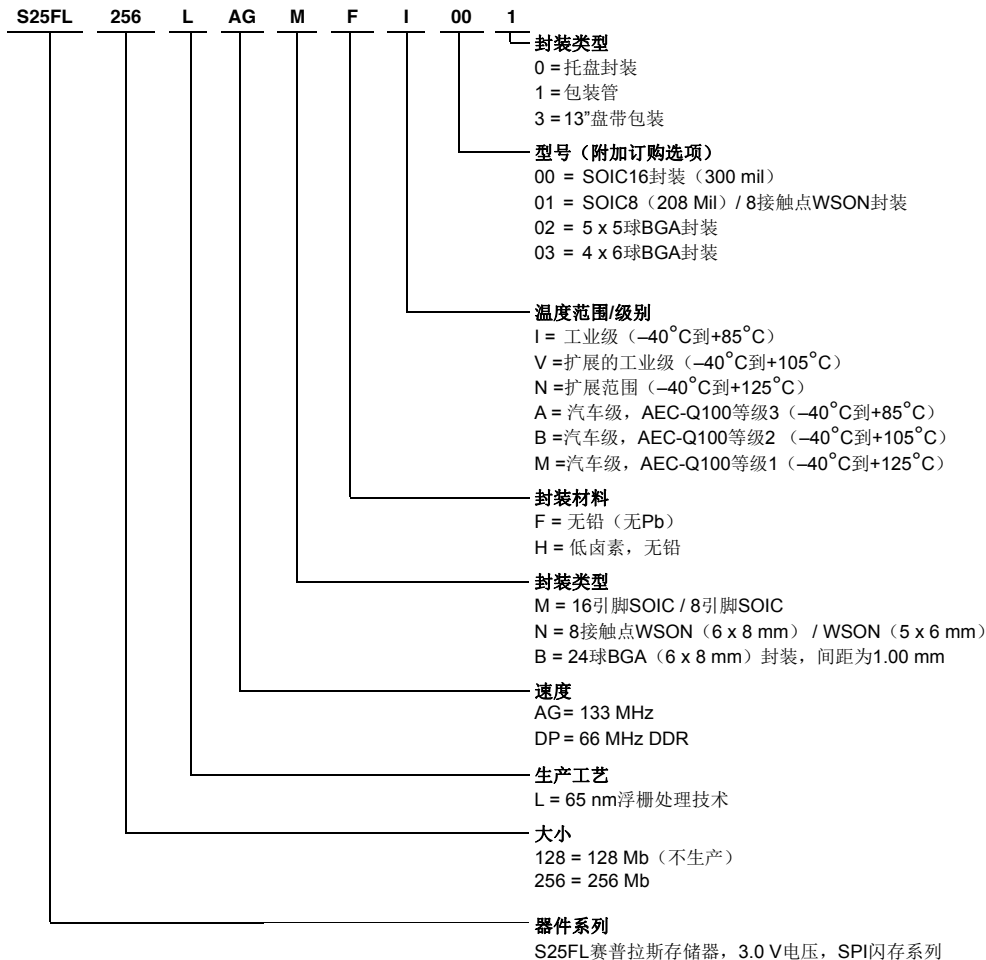
表 12.7 编程或擦除挂起交流参数

参数	典型值	最大值	单位	注释
挂起延迟时间 (t_{SL})	–	40	μs	从发出挂起命令到WIP位为0的时间。
恢复操作到下一次挂起操作的时间 (t_{RNS})	100	–	μs	发送下一个挂起命令所需的时间。

13. 订购信息

13.1 订购器件型号

订购器件型号由以下有效组合构成：



有效组合 — 标准

有效组合列出了计划针对该器件批量支持的配置。如要确认特定有效组合的可用性和了解最新推出的组合，请咨询您当地的销售办事处。

表 13.1 有效组合 — 标准

有效组合 — 标准					
基本订购器件型号	速度选项	封装和温度	模型编号	封装类型	封装标识
S25FL128L (非生产状态)	AG	MFI、MFV、MFN	00	0、1、3	(基本型号) + A+ (温度) + F+ (封装型号)
	AG	NFI、NFV、NFN	01	0、1、3	(基本型号) + A+ (温度) + F+ (封装型号)
	AG	BHI、BHV、BHN	02、03	0、3	(基本型号) + A+ (温度) + H+ (封装型号)
	DP	MFI、MFV、MFN	00	0、1、3	(基本型号) + D+ (温度) + F+ (封装型号)
	DP	NFI、NFV、NFN	01	0、1、3	(基本型号) + D+ (温度) + F+ (封装型号)
	DP	BHI、BHV、BHN	02、03	0、3	(基本型号) + D+ (温度) + H+ (封装型号)
S25FL256L	AG	MFI、MFV、MFN	00	0、1、3	(基本型号) + A+ (温度) + F+ (封装型号)
	AG	NFI、NFV、NFN	01	0、1、3	(基本型号) + A+ (温度) + F+ (封装型号)
	AG	BHI、BHV、BHN	02、03	0、3	(基本型号) + A+ (温度) + H+ (封装型号)
	DP	MFI、MFV、MFN	00	0、1、3	(基本型号) + D+ (温度) + F+ (封装型号)
	DP	NFI、NFV、NFN	01	0、1、3	(基本型号) + D+ (温度) + F+ (封装型号)
	DP	BHI、BHV、BHN	02、03	0、3	(基本型号) + D+ (温度) + H+ (封装型号)

有效组合 — 汽车级/AEC-Q100

下表列出了汽车级/符合AEC-Q100标准且准备批量上市的配置。该表会随着新组合的推出而不断更新。如要确认特定组合的供货情况并了解最新推出的组合，请咨询您当地的销售代表。

生产件批准程序（PPAP）支持仅针对AEC-Q100级产品。

要求符合ISO/TS-16949标准，并且使用于终端应用的产品必须符合PPAP的AEC-Q100等。对于非AEC-Q100级的产品，将不许投产或不提供符合ISO/TS-16949标准的全部材料。

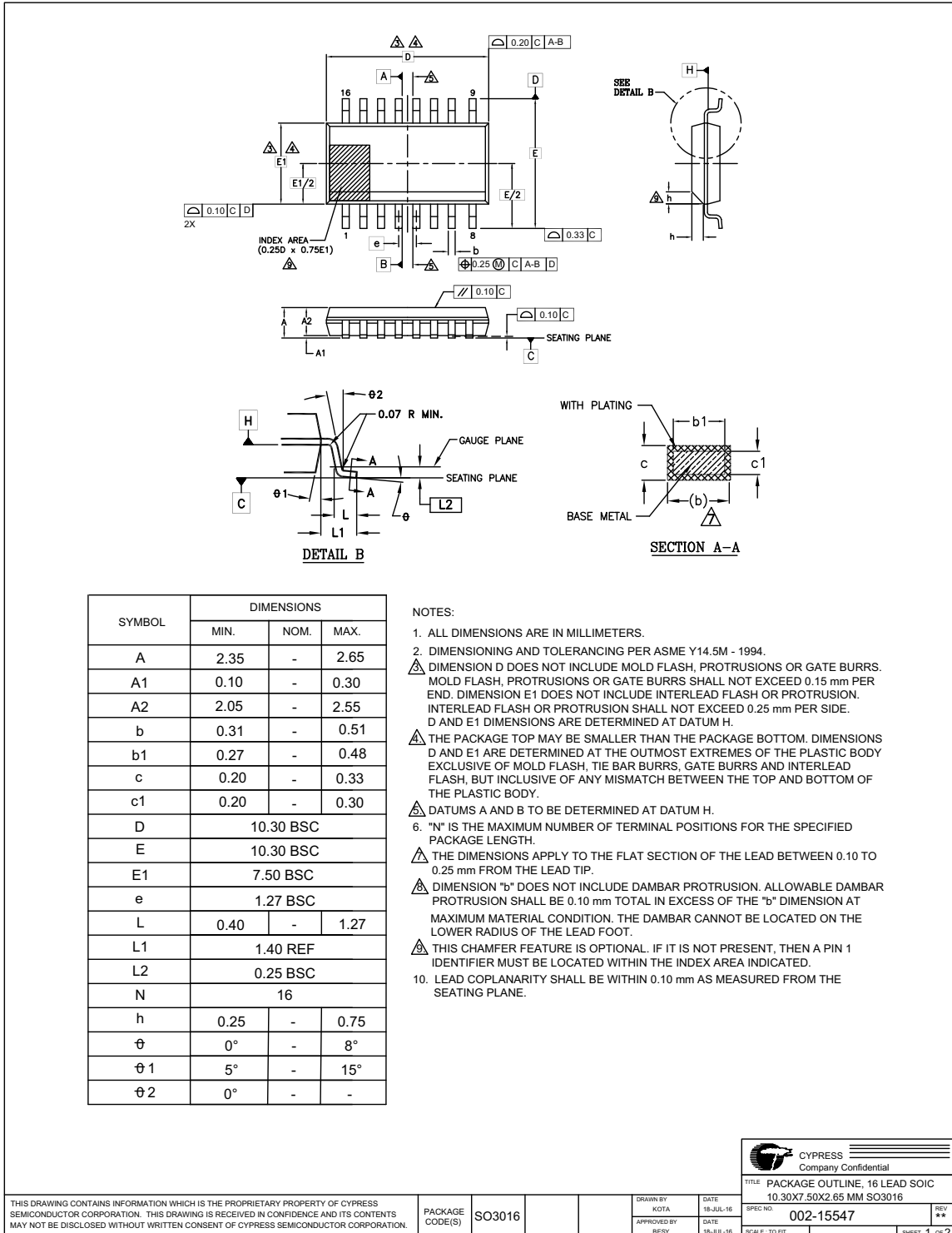
此外，对于不要求符合ISO/TS-16949标准的终端应用，也会提供无PPAP支持的EC-Q100级产品。

表 13.2 有效组合 — 汽车级 /AEC-Q100

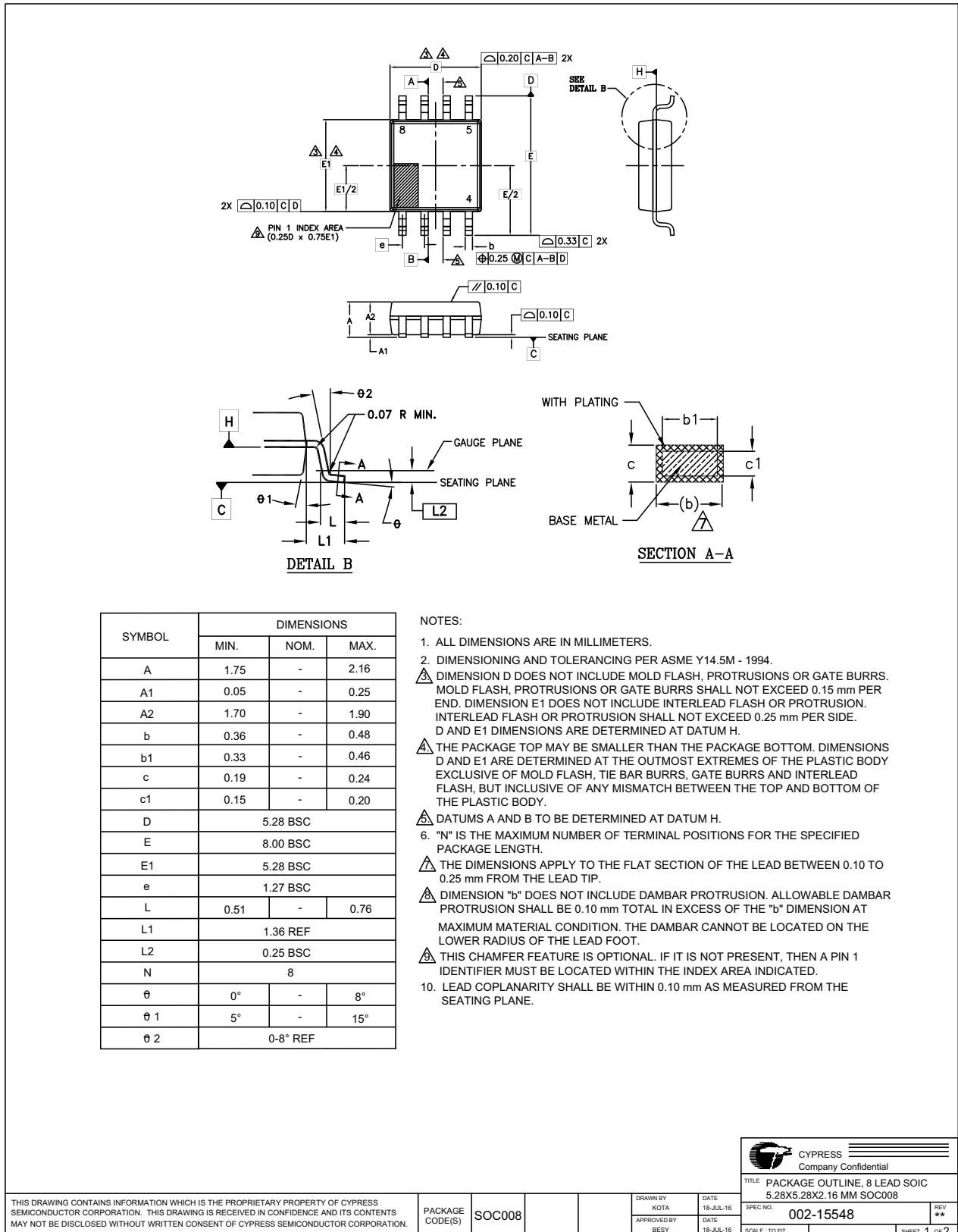
有效组合 — 汽车级/AEC-Q100					
基本订购器件型号	速度选项	封装和温度	模型编号	封装类型	封装标识
S25FL128L (不在生产中)	AG	MFA、MFB、MFM	00	0、1、3	(基本型号) + A + (温度) + F + (封装型号)
	AG	NFA、NFB、NFM	01	0、1、3	(基本型号) + A + (温度) + F + (封装型号)
	AG	BHA、BHB、BHM	02、03	0、3	(基本型号) + A + (温度) + H + (封装型号)
	DP	MFA、MFB、MFM	00	0、1、3	(基本型号) + D + (温度) + F + (封装型号)
	DP	NFA、NFB、NFM	01	0、1、3	(基本型号) + D + (温度) + F + (封装型号)
	DP	BHA、BHB、BHM	02、03	0、3	(基本型号) + D + (温度) + H + (封装型号)
S25FL256L	AG	MFA、MFB、MFM	00	0、1、3	(基本型号) + A + (温度) + F + (封装型号)
	AG	NFA、NFB、NFM	01	0、1、3	(基本型号) + A + (温度) + F + (封装型号)
	AG	BHA、BHB、BHM	02、03	0、3	(基本型号) + A + (温度) + H + (封装型号)
	DP	MFA、MFB、MFM	00	0、1、3	(基本型号) + D + (温度) + F + (封装型号)
	DP	NFA、NFB、NFM	01	0、1、3	(基本型号) + D + (温度) + F + (封装型号)
	DP	BHA、BHB、BHM	02、03	0、3	(基本型号) + D + (温度) + H + (封装型号)

14. 物理框图

14.1 SOIC 16引脚（300 mil）封装（SO3016）




14.2 8引脚SOIC (208 mil) 封装 (SOC008)



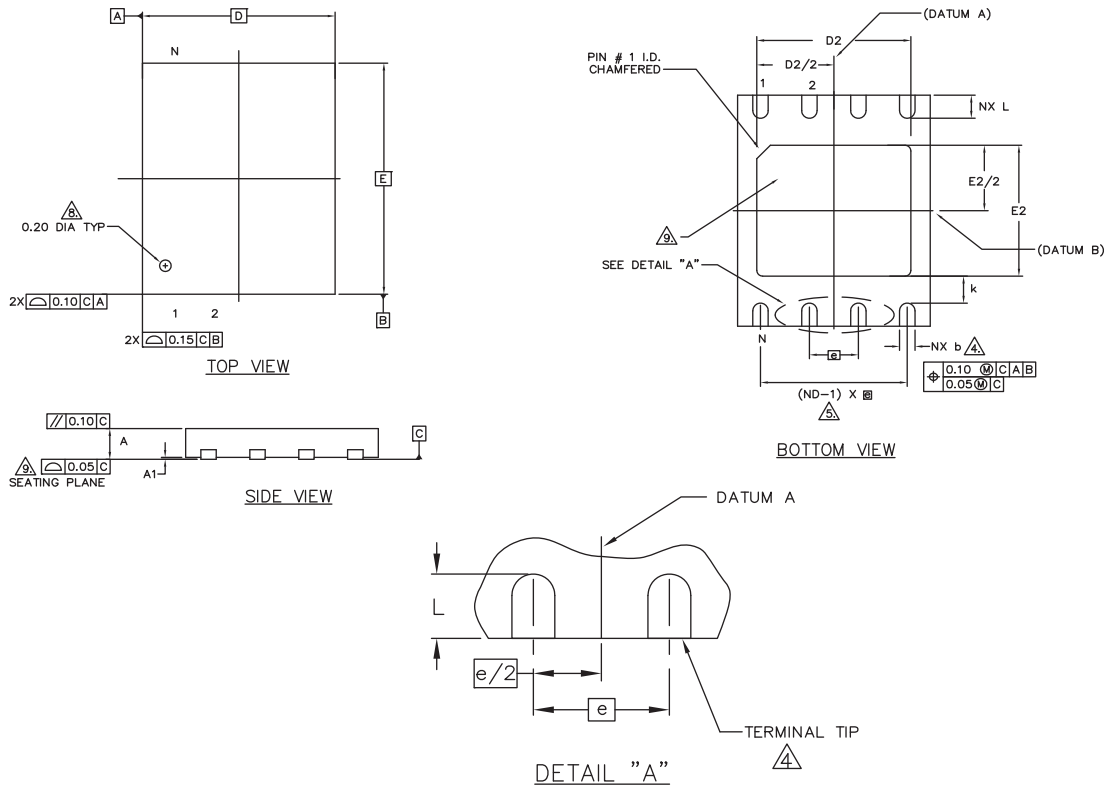
THIS DRAWING CONTAINS INFORMATION WHICH IS THE PROPRIETARY PROPERTY OF CYPRESS SEMICONDUCTOR CORPORATION. THIS DRAWING IS RECEIVED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED WITHOUT WRITTEN CONSENT OF CYPRESS SEMICONDUCTOR CORPORATION.

PACKAGE CODE(S) SOC008

DRAWN BY: KOTA
DATE: 18-JUL-16
APPROVED BY: BESY
DATE: 18-JUL-16

 <p>CYPRESS Company Confidential</p>			
<p>TITLE PACKAGE OUTLINE, 8 LEAD SOIC 5.28X5.28X2.16 MM SOC008</p>			
SPEC NO. 002-15548		REV **	
SCALE: 1:1		SHEET 1 OF 2	

14.3 无引脚并包含8个连接器的WSON，其大小为5 x 6 mm（WND008）



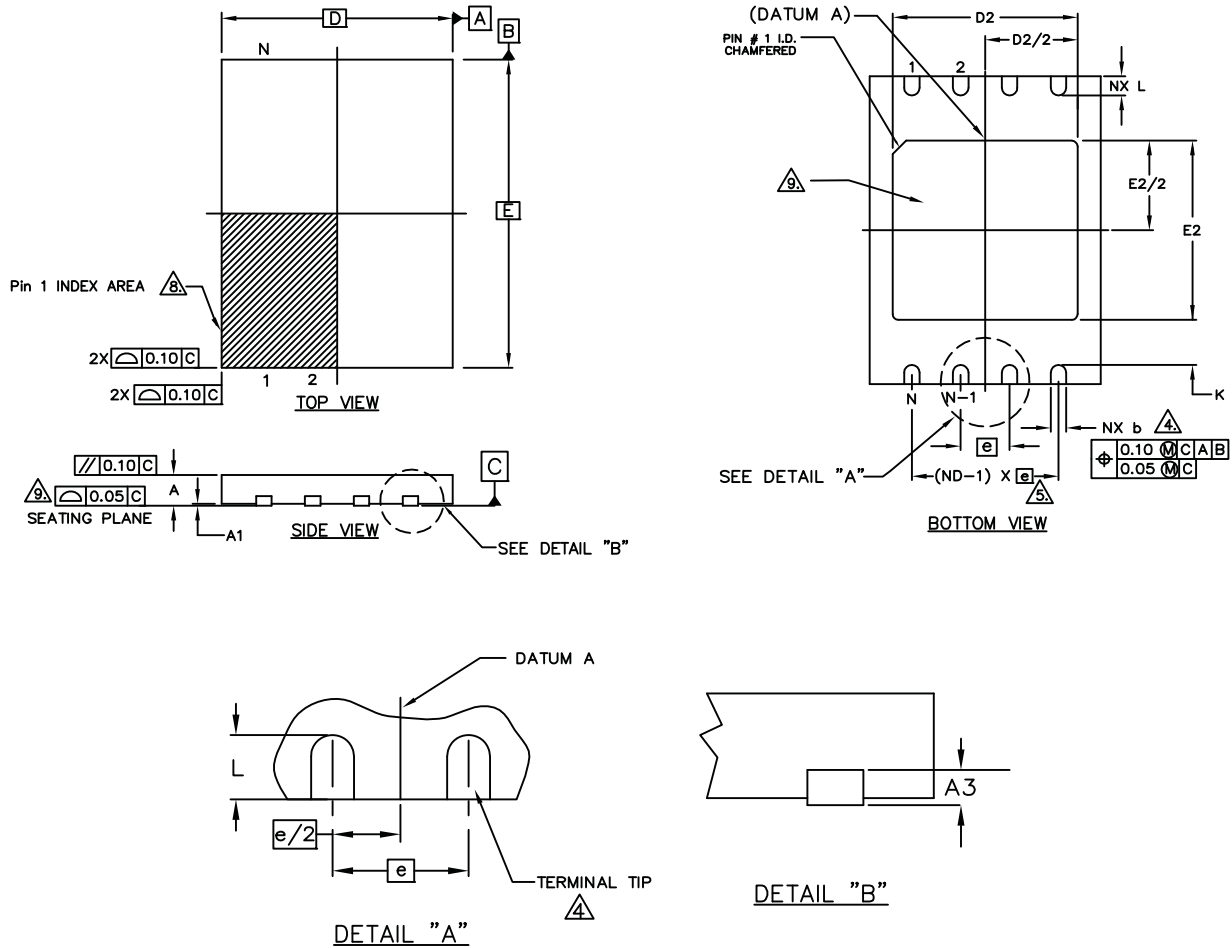
PACKAGE	WND008			
SYMBOL	MIN	NOM	MAX	NOTES
e	1.27 BSC.			
N	8			3
ND	4			5
L	0.55	0.60	0.65	
b	0.35	0.40	0.45	4
D2	3.90	4.00	4.10	
E2	3.30	3.40	3.50	
D	5.00 BSC			
E	6.00 BSC			
A	0.70	0.75	0.80	
A1	0.00	0.02	0.05	
K	0.20 MIN.			

NOTES:

1. DIMENSIONING AND TOLERANCING CONFORMS TO ASME Y14.5M - 1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. N IS THE TOTAL NUMBER OF TERMINALS.
4. DIMENSION "b" APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15 AND 0.30mm FROM TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION "b" SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
5. ND REFERS TO THE NUMBER OF TERMINALS ON D SIDE.
6. MAX. PACKAGE WARPAGE IS 0.05mm.
7. MAXIMUM ALLOWABLE BURR IS 0.076mm IN ALL DIRECTIONS.
8. PIN #1 ID ON TOP WILL BE LASER MARKED.
9. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

g1071 \ 16-038.30 \ 02.22.12

14.4 无引脚并包含8个连接器的WSON，其大小为6 x 8 mm (WNG008)



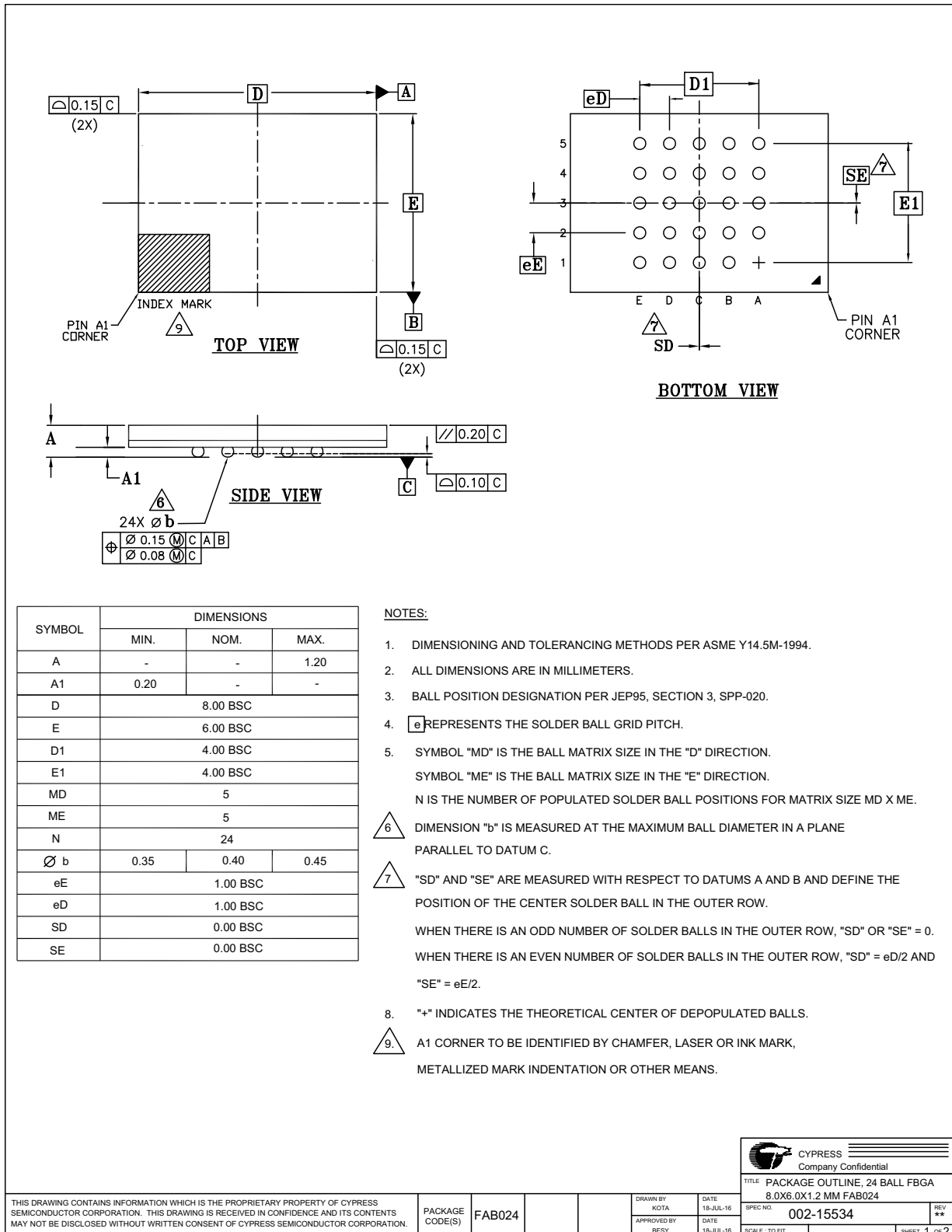
PACKAGE	WNG008			
SYMBOL	MIN	NOM	MAX	NOTE
e	1.27 BSC.			
N	8			3
ND	4			5
L	0.45	0.50	0.55	
b	0.35	0.40	0.45	4
D2	4.70	4.80	4.90	
E2	4.55	4.65	4.75	
D	6.00 BSC			
E	8.00 BSC			
A	0.70	0.75	0.80	
A1	0.00	0.02	0.05	
A3	0.20 REF			
K	0.20 MIN			

NOTES:

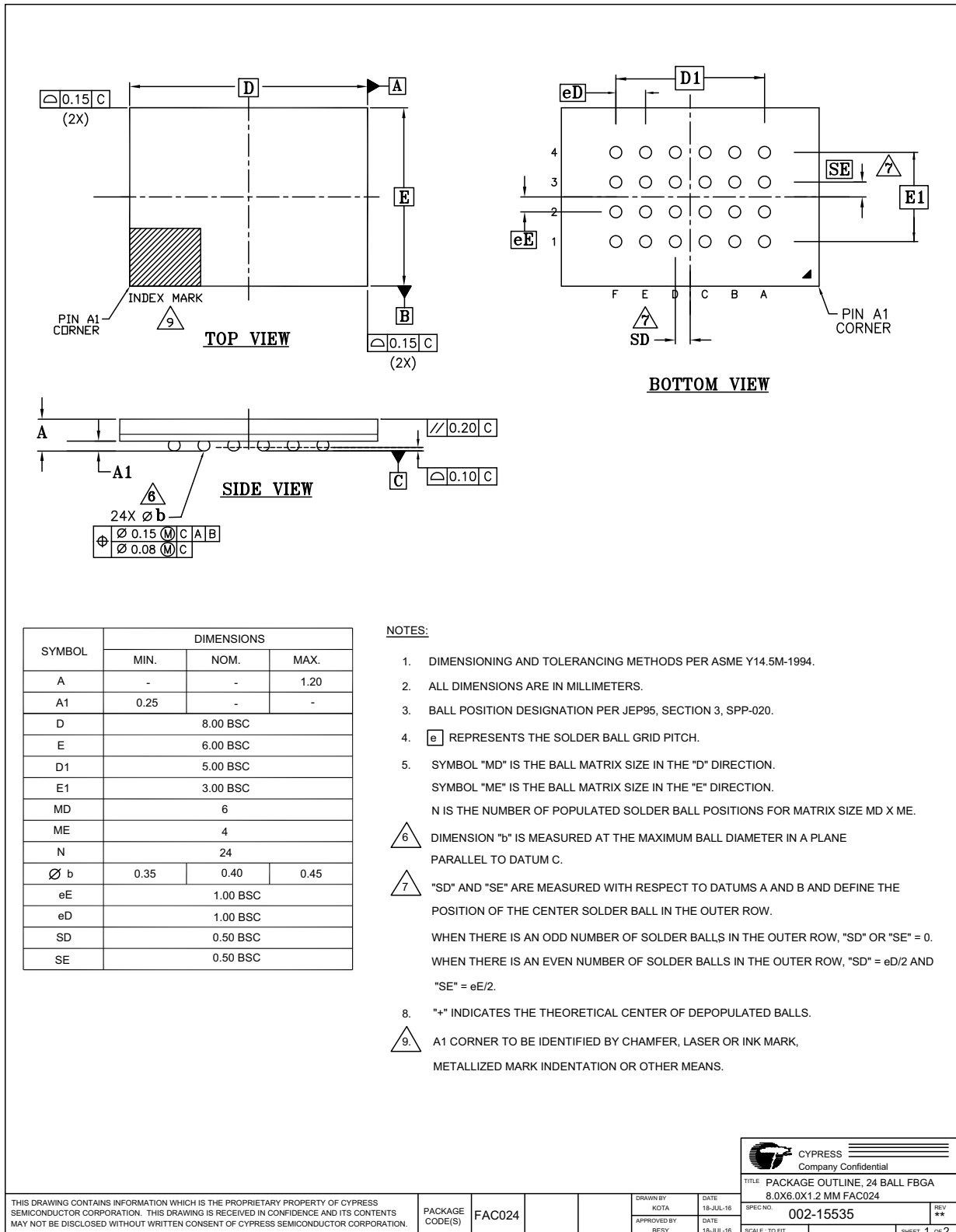
1. DIMENSIONING AND TOLERANCING CONFORMS TO ASME Y14.5M - 1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. N IS THE TOTAL NUMBER OF TERMINALS.
4. DIMENSION "b" APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15 AND 0.30mm FROM TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION "b" SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
5. ND REFERS TO THE NUMBER OF TERMINALS ON D SIDE.
6. MAX. PACKAGE WARPAGE IS 0.05mm.
7. MAXIMUM ALLOWABLE BURR IS 0.076mm IN ALL DIRECTIONS.
8. PIN #1 ID ON TOP WILL BE LOCATED WITHIN THE INDICATED ZONE.
9. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

g50311 \ 16-038.30 \ 03.25.14

14.5 球栅阵列24球型焊盘，其大小为6 x 8 mm (FAB024)



14.6 球栅阵列24球型焊盘，其大小为6 x 8 mm (FAC024)



15. 其他资源

15.1 术语表

BCD	二进制十进数。每四位代表一个十进制数的值。
命令	CS# 为低电平时，在一个周期内主机系统与存储器间传输的所有信息。信息包括指令（有时称为操作代码或操作码）和全部所需地址、模式位、延迟周期或数据。
DDP	双晶片封装 = 两个晶片在同一个封装中堆叠，以增加单个封装的存储容量。有时被称为多芯片封装（MCP）。
DDR	双倍数据速率 = 输入和输出在 SCK 的每一个边沿上被锁存。
闪存	它是一种电可擦可编程只读存储器（EEPROM）的名称，能够同时擦除大量的存储器位，这样可使擦除速度比早期的EEPROM快得多。
高电平	高于 V_{IH} 的信号电压电平，或二进制“1”所代表的逻辑电平。
指令	一个8位代码，它指明了命令需要执行的功能（有时称为操作代码或操作码）。指令始终是主机系统传送给存储器的命令中最靠前的8位。
低电平	$\leq V_{IL}$ 的信号电压电平，或由二进制“0”所代表的逻辑电平。
LSB	最低有效位 = 寄存器或数据值位组中单位数值最小的位，通常是最右边的位。
MSB	最高有效位 = 寄存器或数据值位组中单位数值最大的位，通常是最左边的位。
N/A	不适用。某个值不适用于所述条件。
非易失性	无需使用电源，但仍能保持存储器中的数据。
OPN	订购器件型号 = 指出存储器件类型、容量、封装、工厂非易失性配置等信息的字母数字，用于选择所需器件。
QPI	四线外设接口
页	256字节大小的对齐数据组。
PCB	印刷电路板
寄存器位参考	格式为：Register_name[bit_number]或Register_name[bit_range_MSB: bit_range_LSB]
扇区	擦除单位大小；根据器件模型和扇区位置，扇区大小可为4 KB、32 KB或64 KB
SDR	单倍数据速率 = 输入和输出分别在 SCK 的上升和下降沿上被锁存。
写入	该操作用于修改易失性或非易失性寄存器位中或非易失性闪存中的数据。改变非易失性数据时，会对不发生改变的非易失性数据进行擦除和重新编程操作，这样可以使单一的写入操作改变非易失性数据，可以使用同样的方式来改变易失性数据。非易失性数据显示在主机系统内，并会经过单一的写入命令得到更新，这便无需分开使用不同命令来擦除和编程各个相邻但无受影响的数据。

15.2 赛普拉斯产品路线图链接

www.cypress.com/Flash-Roadmap

15.3 软件链接

www.cypress.com/software-and-drivers-cypress-flash-memory

15.4 应用笔记链接

www.cypress.com/cypressappnotes

13. 文档修订记录

文档标题: S25FL256L/S25FL128L, 256 Mbit (32 Mbyte) /128 Mbit (16 Mbyte), 3.0 V FL-L Flash存储器				
文档编号: 002-03823				
版本	ECN编号	变更者	提交日期	变更说明
**	5001771	YOQI	11/05/2015	本文档版本号为Rev**, 译自英文版002-00124 Rev**。
*A	5174697	YOQI	03/17/2016	本文档版本号为Rev*A, 译自英文版002-00124 Rev*A。
*B	5519738	RZZH	11/15/2016	本文档版本号为Rev*B, 译自英文版002-00124 Rev*C。

销售、解决方案和法律信息

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、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	cypress.com/automotive
时钟与缓冲区	cypress.com/clocks
接口	cypress.com/interface
物联网 (IoT)	cypress.com/iot
照明与电源控制	cypress.com/powerpsoc
存储器	cypress.com/memory
PSoC	cypress.com/psoc
触摸感应	cypress.com/touch
USB控制器	cypress.com/usb
无线/射频	cypress.com/wireless

PSoC®解决方案

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#)

赛普拉斯开发者社区

[论坛](#) | [项目](#) | [视频](#) | [博客](#) | [培训](#) | [组件](#)

技术支持

cypress.com/support

赛普拉斯半导体公司，2015-2016年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC (“赛普拉斯”) 的财产。本文件，包括其包含或引用的任何软件或固件 (“软件”)，根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性的、非独家且不可转让的如下许可 (无再许可) (1) 在赛普拉斯特软件著作权项下的下列许可 (一) 对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和 (二) 仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供 (无论直接提供或通过经销商和分销商间接提供)，和 (2) 在被软件 (由赛普拉斯公司提供，且未经修改) 侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用者应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统 (包括急救设备和手术植入物)、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途 (“非预期用途”)。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担任何全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion徽标，及上述项目的组合，WICED，及PSoC、CapSense、EZ-USB、F-RAM和Traveo应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。