

一、概述

SPI (Serial Peripheral Interface, 串行外设接口)是一种工业协议,广泛用于各种设备的连接。它还有很多优点,比如低成本、低引脚数、紧凑等。在访问 NAND 闪存时,很容易采用其性质。

SPI NAND 是一种 SPI 接口且易于使用的 NAND 闪存。与一般 FLASH 存储器具有高引脚数和复杂的访问方式相比, SPI NAND 具有易于访问的工业标准接口。此外, SPI NAND 还内置了内存块管理。它提供了一种非常直观的访问方式,而无需担心闪存管理,如坏块或下降。SPI NAND 还扩展了 SPI 接口的访问数据位宽度。它提供了一位模式和四位模式,分别代表 SPI 接口上的 1 位和 4 位访问。

One bit mode(标准)- 数据被一个输入和一个输出引脚访问。

Quad bit mode - 数据通过四个数据引脚访问,它们是双向的。

由于 SPI 的性质, SPI NAND 有 8 个引脚,包括两个电源引脚,它们在下一章中列出。SPI NAND 也有不同的页面大小,这是 NAND 闪存的访问单元。页面大小根据不同的模型进行分类,具有不同的性能。块大小是 NAND 闪存的擦除单元,它由数十或数百页组成。NAND 闪存的总大小由块数、页数和页大小计算。此外, SPI NAND 还具有快速访问用户数据的缓存。用户可以存储数据到 SPI NAND 无需编程或读取数据从 SPI NAND 缓存。

SPI NAND 闪存还内置 ECC 引擎,因此它提供了保护用户数据的能力。用户可选择内部 ECC 或自定义 ECC。

自定义 ECC - 当用户将数据写入 SPI NAND 时,采用 ECC 编码。编码的奇偶校验数据位于用户编程的数据中。

内部 ECC - 用户编程数据无需处理 ECC。当数据被编程时, SPI NAND 将保护奇偶性添加到 NAND 闪存中。

如上所述, SPI NAND 提供了许多易于使用,紧凑和廉价的存储选择方式。它为用户提供了一种简单的方法来使用存储,而且不需要考虑 FLASH 管理。

二、特性说明

- 标准, 双线和四线 SPI
 - 标准: SCLK, CS, SI, SO, WP, HOLD
 - 双线: SCLK, CS, SI00, SI01, WP, HOLD
 - 四线: SCLK, CS, SI00, SI01, SI02, SI03
 - 时钟频率: 104MHz (最大值)

- 增强的访问性能
 - 可快速随机读取的 4KB 高速缓存
 - 读缓存和写缓存

- NAND 闪存接口
 - 先进的动态 ECC, 提高可靠性
 - 带有 ECC 的内部数据页面移动
 - 支持数据随机化
 - 支持编程故障恢复
 - 自动省电操作
 - 高于 100000 次擦除/编程次数

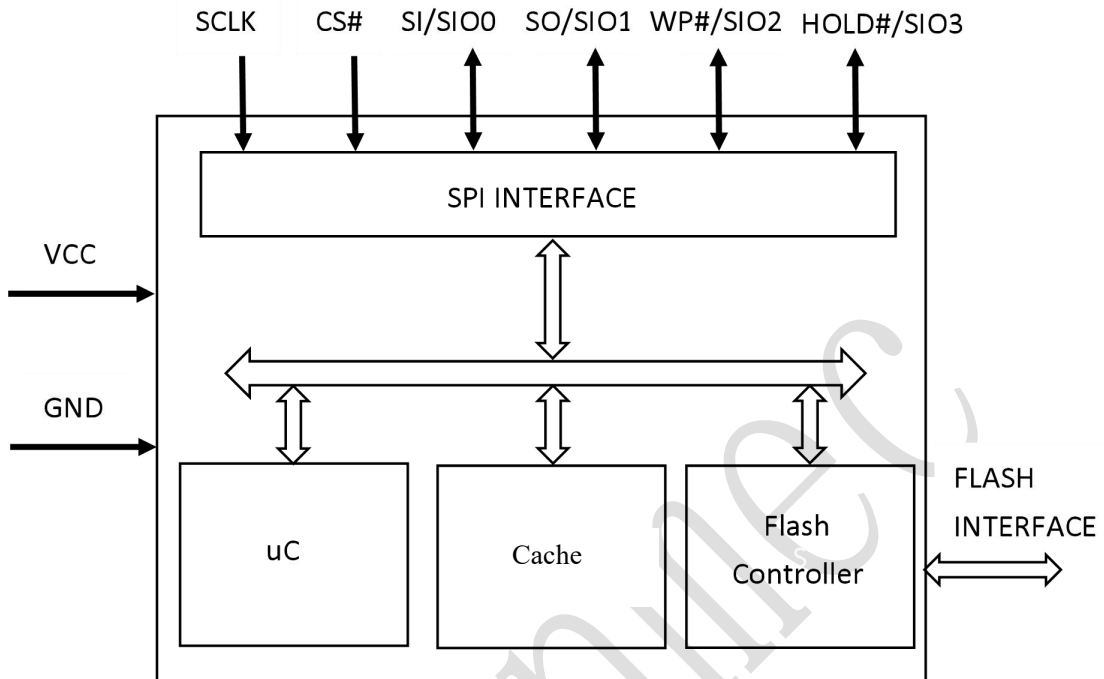
- 电源电压和功耗
 - 电压范围: 2.7V ~ 3.6V
 - 工作电流: 40mA (最大值)
 - 待机电流: 70uA (最大值)

- 软/硬件写保护
 - 通过软件对全部或选中的寄存器进行写保护
 - 通过 WP 引脚使能或失能保护

- 安全性
 - 16KB OTP 区域

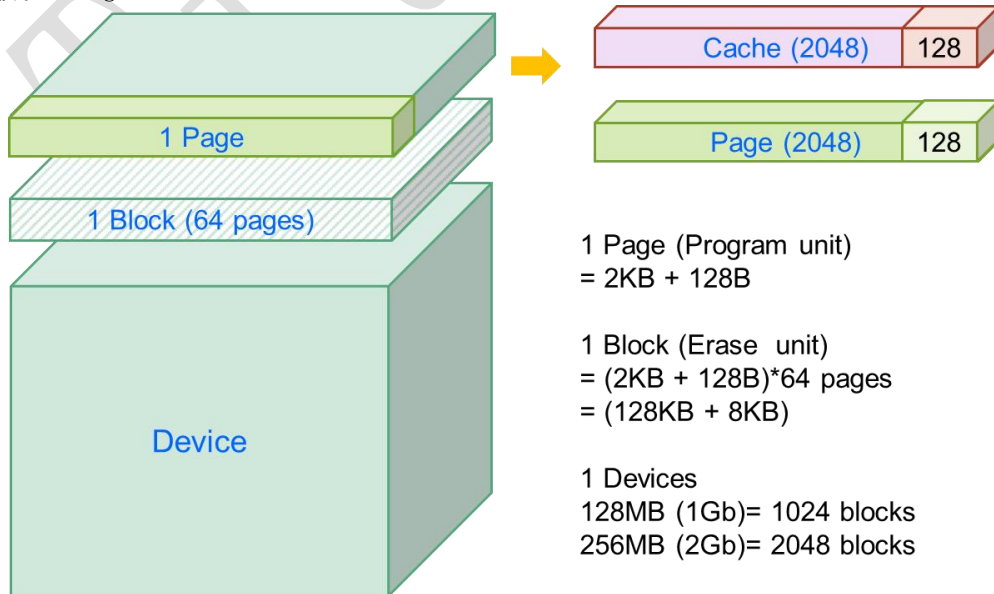
- 温度范围
 - 工作温度: -40°C to +85°C
 - 储存温度: -65°C to +150°C

三、块结构



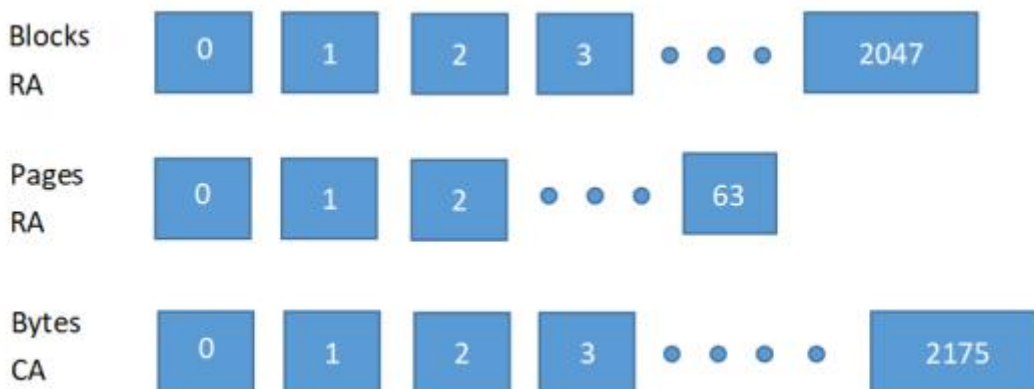
四、闪存结构

举例：假设 1Page= 2KB+128B



五、内存地址映射

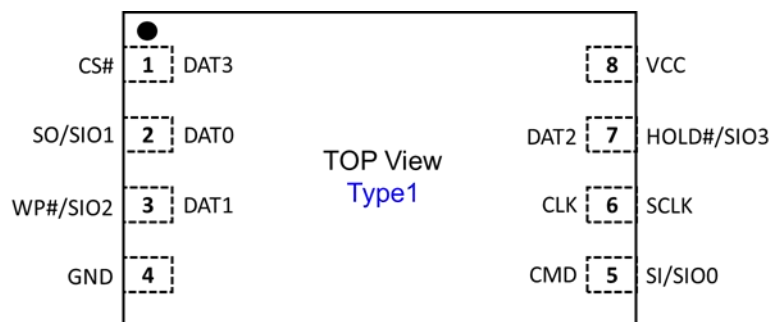
以下是举例。



说明:

1. RA: 行地址用于寻址页或块位置
RA[5:0]: 页面范围 0 ~ 63
RA[16:6]: 块范围 0 ~ 2047
2. CA: 列地址用于寻址字节位置。CA 取值范围 0~2175。

六、管脚定义



七、管脚功能定义

管脚名称	输入输出类型	说明
CS#	输入	SPI 片选
SO/SIO1	输入/输出	SPI 输出 (one bit mode) SPI 数据输入输出 1 (Quad mode)
WP#/SIO2	输入/输出	SPI 写保护 (one bit mode) SPI 数据输入输出 2 (Quad mode)
GND	地	地
SI/SIO0	输入/输出	SPI 输入 (one bit mode) SPI 数据输入输出 0 (Quad mode)
SCLK	输入	SPI 时钟
HOLD#/SIO3	输入/输出	SPI 停止 (one bit mode) SPI 数据输入输出 3 (Quad mode)
VCC	电源	电源

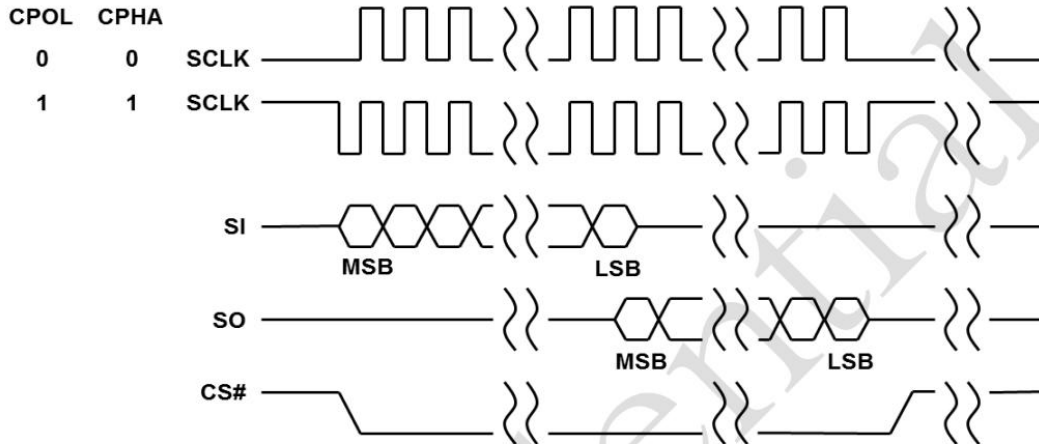
八、SPI 协议

SPI NAND 支持 2 种 SPI 模式：

- CPOL = 0, CPHA = 0 (模式 0)
- CPOL = 1, CPHA = 1 (模式 3)

在模式 0 和模式 3 中，SCLK 的上升沿锁存输入数据，SCLK 的下降沿输出数据有效。

下图展示的是模式 0 的时序



SCLK	SCLK 为 SPI NAND 提供接口时钟 SCLK 上升沿锁存地址、数据和命令。SCLK 的下降沿数据输出到 SO 上。
CS#	当 CS# = 0 时，设备处于使能模式 当 CS# = 1 时，设备处于失能模式，SO 为高阻态

标准 SPI (1 位模式)：

标准 SPI 的 4 信号总线：系统时钟(SCLK)，片选 (CS#)，串行数据输入(SI) 和串行数据输出(SO)。

双线 SPI：

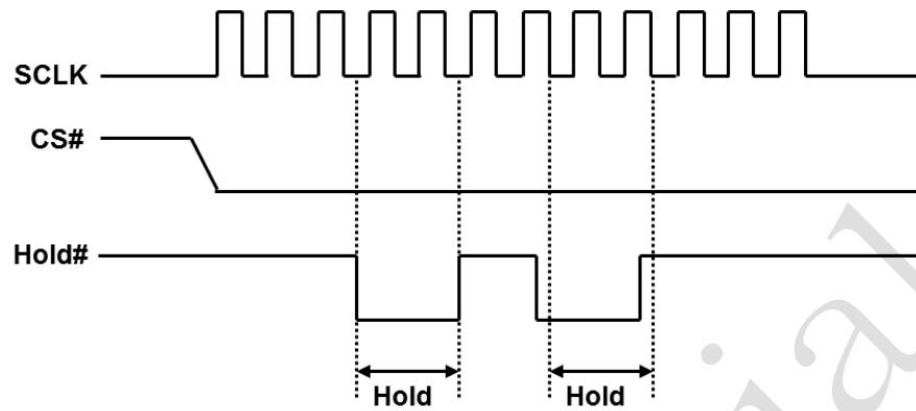
SPI 工作在这种模式下可以用 x2 数据总线宽度的传输数据。这些命令允许数据以标准模式 2 倍的速度传输。SI 和 SO 成为双向 I/O 引脚：SI00 和 SI01。

四线 SPI：

在这种模式下，HOLD#和 WP#成为两个额外的数据访问引脚，使 SPI 允许传输比标准模式多 4 倍的数据。在此模式下共有 4 个数据引脚，SI 表示 SI00，SO 表示 SI01，WP#表示 SI02，HOLD#表示 SI03，而且都是双向的。

保持模式：

HOLD#信号为低时，停止与设备的任何串行通信，但不停止正在进行的写状态寄存器、编程或擦除操作。



HOLD	<p>如果 HOLD#下降沿时 SCLK 也为低，则保持模式从 HOLD #的下降沿开始。 如果 HOLD#下降沿时 SCLK 为高，则保持模式在 SCLK 的下一个下降沿后开始</p>
------	---

写保护模式：

写保护(WP#)提供硬件保护模式。WP#可以防止块锁定位(BP0、BP1 和 BP2)被覆盖。如果 BRWD 位设置为 1,WP#为低，则不能更改块保护位。

九、命令列表

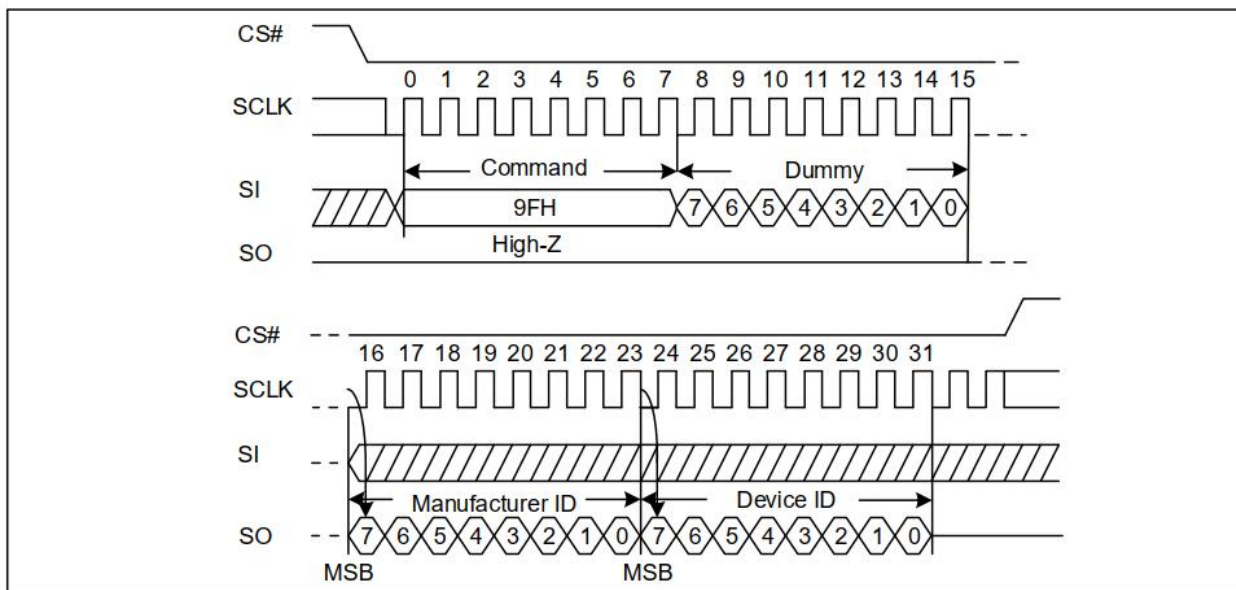
命令名称	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte N
写使能	06H					
写失能	04H					
获取特性	0FH	A7-A0	(D7-D0)			
设置特性	1FH	A7-A0	(D7-D0)	dummy		
页读取 (到缓存)	13H	A23-A16	A15-A8	A7-A0		
从缓存快速读取	03H/0BH	A15-A8	A7-A0	dummy	(D7-D0)	
从缓存读取 x2	3BH	A15-A8	A7-A0	dummy	(D7-D0)	
从缓存读取 x4	6BH	A15-A8	A7-A0	dummy	(D7-D0)	
从缓存读取 (双 I/O)	BBH	A15-A0	dummy	(D7-D0)		
从缓存读取 (4 I/O)	EBH(3)	A15-A0	(D7-D0)			
读 ID	9FH	dummy	(DID)	(DID)		
编程加载	02H	A15-A8	A7-A0	(D7-D0)	Next Byte	Byte N
编程加载 x4	32H	A15-A8	A7-A0	(D7-D0)	Next Byte	Byte N
编程执行	10H	A23-A16	A15-A8	A7-A0		
编程加载随机数据	84H(2)	A15-A8	A7-A0	(D7-D0)	Next Byte	Byte N
编程加载随机数据 x4	C4H	A15-A8	A7-A0	(D7-D0)	Next Byte	Byte N
编程加载随机数据 (4 I/O)	72H(2)(3)	A15-A0	(D7-D0)	Next Byte	Byte N	
复位	FFH					

注释:

1. dummy 是所有字节单位, 而不是时钟周期。例如, dummy 在标准模式是 1 字节 8 时钟周期, 在双线模式是 1 字节 4 时钟周期, 在四线模式是 1 字节 2 时钟周期。
2. 这些命令仅用于内部数据移动。
3. 执行这些命令前 QE 位需要先置位。

十、读 ID (9FH)

读 ID 命令用于识别 SPI NAND 的部件类别。用户可以使用此命令检查 ID 是否与自己的应用匹配。



ID 表

PART NO	MID	DID15 - DID0
TM1F512UAI	3Dh	0030h
TM1F01GUAI	3Dh	0031h
TM1F02GUAI	3Dh	0032h
TM1F04GUAI	3Dh	0034h

十一、写使能/失能

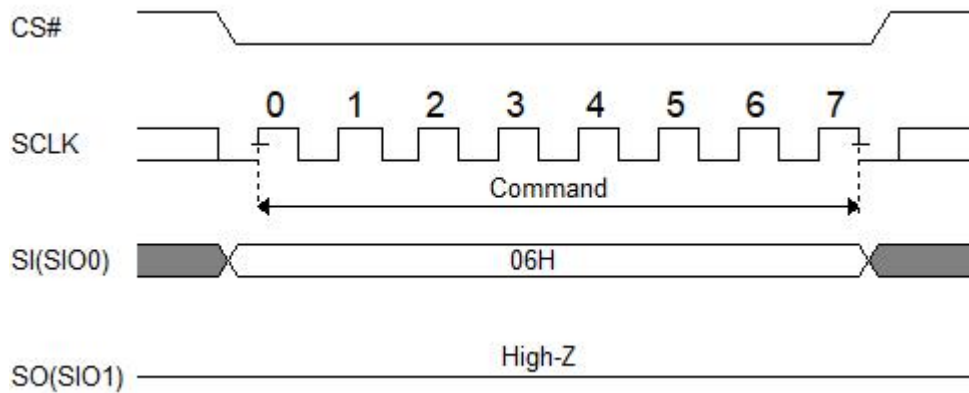
(1) 写使能 (06H)

在开始编程 SPI FLASH 之前，应该执行写使能。这是为了保护 SPI NAND 免受泄漏和一些未知的损坏。

写使能是在以下命令之前发出的

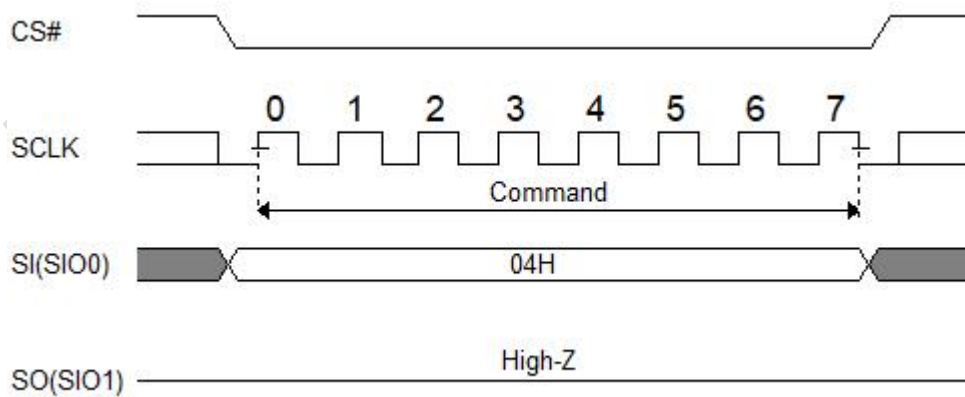
- 页编程
- OTP 编程/OTP 保护
- 块擦除

执行复位命令后，写使能会被清除。



(2) 写失能 (04H)

要重置“写使能”命令，用户可以下发“写失能”命令。



十二、SPI NAND 特性

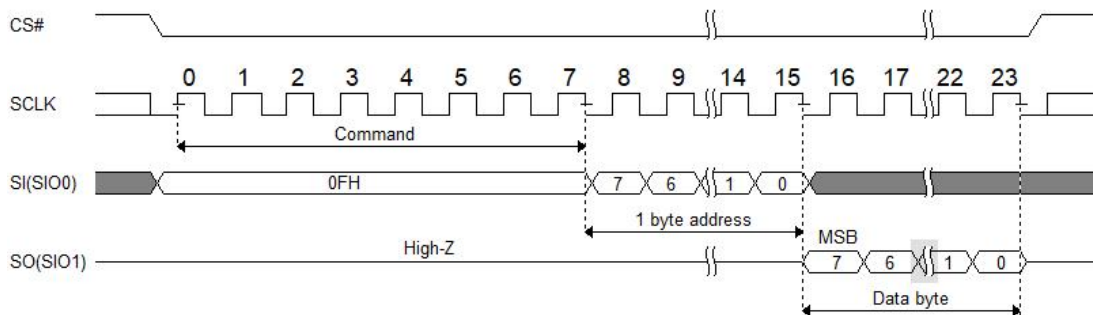
使用获取特性 (0FH) 和设置特性 (1FH) 命令可访问 SPI NAND 特性。这些命令用于监控设备状态，改变设备行为。这些命令使用一个字节的特性地址来确定要更改或读取哪个特性。

支持的特性如下所示

Register	Addr	7	6	5	4	3	2	1	0
Protection	AOH	BRWD	Reserved	BP2	BP1	BPO	INV	CMP	Reserved
Featured	BOH	OTP_PRT	OTP_EN	Reserved	ECC_EN	Reserved	Reserved	Reserved	QE
Status	COH	Reserved	ECCS2	ECCS1	ECCS0	P_FAIL	E_FAIL	WEL	OIP
Strength	DOH	Reserved	DS_S1	DS_S0	Reserved	Reserved	Reserved	Reserved	Reserved

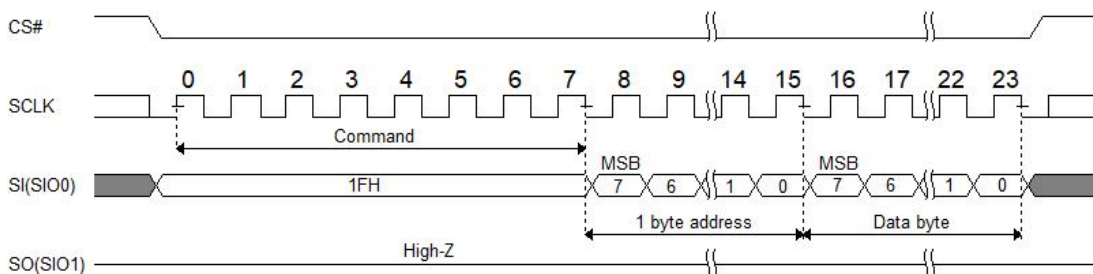
(1) 获取特性 (0FH)

获取特性 (0FH) 后面跟着一个字节的数据来监控内部寄存器的状态。



(2) 设置特性 (1FH)

设置特性 (1FH) 发出一个字节的数据来控制 SPI NAND 的内部寄存器。



十三、读取

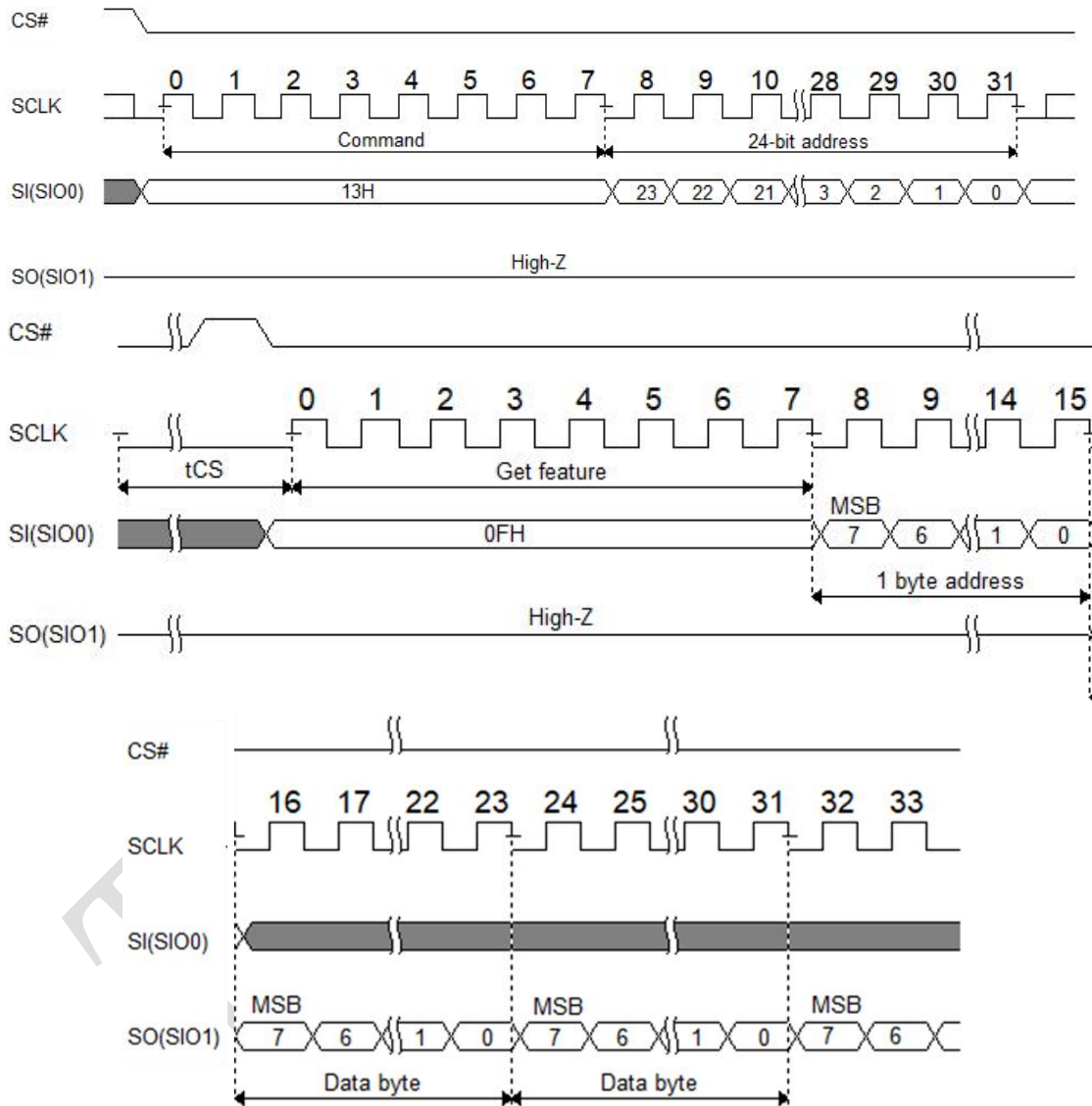
读取命令用于为 SPI 检索数据。用户可以根据应用场景发出以下命令，以不同的方式获取数据。

- 13H (页读取到缓存)
- 0FH (获取特性命令用来读取状态)
- 03H or 0BH (从缓存读取数据) / 3BH(从缓存读取数据 x2) / 6BH(从缓存读取数据 x4) / BBH (双线方式从缓存读取数据) / EBH(四线方式从缓存读取数据)

页地址有 24 位宽，可用于分配所需地址。在块/页寻址完成之后，设备开始从主阵列传输到缓存寄存器，并且在 tRD 时间内处于繁忙状态。在此期间，可以发送获取特性(0FH)命令来监视状态。当页面读取命令完成后，可以通过任意的数据读取命令从缓存中读出所需的数据。一旦达到一个页面的最大字节，下面的输出数据将循环，直到 CS#被释放到高。

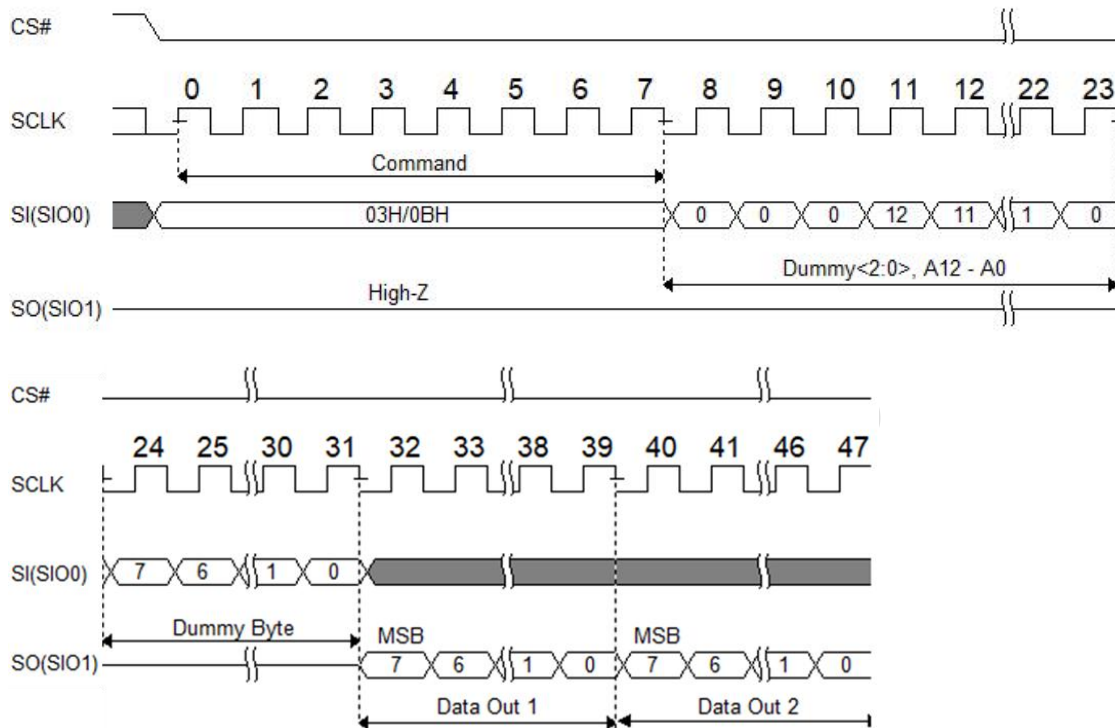
(1) 读到缓存 (13H)

读到缓存主要用于将 NAND 闪存中的数据移动到控制器的缓存中。当用户发出此命令时，控制器将地址转换为 NAND 地址并发送到 NAND 接口以定位用户数据。一旦数据被定位，它将从 NAND 转移到控制器缓存中。



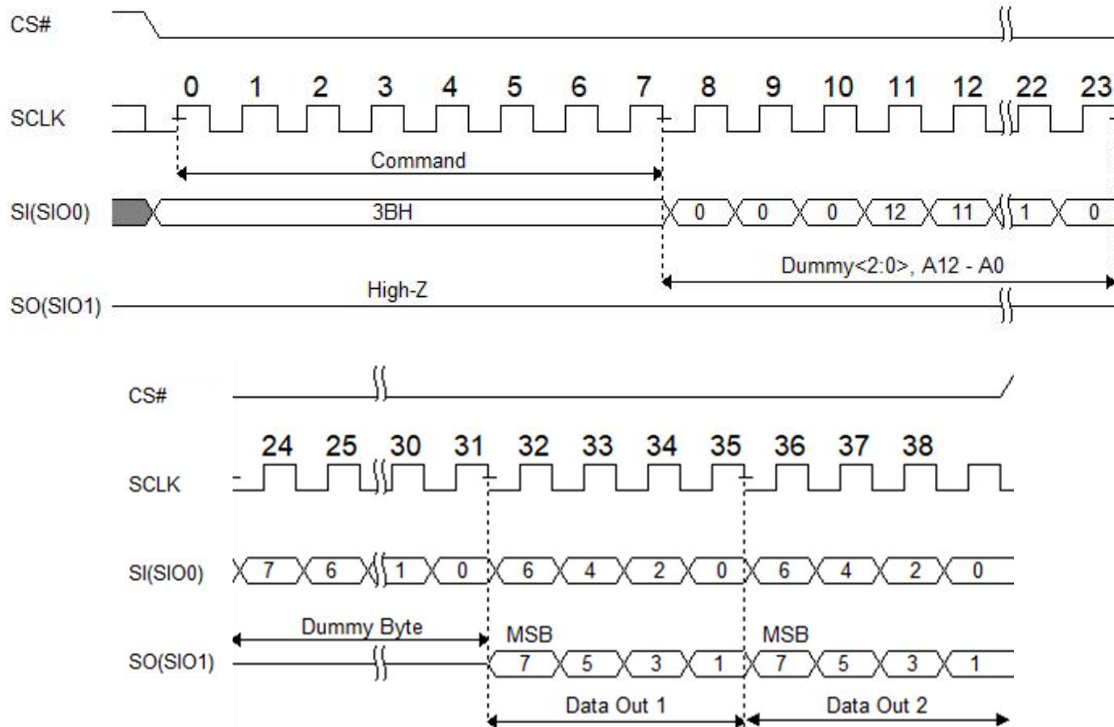
(2) 从缓存读取 (03H or 0BH)

SPI NAND 中的缓存可以看作是要访问的缓冲区。用户可以使用缓存来存储或检索数据。一旦数据被读取到缓存中（用户应该注意它），可以使用从缓存读取命令通过 SPI 接口获取缓存中的数据。建议采用字节对齐的方式获取缓存数据，以防止出现不可预测的情况。在休眠模式下缓存将被清空。



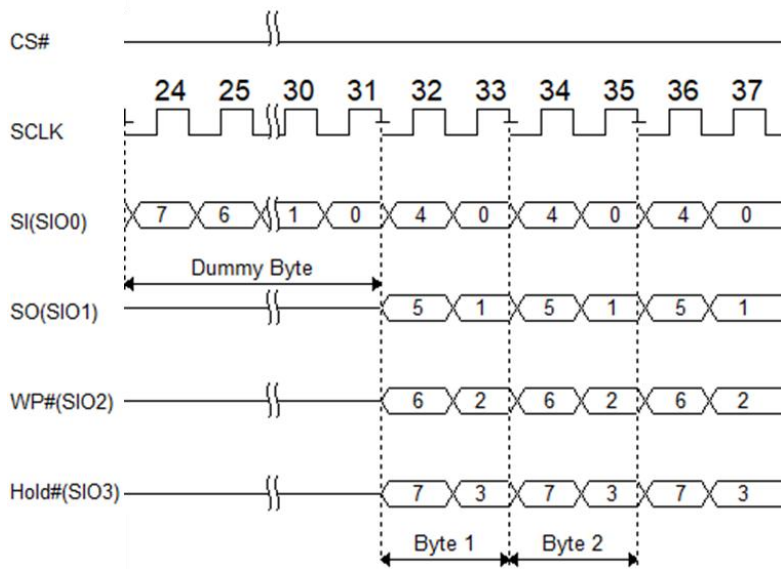
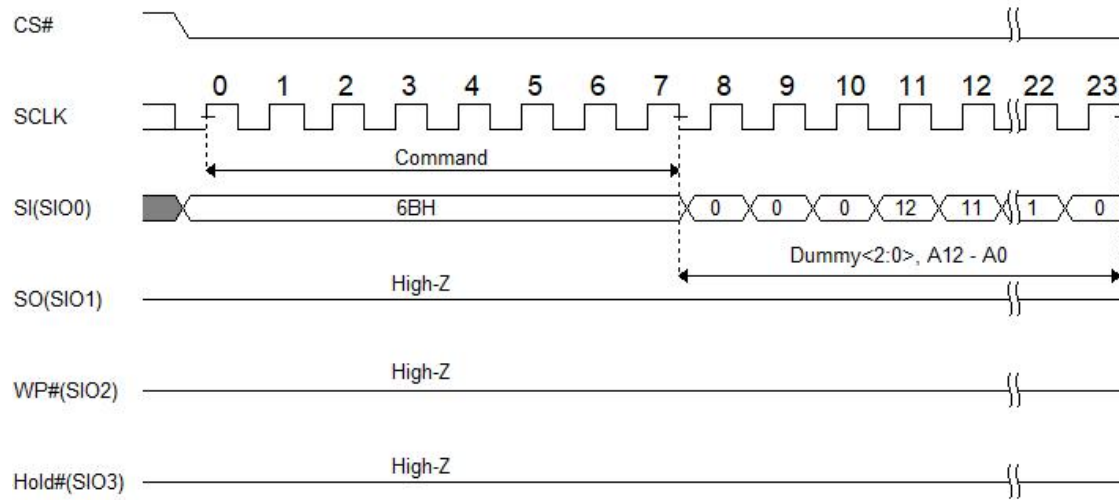
(3) 从缓存读取 x2 (3BH)

为了提高总线效率，SPI NAND 提供一种以两倍总线宽度访问缓存的方法。在这种模式下，SI 和 SO 被看作 SI00 和 SI01，这意味着数据可以在这两个引脚中输入或输出。因此，数据速率将是标准模式的两倍。



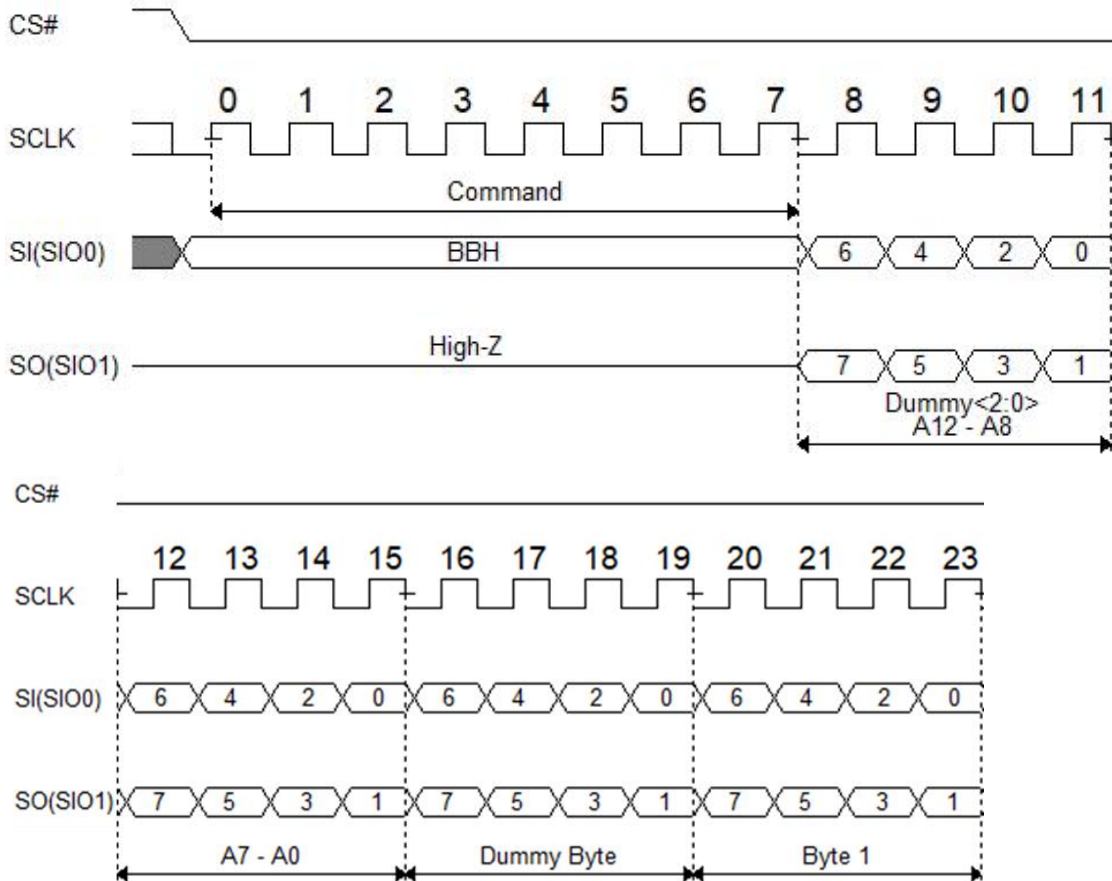
(4) 从缓存读取 x4 (6BH)

访问 SPI NAND 的最终方法是使用所有数据引脚。为了最大限度地利用数据引脚，SPI NAND 可以通过其所有引脚访问缓存中的数据。换句话说，它将数据速率提高了 4 倍。为了实现此功能，WP#和 HOLD#引脚将作为数据引脚，HOLD 和 WP 功能暂时不可用。



(5) 从缓存中读取 双 IO (BBH)

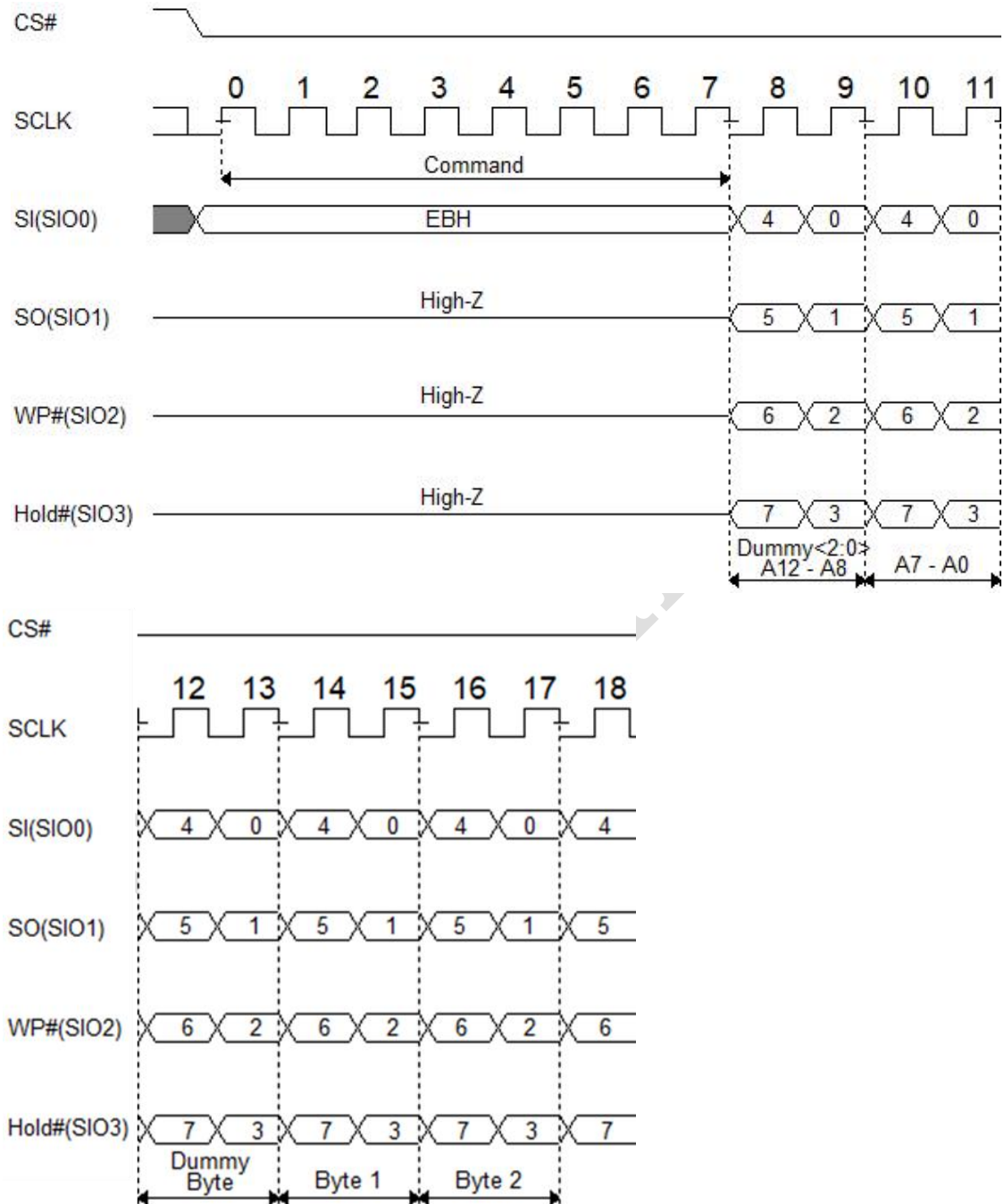
为了最大限度提高总线效率，在双 IO 模式下这些命令被看作是地址。因此，双 IO 模式可以在 SI00 和 SO1 上发送地址，如下所示。



请注意，这里的 Dummy Byte 是四个时钟周期。

(6) 从缓存中读取 4IO (EBH)

与双 IO 模式一样，4IO 模式可以将地址扩展到 4 个数据引脚，以扩展总线推进。



请注意，这里的 Dummy Byte 是两个时钟周期。

十四、编程

编程命令用于将数据写入 SPI NAND。由于在 SPI 接口和 NAND 闪存之间有一个缓存作为缓冲区，编程命令被划分为加载和执行命令。编程顺序可以分为以下两个流程。

- 02H (编程加载)/32H (编程加载 X4)
- 06H (写使能)
- 10H (编程执行)
- 0FH (通过获取特性命令读取状态)

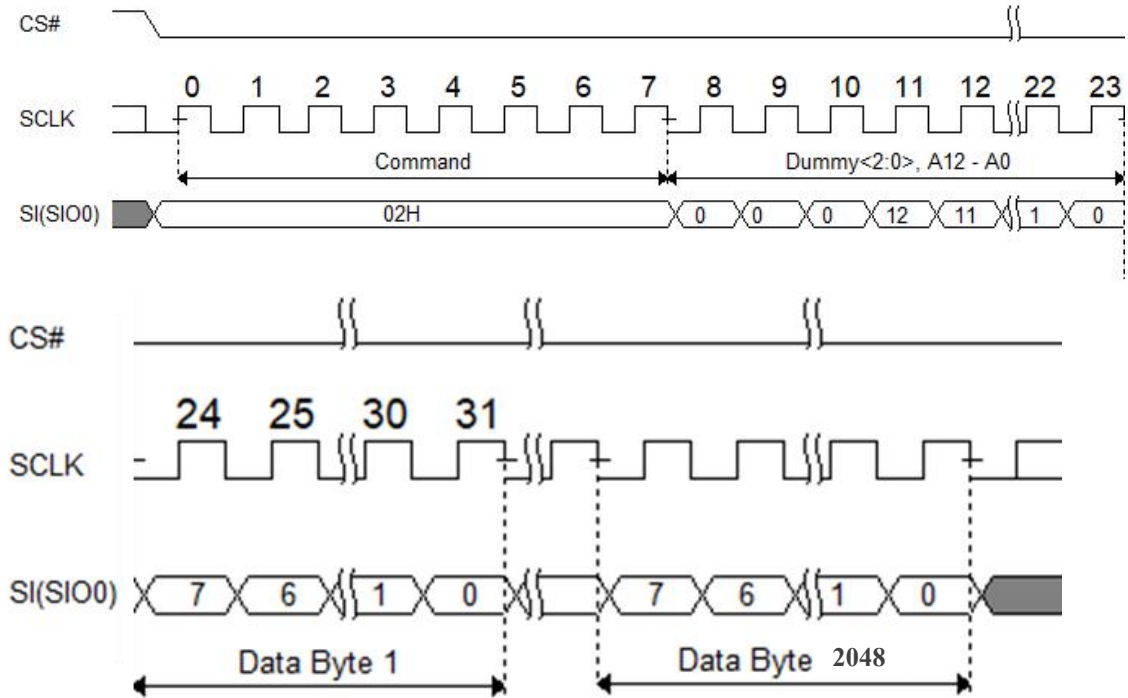
或者：

- 06H (写使能)
- 02H (编程加载)/32H (编程加载 x4)
- 10H (编程执行)
- 0FH (通过获取特性命令读取状态)

编程的第一个命令是编程加载 (02H/32H) 命令。它由一个 8 位的命令和后面的 3 个 dummy 位+13 位的列地址和数据字节组成。根据缓存大小的限制，将数据字节装入缓存寄存器。超过缓存大小的数据将被忽略。根据缓存大小的限制，将数据字节装入缓存寄存器。超过缓存大小的数据将被忽略。当 CS# 从低转到高时，命令序列结束。在执行编程执行之前，必须发送写使能 (06H) 命令。与任何改变内存内容的命令一样，必须执行写使能才能设置 WEL 位。如果未发出此命令，编程序列的其余部分将被忽略。

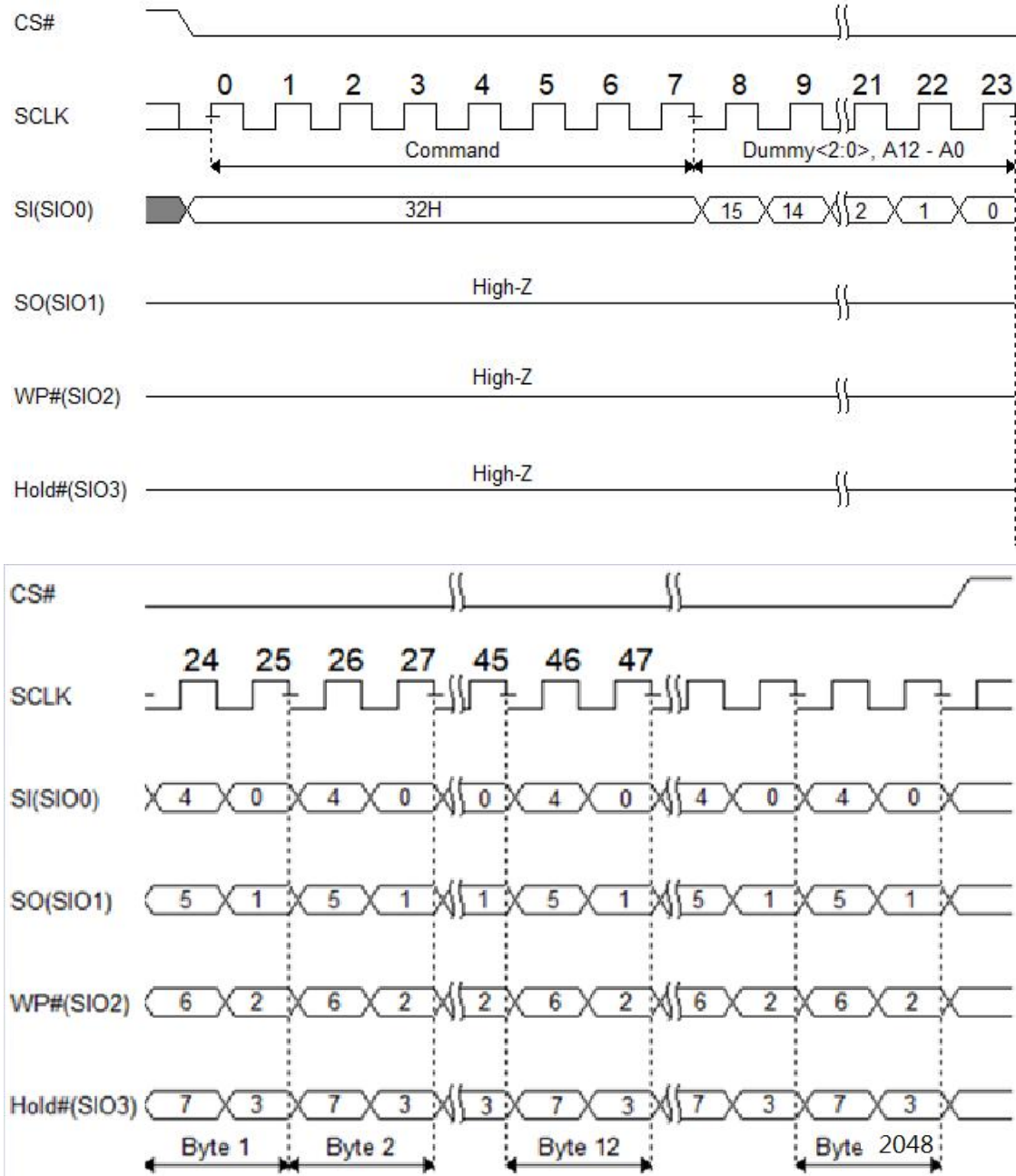
(1) 编程加载 (02H)

编程加载命令使用户能够将数据从 SPI 接口传输到控制器。直到用户发出编程执行前，数据将被存放在控制器。



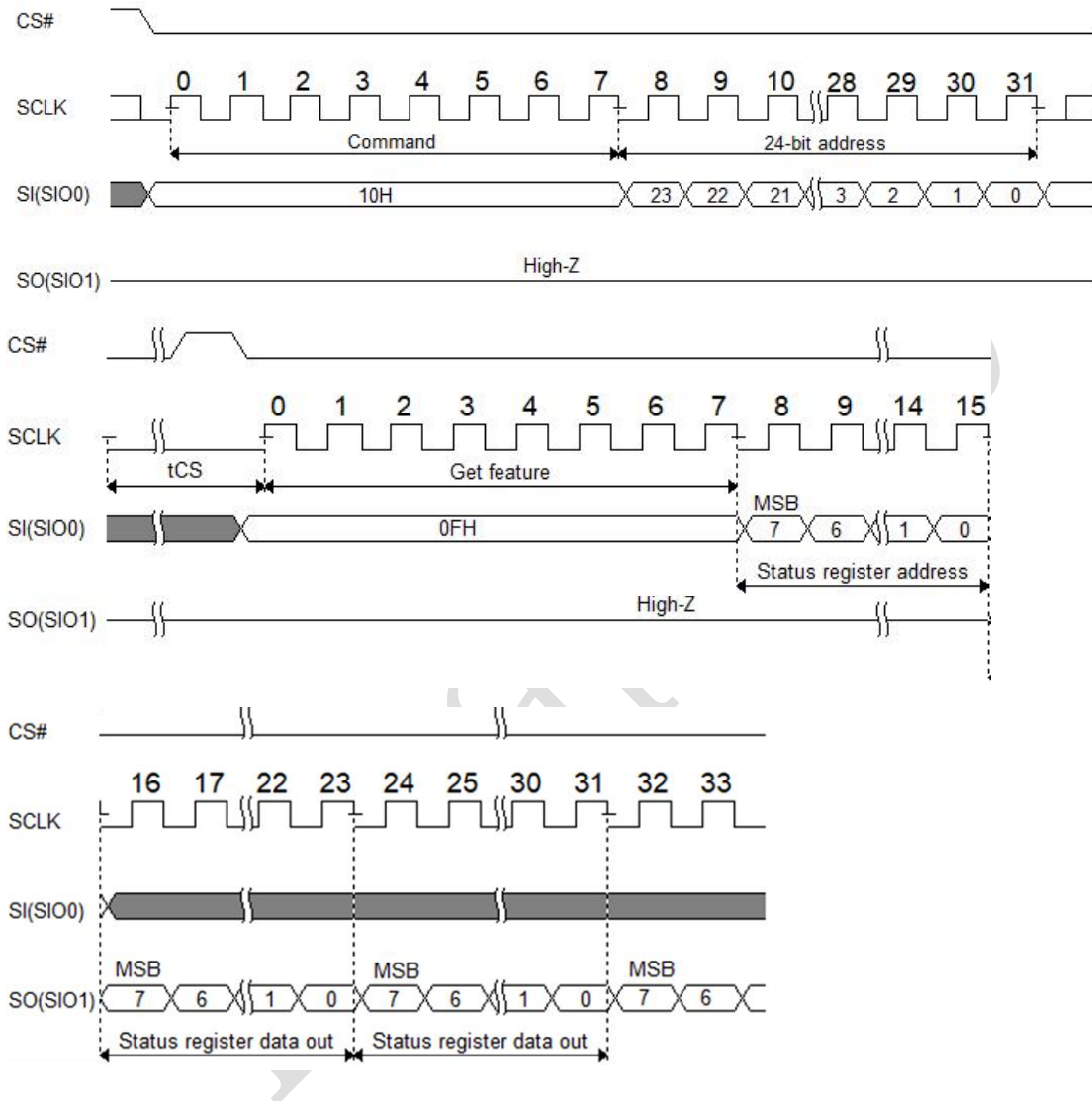
(2) 编程加载 x4 (32H)

在这种模式下，数据在 4 个数据管脚中传输，数据速率可以相应提高。在使用此模式之前，应提前将 QE 位设置为高。



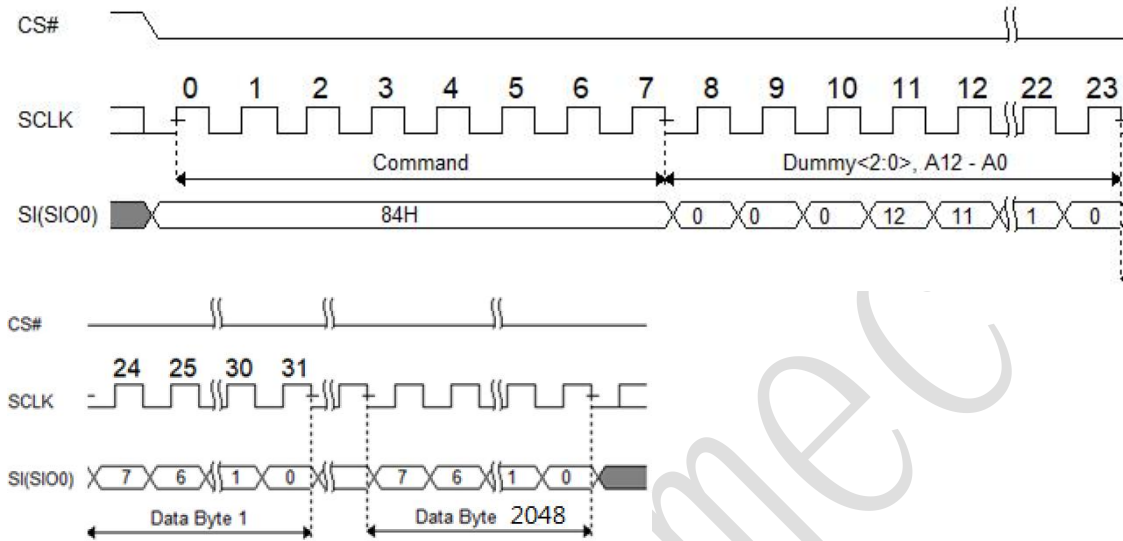
(3) 编程执行 (10H)

当用户发出程序执行时，控制器开始对 NAND 闪存执行闪存编程命令。编程将持续一段时间，时长取决于参数 t_{PROG}。在编程忙期间，用户可以继续监控状态，以检查编程是否完成。直到编程执行完成前，不能发送下一个编程加载命令。

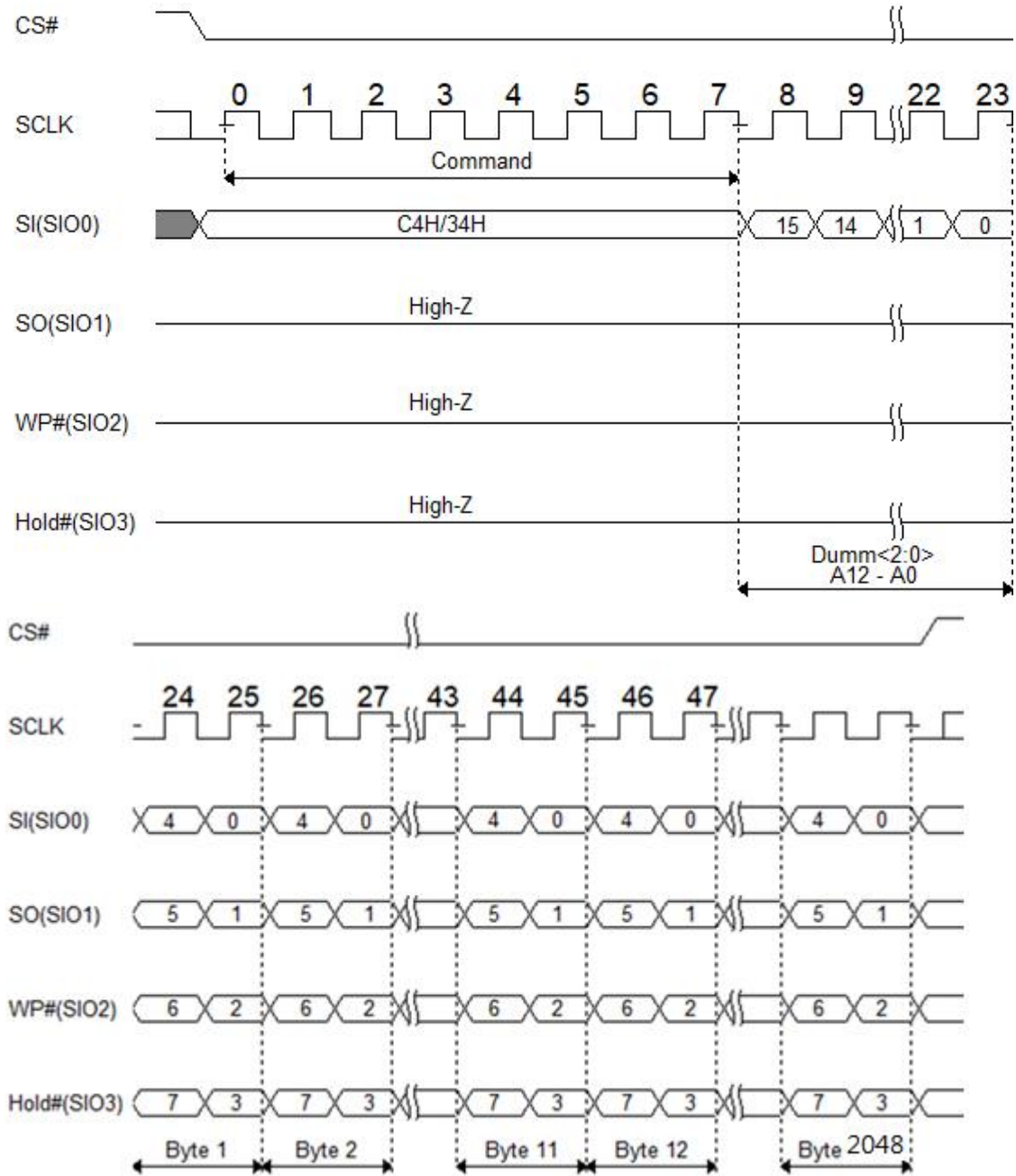


(4) 程序加载随机数据移动 (84H)

SPI NAND 缓存可以看作是一个缓冲区，因此用户可以根据地址将数据迁入或迁出。内部数据移动用于在页内附加或删除数据。操作地址应该在一个页面大小内。尽管可以自由使用内部数据，但请以字对齐的方式访问，以防止不可预测的情况。

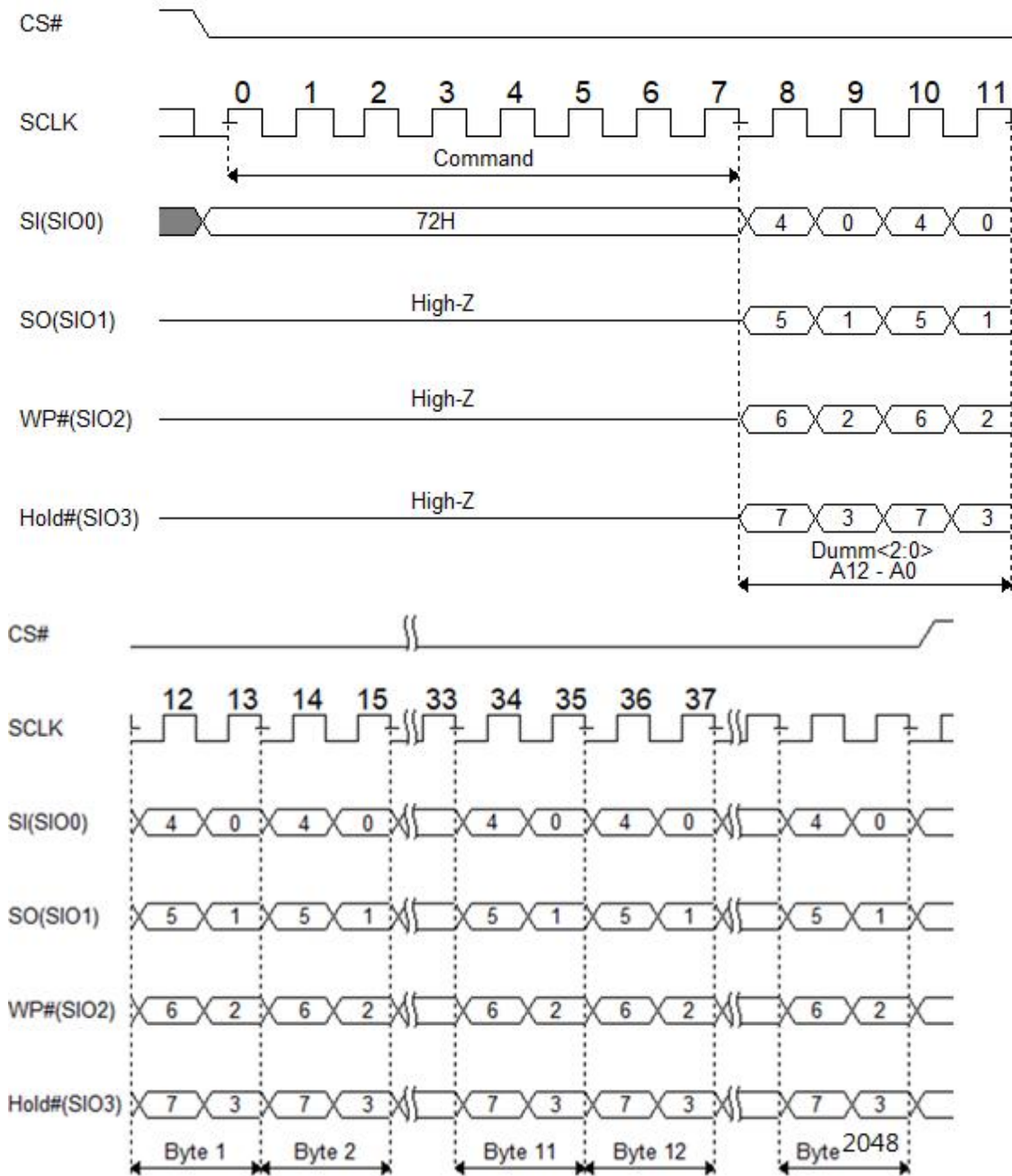


(5) 程序加载随机数据移动 x4 (C4/34H)
内部数据以 4 总线宽度移动。



(6) 程序加载随机数据移动 4I0 (72H)

内部数据以 4 总线宽度移动，以 4 数据宽度处理地址字段。

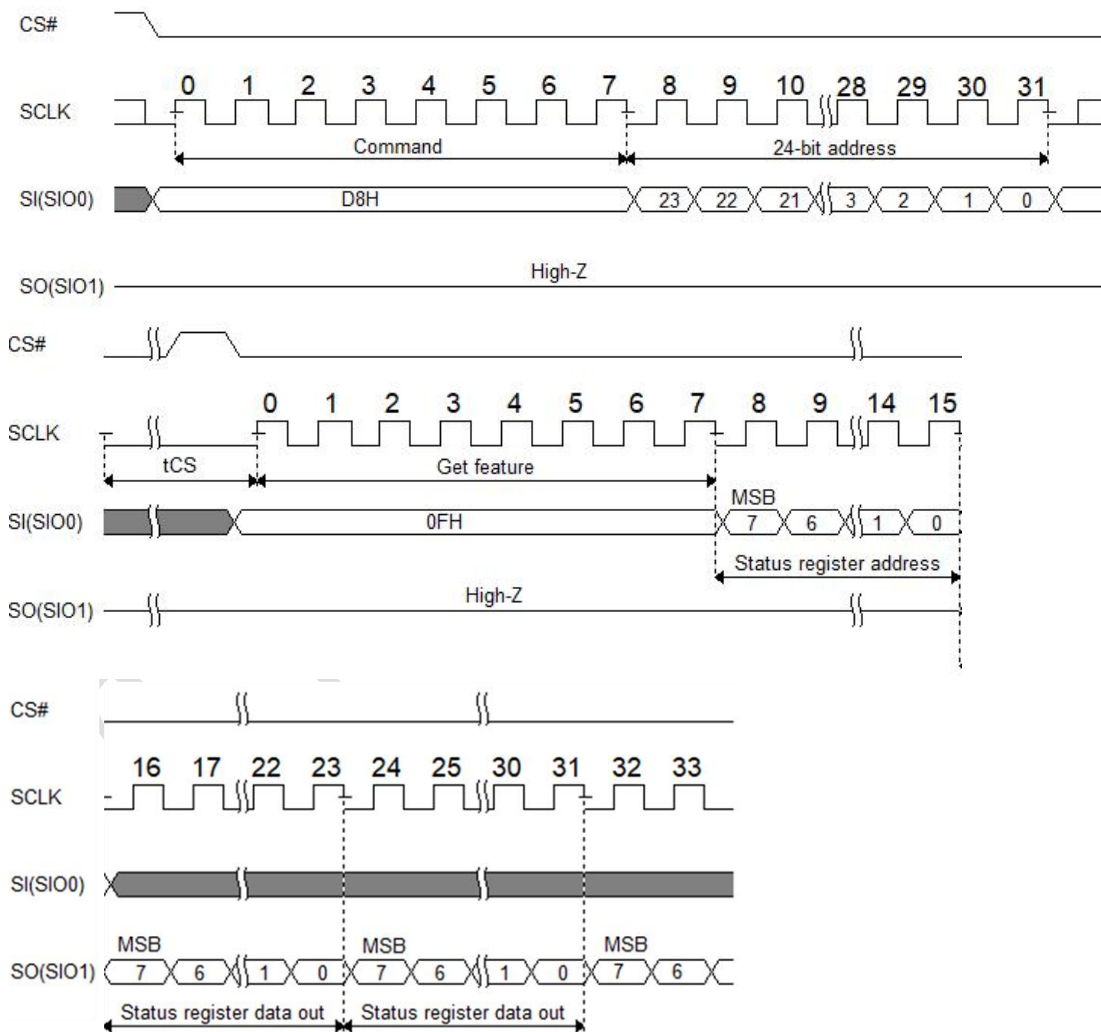


十五、块擦除 (D8H)

块擦除 (D8H) 命令用于在块级别擦除。块擦除命令按以下顺序操作

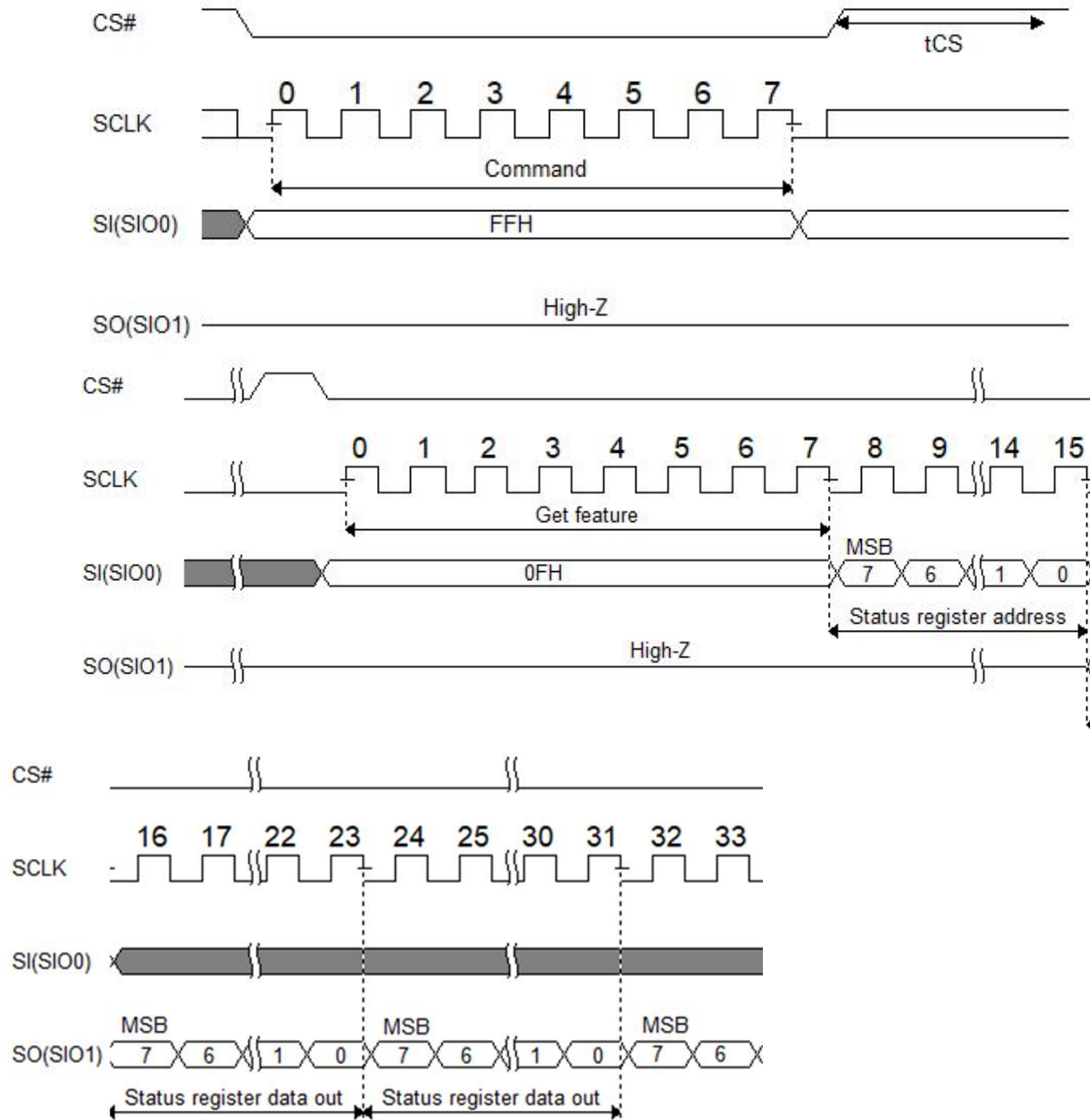
- 06H (写使能命令)
- D8H (块擦除命令)
- 0FH (通过获取特性命令读取状态)

擦除操作从写使能 (06H) 命令开始, 将“WEL 位”设置为 1。执行写使能命令后, 可以发送块擦除 (D8H) 命令。块擦除 (D8H) 需要一个由 dummy 和行地址组成的 24 位地址 (行地址中的页地址将被自动忽略)。发送获取特性命令 (0FH) 来监视发出块擦除命令后的擦除操作。状态寄存器 E_FALL 位可以反应块是否擦除成功。



十六、复位 (FFH)

复位 (FFH) 命令用来停止所有的操作。如果有先前的操作正在执行或挂起在缓存中，它将停止执行。



十七、高级功能

(1) 内部 ECC

SPI NAND 闪存提供内部 ECC，无需其他控制序列即可保护用户数据。ECC 可以使用 NAND 闪存的空闲区域，占用的大小由+ECC 保护能力决定。

设备上电后使能 ECC。

在编程操作期间，设备在缓存寄存器中的 2K 页上计算 ECC 编码，然后将该页写入 NAND 闪存阵列。

使用内部 ECC 时，用户必须满足以下要求：

- 备用区域定义见以下 ECC 保护表。
- ECC 可根据主备区域进行保护。对 ECC 区域的写将被忽略。

ECC 保护和备用区域 (Ex: Page= 2KB + 64B)

最大字节地址	最小字节地址	ECC 保护	区域	描述
3FFh	000h	YES	Main 0	User data 0
403h	400h	YES	Spare 0	User meta data 0
41Fh	404h	YES	Spare Area	Internal ECC parity data
81Fh	420h	YES	Main 1	User data 1
823h	820h	YES	Spare 1	User meta data 1
83Fh	824h	YES	Spare Area	Internal ECC parity data

ECC 保护和备用区域 (Ex: Page= 2KB + 128B)

最大字节地址	最小字节地址	ECC 保护	区域	描述
3FFh	000h	YES	Main 0	User data 0
403h	400h	YES	Spare 0	User meta data 0
42Dh	404h	YES	Spare Area	Internal ECC parity data
82Dh	42Eh	YES	Main 1	User data 1
831h	82Eh	YES	Spare 1	User meta data 1
85Bh	832h	YES	Spare Area	Internal ECC parity data

(2) 状态寄存器

设备具有一个 8 位状态寄存器，在设备运行期间可以通过软件读取该寄存器以查询运行状态查询。可以通过发送获取特性（0FH）命令来读取状态寄存器，后面跟着特性地址 COH（详情见特性描述）。

Bit	Bit Name	Description
P_FAIL	编程失败	这个位表示程序发生了故障 (P_FAIL 被置 1)。如果用户试图对无效地址或受保护区域 (包括 OTP 区域) 进行编程，此位也会被置 1。这个位在编程执行命令或复位命令期间被清除 (P_FAIL=0) 这个位只在页面编程命令之后有效。
E_FAIL	擦除失败	这个位表示擦除失败 (E_FAIL 被置 1)。如果用户试图擦除锁定区域，此位也会被置 1。这个位在块擦除命令或复位命令开始时被清零 (E_FAIL = 0)。
WEL	写使能锁存器	这个位表示写使能锁存器的当前状态 (WEL)，必须在发送编程执行命令或块擦除命令前置 1 (WEL = 1)。可以通过发送写使能命令将它置 1。也可以通过发送写使能命令来禁用 WEL (WEL = 0)
OIP	正在执行的操作	该位表示正在执行编程执行，页读取，块擦除或复位操作。
ECCS2 ~ ECCS0	ECC 状态	ECCS 提供 ECC 以下状态： 在复位之后或读取开始时，将 ECCS _x 设置为 0000b。然后在设备完成一个有效的读操作后更新它。 如果内部 ECC 被禁用 (通过设置特性命令将 ECC_EN 重置为 0)，则 ECCS 无效。上电复位后，ECC 状态将反映 0 块 0 页的内容。ECC1/0 位仅在页读取命令 (0x13) 之后有效。

下面是 ECC 状态的示例

示例 1: ECC 状态 (Page= 2KB + 64B)

ECCS2	ECCS1	ECCS0	描述
0	0	0	无误码。
0	0	1	误码 (<=3) 被检测并纠正。
0	1	0	误码 (<=6) 被检测并纠正。
0	1	1	误码 (<=9) 被检测并纠正。
1	0	0	误码 (<=12) 被检测并纠正。
1	0	1	误码 (<=15) 被检测并纠正。
1	1	0	误码 (=16) 被检测并纠正。
1	1	1	误码 (>=17) 超出范围并且无法被纠正。

示例 1: ECC 状态 (Page= 2KB + 128B)

ECCS2	ECCS1	ECCS0	描述
0	0	0	无误码。
0	0	1	误码 (<=4) 被检测并纠正。
0	1	0	误码 (<=8) 被检测并纠正。
0	1	1	误码 (<=12) 被检测并纠正。
1	0	0	误码 (<=16) 被检测并纠正。
1	0	1	误码 (<=20) 被检测并纠正。
1	1	0	误码 (<=24) 被检测并纠正。
1	1	1	误码 (>=25) 超出范围并且无法被纠正

(3) 强度寄存器

DS_S1	DS_S0	驱动强度
0	0	50%
0	1	25%
1	0	75%
1	1	100%

十八、电气特性

(1) 极限参数

参数	范围
存储温度	-65°C ~ +150°C
环境工作温度	-40°C ~ +85°C
应用输入/输出电压	0V ~ VCC*1.1

(2) 直流特性

参数	最小	典型	最大	单位
供电电压 (VCC)	2.7	3.3	3.6	V
输入漏电流			2	uA
输出漏电流			2	uA
工作电流 (读)		30	40	mA
待机电流			70	uA

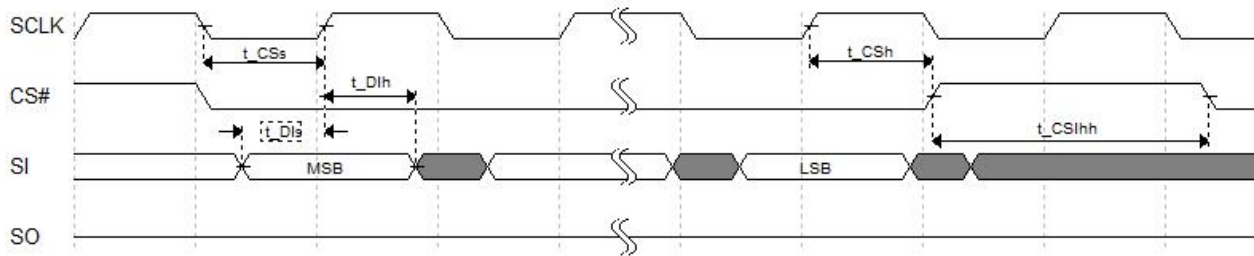
(3) 编程和擦除时间

参数	最小	典型	最大	单位
从阵列中读取			80	us
页编程时间		0.4	0.7	ms
块擦除时间		3	5	ms
复位后的下一条命令			500	us

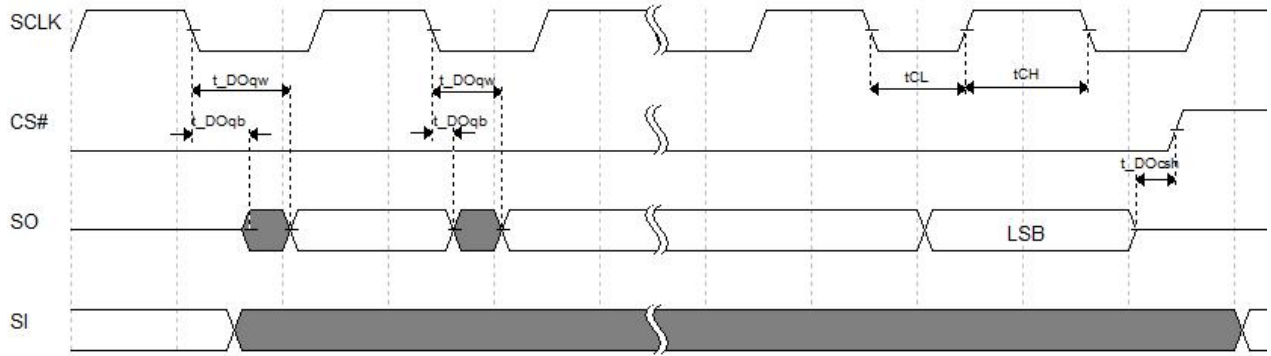
十九、时序特性

符号	参数	最小	典型	最大	单位
Fop	串行时钟频率			104	Mhz
t_CH	时钟高电平占比	4		4.8	ns
t_CL	时钟低电平占比	4		4.8	ns
t_CSs	CS# 建立时间	2			ns
t_CSh	CS# 保持时间	4			ns
t_D0s	数据输入建立时间	2			ns
t_D0h	数据输入保持时间	2			ns
t_CS1hh	CS# 从低到高时间	5			ns
t_DOqw	时钟到数据输出的最大延迟			8	ns
t_DOqb	时钟到数据输入的最小延迟	6			ns
t_D0csh	数据输出失效时间			20	ns
t_CHh1	HOLD# 高保持时间	5			ns
t_CHhh	HOLD# 低保持时间	5			ns
t_HHch	HOLD# 高建立时间	5			ns
t_HLch	HOLD# 低建立时间	5			ns
t_HHqx	HOLD# 输出从高到低阻			15	ns
t_HLqz	HOLD# 输出从高到高阻			15	ns

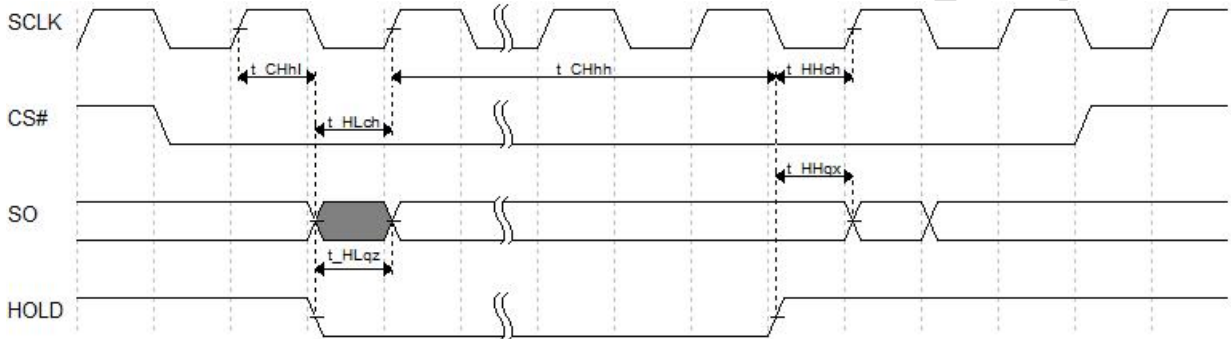
SPI 输入时序图



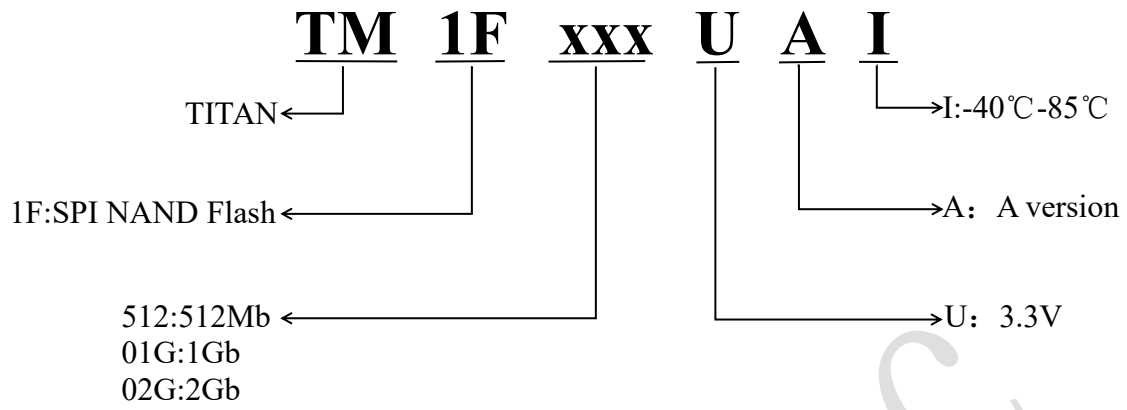
SPI 输出时序图



SPI HOLD 时序图

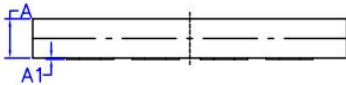
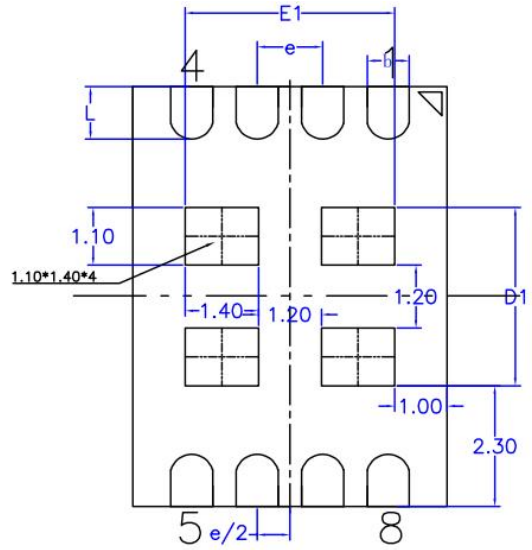
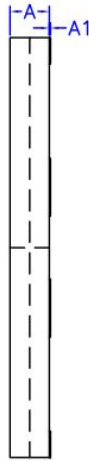
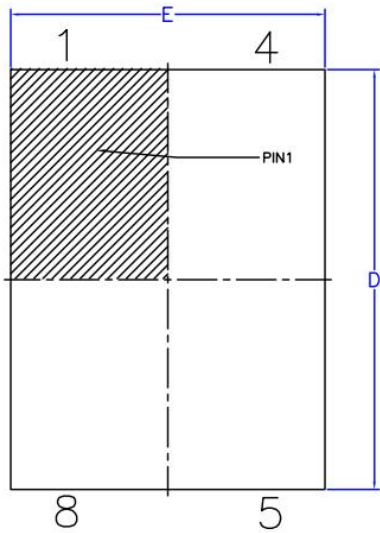


二十、订购信息



二十一、封装信息

LGA-8LD-6x8 的封装尺寸:



Dimensions

Item	D	E	D1	E1	A	A1	b	L	e
Unit	mm	mm	mm	mm	mm	mm	mm	mm	mm
Spec	8.10 (8.00) 7.90	6.10 (6.00) 5.90	3.45 (3.40) 3.35	4.05 (4.00) 3.95	0.80 (0.75) 0.70	0.05 (0.02) 0.00	0.85 (0.80) 0.75	1.05 (1.00) 0.95	125BSC