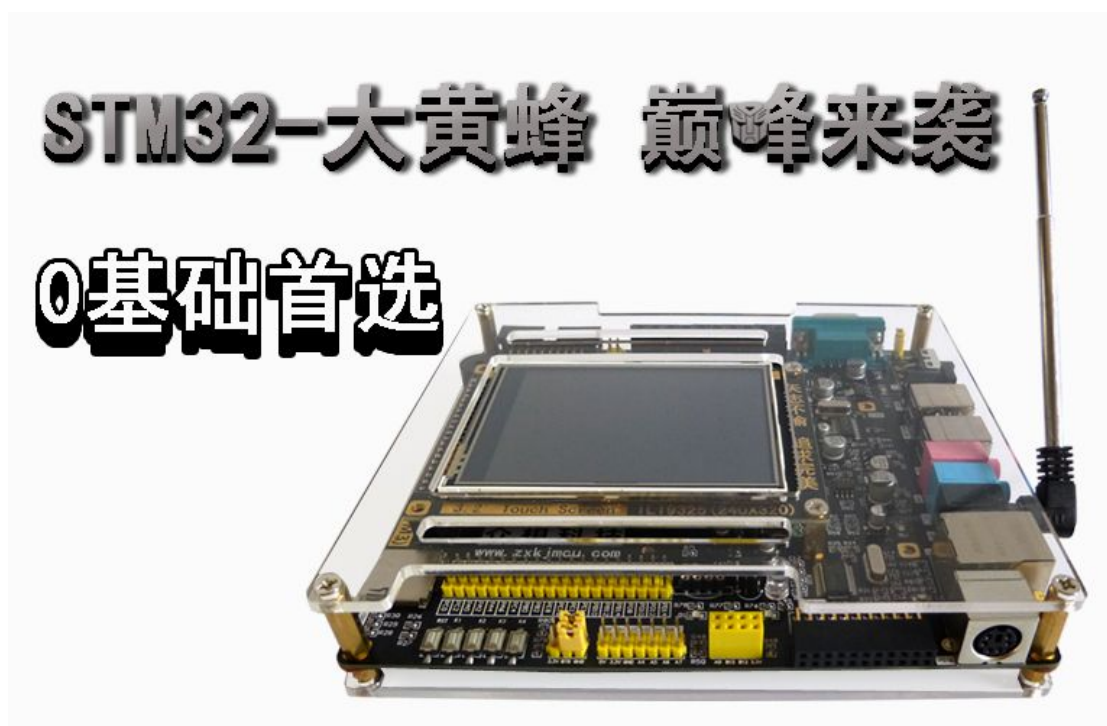


学 ARM 从 STM32 开始

STM32 开发板库函数教程—实战篇



官方网站: <http://www.zxkjmcu.com>

官方店铺: <http://zxkjmcu.taobao.com>

官方论坛: <http://bbs.zxkjmcu.com>

刘洋课堂: <http://school.zxkjmcu.com>

4.27 STM32 SST25VF016B 工作原理 (SPI 接口 FLASH)

4.27.1 概述

16 Mbit SPI 串行闪存。

4.27.2 SST25VF016B 特性

- 单电压读写操作
 - 2.7-3.6V
- 串行接口架构
 - 兼容 SPI 模式 0 和模式 3
- 高速时钟频率
 - 最高 50 MHz
- 卓越的可靠性
 - 可擦写次数: 100,000 周期 (典型值)
 - 大于 100 年数据保存期
- 低功耗
 - 有效的读操作电流: 10 mA (典型)
 - 待机电流: 5 μ A (典型)
- 灵活的擦除功能
 - 均一 4 KB 扇区
 - 均一 32 K 字节块重叠

- 均一 64 K 字节块重叠
- 快速擦除和字节编程
 - 芯片擦除时间: 35 ms (典型)
 - Sector-/Block-Erase 擦除时间: 18 ms (典型)
 - 字节编程时间: 7 μ s (典型)
- 编程自动地址递增(AAI)
 - 减少字节编程操作期间的整个芯片编程时间
- End-of-Write 检测
 - 软件查询中状态 BUSY 位寄存器
 - AAI 模式下 SO 引脚上的忙状态读出
- 保持引脚 (按住)
 - 挂起到内存串行序列该设备没有取消选择
- 写保护(WP#)
 - 启用/禁用的锁断功能 status 寄存器
- 软件写保护
 - 通过写块保护 bits 在保护 status 寄存器
- 温度范围
 - 商业 0° C 到+70° C
 - 工业 -40° C 到+85° C
- 封装形式
 - 8-lead SOIC (200 mils)
 - 8-contact WSON (6mm x 5m

4.27.3 SST25VF016B 引脚说明

表 1：引脚说明

符号	引脚名称	功能
SCK	串行时钟	用于提供串行接口的时序。 命令、地址或者输入数据在时钟输入的上升沿进行锁存，而输出数据在时钟输入的下降沿移出。
SI	串行数据输入	用于以串行方式将命令、地址或者数据传送到器件中。 输入在串行时钟的上升沿进行锁存。
SO	串行数据输出	用于以串行方式将数据传送出器件。 数据在串行时钟的下降沿移出。 当重新配置为 RY/BY# 引脚后，在 AAI 编程期间输出闪存忙状态。有关详细信息，请参考后续讲解。
CE#	芯片使能	CE# 上高电平到低电平的转换可使能器件，在任何命令序列的持续时间内，CE# 都必须保持低电平。
WP#	写保护	写保护 (WP#) 引脚用于使能/禁止状态寄存器中的 BPL 位。
HOLD#	保持	用于在不复位器件的情况下暂时停止与 SPI 闪存存储器的串行通讯。
VDD	电源	电源电压
VSS	地	参考地

4.27.4 SST25VF016B 模式

SST25VF016B 支持 SPI 总线操作模式 0 (0,0) 和模式 3 (1,1)。两种模式之间的差异在于当总线主器件处于待机模式并且没有数据传送时的 SCK 信号状态。如图 1 所示，SCK 信号在模式 0 时为低电平，在模式 3 时为高电平。对于这两种模式，串行数据输入 (SI) 在 SCK 时钟信号的上升沿被采样，串行数据输出 (SO) 在 SCK 时钟信号的下降沿之后被驱动。

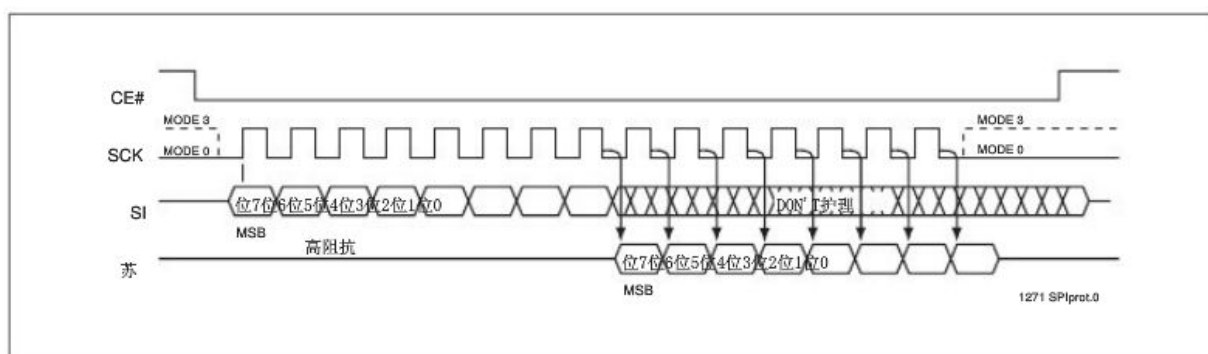


图 1 SPI 时序

4.27.5 SST25VF016B 写保护

SST25VF016B 提供软件写保护。写保护引脚 (WP#) 用于使能或者禁止状态寄存器的锁定功能。状态寄存器中的块保护 bits (BP3、BP2、BP1、BP0 和 BPL) 提供对存储器阵列和状态寄存器的写保护。见表 4 的块保护描述。

4.27.5.1 写保护 (WP#) 引脚

写保护 (WP#) 引脚用于使能状态寄存器的 BPL 位 (bit 7) 的锁定功能。当 WP#驱动为低电平时，是否执行写状态寄存器 (WRSR) 指令由 BPL 位的值 (见表 2) 决定。当 WP#为高电平时，BPL 位的锁定功能被禁止。

表 2: 执行写状态寄存器 (WRSR) 指令的条件

WP#	BPL	执行 WRSR 指令
L	1	不允许
L	0	允许
H	x	允许

4.27.6 状态寄存器

软件状态寄存器状态是提供有关闪存存储器阵列是否可进行任何读或写操作的状态、器件是否写使能以及存储器写保护状态。在内部擦除或编程操作期间，可以对状态寄存器只进行读操作，以确定正在进行的操作的完成状态。详细见表 3。

4.27.6.1 Busy 位

Busy 位确定是否有内部擦除或正在进行的程序编程操作。Busy 位为“1”表示器件正忙于操作；Busy 位为“1”表示器件准备好进行下一个有效操作。

4.27.6.2 写使能锁存器 (WEL)

写使能锁存器 (Write-Enable-Latch) 位表示内部寄存器的写使能锁存器状态。如果 Write-Enable-Latch 位设置为“1”,它表明器件写使能。如果该位设置为“0” (复位), 它表明器件未写使能, 不接受任何存储器写 (写入 程序/擦除) 命令。在下列条件下, 写使能锁存器位自动复位:

上电

写禁用(WRDI)指令完成

字节编程指令完成

自动地址递增(AAI)编程完成或达到最高不受保护内存地址

扇区擦除指令完成

块擦除指令完成

芯片擦除指令完成

Write-Status-Register 指令

表 3: 软件状态寄存器

bit	名称	功能	上电时默认值	读/写
0	BUSY	1=内部写操作正在进行中 0=没有内部写操作正在进行中	0	R
1	WEL	1 =器件处于存储器写使能状态 0 =器件未处于存储器写使能状态	0	R
2	BP0	说明当前的块写保护级别	1	R/W
3	BP1	说明当前的块写保护级别	1	R/W
4	BP2	说明当前的块写保护级别	1	R/W
5	BP3	说明当前的块写保护级别	0	R/W
6	AAI	编程状态自动地址递增 1 = AAI 编程模式	0	R

		0 = 字节编程模式		
7	BPL	1 = BP3, BP2, BP1, BP0 是只读 bits 0 = BP3, BP2, BP1, BP0 可读/写	0	R/W

4.27.6.3 块保护 (BP3、BP2、BP1 和 BP0)

块保护 (BP3、BP2、BP1 和 BP0) 位定义了要通过软件保护方式防止任何存储器写 (编程或者擦除) 操作的存储区域的大小, 只要 WP# 为高电平或块保护锁定 (Block-Protect-Lock, BPL) 位为 0, 就可以使用写状态寄存器 (WRSR) 指令编程 BP3、BP2、BP1 和 BP0 位。仅当块保护全部为 0 时, 才能执行全片擦除。上电后, BP3、BP2、BP1 和 BP0 置为 1。

表 4: 软件状态寄存器块保护

保护级别	状态寄存器				受保护的存储器地址
	BP3	BP2	BP1	BP0	16Mb
无	x	0	0	0	无
前 1/32	x	0	0	1	1F0000H-1FFFFFFH
前 1/16	x	0	1	0	1E0000H-1FFFFFFH
前 1/8	x	0	1	1	1C0000H-1FFFFFFH
前 1/4	x	1	0	0	180000H-1FFFFFFH
前 1/2	x	1	0	1	100000H-1FFFFFFH
所有快	x	1	1	0	000000H-1FFFFFFH
所有快	x	1	1	1	000000H-1FFFFFFH

1.x=无关 (保留), 默认值为 “0”。

2.上电时, BP2、BP1、BP0 的默认值为 “111” (所有块均受保护)。

4.27.7 指令

指令用于读、写 (擦除和编程) 和配置总线周期是 8 个表示命令 (操作码)、数据和地址的位。在执行任何字节编程、自动地址递增 (AAI) 编程、扇区擦除、块擦除、写状态寄存器或者全片擦除之前, 必须先执行写使能 (WREN) 指令。表 5 提供了完整的指令列表。所有指令在 CE# 从高电平转换到低电平时同步。在 SCK 的上升沿从最高有效位开始接受输入。在输入

指令之前，CE#必须驱动为低电平，而在输入指令的最后一位，CE#必须驱动为高电平（读、读 ID 和读状态寄存器指令除外）。在接收到指令总线周期的最后一位之前，CE#上任何电平到高电平的转换都将终止正在进行的指令，并将器件恢复成待机模式。指令命令（操作码）、地址和数据都先从最高有效位输入。

表 5：器件操作指令

指令	说明	操作码周期	地址周期	空周期	数据周期	最大频率
读	以 25MHz 的频率读存储器	0000 0011b (03H)	3	0	1 至 ∞	25MHz
高速读	以 80MHz 的频率读存储器	0000 1011b (0BH)	3	1	1 至 ∞	80MHz
4K 扇区擦除	擦除 4KB 的存储器阵列	0010 0000b (20H)	3	0	0	80MHz
32K 扇区擦除	擦除 32KB 的存储器阵列	0101 0010b (52H)	3	0	0	80MHz
64K 扇区擦除	擦除 64KB 的存储器阵列	1101 1000b (D8H)	3	0	0	80MHz
全片擦除	擦除全部存储器阵列	0110 0000b (60H) 1100 0111b (C7H)	3	0	1	80MHz
字节编程	编程一个数据字节	0000 0010b (02H)	3	0		80MHz
AAI 字编程	自动地址递增编程	1010 1101b (ADH)	3	0	2 至 ∞	80MHz
RDSR	读取状态寄存器	0000 0101b (05H)	0	0	1 至 ∞	80MHz
EWSR	使能写状态寄存器	0101 0000b (50H)	0	0	0	80MHz
WRSR	写状态寄存器	0000 0001b (01H)	0	0	1	80MHz
WREN	写使能	0000 0110b (06H)	0	0	0	80MHz
WRDI	写禁止	0000 0100b (06H)	0	0	0	80MHz
RDID	读 ID	1001 0000b (90H) 1010 1011b (ABH)	3	0	1 至 ∞	80MHz
JEDEC-ID	JEDEC ID 读	1001 1011b (9FH)	0	0	3 至 ∞	80MHz
EBSY	在 AAI 编程器期间使能 SO 以输出 RY/BY 状态	0111 0000b (70H)	0	0	0	80MHz
DBSY	在 AAI 编程器期间禁止 SO 为 RY/BY 状态	1000 0000b (80H)	0	0	0	80MHz

4.27.7.1 读

读指令（03H）支持最高为 25MHz 的读操作。器件从指定的地址单元

开始输出数据。数据输出流连续遍历所有地址，直到被 CE# 上低电平到高电平的转换终止。内部地址指针将自动递增，直到达到最高的存储器地址为止。达到最高的存储器地址后，地址指针将自动递增到地址空间的开始位置（回绕）。从地址单元 1FFFFFFH 读取数据之后，下一次输出将来自地址单元 000000H。通过执行 8 位命令 03H（后面紧跟着地址位【A23-A0】）来启动读指令。在读周期的持续时间内，CE# 必须保持有效低电平。

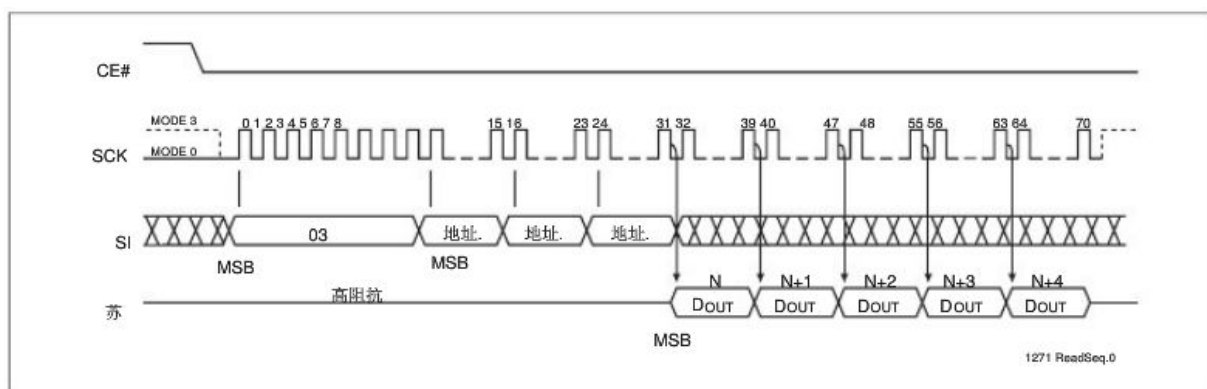


图 2 读指令时序

4.27.7.2 高速读[High-Speed-Read (80 MHz)]

该 High-Speed-Read 指令支持高达 80 MHz 读操作，是通过执行一个 8-bit 命令 0BH（后面紧跟地址位 [A23-A0] 和一个空字节）启动。CE# 必须保持有效低电平。有关高速读序列，请参见图 3 的 High-Speed-Read 序列。经过一个空周期，High-Speed-Read 指令从指定的地址单元开始输出数据。数据输出流是连续通过所有地址，直到被 CE# 上低电平到高电平的跳变终止。内部地址指针会自动递增，直至达到最高的内存地址。一旦达到最高的内存地址后，地址指针会自动递增至地址空间的开始位置（）。一旦数据从地址位置 1FFFFFFH 被读取后，下一次输出将来自地址单元 000000H。

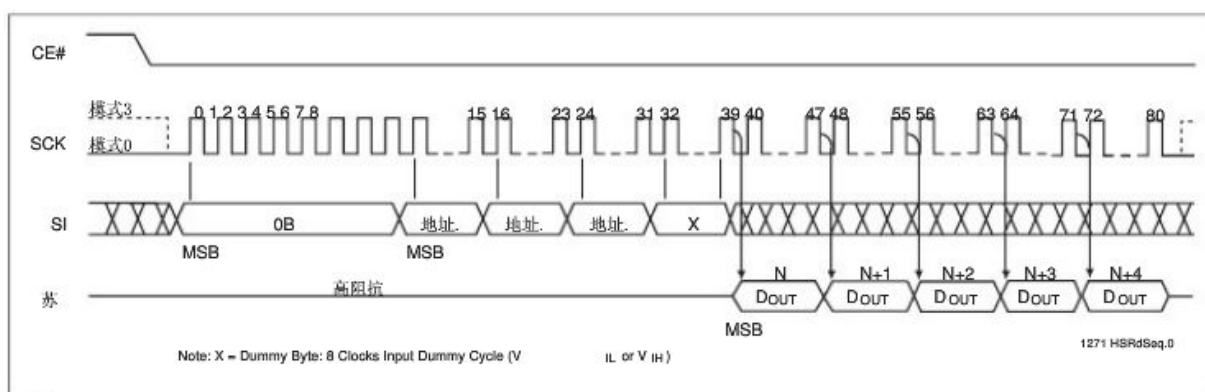


图 3 高速读指令时序

4.27.7.3 字节编程

字节编程指令计划在 bits 选择所需的数据字节。当启动编程操作时，所选择字节必须处于擦出状态（FFH）。一个应用到受保护存储器区域的字节编程指令会被忽略。在执行任何写操作之前，必须先执行写使能（WREN）指令。在字节编程指令持续的时间内，CE#必须保持低电平。通过执行一个 8-bit 命令 02H（后面紧跟地址 bits [A23-A0]）来启动字节程序指令。在地址后面，数据输入从 MSB（bit7）到 LSB（bit 0）的顺序输入。在指令执行之前，CE#必须被驱动为高电平。用户可以轮询软件状态寄存器中的 BUSY 位，或等待 TBP（10us）让内部完成自定时字节的编程操作。参见图 4 字节编程序列。

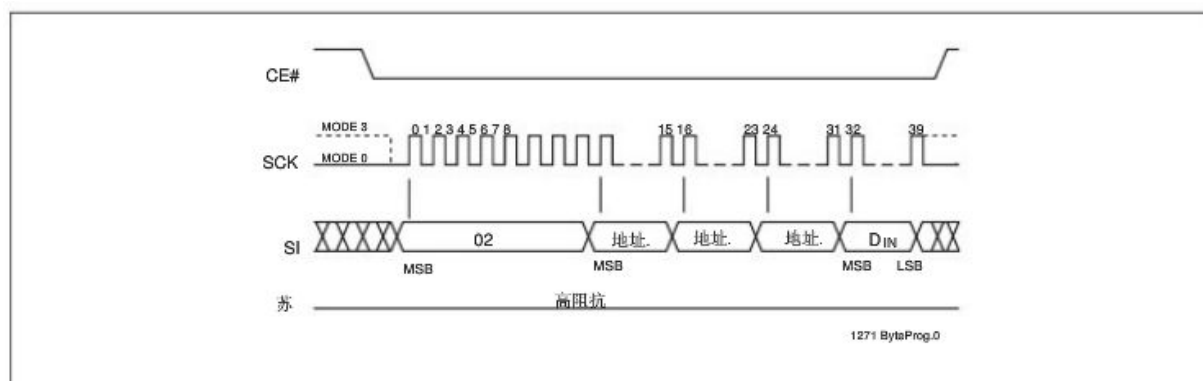


图 4 字节编程时序

4. 27. 7. 4 自动地址递增 (AAI)

AAI 程序指令允许对多个字节的数据进行编程，而无需重新发出下一个顺序地址单元。当要对多个字节或整个存储器阵列进行编程时。该特性可以减少总编程时间。指向受保护的存储器区域的 AAI 字编程指令会被忽略。选定地址范围必须在擦除状态时(FFH) initiating 一 AAI 字操作才被允许。参见图 5.1 和图 5.2 使用硬件写操作结束检时的自动地址递增 (AAI) 字编程序列。

处于 AAI 字编程序列时，唯一有效的说明：

1、软件写操作结束检测指令

AAI 字(ADH)、读取状态寄存器 RDSR (05H)、写禁止 WRDI (04H)。

用户有三种选择 以确定每个完成 AAI 字编程周期。

2、硬件写操作结束检测指令

AAI 字 (ADH) 、写禁止 WRID (04H) 。

在执行任何写操作之前，必需先执行写使能 (WREN) 指令。先执行一个 8-bit 命令 ADH (后面紧跟地址 bits [A23-A0]) 来启动 AAI 字编程指令。在地址后面，两个字节的数据连续输入，每一个字节都从 MSB (bit7) 到 LSB (bit0)。在执行 AAI Word 编程指令之前，CE#必须驱动为高电平。在输入下一个有效命令之前，请检查 BUSY 状态。一旦器件指示它不再忙碌，就可以对下两个连续的地址的数据进行编程，然后对再下两个连续的地址的数据进行编程，以此类推。

当编程最后一个所需字或者最高的不受保护存储器地址时，通过硬件或软件 (RDSR 指令) 方法检查忙状态，以确定编程是否完成。编程完成之后

使用适当的方法终止 AAI。如果器件处于软件写操作结束检测模式，则执行写禁止 (WRDI) 指令 04H。再执行 8 位 DBSY 命令 80H。在达到最高的未受保护的存储器地址之后，AAI 编程期间不存在回绕模式。

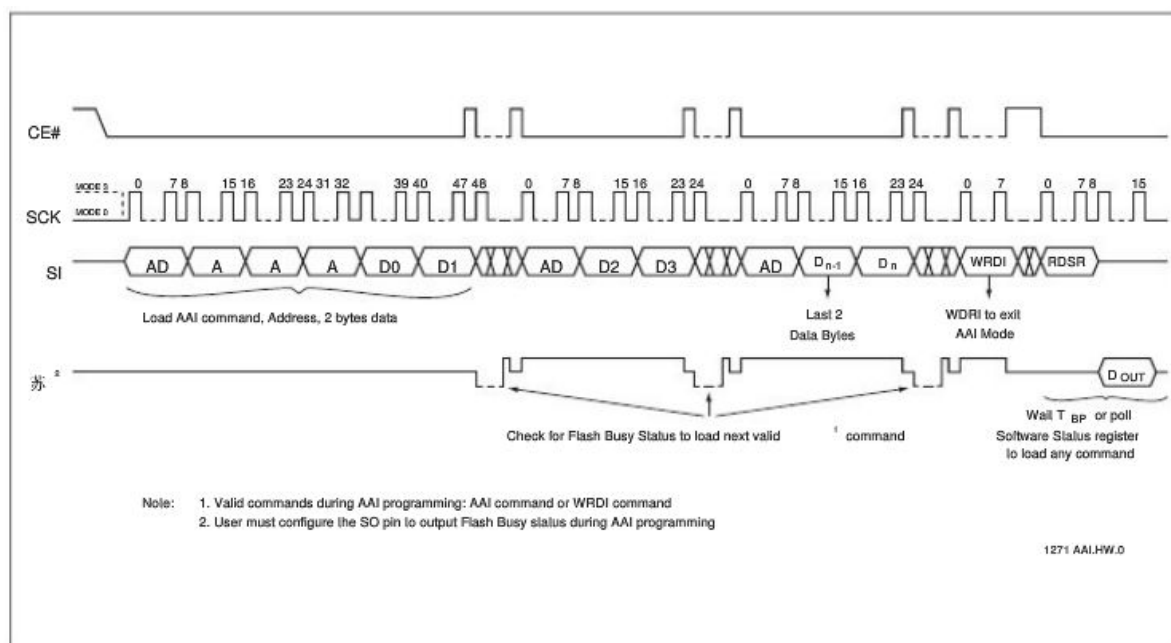


图 5.1 使用硬件写操作结束检时的自动地址递增 (AAI) 字编程序列

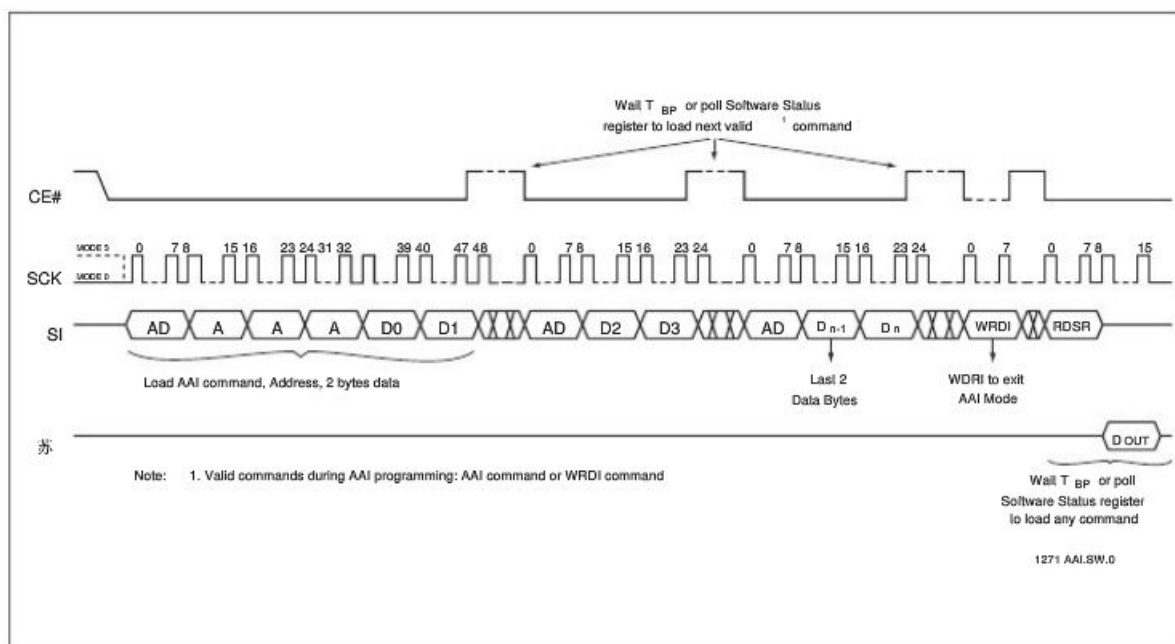


图 5.2 使用软件写操作结束检时的自动地址递增 (AAI) 字编程序列

4.27.7.5 4-KByte 扇区擦除

扇区擦除指令将所有选定 bits 4K 字节清除为 FFH。应用到受保护存储器区域的扇区擦除指令将被忽略。在执行任何写操作之前，写使能（WREN）指令必须先执行。在任何命令序列的持续时间内，CE#必须保持低电平。通过执行一个 8-bit 命令 20H（后面紧跟地址位[A23-A0]）来启动扇区擦除指令。在执行指令之前，CE#必须驱动为高电平。用户可以轮询软件状态寄存器中的 BUSY 位，或等待 TSE（25ms）时间让内部自定时扇区扇区擦除周期完成。见图 6 4K 扇区擦除时序。

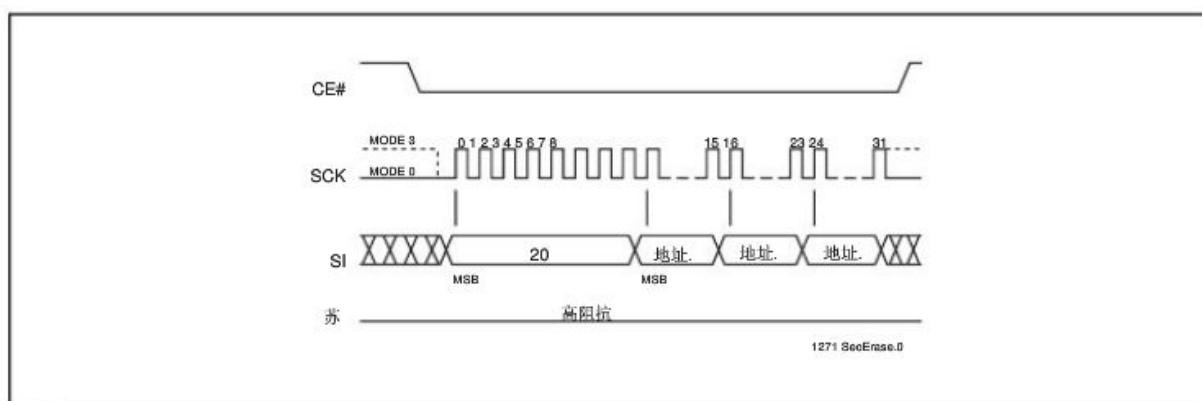


图 6 4K 扇区擦除时序

4.27.7.6 32(64)-KByte 扇区擦除

扇区擦除指令将所有选定 bits 32K 字节清除为 FFH。应用到受保护存储器区域的扇区擦除指令将被忽略。在执行任何写操作之前，写使能（WREN）指令必须先执行。在任何命令序列的持续时间内，CE#必须保持低电平。通过执行一个 8-bit 命令 52H（后面紧跟地址位[A23-A0]）来启动 32K 扇区擦除指令。（通过执行一个 8-bit 命令 D8H（后面紧跟地址位[A23-A0]）来启动 64K 扇区擦除指令。）在执行指令之前，CE#必须驱动为高电平。用户可以轮询软件状态寄存器中的 BUSY 位，或等待 TSE（50ms）时间让内部自定时

扇区擦除周期完成。见图 7 32K 扇区擦除时序，见图 8 64K 扇区擦除时序。

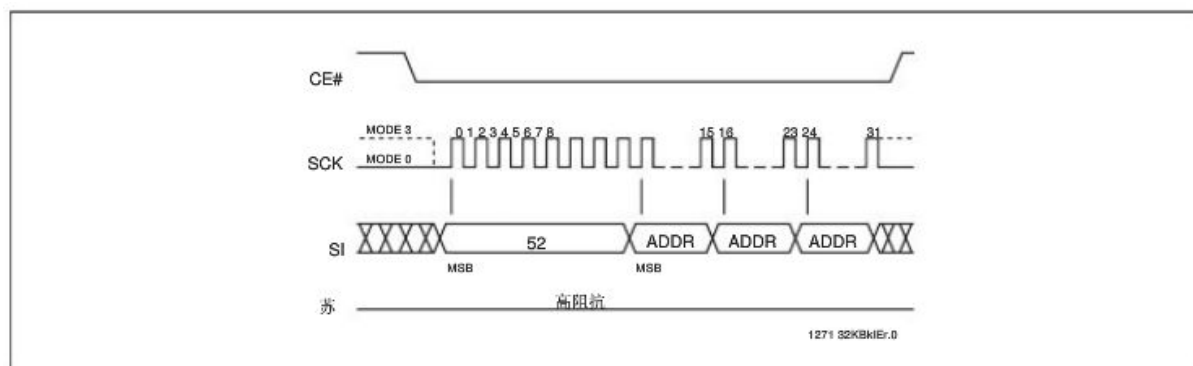


图 7 32K 扇区擦除时序

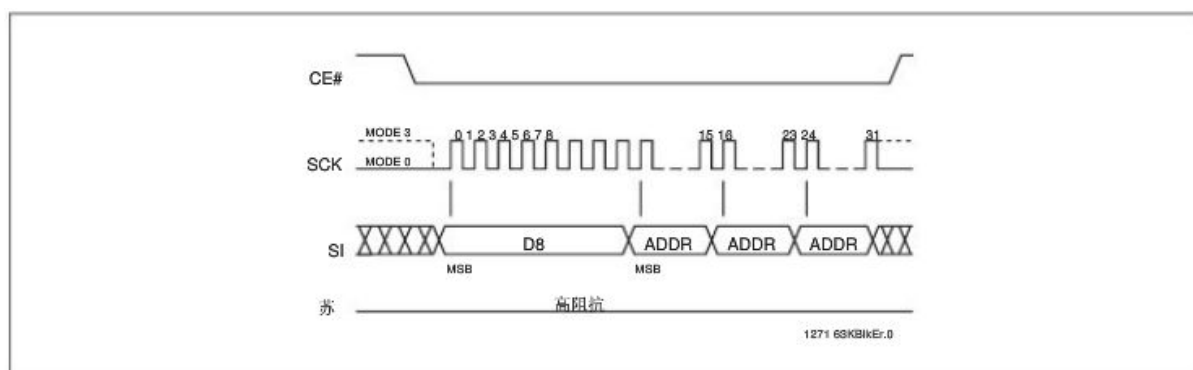


图 8 64K 扇区擦除时序

4.27.7.7 全扇区擦除

全扇区擦除指令将器件中的所有位全部清除为 FFH。应用到受保护存储器区域的扇区擦除指令将被忽略。在执行任何写操作之前，写使能（WREN）指令必须先执行。在任何命令序列的持续时间内，CE#必须保持低电平。通过执行一个 8-bit 命令 60H 或 C7H（后面紧跟地址位[A23-A0]）来启动全扇区擦除指令。在执行指令之前，CE#必须驱动为高电平。用户可以轮询软件状态寄存器中的 BUSY 位，或等待 TSE（50ms）时间让内部自定时扇区擦除周期完成。见图 9 全扇区擦除时序

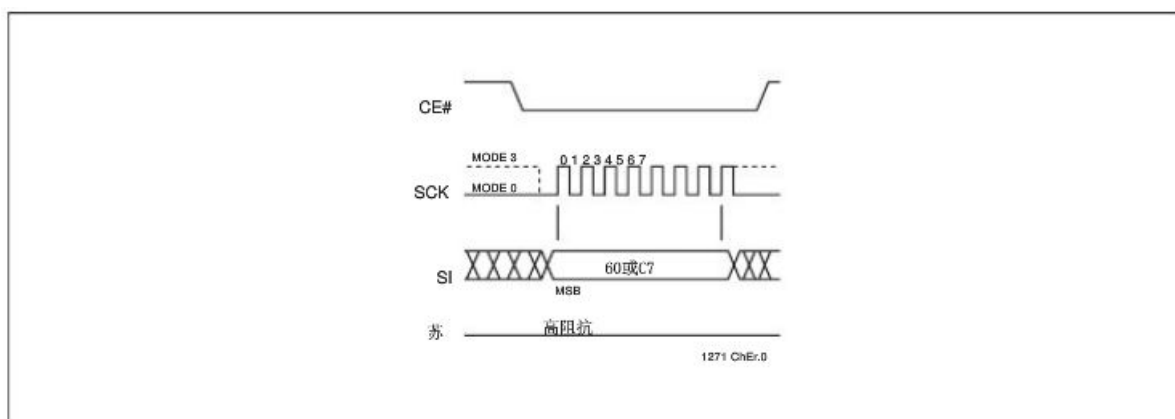


图 9 全扇区擦除时序

4.27.7.8 Read-Status-Register (RDSR)

Read-Status-Register (RDSR) 指令允许读取状态寄存器。可在任何时间读取状态寄存器。甚至是在写（编程/擦除）操作期间。当一个写操作正在进行中，应发出任何新的命令前检查 BUSY 位，以确保新的命令会被器件正确接收。在输入 RDSR 指令之前，CE# 必须驱动为低电平，并保持低电平直到状态数据被读取为止。读状态寄存器继续使用当前时钟周期，直到被 CE# 上低电平到高平的转换终止。见图 10 RDSR 指令序列图。

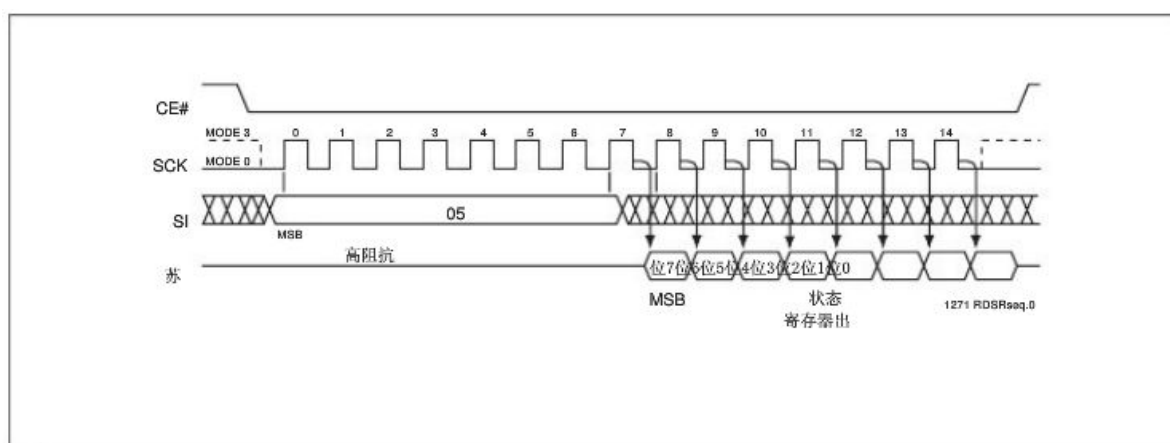


图 10 RDSR 指令序列图

4.27.7.9 写使能

写使能 (WREN) 指令可将状态寄存器中的写使能锁存器位设置为“1”，以允许进行写操作。在执行任何写 (编程/擦除) 操作之前。都必须先执行 WREN 指令。在执行 WREN 指令之前，CE# 必须驱动为高电平，见图 11 写使能时序。

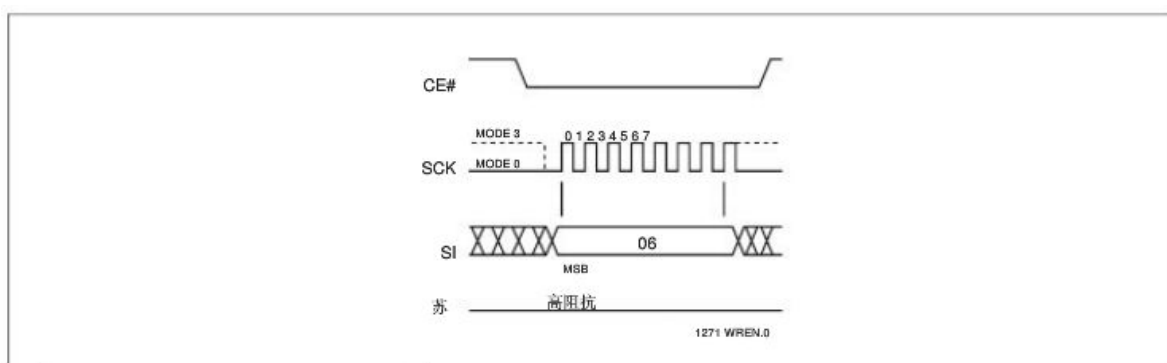


图 11 写使能时序

4.27.7.10 写禁止 (WRDI)

写禁止 (WRDI) 指令可将写使能锁存器位和 AAI 位复位为 0，以禁止任何新的写操作发生。WRDI 指令不会终止任何正在进行的编程操作。执行 WRDI 指令后，所有正在进行的编程操作都可以持续到 TBP。在执行 WRDI 指令之前，CE# 必须驱动为高电平。

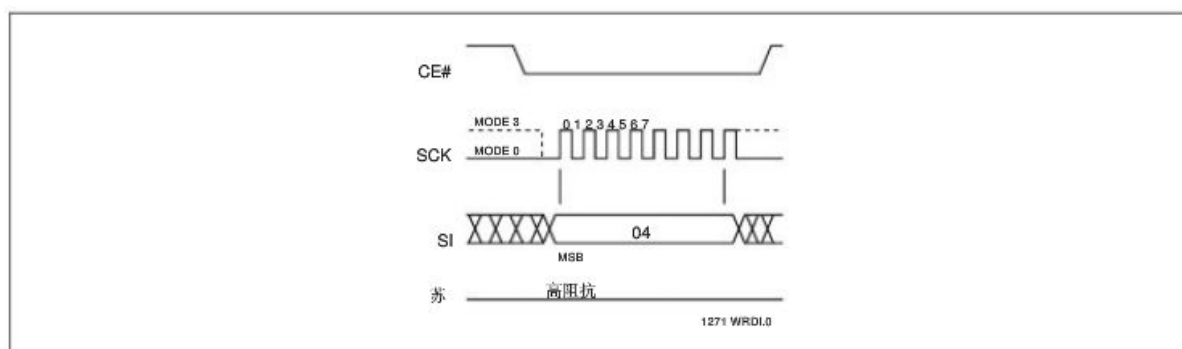


图 12 写禁止时序

4.27.7.11 Enable-Write-Status-Register (EWSR)、写状态寄存器 WRSR

写状态寄存器指令将新值写入状态寄存器的 BP3、BP2、BP1、BP0 和 BPL

位。在输入 WRSR (0x01) 指令的命令序列之前，需要先输入 EWSR (0x50) 使能写状态寄存器。

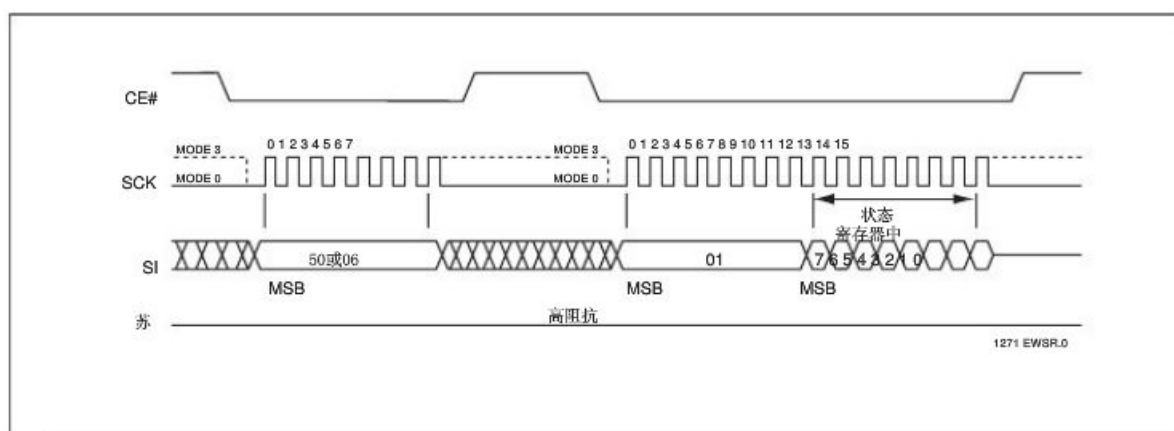


图 13 使能写状态寄存器时序

4.27.7.12 ID 号 (EWSR)、写状态寄

读 ID 指令 (RDID) 将器件标示为 SST25VF016B，将制造商标示为 SST。此命令向后兼容所有 SST25xFxxxA 器件，当在一个设计中使用多个版本的 SPI 串行闪存器件时，应将此命令用作默认器件标示命令。通过执行 8 位命令 90H 或者 ABH（后面紧跟地址位[A23-A0]）来读取器件信息。执行读 ID 指令之后，制造商 ID 位于地址 00000H，器件 ID 位于地址 00001H。

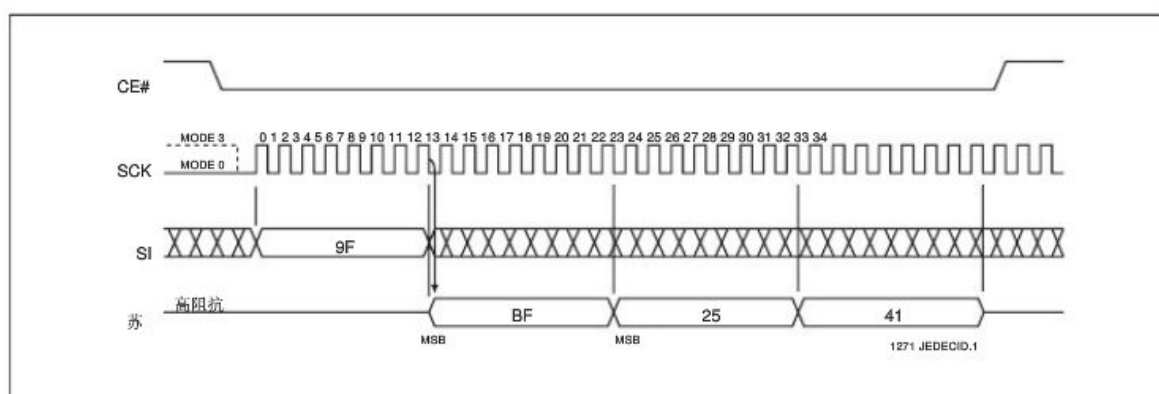


图 14 ID\写状态寄存器时序

器件处于读 ID 模式之后，制造商 ID 和器件 ID 输出数据在地址 00000H 和 00001H 之间翻转，直到被 CE#上低电平到高电平的转换终止。具体时序

参考图 14 ID\写状态寄存器时序。

4.27.8 产品订购信息

下面是芯片选型的详细信息，或者查找相关技术资料。

SST 25 VF 016B - 50 - 4C - S2AF

AA BB CCCC - DD - EE - FFFF

AA: 产品系列 (25=串行外设接口闪存)

BB: 电压 ($V=2.7\sim 3.6V$)

CCC: 器件容量 (016=16Mb)

DD: 工作频率 (50=50MHz)

EE: 最低可擦写次数 (4=10000 次)，温度范围 (C=商业级别)

FFFF: 封装类型 (S2=SOIC)，封装说明符 (A=8 引脚或触点)，环保特性 (F=无铅/无锡触点加工)