

W25X16/W25X32 中文资料

SPI 通信接口的 FLASH 中文的不是很多，虽然网络上例程很多，你也可以读写数据，你真的了解这个 FLASH 操作的细节吗？你使用到了最佳状态了吗？整理了下分享出来，希望对你有帮助。欢迎指正，共同学习。

W25X16 和 W25X32 都是 4K 字节的扇区。基本上都是一样的,封装和引脚都是兼容的。工作电压 (2.7V-3.6V)。

W25X16 为 16M 位 FLASH，由 8192 个 256 字节页组成，用页编程指令最多一次可以编程 256 个字节到 FLASH,可以通过扇区擦除指令一次可擦除 16 个页。块擦除指令一次可以擦除 256 个页。芯片擦除指令可以擦除全部的芯片存储空间。

W25X32 为 32M 位 FLASH，由 16384 个 256 字节页组成，用页编程指令最多一次可以编程 256 个字节到 FLASH,可以通过扇区擦除指令一次可擦除 16 个页。块擦除指令一次可以擦除 256 个页。芯片擦除指令可以擦除全部的芯片存储空间。

3. PIN CONFIGURATION 208-MIL

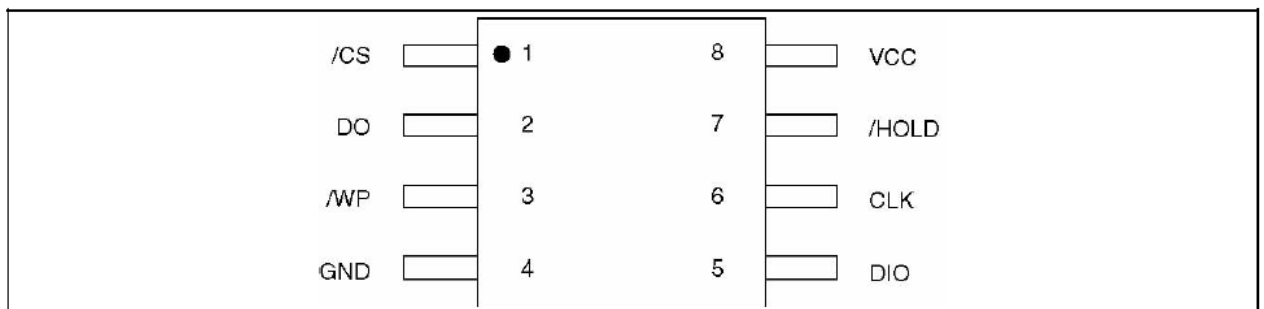


Figure 1a. W25X16 Pin Assignments, 8-pin SOIC (Package Code SS)

4. PIN DESCRIPTION 208-MIL

PAD NO.	PAD NAME	I/O	FUNCTION
1	/CS	I	Chip Select Input
2	DO	O	Data Output
3	/WP	I	Write Protect Input
4	GND		Ground
5	DIO	I/O	Data Input / Output
6	CLK	I	Serial Clock Input
7	/HOLD	I	Hold Input
8	VCC		Power Supply

6.2 片选引脚 (/CS)

CS 引脚高电平设备处于未选中状态：数据输出脚（DO）处于高阻状态，FLASH 为待机状态的功耗（除非内部在擦除，编程，或者状态寄存器处于编程状态）。

当 CS 脚跳变到低电平状态，FLASH 将被选中，功耗将会增加到激活水平，可以写入指令或数据至 FLASH 或从 FLASH 中读取数据或状态。在上电后，在一个新的指令接收前，CS 脚必须从高向低跳变。必要时在 CS 脚加一上拉电阻到 VCC（FLASH 上电期间有个阶段 CS 必须要维持 VCC 电压值 10us）。

6.3 串行数据输出脚（DO）

DO 脚用于输出 FLASH 的数据和状态，在时钟下降沿输出数据。

6.4 写保护引脚（/WP）

防止状态寄存器被意外写入。WP 脚结合（BP2，BP1 和 BP0）位和状态寄存器保护位（SRP），部分或者是全部的存储区域可以被硬件保护。WP 低电平有效。

6.5 维持引脚（/HOLD）

HOLD 引脚允许 FLASH 被选中的情况下暂停操作。当 CS 为低电平时，HOLD 跳变为低电平，DO 引脚将处于高阻状态，并且 DIO 和 CLK 脚将会忽略。当 HOLD 跳变为高电平，FLASH 恢复操作。当多个设备共享 SPI 信号线时，HOLD 功能比较有用。单个 FLASH 情况下直接接 VCC。

6.6 串行时钟引脚（CLK）

CLK 输入脚提供串行数据输入和输出操作的时钟。

6.7 串行数据输入/输出引脚（DIO）

DIO 脚用于指令，地址和数据的输入。数据在 CLK 上升沿锁存。当双输出快读指令（Fast read dual）被执行 DIO 也用做数据输出。

8.功能描述

8.1.2 双输出 SPI

用“双输出快读（0x3B）”指令，可以使用双数据输出模式。在上电后双输出能快速的将 FLASH 中的数据下载到 RAM 中。上电后通过这个指令快速加载数据到 RAM 是个不错注意。

8.1.3 维持功能

初始化进入维持状态（HOLD）：当 CS 为低电平，HOLD 下降沿进入维持（HOLD）状态。如果 HOLD 下降沿时 CS 不是低电平，维持状态（HOLD）将会在下一次时钟下降沿被激活。

退出维持功能状态（HOLD）：当 CS 为低电平，HOLD 上升沿退出维持（HOLD）转态。如果 HOLD 上升沿时 CS 不为低电平，下一个时钟下降沿将退出维持状态（HOLD）

8.2 写保护功能

应用程序使用非易失 FLASH 存储时，噪音和意外的系统状态可能会破坏数据的完整性。W25X16 提供了一些方法用于保护数据，防止意外的写入。

8. 2.1 写保护特点

W25X16 在上电或者掉电过程中，当 VCC 电压低于阈值电压 VWI (2V),将维持复位状态。当处于复位状态，所有的操作将被禁止并且不接收任何的指令。在上电 VCC 电压超过 VWI(2V)后，在 TPUW(10ms)这段时间所有的编程和擦除相关指令将会被忽略，这些指令包含写使能，页编程，扇区擦除，块擦除，芯片擦除和写状态寄存器指令。CS 脚必要的时候需要上拉一电阻到 VCC。

FLASH 上电后自动处于禁止写入状态（通过将状态寄存器的写锁存位（WEL）设置为 0）。在页编程，扇区擦除，芯片擦除或者写状态寄存器指令前，写使能指令必须先执行。在完成编程操作或者写指令后，写使能锁存位（WEL）自动清 0，继续处于写保护状态。

9.1 状态寄存器

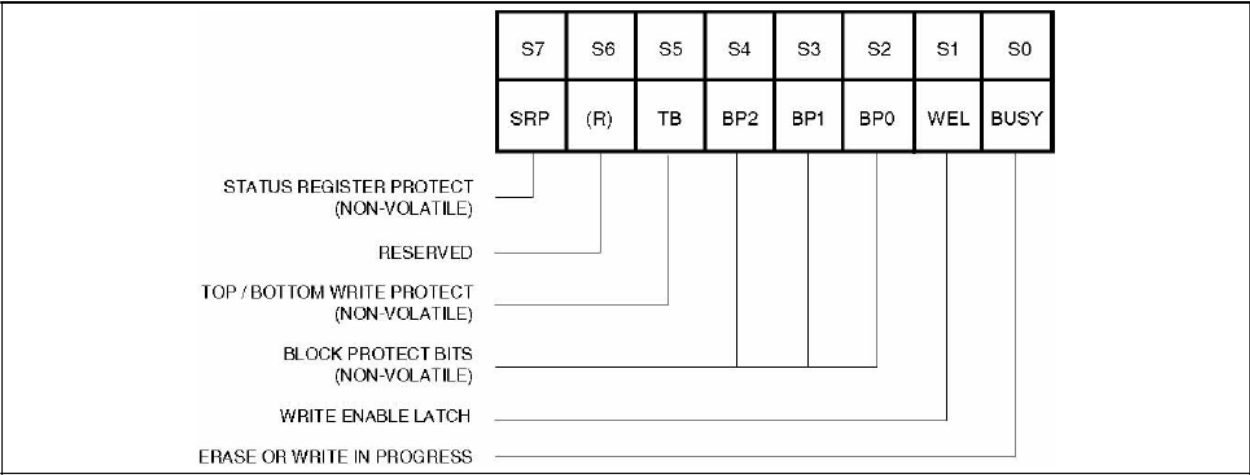


Figure 3. Status Register Bit Locations

9.1.1 忙标志位（BUSY）

在状态寄存器中 BUSY 位（S0）为只读位，当 FLASH 执行页编程，扇区擦除，块擦除，芯片擦除或者写状态寄存器指令时，BUSY 为 1。在 BUSY 为 1 的这段时间将会忽略除读状态寄存器指令外的所有指令。当编程，擦除，写状态寄存器指令完成，BUSY 自动清 0 指示设备已经准备好接收新的指令。

9.1.2 写使能锁存位（WEL）

在状态寄存器中 WEL 位（S1）为只读位，在执行写使能命令后 WEL 设置为 1。当设备写除能时 WEL 位清 0。上电或者写，除能，页编程，扇区擦除，块擦除，芯片擦除或者写状态寄存器后 WEL 位自动清 0。

9.1.3 块保护位（BP2，BP1，BP0）

块保护位在状态寄存器中的 (S4, S3, S2) 位是非易失可读写位, 提供了写保护控制和状态。通过写状态寄存器指令可以设置块保护位。出厂默认设置块保护位为 0, 存储阵列全部处于非保护状态。如果 WP 脚为低电平并且状态寄存器保护位 (SRP) 被设置为 1 时, 块保护位可以禁止写入。

9.1.4 顶部/底部块保护位 (TB)

块保护从顶部/底部开始保护数据。TB 位默认设置为 0。如果状态保护寄存器 (SRP) 位为 1 并且 WP 脚为低电平, TB 位将不能被写入。

9.1.6 状态保护寄存器位 (SRP)

SRP 位在状态寄存器中 (S7) 位是非易失性可读写位, SRP 可以结合 WP 脚禁止写状态寄存器。默认出厂 SRP 为 0, WP 引脚不控制状态寄存器读写操作。当 SRP 为 1 时, WP 位低电平将禁止写状态寄存器; 只有 WP 为高电平时才允许写状态寄存器指令。

STATUS REGISTER ⁽¹⁾				W25X16 (16M-BIT) MEMORY PROTECTION			
TB	BP2	BP1	BP0	BLOCK(S)	ADDRESSES	DENSITY (KB)	PORTION
x	0	0	0	NONE	NONE	NONE	NONE
0	0	0	1	31	1F0000h - 1FFFFFFh	512K-bit	Upper 1/32
0	0	1	0	30 and 31	1E0000h - 1FFFFFFh	1M-bit	Upper 1/16
0	0	1	1	28 thru 31	1C0000h - 1FFFFFFh	2M-bit	Upper 1/8
0	1	0	0	24 thru 31	180000h - 1FFFFFFh	4M-bit	Upper 1/4
0	1	0	1	16 thru 31	100000h - 1FFFFFFh	8M-bit	Upper 1/2
1	0	0	1	0	000000h - 00FFFFh	512K-bit	Lower 1/32
1	0	1	0	0 and 1	000000h - 01FFFFh	1M-bit	Lower 1/16
1	0	1	1	0 thru 3	000000h - 03FFFFh	2M-bit	Lower 1/8
1	1	0	0	0 thru 7	000000h - 07FFFFh	4M-bit	Lower 1/4
1	1	0	1	0 thru 15	000000h - 0FFFFFFh	8M-bit	Lower 1/2
x	1	1	x	0 thru 31	000000h - 1FFFFFFh	16M-bit	ALL

Note:

1. x = don't care

9.2 指令

W25X16 一套指令包含 15 个基本的指令, 这些指令完全通过 SPI 通信线进行控制。在 CS 的下降沿指令被初始化。输入 DIO 的第一个字节提供了指令代码。DIO 上输入的数据在 CKL 上升沿被采样。传输首位为高字节 (MSB)。

CS 上升沿指令完成。所有的读指令可以在任何时钟内完成。写, 编程或者擦除指令在一个字节的边界完成 (等待 8 个时钟输入后 CS 变高), 否则指令将会被终止。这个特征将保护 FLASH 被意外写入。当 FLASH 正在编程, 擦除或者状态寄存器正在被写, 除读状态寄存器指令外所有的指令将会被忽略, 一直到工作周期结束。

厂商 ID 和设备 ID

9.2.1 Manufacturer and Device Identification

MANUFACTURER ID	(M7-M0)	
Winbond Serial Flash	EFH	
Device ID	(ID7-ID0)	(ID15-ID0)
Instruction	ABh, 90h	9Fh
W25X16	14h	3015h
W25X32	15h	3016h

指令集

9.2.2 Instruction Set ⁽¹⁾

INSTRUCTION NAME	BYTE 1 CODE	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	N-BYTES
Write Enable	06h						
Write Disable	04h						
Read Status Register	05h	(S7-S0) ⁽¹⁾					(2)
Write Status Register	01h	S7-S0					
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	continuous
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(Next Byte) continuous
Fast Read Dual Output	3Bh	A23-A16	A15-A8	A7-A0	dummy	I/O = (D6,D4,D2,D0) O = (D7,D5,D3,D1)	(one byte per 4 clocks, continuous)
Page Program	02h	A23-A16	A15-A8	A7-A0 ⁽³⁾	(D7-D0)	(Next byte)	Up to 256 bytes
Block Erase (64KB)	D8h	A23-A16	A15-A8	A7-A0			
Sector Erase (4KB)	20h	A23-A16	A15-A8	A7-A0			
Chip Erase	C7h						
Power-down	B9h						
Release Power-down / Device ID	ABh	dummy	dummy	dummy	(ID7-ID0) ⁽⁴⁾		
Manufacturer/Device ID ⁽³⁾	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	
JEDEC ID	9Fh	(M7-M0) Manufacturer	(ID15-ID8) Memory Type	(ID7-ID0) Capacity			

注意：读状态寄存器指令 和 设备 ID 指令 将会持续的返回数据直到 CS 脚终止命令

9.2.3 除能写指令(0x04)

写使能指令设置状态寄存器的 WEL 位为 1。在页编程，扇区擦除，块擦除，芯片擦除或写状态寄存器指令前必须将 WEL 位设置为 1。

写使能指令进入系列：通过将 CS 驱动到低电平，将指令代码 (0x06) 串行输入 DI 脚，在每个 CLK 上升沿输入数据，最后将 CS 驱动为高电平。

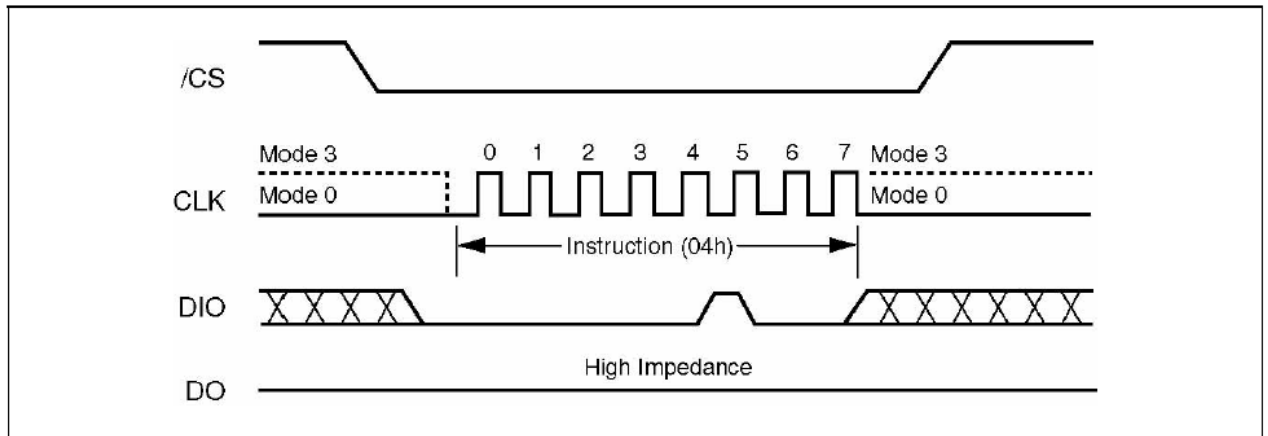


Figure 4. Write Disable Instruction Sequence Diagram

9.2.4 使能写指令 (0x06)

除能写指令复位状态寄存器的 WEL 位。除能写指令进入系列：驱动 CS 为低电平，发送指令代码 (0x04) 到 DIO 脚后驱动 CS 为高电平。上电后 WEL 为自动复位，完成写状态寄存器，编程数据，扇区擦除，块擦除和芯片擦除指令后 WEL 自动复位。

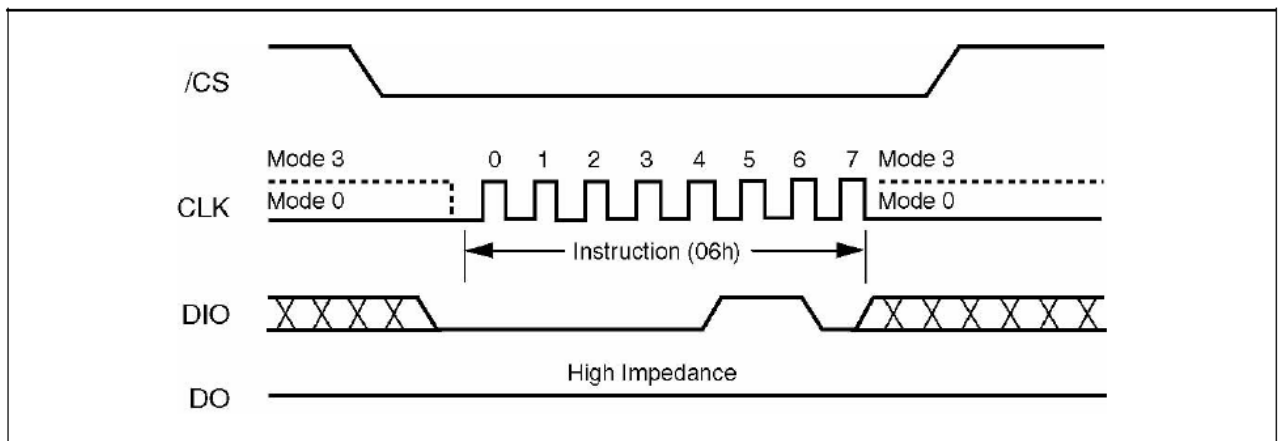


Figure 5. Write Enable Instruction Sequence Diagram

9.2.5 读状态寄存器 (0x05)

读状态寄存器指令允许读出 8 位状态寄存器值。

读状态寄存器指令序列：驱动 CS 为低电平，发送指令码 (0x05) 到 DIO 脚，CLK 上升沿输入数据；然后状态寄存器数据位通过 DO 脚移出，每个时钟的下降沿输出数据，高位 (MSB) 作为首位。

读状态寄存器指令可以用于任何时候，甚至编程，擦除，或者写状态寄存器正在处理的期间都可以执行读状态寄存器指令。BUSY 状态标志位用于检测内部操作周期是否已经完成，是否可以接受新的指令。读状态寄存器可以连续读回状态数据，通过将 CS 驱动为高电平结束这个指令。

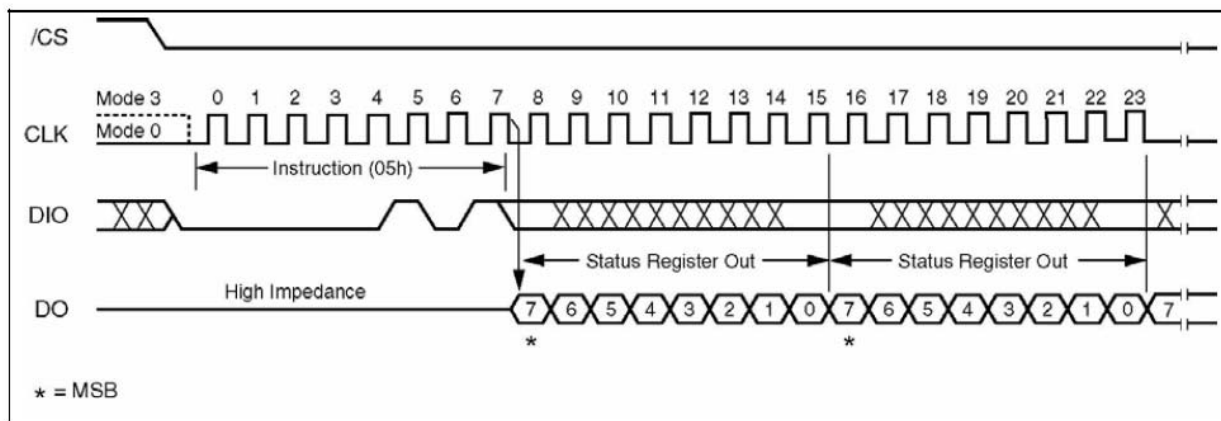


Figure 6. Read Status Register Instruction Sequence Diagram

9.2.6 写状态寄存器指令 (0x01)

写状态寄存器指令允许状态寄存器被写入。写状态寄存器前，写使能指令必须先执行。一旦写使能，通过驱动 CS 为低电平，发送指令代码 (0x01)，然后将数据写入状态寄存器。

仅仅非易失性状态寄存器位 SRP, TB, BP2, BP1 和 BP0 可以被写入。其它状态寄存器位为只读位。

在最后一个字节的第 8 位被锁存后，CS 脚必须驱动为高电平。如果不是这样做的，写状态寄存器指令将不会执行。在 CS 驱动为高电平后，内部的写状态寄存器周期将开始，并维持 t_w (5~15ms) 时间。当写状态寄存器指令周期正在进行，读状态寄存器指令任然可以执行检查 BUSY 位，在写状态寄存器周期期间 BUSY 为 1，完成写状态寄存器周期后 BUSY 为 0，此时可以接收新的指令。在写寄存器周期完成后状态寄存器中的 WEL 位将会自动清 0。

写状态寄存器指令通过 TB, BP2, BP1, BP0 位设置来配置存储区域是否被保护，或者部分存储区被保护，或者全部存储区域处于非保护状态。被保护的区域将会变为只读状态。

写状态寄存器指令也允许设置寄存器 (SRP)。SRP 为结合 WP 引脚除能写状态寄存器。当 SRP 位设置为 0 (出厂默认状态) WP 脚将不控制状态寄存器。当 SRP 为 1，WP 引脚为低电平，写状态寄存器指令将被忽略。当 SRP 为 1，WP 为高电平才允许写状态寄存器指令。

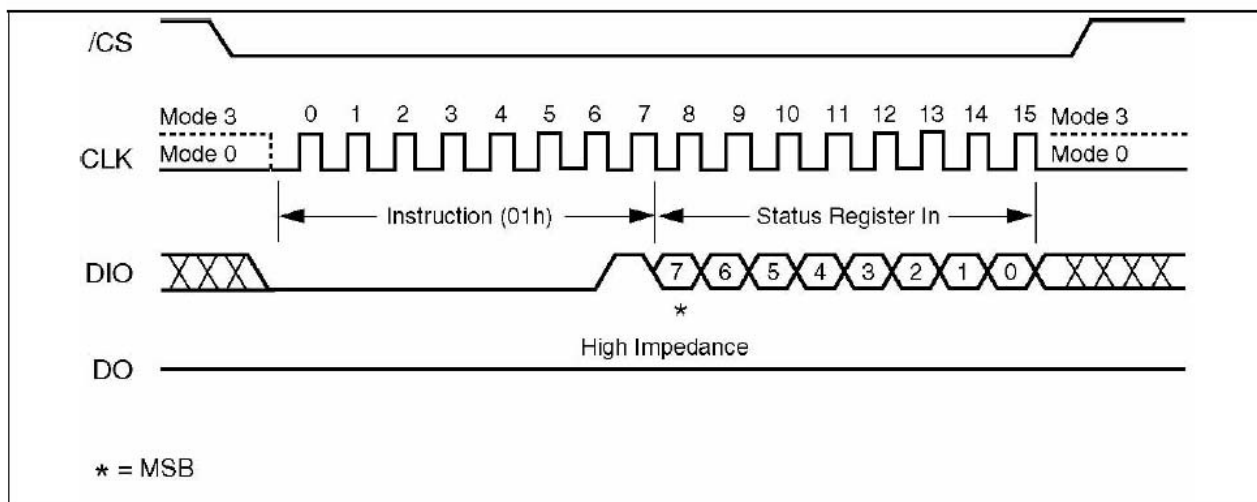


Figure 7. Write Status Register Instruction Sequence Diagram

9.2.7 读数据指令 (0x03)

读数据指令允许从 FLASH 读取连续的一个个字节。通过将 CS 驱动为低电平，然后发送指令代码 (0x03) 和 24 位地址到 DIO 引脚。发送的指令码和地址位在 CLK 上升沿被锁存。在地址被接受到后，该地址处的数据将会通过 DO 脚在 CLK 的下降沿输出，高位(MSB)在前。在每个数据字节被发送后地址会自动加 1 到下一个更高的地址。从而允许连续的数据流传输。允许通过提供连续 CLK 可以读取更多的数据。通过驱动 CS 为高电平结束此指令。当擦除，编程或者写周期正在进行 (BUSY=1)，读状态指令被执行将不会影响到当前的工作周期。

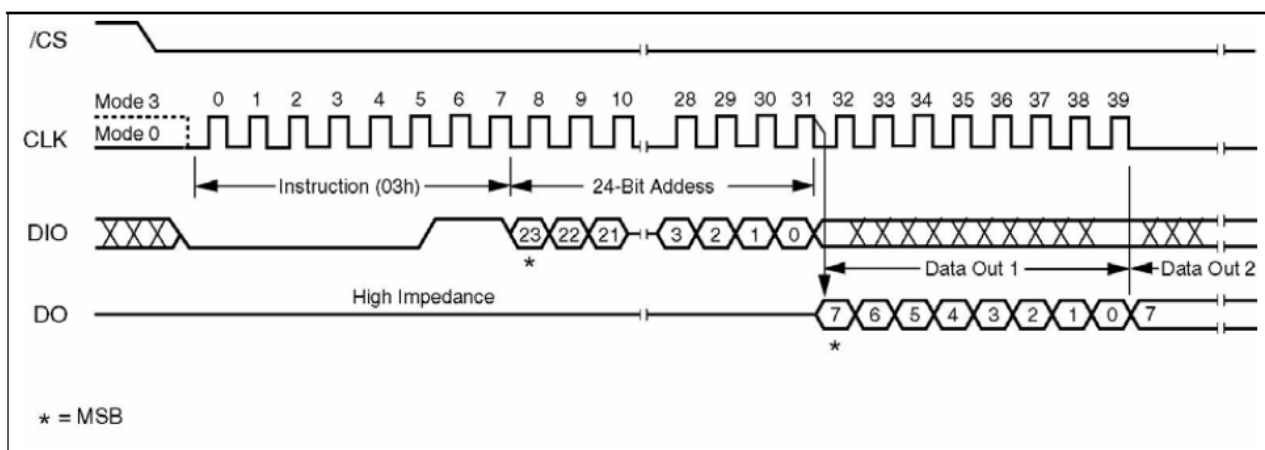


Figure 8. Read Data Instruction Sequence Diagram

9.2.8 快速读指令 (0x0b)

快速指令和读数据指令非常相似，快速指令可以工作在比较高的频率。通过在 24 位地址后加入 8 个时钟，进入快速读模式。发送 8 个多余的脉冲时，不用担心 DIO 上输入的数据。这些数据都会被忽略。

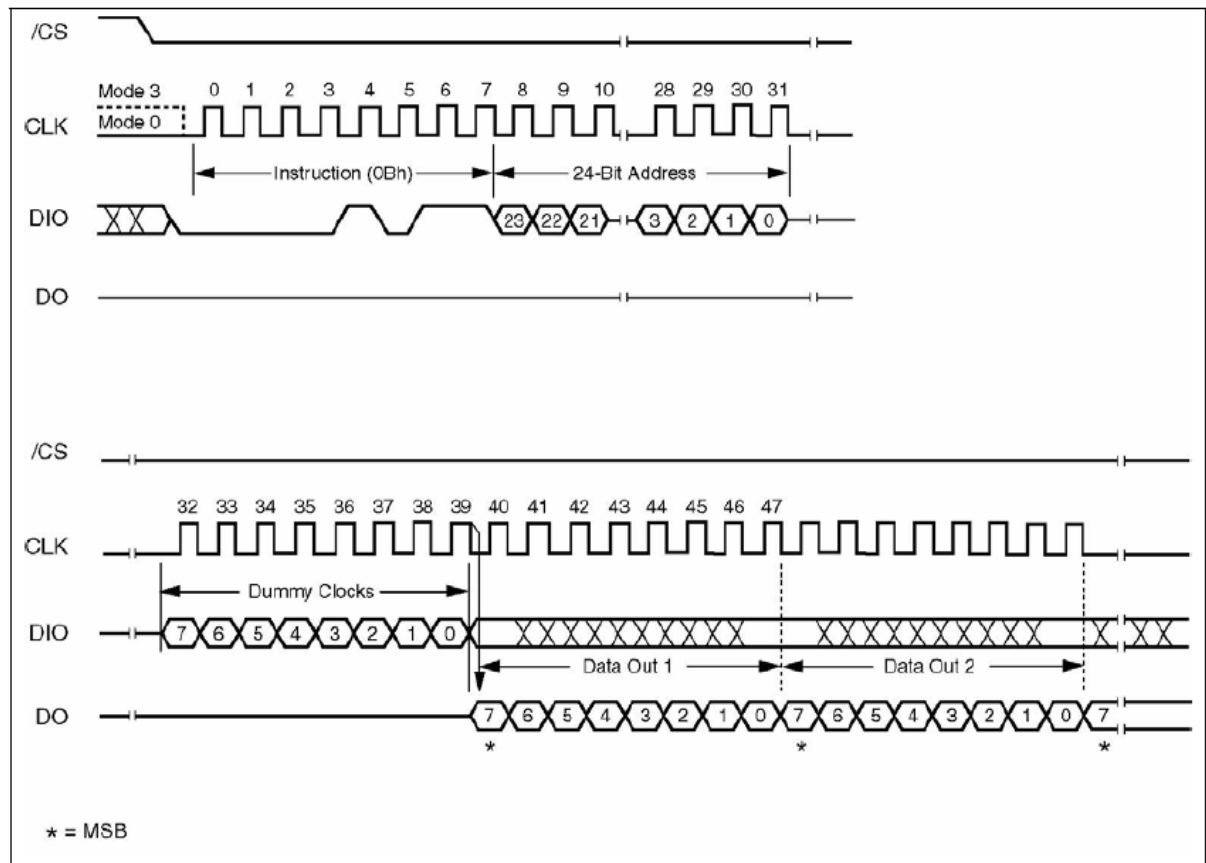
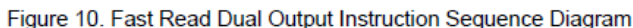


Figure 9. Fast Read Instruction Sequence Diagram

9.2.9 双输出快读 (0x3b)

双输出快读和标准的快读 (0x0b) 很相似，只是在 DO 和 DIO 两个引脚上同时输出数据。这样将是标准的设备速度的两倍。与标准快读一样，双输出快读指令工作在高频率下。这个通过在 24 位地址后加入额外的 8 个时钟。发送 8 个多余的脉冲时，不用担心 DIO 上输入的数据。这些数据都会被忽略。

在第一个数据输出时钟的下降沿前，DIO 脚将会处于高阻状态。



页编程指令允许最多 256 字节编程到已经擦除过的地址空间中。在执行页编程指令前，写使能指令必须先执行（状态寄存器的 WEL 位必须为 1）。通过驱动 CS 为低电平，然后发送指令代码（0x02）加 24 位地址加最少一个字节数据至 DIO 脚。CS 脚必须维持低电平在整个指令周期，直到数据已经发送到 FLASH。

同写指令和擦除指令一样，在最后字节的第 8 位被锁存后，CS 脚必须驱动为高电平。如果没有这样做页编程指令将不会执行。在 CS 驱动为高后，内部的页编程指令开始并维持 TPP(1.5-5ms)这段时间。当写周期正在进行，读状态寄存器指令任然可以执行检查 BUSY 位。在编程状态 BUSY 为 1。当编程周期结束 BUSY 将变为 0，此时 FLASH 已经准备好接收新的指令。在页编程指令周期结束后，在状态寄存器中的写保护锁存位 (WEL) 自动清零。如果地址页被写保护 (BP2, BP2, BP0)，页编程指令将不会执行。

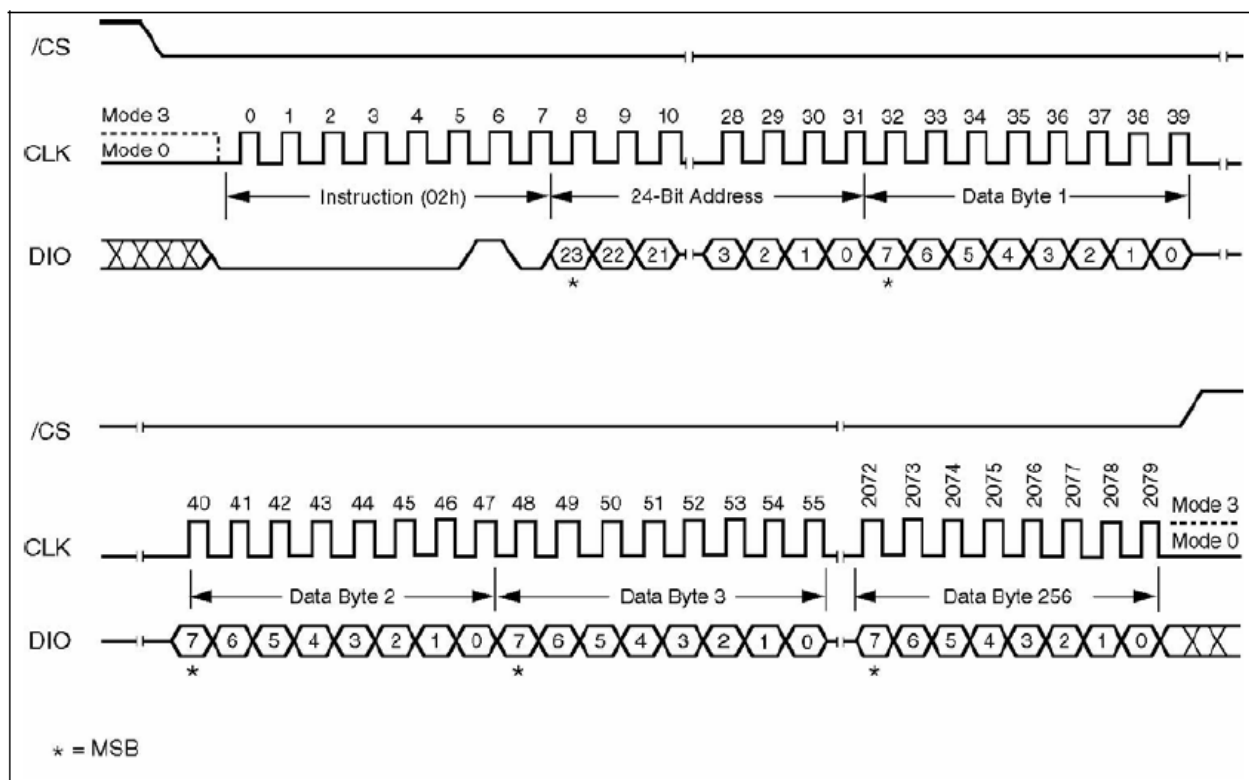


Figure 11. Page Program Instruction Sequence Diagram

9.2.11 扇区擦除 (0x20)

扇区擦除指令设置指定扇区 (4K) 存储的状态为 0xff。在扇区指令执行前, 必须执行写使能指令 (状态寄存器的 WEL 位必须为 1)。

扇区擦除指令进入序列: 通过驱动 CS 为低电平, 发送指令代码 (0x20) 和 24 位地址到 DIO 引脚。

在最后一个字节的第 8 位被锁存时, 必须驱动 CS 为高电平。如果不这样做, 扇区擦除指令将不会执行。在 CS 驱动为高后, 内部扇区擦除指令开始并维持 TSE(150ms-300ms) 这段时间。当扇区擦除周期正在执行, 读状态寄存器指令可以执行读取 BUSY 标志位。在扇区擦除周期期间 BUSY 为 1, 当扇区擦除周期结束 BUSY 变为 0, 此时 FLASH 可以接收新的指令。在扇区擦除周期已经完成后, 状态寄存器中的写使能位 (WEL) 自动清 0。如果页地址被保护 (TB, BP2, BP1, BP0) 扇区擦除指令将不会执行。

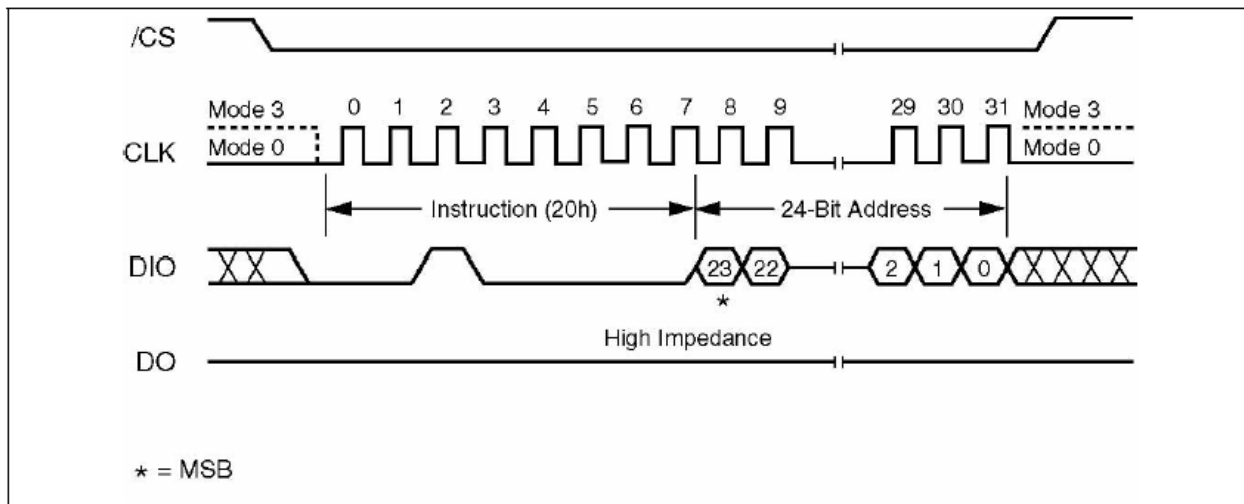


Figure 12. Sector Erase Instruction Sequence Diagram

9.2.12 块擦除指令 (0xd8)

块擦除指令设置指定的块存储区为 0xff。块擦除指令执行前，必须先执行写使能指令（状态寄存器中的 WEL 必须为 1）。

块擦除指令进入序列：通过驱动 CS 脚为低电平，发送指令代码 (0xd8) 和 24 位地址到 DIO 引脚。

在最后一个字节的第 8 位被锁存时，必须驱动 CS 为高电平。如果不这样做，块擦除指令将不会执行。在 CS 驱动为高后，内部块擦除指令开始并维持 TBE(1-2S)这段时间。当块擦除周期正在执行，读状态寄存器指令可以执行以读取 BUSY 标志位。在块擦除周期期间 BUSY 为 1，当块擦除周期结束 BUSY 变为 0，此时 FLASH 可以接收新的指令。在块擦除周期已经完成后，状态寄存器中的写使能位 (WEL) 自动清 0。如果页地址被保护 (TB, BP2, BP1, BPO) 块擦除指令将不会执行。

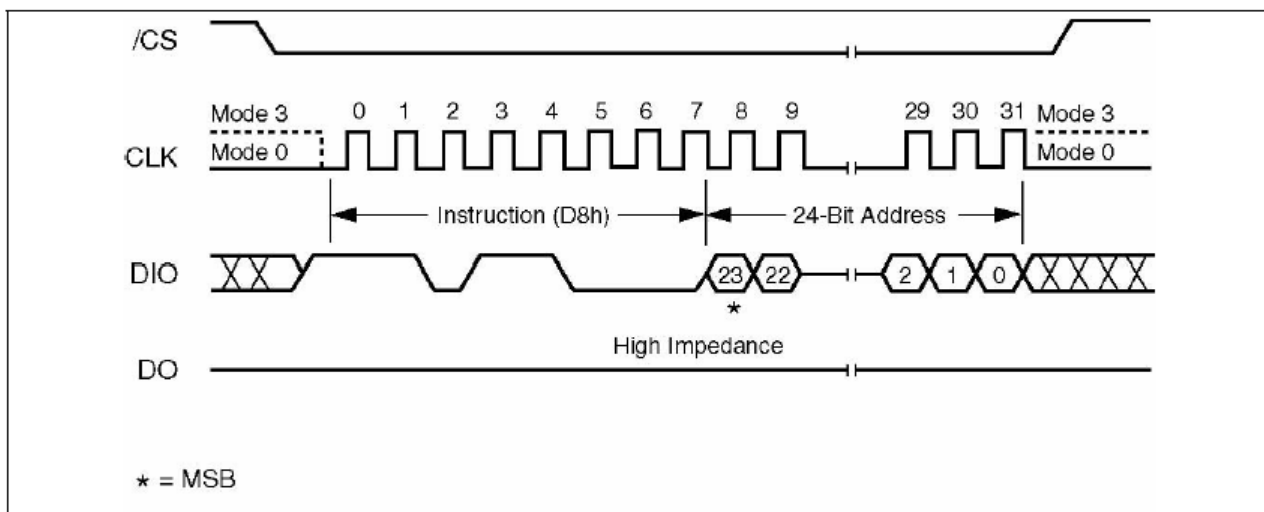


Figure 13. Block Erase Instruction Sequence Diagram

9.2.13 芯片擦除 (0xc7)

<https://shop60428517.taobao.com>

欢迎加 QQ 群: 81149997

芯片擦除指令设置 FLASH 中所有存储区为 0xff。芯片擦除指令被接收前，必须先执行写使能指令（状态寄存器中的 WEL 必须为 1）。

芯片擦除指令进入序列：通过驱动 CS 脚为低电平，发送指令代码（0xc7）和 24 位地址到 DIO 引脚。

在最后一个字节的第 8 位被锁存时，必须驱动 CS 为高电平。如果不这样做，芯片擦除指令将不会执行。在 CS 驱动为高后，内部芯片擦除指令开始并维持 TCE(W25X16 需要 15-40s, W25X32 需要 25-80s)这段时间。当芯片擦除周期正在执行，读状态寄存器指令可以执行读取 BUSY 标志位。在芯片擦除周期期间 BUSY 为 1，当芯片擦除周期结束 BUSY 变为 0，此时 FLASH 可以接收新的指令。在芯片擦除周期已经完成，状态寄存器中的写使能位（WEL）自动清 0。如果页地址被保护（BP2, BP1, BP0）芯片擦除指令将不会执行。

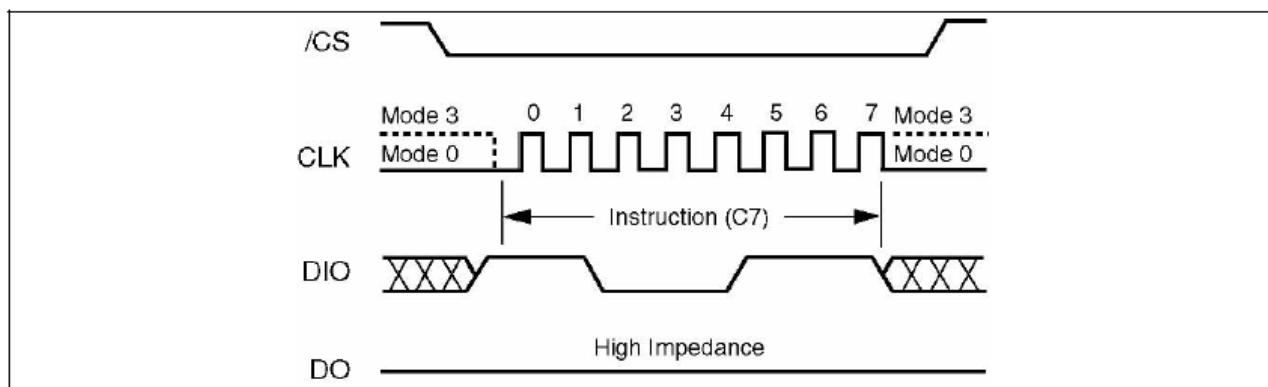


Figure 14. Chip Erase Instruction Sequence Diagram

9.2.14 电源休眠指令（0xb9）

尽管待机模式在常规运行下电流已经相当低，待机电流可以通过电源休眠指令进一步减小。电源休眠指令的低能量低功耗对电池供电应用非常有用。

电源休眠指令进入序列：通过驱动 CS 脚为低电平，发送指令代码（0xb9）。

在字节的第 8 位被锁存后，CS 脚必须驱动为高电平。如果不这样做，电源休眠指令将不会执行。在 CS 为高电平后，电源休眠状态将在 TDP(3ms)时间内进入。在电源休眠状态仅仅能接受**释放电源休眠/设备 ID (0xab) 指令**，包含读状态寄存器指令内的其它的指令都被忽略（在常规模式下读状态寄存器指令通常都是有效的，但是电源休眠状态下也是出于无效的）。**释放电源休眠/设备 ID (0xab) 指令**用于让 FLASH 恢复常规模式。FLASH 上电总是出于常规模式。

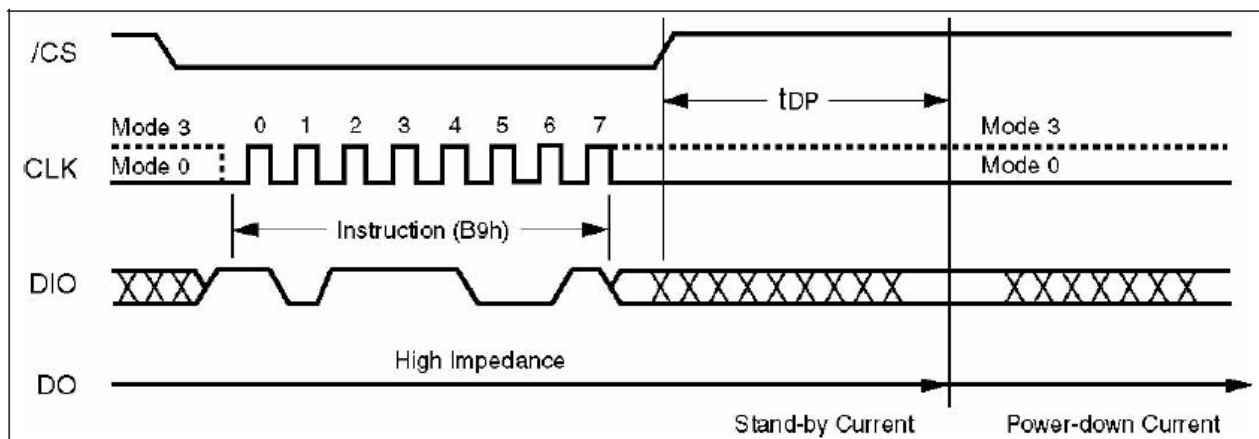


Figure 15. Deep Power-down Instruction Sequence Diagram

9.2.15 释放电源休眠/设备 ID 指令 (0xab)

释放电源休眠/设备 ID 指令是一个多功能指令。可以用来释放 FLASH 的电源休眠状态，获取设备 ID 号。

当仅仅用于释放 FLASH 的电源休眠状态，这个指令通过驱动 CS 为低电平，发送指令代码 (0xab) 然后驱动 CS 为高电平。在 TRES1(3ms)这段时间设备将恢复常规模式运行，并且其它的指令将会被接受。在 TRES1 期间，CS 脚必须维持为高电平。

当仅仅用于获取设备 ID 并且不是电源休眠状态，通过驱动 CS 为低电平，发送指令代码 (0xab) 和 3 个字节数据。设备 ID 位会在 CLK 的下降沿输出 (MSB 高位在前)。设备 ID 可以连续的读取，通过驱动 CS 为高电平结束此指令。

当用这个指令释放电源休眠状态和获取设备 ID,指令和前面的描述一样，只是 CS 驱动为高电平后必须维持 TRES2(1.8ms)这段时间。在这段时间后设备将会恢复到常规模式并且其它指令将会被接收。

在擦除，编程或者写操作周期内，如果释放电源休眠/设备 ID 指令被执行，这个指令将被忽略，并且不会影响到当前的工作周期。

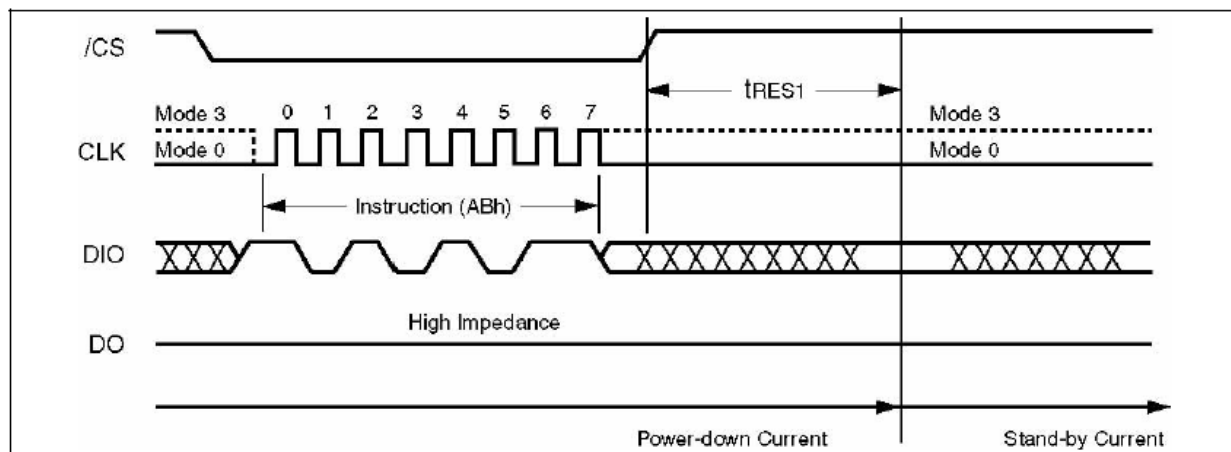
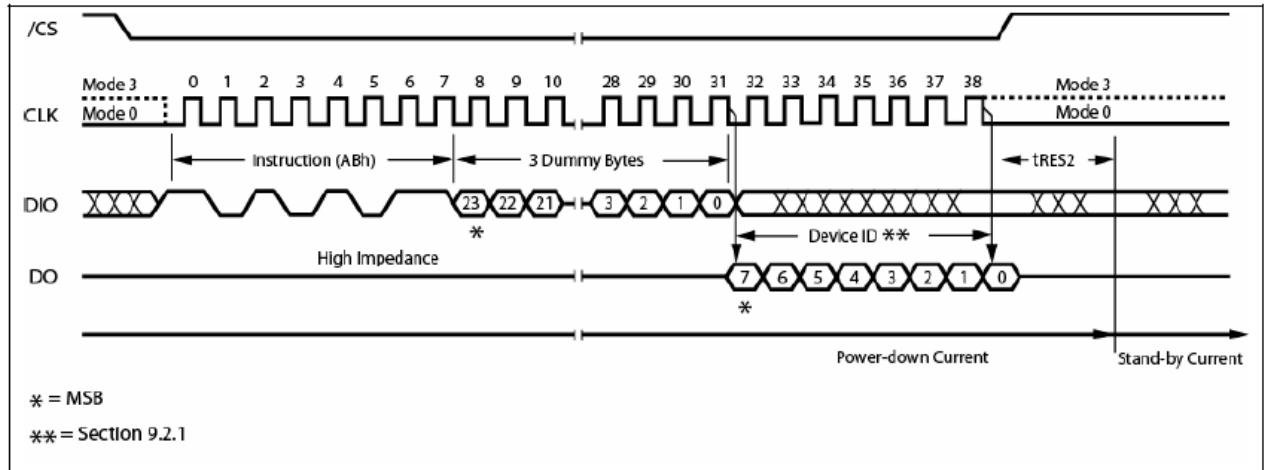


Figure 16. Release Power-down Instruction Sequence



9.2.16 读制造商/设备 ID 指令(0x90)

读制造商/设备 ID 指令和释放电源休眠/设备 ID 指令非常相似。通过驱动 CS 为低电平，发送指令代码 (0x90) 和 24 位地址 (0x000000)。华邦的制造商 ID(0xef)和设备 ID 在时钟的下降沿输出，高位 (MSB) 在前。如果 24 位地址被初始化为 0x000001,将会先读取设备 ID，然后是制造商 ID。设备 ID 和制造商 ID 可以被连续的读出，通过驱动 CS 为高电平结束此指令。

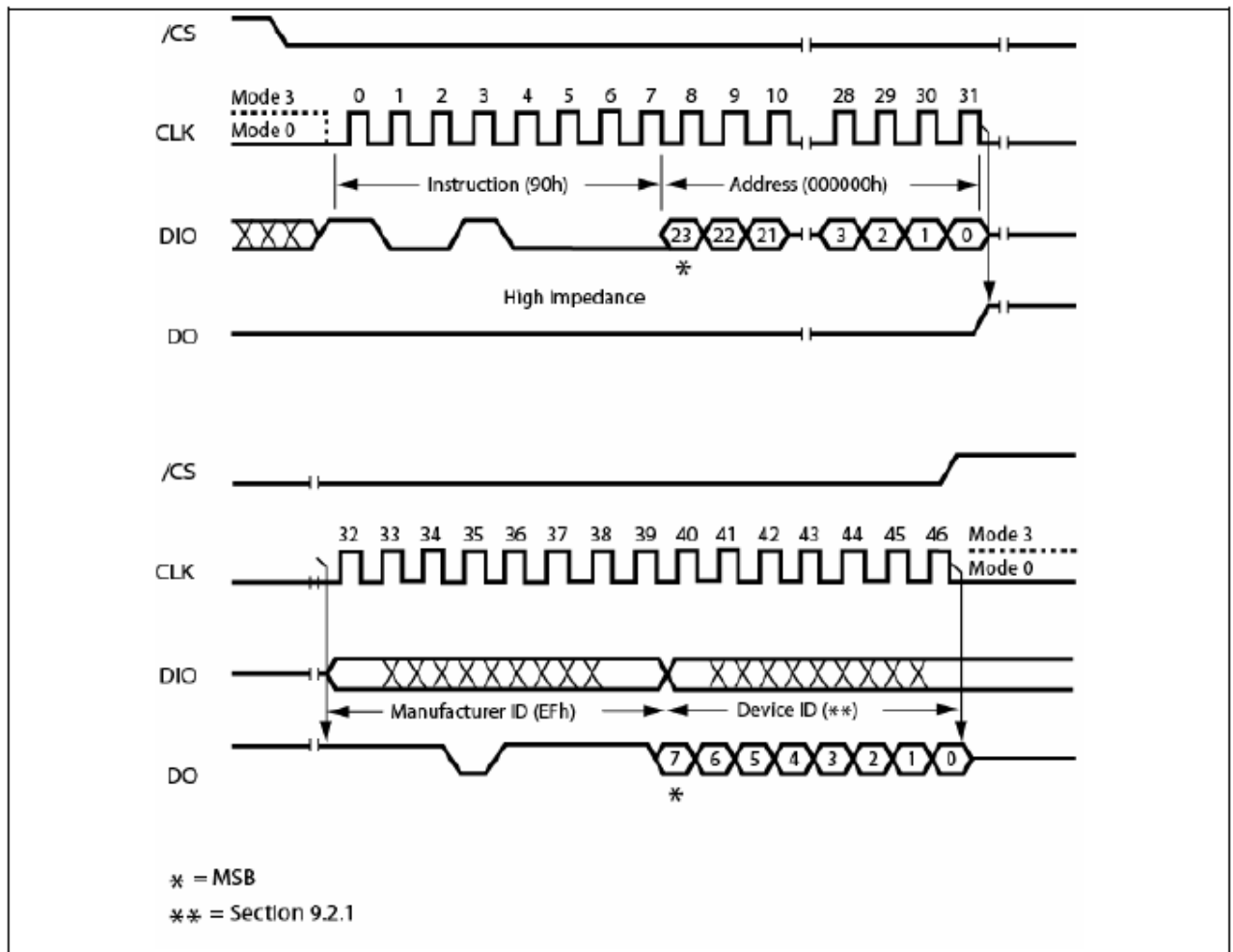


Figure 18. Read Manufacturer / Device ID Diagram

9.2.17 JEDEC ID 指令(0x9f)

为了兼容性，W25X16/32 提供了几种指令来检测设备。

JEDEC ID 指令进入系列：通过驱动 CS 为低电平，发送指令代码（0x9f）。JEDEC 通过 CLK 下降沿输出数据,(MSB)高字节在先。

JEDEC 分布为：一字节厂商(华邦)ID，两字节设备 ID(ID15-ID8 为存储器类型，ID7-ID0 位兼容能力)

9.2.1 Manufacturer and Device Identification

MANUFACTURER ID	(M7-M0)	
Winbond Serial Flash	EFH	
Device ID	(ID7-ID0)	(ID15-ID0)
Instruction	ABh, 90h	9Fh
W25X16	14h	3015h
W25X32	15h	3016h

10.4 上电时间和禁能写门槛值

参数	标识	最小	最大	单位
CS 必须保持 VCC 电压值	TVST	10		us
写指令前延时	TPUW	1	10	ms
禁能写门槛电压	VWI	1	2	v

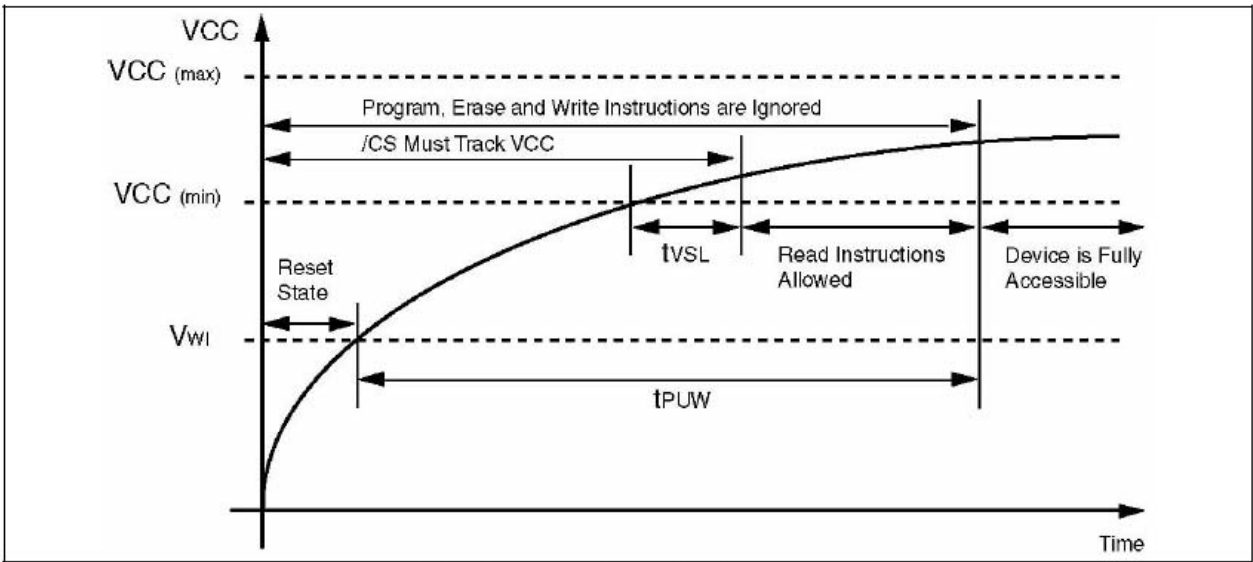


Figure 20. Power-up Timing and Voltage Levels

功耗:

参数	最小	典型	最大	单位
待机模式电流		25	50	uA
电源休眠模式电流		<1	5	uA
1MHZ 读数据		5	8	mA
50MHZ 读数据		20	25	mA
75MHZ 读数据		25	30	mA
页编程		15	25	mA
写状态寄存器		20	25	mA
扇区或块擦除		20	25	mA
芯片擦除		20	25	mA

交流电气特性