深圳市雅创芯瀚电子科技有限公司
SHENZHEN ASTRONG-TECH CO., LTD

AST25QW256S宽电压SPI Flash存储器数据手册

服务电话：13538015750　13691641629

目录

1 产品简介

1.1 概述

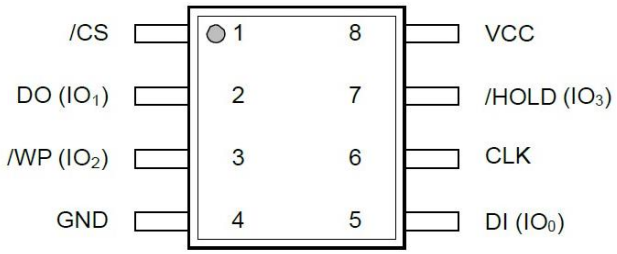
AST25QW256S 是一款容量为 256Mb 的 SPI NOR Flash 存储器，其工作电压为 1.65～3.6V。指令、地址和输入数据在时钟信号的上升沿输入芯片，经过内部控制逻辑处理后，输出数据在时钟信号的下降沿输出芯片。芯片支持标准 SPI、双 SPI 和四 SPI 指令，读指令(03h/13h)的最大工作频率为 66MHz，其它指令的最大工作频率可达 133MHz，可通过双 SPI、四 SPI 指令进一步提高数据输出带宽。芯片支持 4KB 扇区、32KB 扇区、64KB 扇区和全芯片擦除指令，通过页编程指令可进行 1~256 字节数据编程，在芯片工作过程中，可通过读取状态寄存器的值来查询芯片工作状态。

AST25QW256S 兼容华邦公司的 W25Q256JV， W25Q256JW 同封装形式的芯片。

1.2 产品特性

- 工作电压：1.65 V～3.6V
- 工作温度：-55℃～+125℃
- 支持标准 SPI/双 SPI/四 SPI 指令
- 支持 4 字节地址指令和 4 字节地址模式
- 灵活的扇区结构(4KB/32KB/64KB)
- 支持 1～256 字节大小的页编程
- 最高工作频率可达 133MHz
- 用户可配置读等待周期
- 256 字节页编程时间：0.5ms（典型值）
- 4KB 扇区擦除时间：40ms（典型值）
- 32KB 扇区擦除时间：120ms（典型值）
- 64KB 扇区擦除时间：250ms（典型值）
- 全芯片擦除时间：100s（典型值）
- 擦/写循环次数：10 万次
- 数据保持时间：20 年
- 支持 SOP8/SOP16/DFN8 封装形式
- 质量等级：参考 GJB7400《合格制造厂认证用半导体集成电路通用规范》中的 N1 级要求

1.3 引脚排布和说明



SOP8 管脚图（顶视图）

引脚名称	引脚类型	引脚功能描述
/CS	输入	片选信号，低电平有效
CLK	输入	串行时钟
DI(IO0)	双向	SPI：输入；双 SPI：双向；四 SPI：双向
DO(IO1)	双向	SPI：输出；双 SPI：双向；四 SPI：双向
/WP(IO2)	双向	SPI：输入；双 SPI：输入；四 SPI：双向
/HOLD(IO3)	双向	SPI：输入；双 SPI：输入；四 SPI：双向
VCC	—	1.65~3.6 电源
GND	—	地
NC	—	内部无连接

图 1-1 引脚排布和说明

2 功能描述

2.1 寄存器

芯片提供了 4 个寄存器供用户访问，用户可以从这些寄存器中读出芯片的状态信息，也可以向这些寄存器中写入数据以控制芯片的功能，下面分别介绍每个寄存器的含义及用法。

2.1.1 状态寄存器

芯片向用户提供一个 8 位的状态寄存器供用户查询芯片的状态信息，状态寄存器的格式如下所示。

7	6	5	4	3	2	1	0
SRP	TB	BP3	BP2	BP1	BP0	WEL	BUSY

BUSY(只读)：状态寄存器的 BUSY 位是只读位，表示芯片是否正在进行内部写操作。如果用户读出的 BUSY 数据为 1，表示芯片正在进行编程、擦除或写寄存器等操作。在此期间，用户写入的所有指令（“读状态寄存器”指令除外）都将被芯片忽略。当内部写操作完成后，BUSY 位将变为 0，用户可通过查询该位的状态来判断写操作是否完成。

WEL(只读)：状态寄存器的 WEL 位是只读位，当上电复位或软件复位完成后，WEL 位为 0。当芯片成功执行“写使能”(06h)指令后，WEL 位将会被设置为 1。为了防止意外改写芯片中存储的数据，在 WEL 为 0 时用户输入的编程、擦除和写寄存器指令都将被芯片忽略。因此用户在输入相应指令前，应该先使用“写使能”指令将 WEL 设置为 1。当编程/擦除/写寄存器指令正确完成后，WEL 位将被芯片自动设置为 0。

BP3~BP0，**TB**：状态寄存器的 BP 和 TB 位为非易失位，共同实现对主存储区的写保护，具体功能见下表。

TB	BP[3:0]	写保护扇区（64KB）	写保护地址范围
X	0000	无	无
0	0001	511	1FF0000h—1FFFFFFh
0	0010	510—511	1FE0000h—1FFFFFFh
0	0011	508—511	1FC0000h—1FFFFFFh
0	0100	504—511	1F80000h—1FFFFFFh
0	0101	496—511	1F00000h—1FFFFFFh
0	0110	480—511	1E00000h—1FFFFFFh
0	0111	448—511	1C00000h—1FFFFFFh
0	1000	384—511	1800000h—1FFFFFFh
0	1001	256—511	1000000h—1FFFFFFh
1	0001	0	0000000h—000FFFFh
1	0010	0—1	0000000h—001FFFFh
1	0011	0—3	0000000h—003FFFFh
1	0100	0—7	0000000h—007FFFFh

TB	BP[3:0]	写保护扇区（64KB）	写保护地址范围
1	0101	0—15	0000000h—00FFFFFh
1	0110	0—31	0000000h—01FFFFFh
1	0111	0—63	0000000h—03FFFFFh
1	1000	0—127	0000000h—07FFFFFh
1	1001	0—255	0000000h—0FFFFFFh
X	1010~1111	0—511	0000000h—1FFFFFFh

SRP：状态寄存器的 SRP 位为非易失位，结合配置寄存器的 SRL 位和/WP 管脚可实现对状态、配置和控制寄存器的写保护，具体功能如下表所示。

SRL	SRP	/WP	保护状态	说明
0	0	X	未保护	
0	1	0	写保护	
0	1	1	未保护	
1	X	X	写保护	只有掉电复位才能将 SRL 从 1 变为 0

2.1.2 配置寄存器

芯片向用户提供一个 8 位的配置寄存器用于控制芯片的行为，配置寄存器的格式如下所示。

7	6	5	4	3	2	1	0
RSV	CMP	RSV	RSV	RSV	RSV	QE	SRL

SRL：配置寄存器的 SRL 位为易失位，上电复位完成后 SRL 的值为 0，表示未使用该位对状态、配置和控制寄存器进行写保护。当用户使用写配置寄存器指令将 SRL 写为 1 后，无论 SRP 和/WP 为何值，状态、配置和控制寄存器都将被写保护，且用户无法再使用写配置寄存器指令将 SRL 写为 0，只有掉电然后重新上电才可以将 SRL 值改写为 0。

QE：配置寄存器的 QE 位为非易失位，用于使能/WP 和/HOLD 功能。当 QE 为 0 时，/WP 和/HOLD 管脚功能有效；当 QE 为 1 时（出厂值）/WP 和/HOLD 功能无效。需要注意的是：在使用四输出或四 I/O 指令之前，必须将 QE 设置为 1。

CMP：配置寄存器的 CMP 位为非易失位，用于提供更灵活的主存储区写保护方案。CMP 需要结合 TB 和 BP 共同实现灵活的主存储区写保护方案。当 CMP 为 0 时，TB 和 BP 的任一组合将会对主存储区中的某部分扇区进行写保护（相应扇区组合假设为 A），而剩下的扇区未被写保护（相应扇区组合假设为 B）。保持 TB 和 BP 不变，然后将 CMP 设置为 1 后，A 将失去写保护，而 B 将被写保护。

RSV：保留位，在读配置寄存器时，忽略这些位的值；在写配置寄存器时，请向这些位写入 0。

2.1.3 控制寄存器

芯片向用户提供一个 8 位的控制寄存器用于控制芯片的行为，控制寄存器的格式如下所示。

7	6	5	4	3	2	1	0
RSV	DRV1	DRV0	DC1	DC0	RSV	ADP	ADS

ADS(只读)：控制寄存器的 ADS 位为只读位，用于表示芯片当前的地址模式，如果 ADS 位为 1，表示芯片处于 4 字节地址模式；如果 ADS 位为 0，表示芯片处于 3 字节地址模式。

ADP：控制寄存器的 ADP 位为非易失位，用于控制芯片上电复位/软件复位后使用的地址模式。如果 ADP 为 0（出厂模式），那么上电复位/软件复位后芯片将处于 3 字节地址模式；如果 ADP 为 1，那么上电复位/软件复位后芯片将处于 4 字节地址模式。

DRV1~DRV0：控制寄存器的 DRV1 和 DRV0 为非易失位，用于控制芯片输出 IO 的驱动能力，具体设置及对应的驱动能力见下表。

DRV1	DRV0	驱动能力
0	0	100%
0	1	75%(出厂模式)
1	0	50%
1	1	25%

DC1~DC0：控制寄存器的 DC1 和 DC0 为非易失位，用于设置双 I/O （BBh/BCh）和四 I/O(EBh/ECh)快速读指令所需的等待周期数。通过设置这两个非易失位，可以满足不同用户对最快时钟频率或最小输出延迟的需求。需要注意的是：快速读(0Bh/0Ch)、2 输出快速读(3Bh/3Ch)和 4 输出快速读(6Bh/6Ch) 指令所需的等待周期数为固定的 8 周期。DC 的具体设置及对应的等待周期数见下表。

DC[1:0]	双 IO 快速读（BBh/BCh）		四 IO 快速度（EBh/ECh）	
	等待周期数	最大频率	等待周期数	最大频率
00(出厂值)	4	108MHz	6	108MHz
01	8	133MHz	4	54MHz
10	4	108MHz	8	133MHz
11	8	133MHz	10	133MHz

RSV：保留位，在读配置寄存器时，忽略这些位的值；在写配置寄存器时，请向这些位写入 0。

2.1.4 地址扩展寄存器

芯片向用户提供一个 8 位的地址扩展寄存器用于后向兼容部分使用 3 字节地址的指令，寄存器的格式如下所示。

7	6	5	4	3	2	1	0

RSV	RSV	RSV	RSV	RSV	RSV	RSV	A24

3 字节地址最大只能访问 128Mb 的存储空间，结合地址扩展寄存器的第 0 位（A24）可实现对 256Mb 存储空间的访问。当 A24 为 0 时，所有 3 字节地址指令将访问芯片的低 128Mb 存储空间。用户可使用“写地址扩展寄存器”指令将 A24 设置为 1，此时所有 3 字节地址指令将访问芯片的高 128Mb 存储空间。当芯片工作于 4 字节地址模式时（ADS=1），所有指令都将使用 4 字节地址，地址扩展寄存器中的值将被忽略。需要注意的是，所有使用 4 字节地址的指令执行后，地址扩展寄存器都将被指令所使用的地址更新，因此从 4 字节地址模式返回到 3 字节地址模式后，用户需仔细检查地址扩展寄存器的 A24 是否是自己所期望的值。

2.2 指令说明

功能	命令（写）	地址（写）	数据（写）	等待周期	数据（读）
写使能	06h	—	—	—	—
写除能	04h	—	—	—	—
读状态寄存器	05h	—	—	—	1 字节
写状态寄存器	01h	—	1 字节	—	—
读配置寄存器	35h	—	—	—	1 字节
写配置寄存器	31h	—	1 字节	—	—
读控制寄存器	15h	—	—	—	1 字节
写控制寄存器	11h	—	1 字节	—	—
读地址扩展寄存器	C8h	—	—	—	1 字节
写地址扩展寄存器	C5h	—	1 字节	—	—
进入低功耗模式	B9h	—	—	—	—
退出低功耗模式	ABh	—	—	—	—
进入 4 字节地址模式	B7h	—	—	—	—
退出 4 字节地址模式	E9h	—	—	—	—
软件复位使能	66h	—	—	—	—
软件复位	99h	—	—	—	—
读数据	03h	3/4 字节	—	—	1-∞字节
	13h	4 字节	—	—	1-∞字节
快速读数据	0Bh	3/4 字节	—	8 周期	1-∞字节
	0Ch	4 字节	—	8 周期	1-∞字节
双输出快速读数据	3Bh	3/4 字节	—	8 周期	1-∞字节
	3Ch	4 字节	—	8 周期	1-∞字节
四输出快速读数据	6Bh	3/4 字节	—	8 周期	1-∞字节
	6Ch	4 字节	—	8 周期	1-∞字节
双 I/O 快速读数据	BBh	3/4 字节	—	4 周期	1-∞字节
	BCh	4 字节	—	4 周期	1-∞字节

功能	命令（写）	地址（写）	数据（写）	等待周期	数据（读）
四 I/O 快速读数据	EBh	3/4 字节	—	6 周期	1-∞字节
	ECh	4 字节	—	6 周期	1-∞字节
单输入页编程	02h	3/4 字节	1-256 字节	—	—
四输入页编程	32h	3/4 字节	1-256 字节	—	—
4KB 扇区擦除	20h	3/4 字节	—	—	—
32KB 扇区擦除	52H	3/4 字节	—	—	—
64KB 扇区擦除	D8h	3/4 字节	—	—	—
全芯片擦除	60h/C7h	—	—	—	—

2.2.1 写使能指令(06h)

为了防止意外改写芯片中存储的数据，在执行编程、擦除或写寄存器操作之前必须使用“06h”指令将状态寄存器的 WEL 设置为 1，否则相应的操作指令将被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“06h”从 IO0 输入芯片，最后将/CS 信号置为高电平。

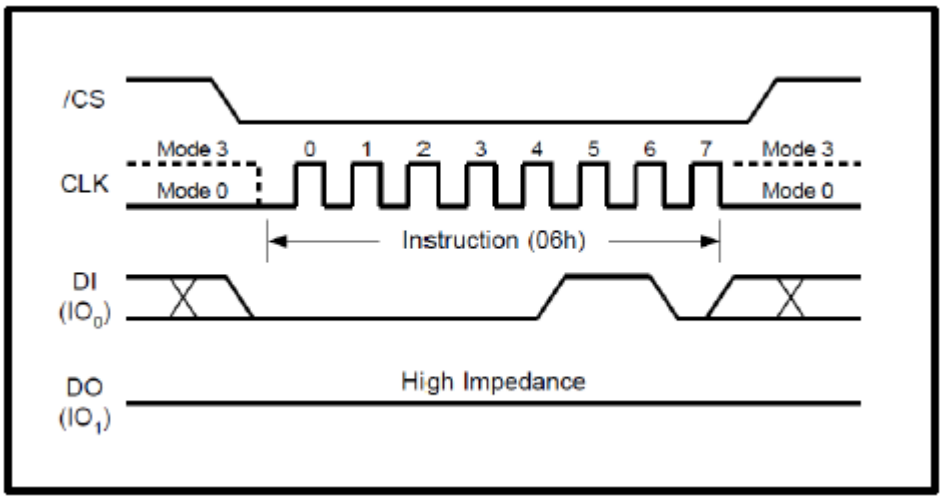


图 2-1 “06h”指令时序图

2.2.2 写除能指令(04h)

“04h”指令可将状态寄存器的 WEL 设置为 0。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“04h”从 IO0 输入芯片，最后将/CS 信号置为高电平。需要注意的是，当上电复位或软件复位完成后， WEL 的状态为 0；当编程、擦除或写寄存器操作完成后，WEL 也将自动被芯片控制逻辑设置为 0。

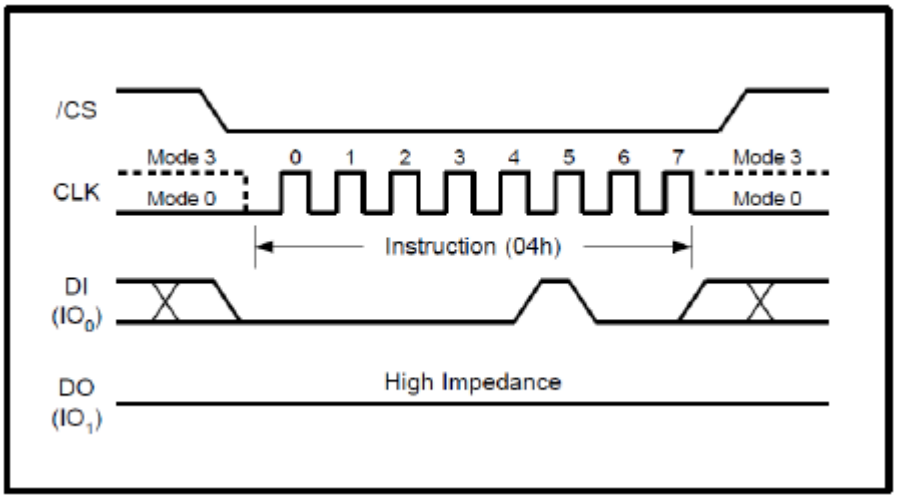


图 2-2“04h”指令时序图

2.2.3 读状态/配置/控制寄存器(05h/35h/15h)

用户可以使用指令来读取芯片状态/配置/控制寄存器的值。指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“05h/35h/15h”从 IO0 输入芯片，随后寄存器的值将在 CLK 信号的下降沿从 IO1 输出芯片（高位先输出），最后将/CS 信号置为高电平完成读操作。需要注意的是，将/CS 置为高电平之前如果一直提供 CLK 信号，那么寄存器中的数据将周期性的从 IO1 输出芯片。

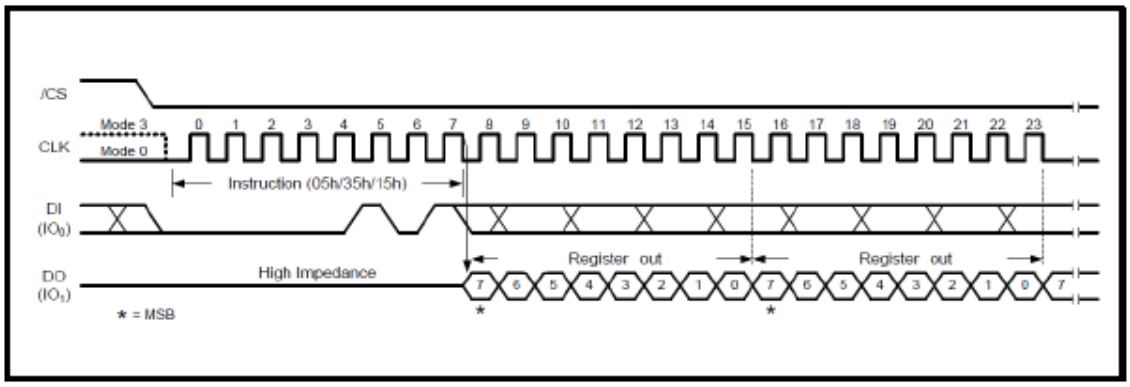


图 2-3 “05h/35h/15h”指令时序图

2.2.4 写状态/配置/控制寄存器(01h/31h/11h)

用户可以使用指令来改变状态/配置/控制寄存器的值，在使用相应指令之前，必须先使用“写使能”指令将状态寄存器的 WEL 设置为 1，否则，后续的写寄存器指令将被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“01h/31h/11h”和数据从 IO0 输入芯片，最后将/CS 信号置为高电平。/CS 信号变为高电平后，芯片内部控制电路便开始使用特定的算法将输入数据写入非易失寄存器，内部写操作将在 t_W 时间内完成，在此期间，用户可以通过检查状态寄存器的 BUSY 位来判断写操作是否完成。

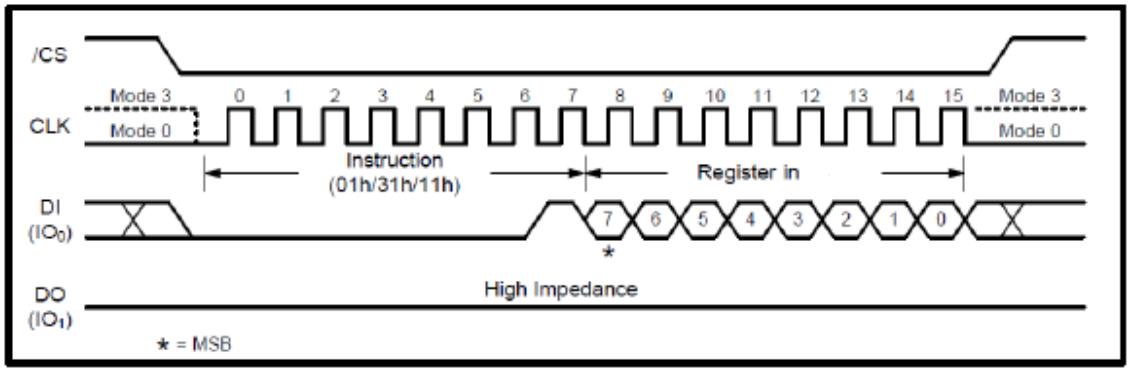


图 2-4 “01h/31h/11h”指令时序图

2.2.5 读地址扩展寄存器(C8h)

当芯片工作于 3 字节地址模式时，地址扩展寄存器可以扩展 3 字节地址指令可访问的地址空间。用户可以使用指令读取地址扩展寄存器的值，指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码 “C8h”从 IO0 输入芯片，随后地址扩展寄存器的值将在 CLK 信号的下降沿从 IO1 输出芯片（高位先输出），最后将/CS 信号置为高电平完成读操作。需要注意的是，将/CS 置为高电平之前如果一直提供 CLK 信号，那么地址扩展寄存器中的数据将周期性的从 IO1 输出芯片。

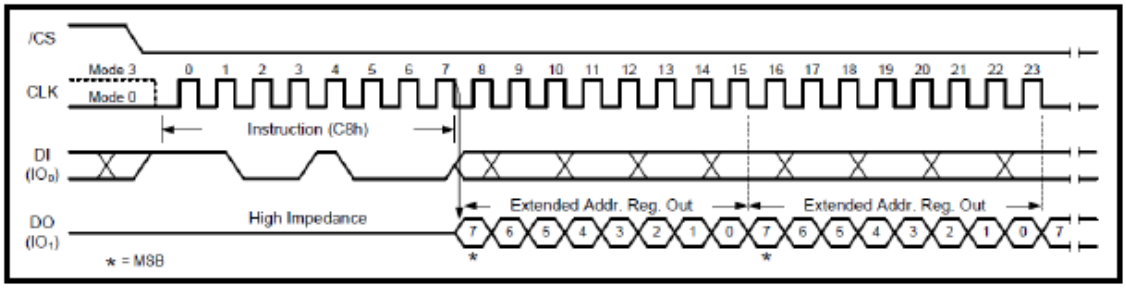


图 2-5 “C8h”指令时序图

2.2.6 写地址扩展寄存器(C5h)

芯片内部的地址扩展寄存器为易失寄存器，可用于扩展 3 字节地址指令可访问的地址空间。当芯片工作于 3 字节地址模式时，用户输入的 3 字节地址结合地址扩展寄存器可实现对 256Mb 地址空间的访问。用户可以使用指令来改变地址扩展寄存器的值，在此之前，需首先使用“写使能”指令将状态寄存器的 WEL 设置为 1。写地址扩展寄存器指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“C5h”和写数据从 IO0 输入芯片，最后将/CS 信号置为高电平。

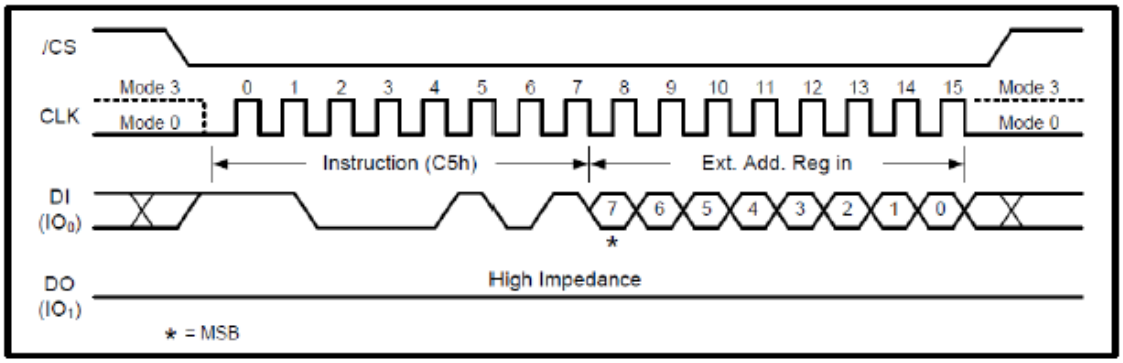


图 2-6 “C5h”指令时序图

2.2.7 进入低功耗模式(B9h)

为了满足对更低静态功耗的需求，用户可以使用指令将芯片置于低功耗模式以进一步降低静态功耗。指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“B9h”从 IO0 输入芯片，最后将/CS 信号置为高电平。需要注意的是，/CS 信号必须在 CLK 的第 8 个上升沿将指令码的最后一个比特锁存入芯片后变为为高电平，否则指令将被芯片忽略。

正确输入指令后，芯片将在 t_{DP} 时间内进入低功耗模式，在低功耗模式下，除“退出低功耗模式”指令之外的所有指令都将被芯片忽略。

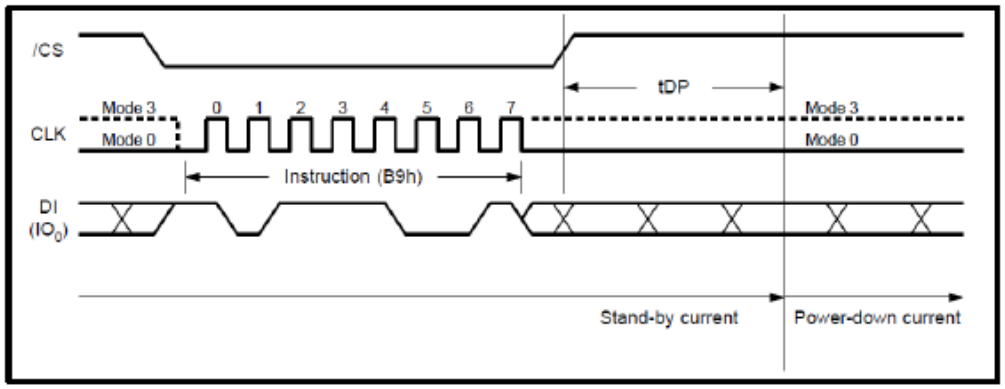


图 2-7“B9h”指令时序图

2.2.8 退出低功耗模式(ABh)

当芯片处于低功耗模式时，用户可以使用指令将芯片返回到正常工作模式。指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“ABh”从 IO0 输入芯片，最后将/CS 信号置为高电平。芯片将在/CS 信号变为高电平后 t_{RES} 时间内返回到正常工作模式。需要注意的是，在 t_{RES} 期间，/CS 信号必须保持高电平。

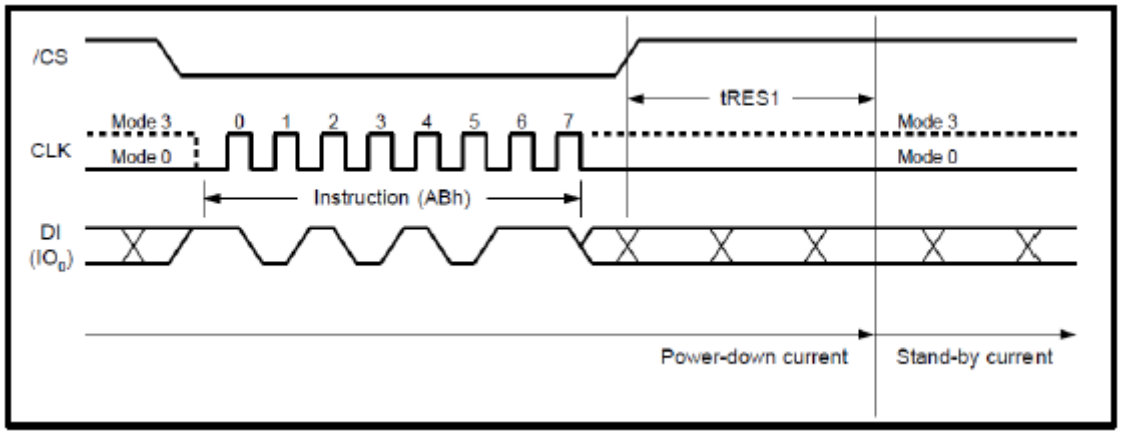


图 2-8“ABh”指令时序图

2.2.9 进入 4 字节地址模式(B7h)

早期 SPI Flash 存储器的存储空间不超过 128Mb，因此其指令使用 3 字节地址来访问存储器。为了后向兼容这些指令且扩展这些指令可访问的地址空间，芯片提供了两种方式供用户选择：地址扩展寄存器和 4 字节地址模式。在上电复位或/软件复位后，芯片默认处于 3 字节地址模式。用户可以使用指令将芯片设置为 4 字节地址模式，在 4 字节地址模式下，传统的 3 字节地址指令也将使用 4 字节的地址（地址扩展寄存器被忽略），从而实现对整个 256Mb 地址空间的访问。指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“B7h”从 IO0 输入芯片，最后将/CS 信号置为高电平。

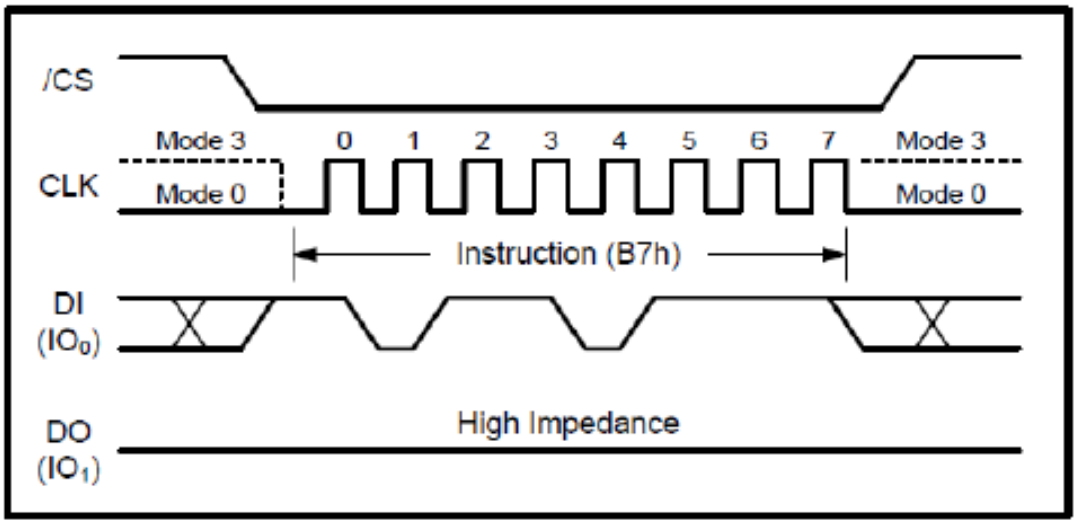


图 2-9“B7h”指令时序图

2.2.10 退出 4 字节地址模式(E9h)

当芯片处于 4 字节地址模式时，用户可以使用指令将芯片返回到 3 字节地址模式。指令时序图如下所示。首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“E9h”从 IO0 输入芯片，最后将/CS 信号置为高电平。

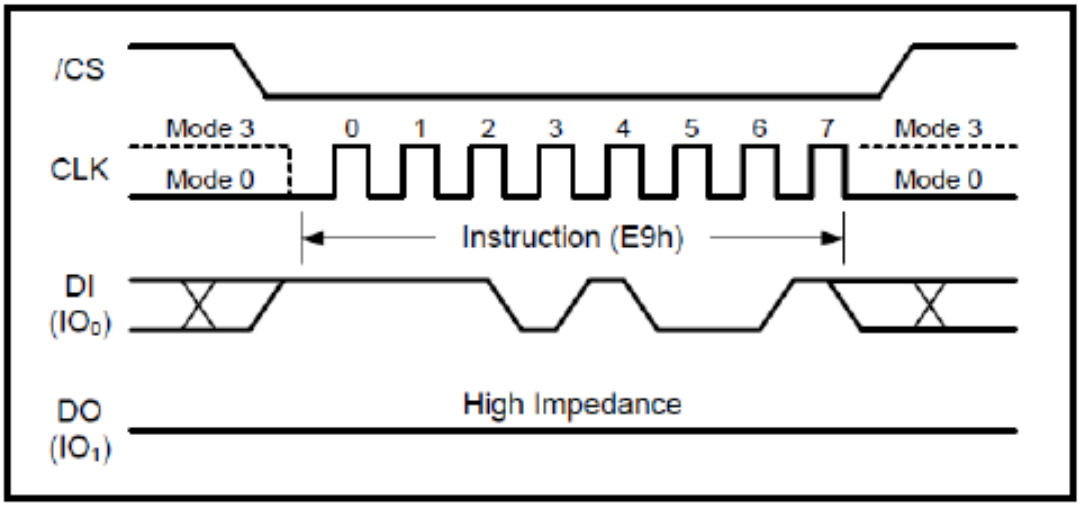


图 2-10“E9h”指令时序图

2.2.11 软件复位(66h+99h)

受小尺寸封装管脚数量的限制，芯片未提供专用的硬件复位管脚。为了满足用户对复位操作的需求，芯片提供了软件复位指令来实现对整个芯片的复位。

为了避免芯片被意外复位，用户需要连续输入“软件复位使能”（66h）和“软件复位”（99h）指令才能启动芯片内部的复位操作。在这两条之间不能插入任何其它指令，否则软件复位操作不会启动，之前输入的 66h 指令也会失效。用户需要重新输入 66h、99h 指令序列才能启动软件复位。当芯片接受了软件复位指令后，所有正在进行的内部操作（如编程、擦除和写寄存器等）都将被异常中断，然后芯片返回到上电复位后的初始状态，所有易失的状态/配置/控制信息都将被复位为初始值。芯片需要 28μs 来完成整个复位操作，在此期间，用户输入的任何指令都将被芯片忽略。

需要特别注意的是，如果芯片正在进行编程/擦除等操作，软件复位有可能导致内部数据变为无效数据，建议在软件复位前仔细确认芯片的状态。软件复位操作时序图如下图所示。

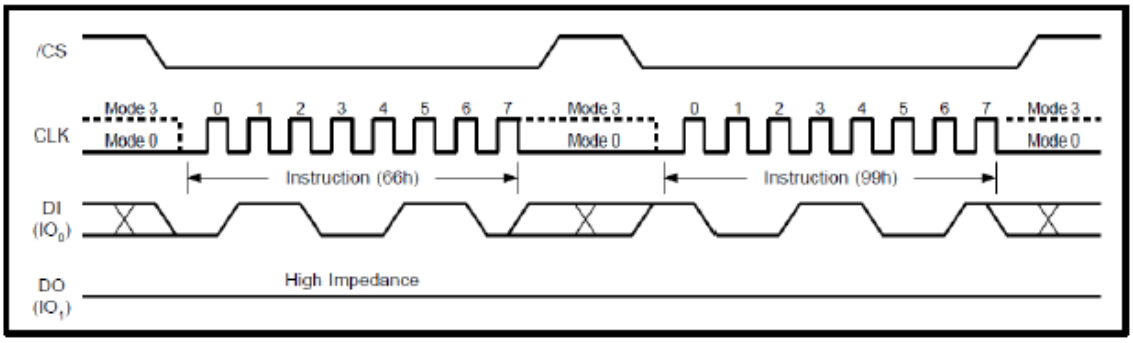


图 2-11 软件复位时序图

2.2.12 三字节地址读数据(03h)

用户可以使用“03h”指令从芯片的主存储区连续读取一个或多个字节的数据。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“03h”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定。当地址的最后一个比特在 CLK 信号上升沿被芯片锁存后，输入地址对应处存储的字节数据将在 CLK 信号下降沿从 IO1 输出芯片（高位先输出），当前地址处存储的数据输出完后，芯片会自动输出下一个地址（当前地址+1）处存储的数据（如果当前地址为最高字节地址，那么当前地址处存储的数据输出完后，芯片会自动输出最低地址处存储的数据）。通过该特性，用户可以方便地一次性读出整个芯片存储的数据。当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

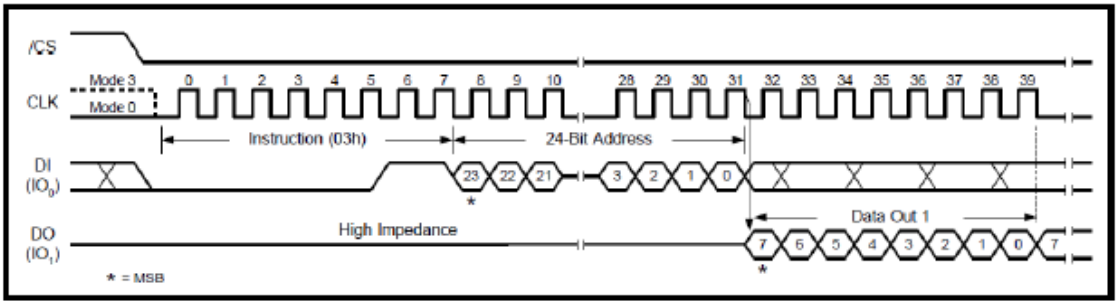


图 2-12 “03h”指令时序图(3 字节地址模式)

2.2.13 四字节地址读数据(13h)

为了方便访问 256Mb 存储空间，芯片向用户提供的“13h”指令可以在不需要地址扩展寄存器的情况下从芯片的主存储区连续读取一个或多个字节的数据。本指令和“03h”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，且地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码 “13h”和“4 字节地址”从 IO0 输入芯片，当地址的最后一个比特在 CLK 信号上升沿被芯片锁存后，输入地址对应处存储的字节数据将在 CLK 信号下降沿从 IO1 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

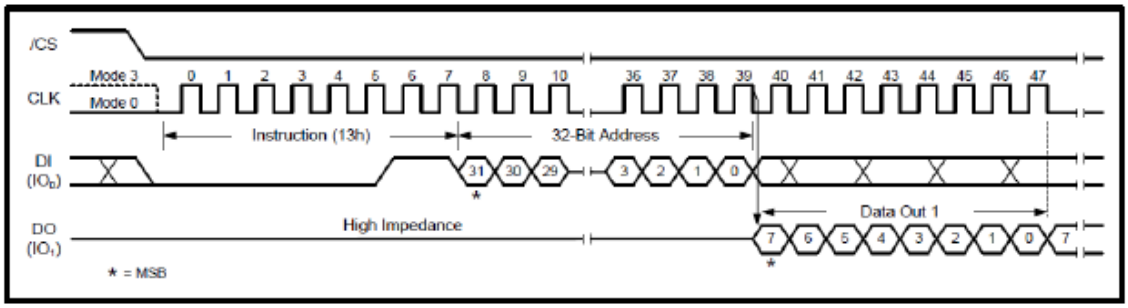


图 2-13 “13h”指令时序图

2.2.14 三字节地址快速读数据(0Bh)

为了满足用户对更高输出带宽的需求，芯片向用户提供的“0Bh”指令能够以更高的时钟频率从芯片的主存储区连续读取一个或多个字节的数据。本指令和 “03h”指令类似，指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“0Bh”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定。和“03”指令不同的是当地址的最后一个比特被芯片锁存后，芯片并不会立即输出数据，需要等待 8 个 CLK 周期，然后输入地址对应处存储的字节数据才会在 CLK 信号下降沿从 IO1 输出芯片（高位先输出），正是由于增加了等待周期，芯片才有更多时间来读取第一个字节数据，从而能够以更高的时钟频率来读取数据。当前地址处存储的数据输出完后，芯片会自动输出下一个地址（当前地址+1）处存储的数据（如果当前地址为最高字节地址，那么当前地址处存储的数据输出完后，芯片会自动输出最低地址处存储的数据）。当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

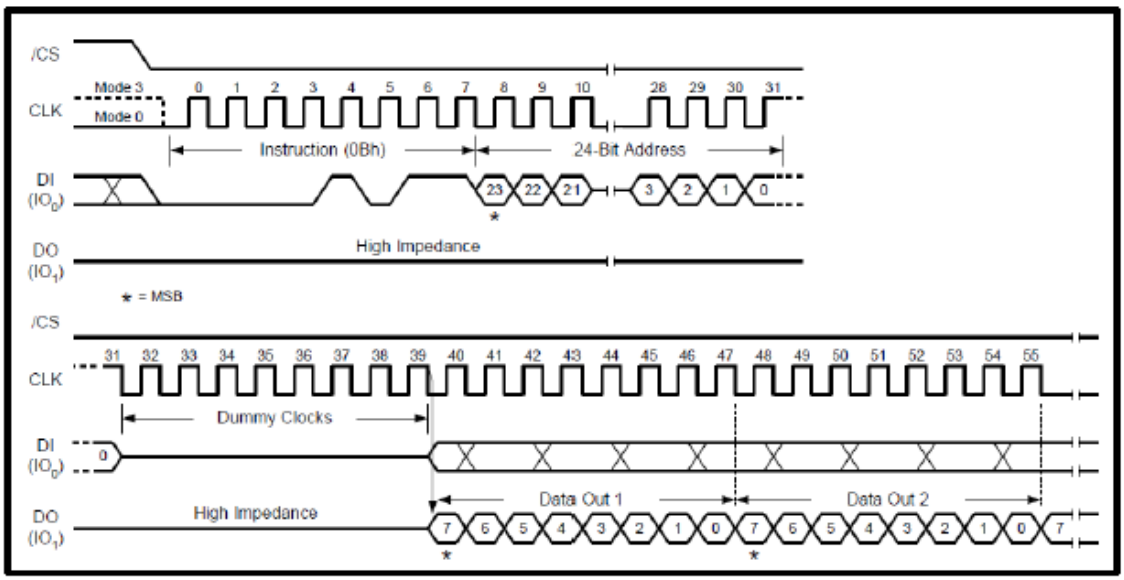


图 2-14 “0Bh”指令时序图(3 字节地址模式)

2.2.15 四字节地址快速读数据(0Ch)

“0Ch”指令是“0Bh”指令的 4 字节地址版本，用户可以在忽略地址扩展寄存器的情况下使用本指令从芯片 256Mb 的主存储区连续读取一个或多个字节的数据。本指令和“0Bh”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，且地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“0Ch”和“4 字节地址”从 IO0 输入芯片，然后等待 8 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO1 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

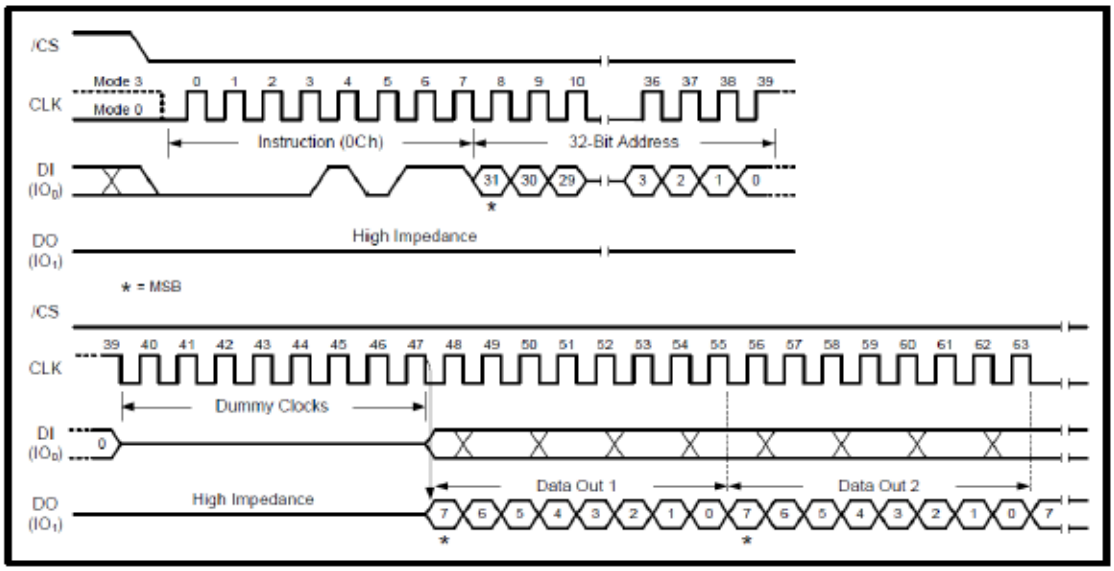


图 2-15 “0Ch”指令时序图

2.2.16 三字节地址双输出快速读数据(3Bh)

为了获得更高的数据输出带宽，用户可以使用“3Bh”指令从芯片的主存储区连续读取一个或多个字节的数据，在本指令的数据输出阶段，芯片可在每个时钟的下降沿输出 2 比特数据，从而提高了输出带宽。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“3Bh”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定。当地址的最后一个比特被芯片锁存并且等待 8 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO1~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

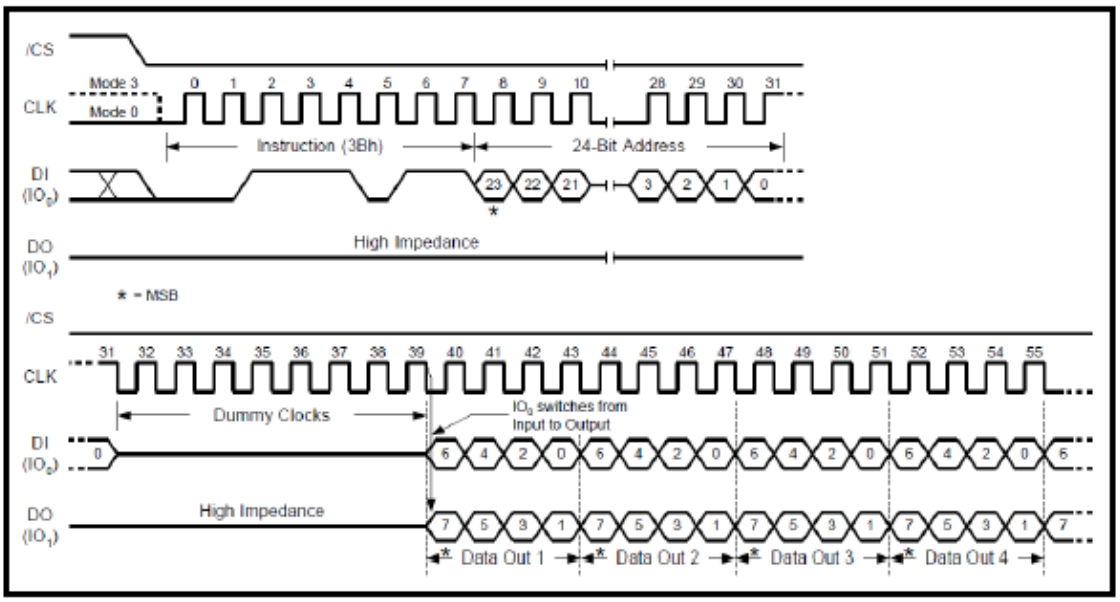


图 2-16 “3Bh”指令时序图(3 字节地址模式)

2.2.17 四字节地址双输出快速读数据(3Ch)

“3Ch”指令是“3Bh”指令的 4 字节地址版本，用户可以在忽略地址扩展寄存器的情况下使用本指令从芯片 256Mb 的主存储区连续读取一个或多个字节的数据。本指令和“3Bh”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“3Ch”和“4 字节地址”从 IO0 输入芯片，等待 8 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO1~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

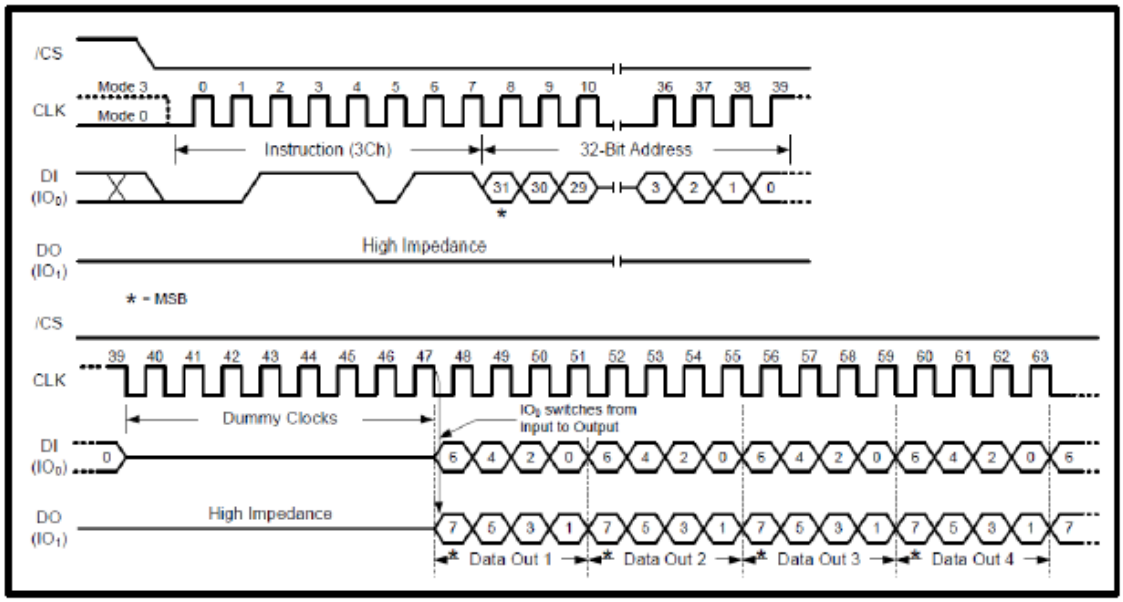


图 2-17 “3Ch”指令时序图

2.2.18 三字节地址四输出快速读数据(6Bh)

为了进一步提高数据输出带宽，用户可以使用“6Bh”指令从芯片的主存储区连续读取一个或多个字节的数据，在本指令的数据输出阶段，芯片可在每个时钟的下降沿输出 4 比特数据，从而进一步提高了输出带宽。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“6Bh” 和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定。当地址的最后一个比特被芯片锁存并且等待 8 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO3~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

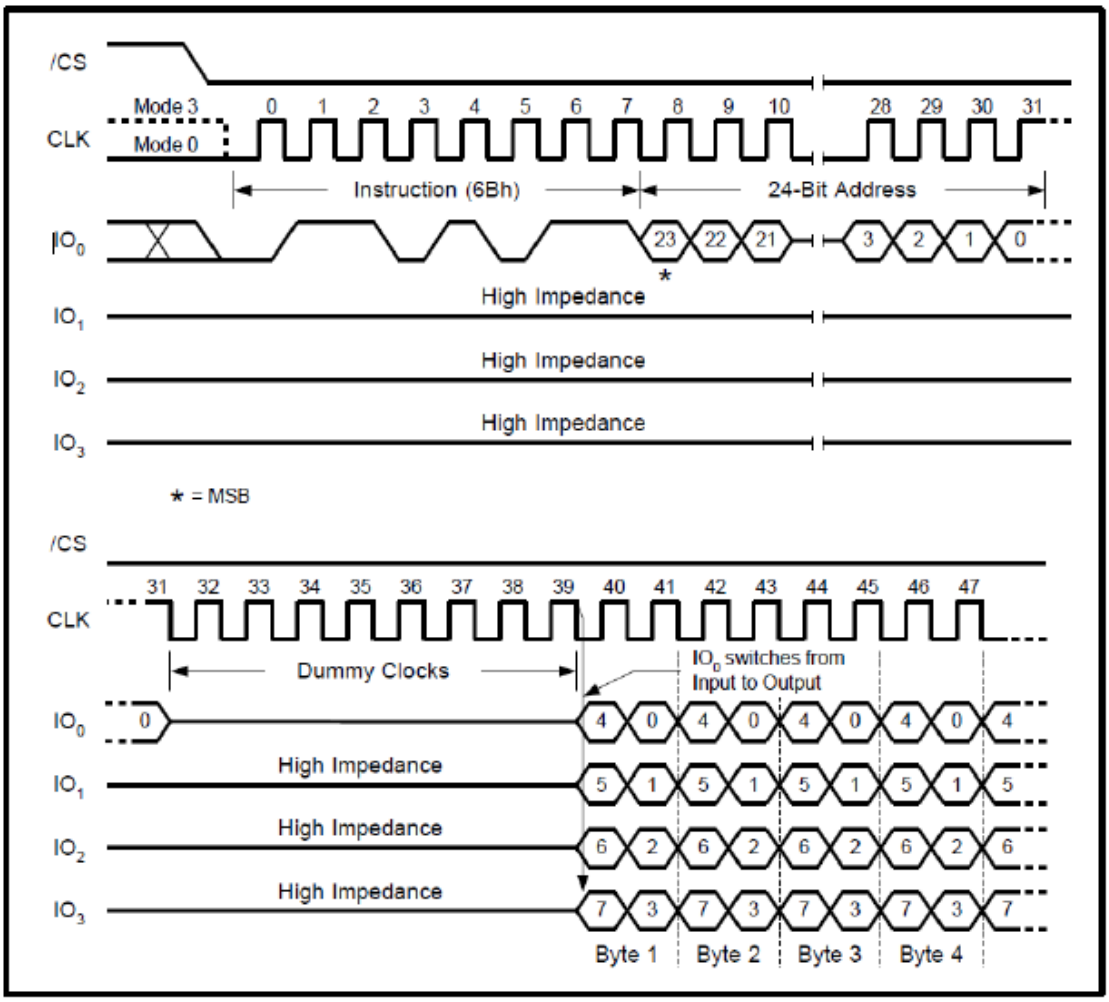


图 2-18 “6Bh”指令时序图(3 字节地址模式)

2.2.19 四字节地址四输出快速读数据(6Ch)

“6Ch”指令是“6Bh”指令的 4 字节地址版本，用户可以在忽略地址扩展寄存器的情况下使用本指令从芯片 256Mb 的主存储区连续读取一个或多个字节的数据。本指令和“6Bh”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，且地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“6Ch”和“4 字节地址”从 IO0 输入芯片，然后等待 8 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO3~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

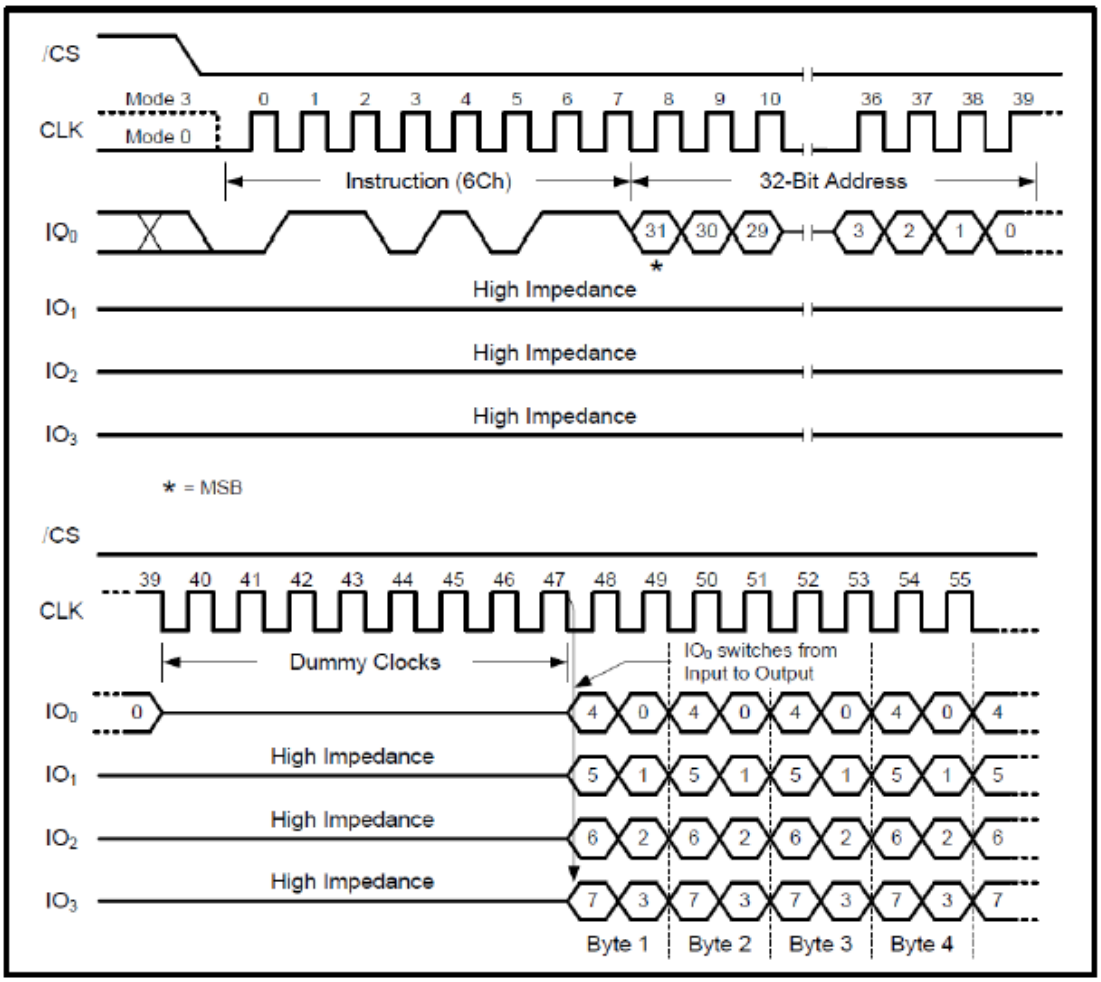


图 2-19 “6Ch”指令时序图

2.2.20 三字节地址双 I/O 快速读数据(BBh)

为了减小数据输出延迟，用户可以使用“BBh”指令从芯片的主存储区连续读取一个或多个字节的数据，在指令的地址输入阶段，芯片可在 CLK 的上升沿锁存 2 比特数据，在指令的数据输出阶段，芯片可在 CLK 的下降沿可输出 2 比特数据，从而可以在保证输出带宽的情况下减小数据输出延迟。

指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“BBh”从 IO0 输入芯片，然后在 CLK 信号的上升沿将“3/4 字节地址”从 IO1~IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定，然后再在 CLK 信号的上升沿将 1 字节的“连续读模式码”（简称 M 码）从 IO1~IO0 输入芯片，在 M 码的最后 2 个比特被芯片锁存后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO1~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

需要注意的是，“BBh”指令的等待周期是用户可配置的，通过设置控制寄存器的 DC1~DC0 位可配置指令所需的等待周期，芯片出厂时设置的等待周期为 4，M 码的输入周期

等效于等待周期，因此在时序图中可见，当通过 4 周期将 M 码输入完成后，输出数据可立即输出芯片。

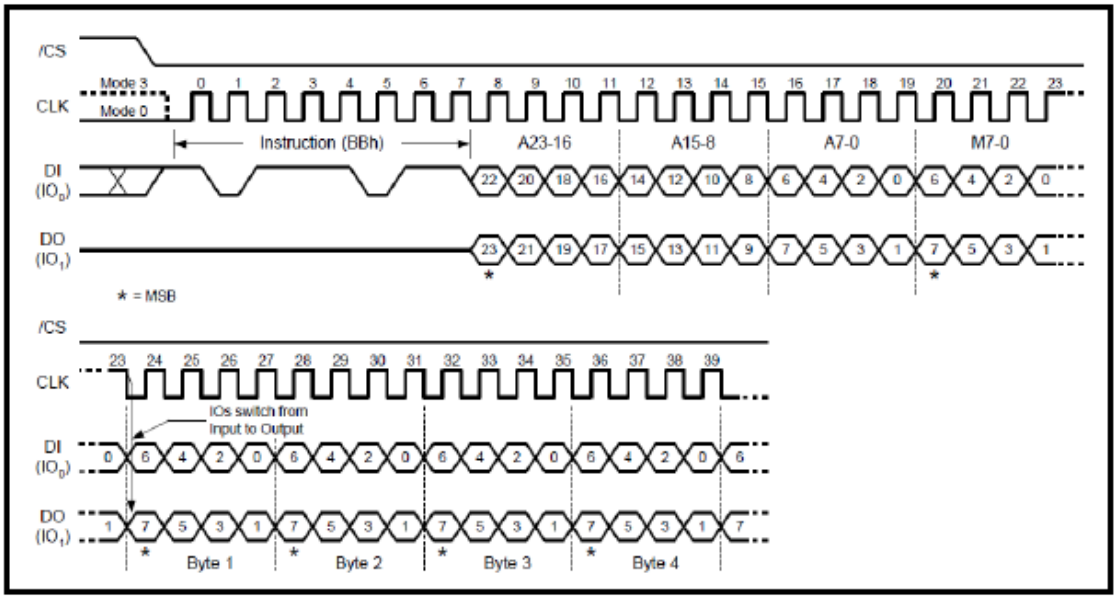


图 2-20 “BBh”指令时序图（首指令）

使用 M 码可以进一步减小数据输出延迟，M 码共有 8 比特，其中 M[5:4] 被用于控制紧随其后的“BBh”指令（其它位未使用），如果当前“BBh”指令的 M[5:4]为(1,0)，那么紧随当前指令之后的“BBh”指令将不需要输入指令码，也就是当前“BBh”指令完成后/CS 变为高电平，然后/CS 变为低电平，然后直接输入新的地址和新的 M 码后芯片即可输出新的数据。相比于正常读，连续读可以减少 8 个指令码输入周期，从而减小数据输出延迟。连续读时序图如下图所示。

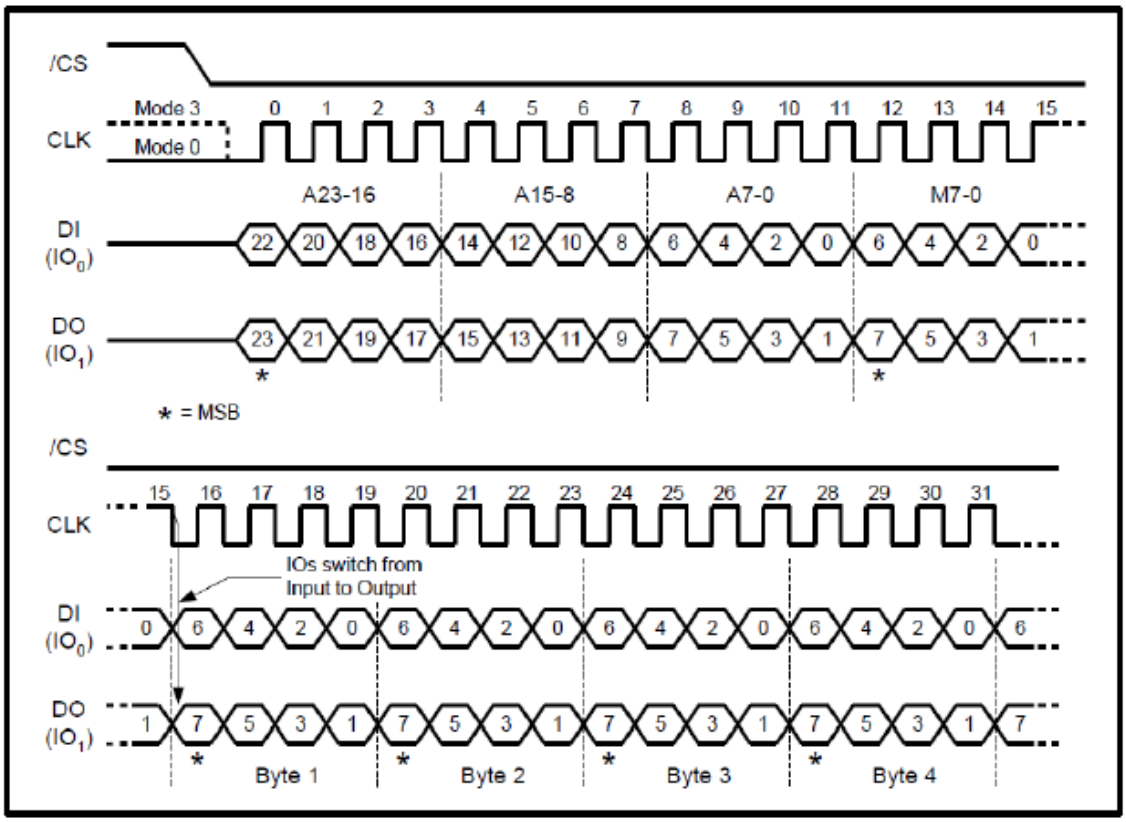


图 2-21 “BBh”指令时序图（连续读）

2.2.21 四字节地址双 I/O 快速读数据(BCh)

“BCh”指令是“BBh”指令的 4 字节地址版本，用户可以在忽略地址扩展寄存器的情况下使用本指令从芯片 256Mb 的主存储区连续读取一个或多个字节的数据。本指令和“BBh”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，且地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“BCh”从 IO0 输入芯片，然后将 4 字节地址和 M 码在 CLK 信号上升沿从 IO1~IO0 输入芯片，然后输入地址对应处存储的字节数据在 CLK 信号的下降沿从 IO1~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

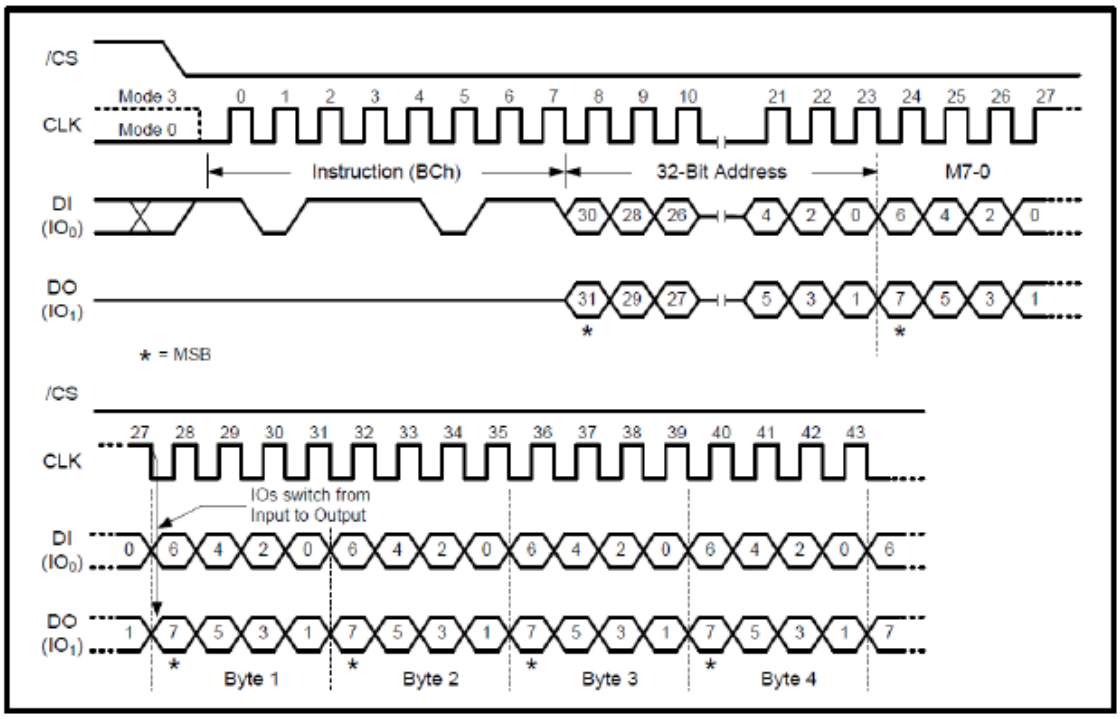


图 2-22 “BCh”指令时序图（首指令）

“BCh”指令也支持连续读模式，在该模式下后续指令在/CS 变为低电平后无需输入指令码，时序图如下所示。

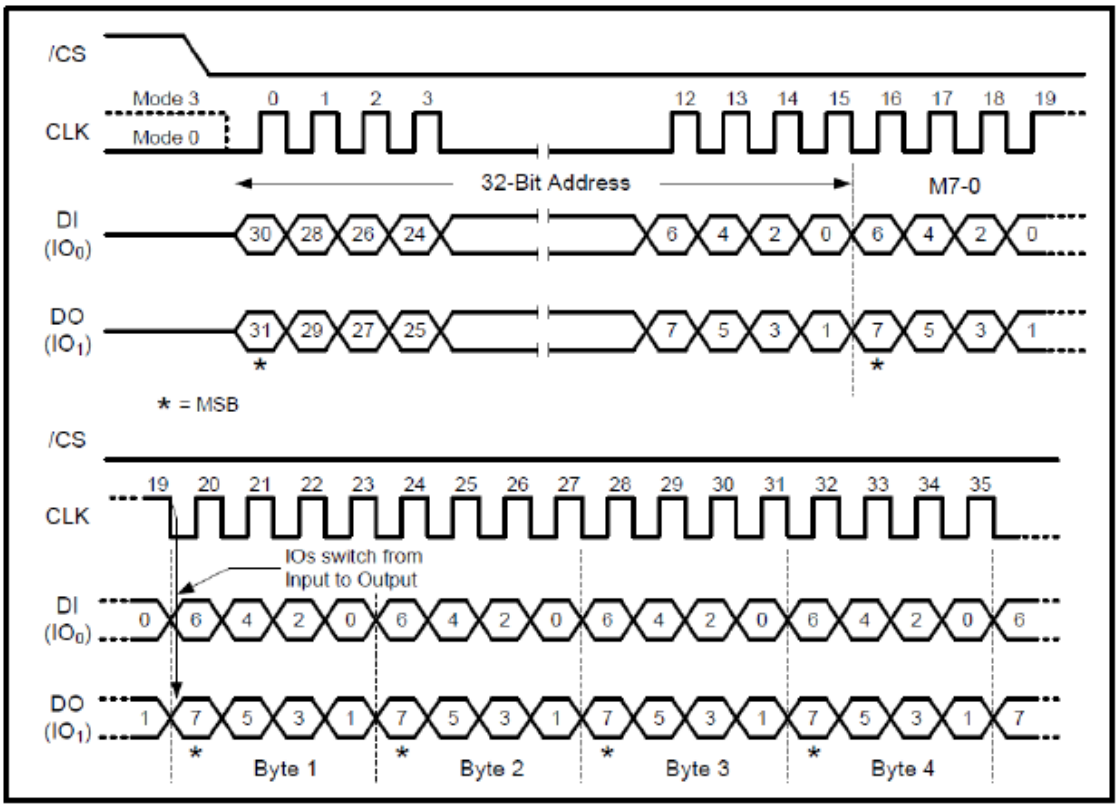


图 2-23 “BCh”指令时序图（连续读）

2.2.22 三字节地址四 I/O 快速读数据(EBh)

为了进一步减小数据输出延迟，用户可以使用“EBh”指令从芯片的主存储区连续读取一个或多个字节的数据，在本指令的地址输入阶段，芯片可在 CLK 的上升沿可锁存 4 比特数据，在本指令的数据输出阶段，芯片可在 CLK 的下降沿可输出 4 比特数据，从而可以在保证高输出带宽的情况下进一步减小数据输出延迟；

指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码“EBh”从 IO0 输入芯片，然后在 CLK 信号的上升沿将“3/4 字节地址”从 IO3~IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定，然后再在 CLK 信号的上升沿将 1 字节的“连续读模式码”（简称 M 码）从 IO3~IO0 输入芯片，在 M 码的最后 4 个比特被芯片锁存并且等待了 4 个 CLK 周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO3~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

需要注意的是，“EBh”指令的等待周期是用户可配置的，通过设置控制寄存器的 DC1~DC0 位可配置指令所需的等待周期，芯片出厂时设置的等待周期为 6，M 码的输入周期等效于等待周期，因此在时序图中可见，当通过 2 周期将 M 码输入完成后，只需等待 4 个周期芯片即可将数据输出芯片。

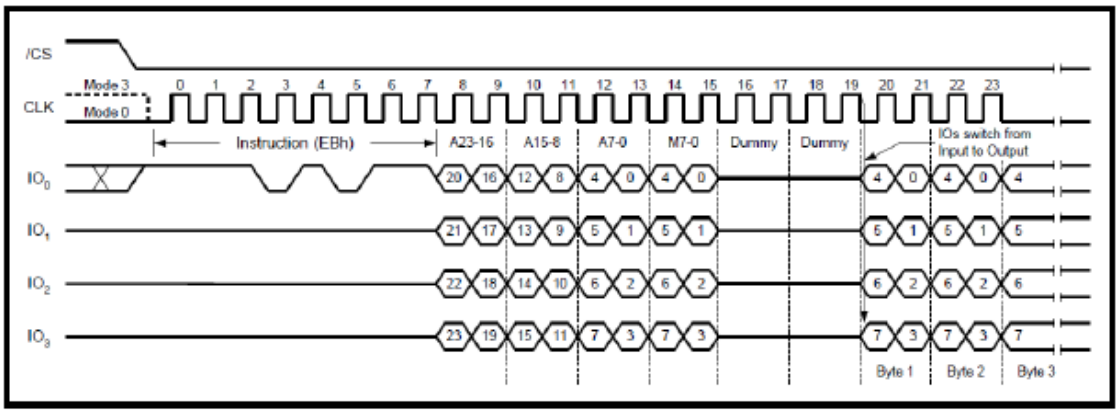


图 2-24 “EBh”指令时序图（首指令）

使用 M 码可以进一步减小数据输出延迟，M 码共有 8 比特，其中 M[5:4] 被用于控制紧随其后的“EBh”指令（其它位未使用），如果当前“EBh”指令的 M[5:4]为(1,0)，那么紧随当前指令之后的“EBh”指令将不需要输入指令码，也就是当前“EBh”指令完成后/CS 变为高电平，然后/CS 变为低电平，然后直接输入新的地址、新的 M 码和等待周期后芯片即可输出新的数据。相比于正常读，连续读可以减少 8 个指令码输入周期，从而减小数据输出延迟。连续读时序图如下图所示。

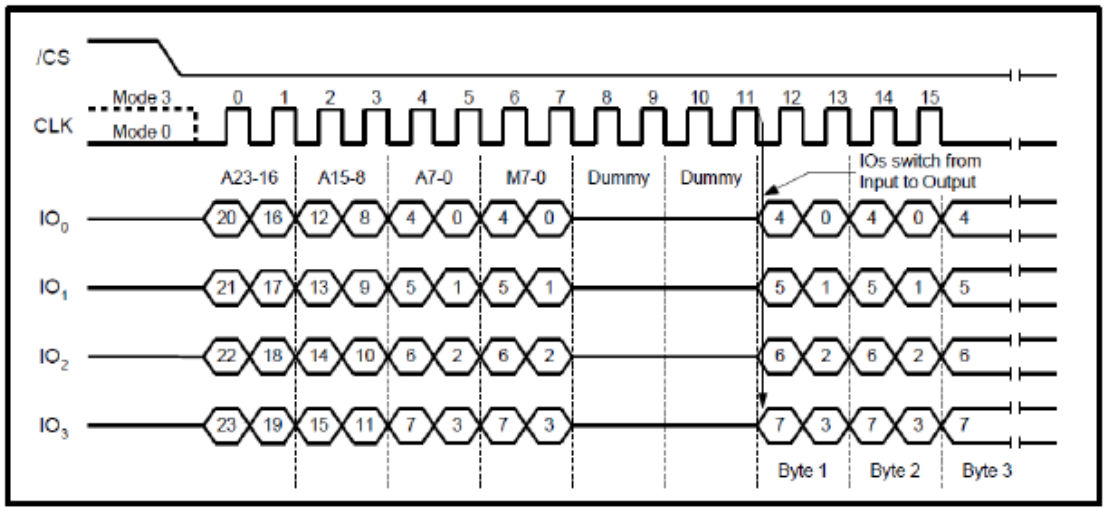


图 2-25 “EBh”指令时序图（连续读）

2.2.23 四字节地址四 I/O 快速读数据(ECh)

“ECh”指令是“EBh”指令的 4 字节地址版本，用户可以在忽略地址扩展寄存器的情况下使用本指令从芯片 256Mb 的主存储区连续读取一个或多个字节的数据。本指令和“EBh”指令类似，区别在于无论芯片当前处于哪种地址模式，指令码后都必须使用 4 字节地址，且地址扩展寄存器中的数据将被忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿将指令码 “ECh”从 IO0 输入芯片，然后将“4 字节地址”和 M 码在 CLK 信号的上升沿从 IO3~IO0 输入芯片，等待 4 个周期后，输入地址对应处存储的字节数据将在 CLK 信号的下降沿从 IO3~IO0 输出芯片（高位先输出），当用户读出所需的数据后，可通过将/CS 信号置为高电平完成此次读操作。

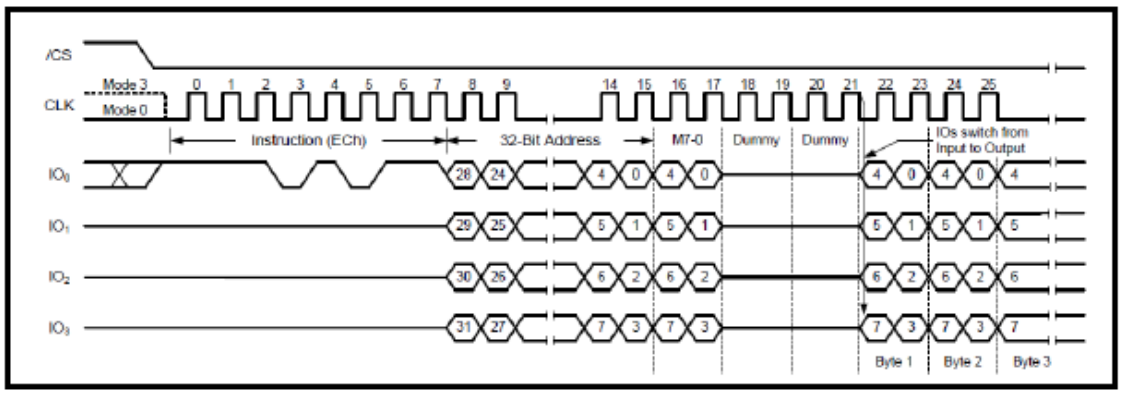


图 2-26 “ECh”指令时序图（首指令）

“ECh”指令也支持连续读模式，在该模式下后续指令在/CS 变为低电平后无需输入指令码，时序图如下所示。

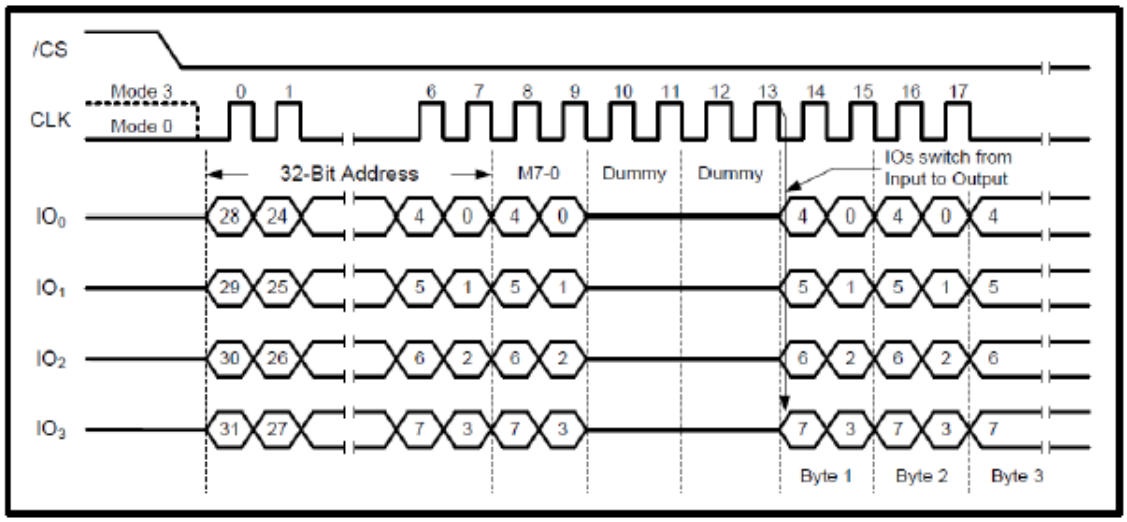


图 2-27 “ECh”指令时序图（连续读）

2.2.24 三字节地址单输入页编程(02h)

用户可以使用“02h”编程指令一次性将 1~256 字节数据写入芯片主存储区。需要注意的是，在进行编程操作之前，必须先将所有编程目标地址擦除为“FFh” 状态。在输入编程指令之前，必须先输入“写使能”指令将状态寄存器的 WEL 设置为 1，否则编程指令将会被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“02h”、“3/4 字节地址”和 “1~256 字节数据”从 IO0 输入芯片，具体使用 3 字节或 4 字节地址由芯片当前的地址模式决定，最后将/CS 信号置为高电平。

芯片内部使用一个地址指针来控制下一个输入字节数据的存储地址，每输入 1 字节数据后，地址指针会加 1，指向下一个存储地址，当地址指针到达页的最高地址后会返回到页的最低地址。如果用户输入的数据超过 256 字节，先输入的数据将会被后输入的数据覆盖，比如用户从页的最低地址开始输入了 257 个数据，那么当第 256 字节数据被写入页的最高地址后，第 257 字节数据将被写入页的最低地址并覆盖之前输入的第 1 字节数据。如果需要一次将整页（256 字节）编程，建议用户将输入地址的最低字节设置为 0，从页的最低字节地址处开始输入数据。

在输入本指令时，/CS 信号必须在最后一个字节输入数据的最后一个比特（字节边界处）被锁存后变为高电平，如果这个条件不能满足，那么芯片将不会执行编程操作。当芯片接受了编程指令后，内部电路便开始相应操作。整个编程操作会在 t_{PP} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定编程操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成编程操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果编程目标地址所在的扇区被写保护，那么编程指令将不会被执行。

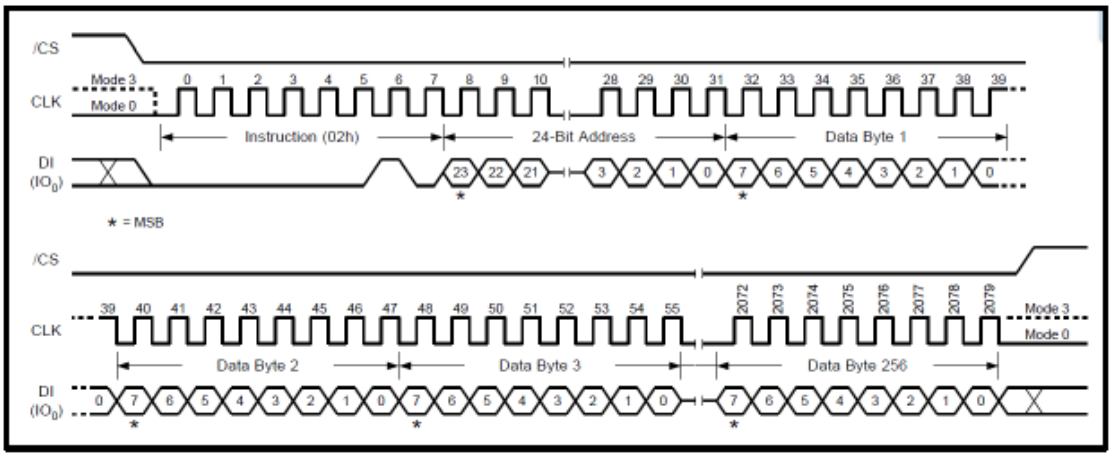


图 2-28 “20h”指令时序图(3 字节地址模式)

2.2.25 三字节地址四输入页编程(32h)

为了提高编程数据输入芯片的效率，用户可以使用“32h”指令将 1~256 字节数据写入芯片主存储区。本指令和“02h”指令类似，区别在于在数据输入阶段，芯片可在 CLK 信号的上升沿将锁存 4 位数据。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号的上升沿依次将指令码“32h”和“3/4 字节地址” 从 IO0 输入芯片，具体使用 3 字节或 4 字节地址由芯片当前的地址模式决定，然后再将 1~256 字节数据在 CLK 信号的上升沿从 IO3~IO0 输入芯片，最后将/CS 信号置为高电平。

输入该指令时，/CS 信号必须在最后一个字节输入数据的最后 4 个比特（字节边界处）被锁存后变为高电平，如果这个条件不能满足，那么芯片将不会执行编程操作。当芯片接受了指令后，内部电路便开始相应操作。整个编程操作会在 t_{PP} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定编程操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成编程操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果编程目标地址所在的扇区被写保护，那么编程指令将不会被执行。

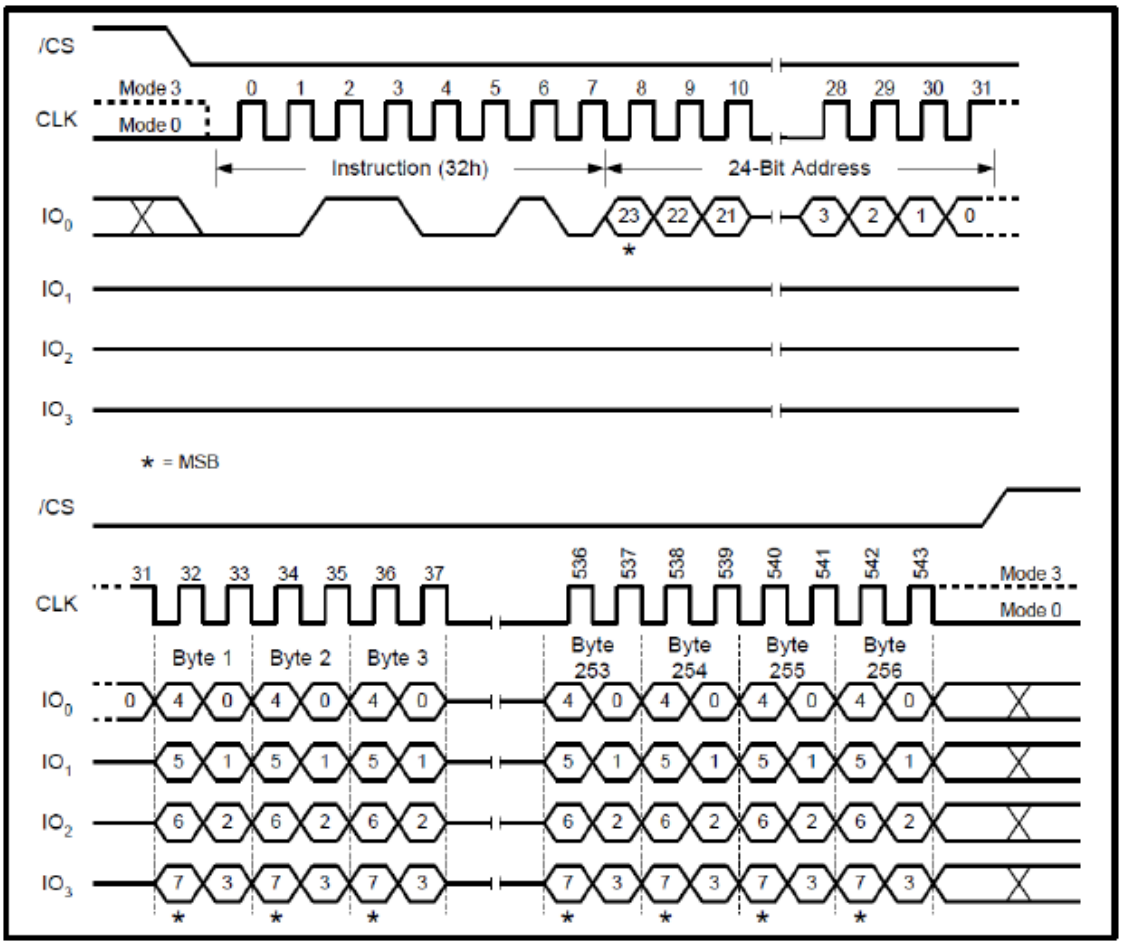


图 2-29 “32h”指令时序图(3 字节地址模式)

2.2.26 三字节地址 4KB 扇区擦除(20h)

用户可以使用“20h”指令将某个 4KB 的扇区擦除。在进行擦除操作之前，用户必须使用“写使能”指令将状态寄存器的 WEL 设置为 1，否则擦除指令将会被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号上升沿依次将指令码“20h”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定，最后将/CS 信号置为高电平。需要注意的是，/CS 信号必须在地址最后一个比特被芯片锁存后变为高电平，否则芯片不会执行擦除操作。

当芯片接受了擦除指令后，内部电路便开始相应操作。整个擦除操作会在 t_{SE} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定擦除操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成擦除操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果被擦除的扇区被写保护，那么擦除指令将不会被执行。

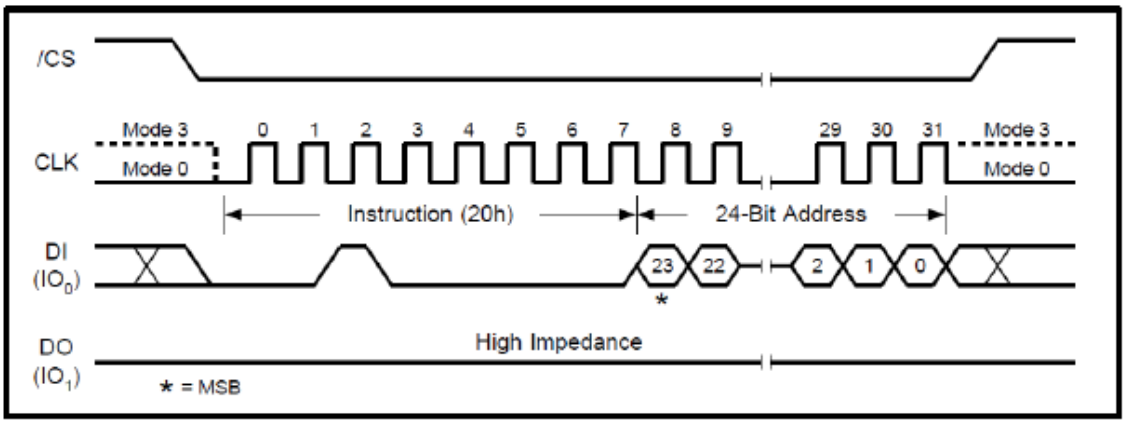


图 2-30 “20h”指令时序图(3 字节地址模式)

2.2.27 三字节地址 32KB 扇区擦除(52h)

用户可以使用“52h”指令将某个 32KB 的扇区擦除。在进行擦除操作之前，用户必须使用“写使能”指令将状态寄存器的 WEL 设置为 1，否则擦除指令将会被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号上升沿依次将指令码“52h”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定，最后将/CS 信号置为高电平。需要注意的是，/CS 信号必须在地址最后一个比特被芯片锁存后变为高电平，否则芯片不会执行擦除操作。

当芯片接受了擦除指令后，内部电路便开始相应操作。整个擦除操作会在 t_{BE1} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定擦除操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成擦除操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果被擦除的扇区处于写保护状态，那么擦除指令将不会被执行。

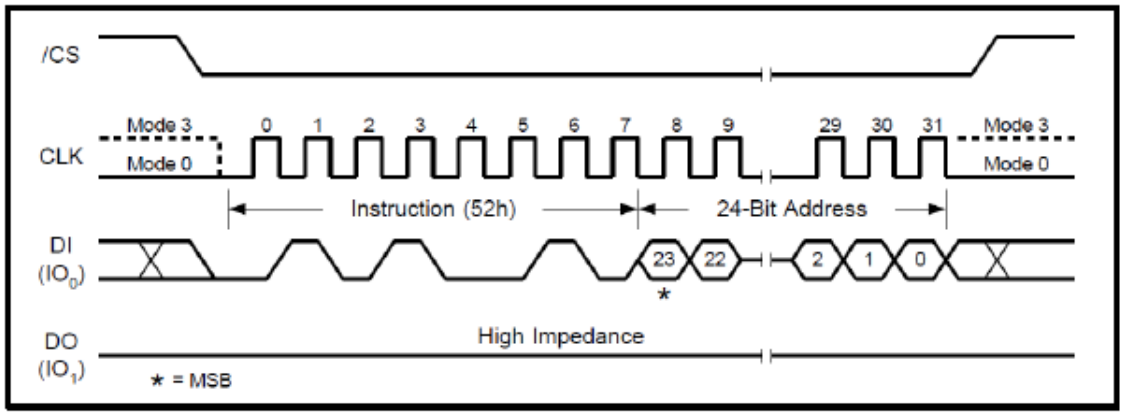


图 2-31 “52h”指令时序图(3 字节地址模式)

2.2.28 三字节地址 64KB 扇区擦除(D8h)

用户可以使用“D8h”指令将某个 64KB 的扇区擦除。在进行擦除操作之前，用户必须使用“写使能”指令将状态寄存器的 WEL 设置为 1，否则擦除指令将会被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号上升沿依次将指令码“D8h”和“3/4 字节地址”从 IO0 输入芯片，具体使用 3 字节地址还是 4 字节地址由芯片当前的地址模式决定，最后将/CS 信号置为高电平。需要注意的是，/CS 信号必须在地址最后一个比特被芯片锁存后变为高电平，否则芯片不会执行擦除操作。

当芯片接受了擦除指令后，内部电路便开始相应操作。整个擦除操作会在 t_{BE2} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定擦除操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成擦除操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果被擦除的扇区被写保护，那么擦除指令将不会被执行。

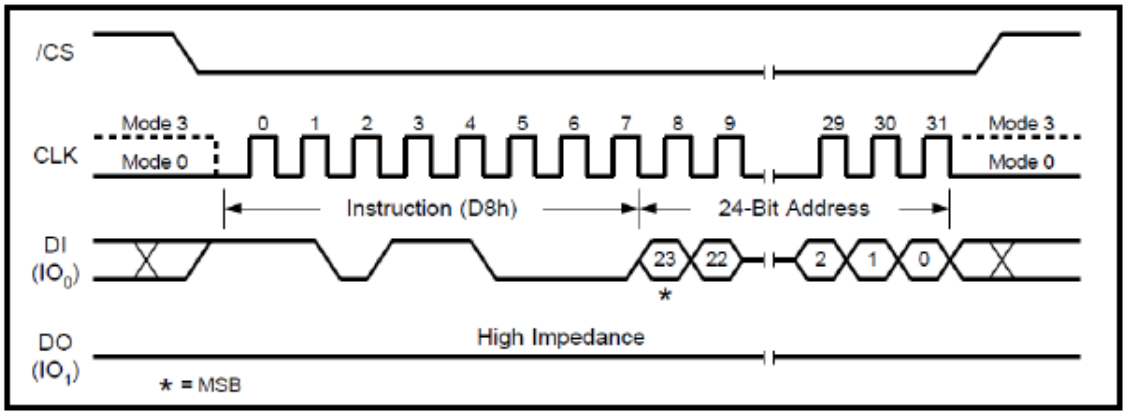


图 2-32 “D8h”指令时序图(3 字节地址模式)

2.2.29 全芯片擦除(C7h/60h)

用户可以使用“C7h/60h”指令将整个主存储区擦除。在进行擦除操作之前，用户必须使用“写使能”指令将状态寄存器的 WEL 设置为 1，否则“全芯片擦除” 指令将会被芯片忽略。指令时序图如下所示，首先将/CS 信号置为低电平，然后在 CLK 信号上升沿将指令码“C7h/60h”从 IO0 输入芯片，最后将/CS 信号置为高电平。需要注意的是，/CS 信号必须在指令码最后一个比特被芯片锁存后变为高电平，否则芯片不会执行全芯片擦除操作。

当芯片接受了擦除指令后，内部电路便开始相应操作。整个擦除操作会在 t_{CE} 时间内完成。在此期间，用户可以通过检查 BUSY 位的状态来确定擦除操作是否完成，如果 BUSY 为 1，表示操作还未完成。当内部逻辑完成擦除操作后，会将 BUSY 设置为 0，同时 WEL 也会被设置为 0。需要注意的是，如果有扇区处于写保护状态，那么“全芯片擦除”指令将不会被执行。

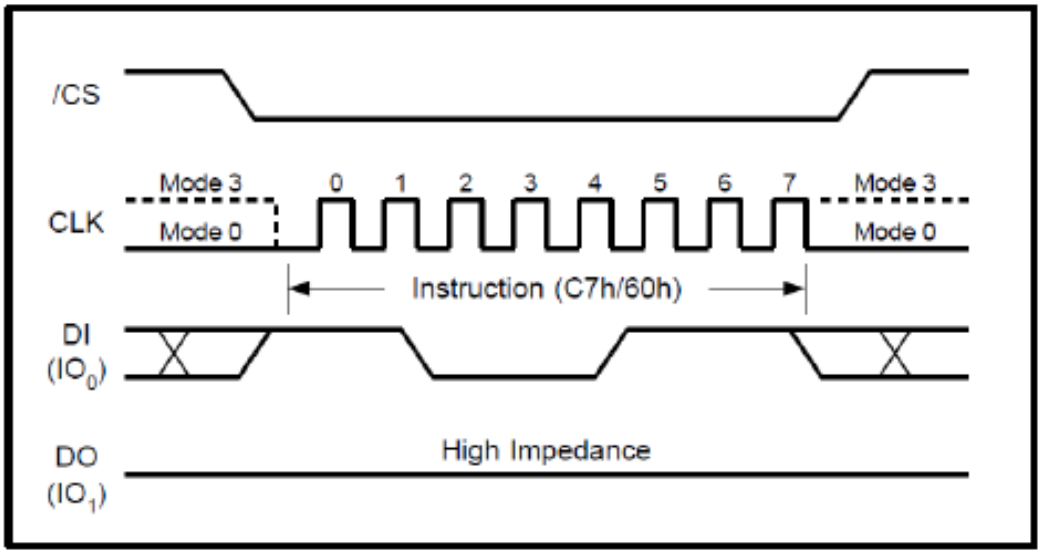


图 2-33 “C7h/60h”指令时序图

3 电气特性

3.1 绝对最大额定值

绝对最大额定值如下：

电 源 电 压 （V_{CC}）……-0.6V～4.6V

IO 端口电压（V_{IO}）……-0.6V～V_{CC}+0.4V

存 储 温 度(T_{stg}) ……-65℃ ～150℃

结 温(T_J)……-65℃ ～150℃

3.2 推荐工作条件

推荐工作条件如下：

电 源 电 压 （V_{CC}）……1.65V~3.6V

工 作 温 度 （T_A）……-55℃ ～125℃

结 温(T_J)……-55℃ ～125℃

3.3 电特性表

3.3.1 直流参数

表 3-1 直流参数表

参数	符号	测试条件 除另有规定外，V_{CC}=1.65V～3.6V，T_A=-55℃～125℃。	最小值	最大值	单位
输入 IO 漏电流	I_{LI}	—	-2	2	μA
双向 IO 漏电流	I_{LO}	—	-2	2	μA
待机电流	I_{CC1}	/CS = V_{CC}	—	50	μA
低功耗模式电流	I_{CC2}	/CS = V_{CC}	—	30	μA
读数据电流	I_{CC3}	f = 66MHz	—	16	mA
		f = 133MHz	—	22	mA
页编程电流	I_{CC4}	/CS= V_{CC}	—	20	mA
写寄存器电流	I_{CC5}	—	—	12	mA
扇区/全芯片擦除电流	I_{CC6}	—	—	20	mA
输入低电平电压	V_{IL}	—	-0.5	V_{CC}×0.2	V
输入高电平电压	V_{IH}	—	V_{CC}×0.8	V_{CC} +0.4	V
输出低电平电压	V_{OL}	I_{OL} = 100μA	—	0.2	V
输出高电平电压	V_{OH}	I_{OH} = -100μA	V_{CC}-0.2	—	V

3.3.2 交流参数

表 3-2 交流参数表

参数	符号	测试条件 除另有规定外，V_{CC}=1.65V～3.6V，T_A=-55℃～125℃。	最小值	最大值	单位
负载电容	C_L	—	—	30	pF
输入上升/下降时间	T_R，T_F	—	—	5	ns
输入时序参考电压	—	—	0.2 V_{CC}～ 0.8 V_{CC}		V
输出时序参考电压	—	—	0.5 V_{CC}～ 0.5 V_{CC}		V
时钟频率	f_C	不包括 03h/13h 指令	—	133	MHz
读时钟频率	f_R	仅限 03h/13h 指令	—	66	MHz
时钟低电平时间	t_{CLL}	—	4	—	ns
时钟高电平时间	t_{CLH}	—	4	—	ns
输出保持时间	t_{CLQX}	—	3.5	—	ns
时钟到输出有效时间	t_{CLQV}	负载 30pf	—	7	ns
/CS 高到输出高阻	t_{SHQZ}	—	—	6	ns
/CS 有效建立时间	t_{SLCH}	—	5	—	ns
/CS 无效建立时间	t_{SHCH}	—	5	—	ns
/CS 有效保持时间	t_{CHSH}	—	5	—	ns
/CS 无效保持时间	t_{CHSL}	—	5	—	ns
连续读指令间隔时间	t_{SHSL1}	—	7	—	ns
连续写指令间隔时间	t_{SHSL2}	—	30	—	ns
进入低功耗模式时间	t_{DP}	—	—	3	μs
退出低功耗模式时间	t_{RES}	—	—	10	μs
写寄存器时间	t_W	—	—	50	ms
页编程时间	t_{PP}	—	—	3	ms
4KB 扇区擦除时间	t_{SE}	—	—	400	ms
32KB 扇区擦除时间	t_{BE1}	—	—	900	ms
64KB 扇区擦除时间	t_{BE2}	—	—	1.8	s
全芯片擦除时间	t_{CE}	—	—	200	s
软件复位时间	t_{SR}	—	—	28	μs

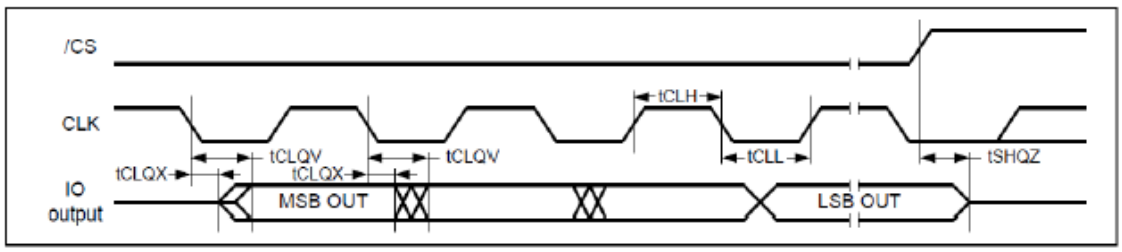


图 3-1 串行输出时序图

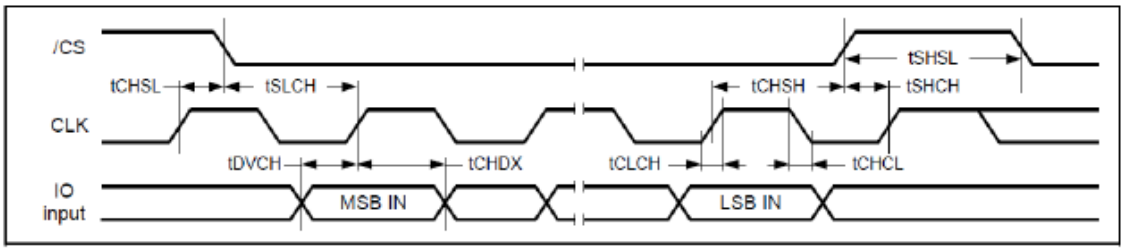

/CS
tCHSL
tSLCH
tSHSL
tCHSH
tSHCH
CLK
tDVCH
tCHDX
tCLCH
tCHCL
IO
input
MSB IN
LSB IN

图 3-2 串行输入时序图

4 说明事项

4.1 运输与储存

芯片在适宜环境下储运。

使用指定的防静电包装盒进行产品的包装和运输。在运输过程中，确保芯片不要与外物发生碰撞。

4.2 开箱与检查

开箱使用芯片时，请注意观察产品标识。确定产品标识清晰，无污迹，无擦痕。同时，注意检查无损坏，无伤痕，引脚整齐，无缺失，无变形。

4.3 使用操作规程及注意事项

器件必须采取防静电措施进行操作。取用芯片时应佩戴防静电手套，防止人体电荷对芯片的静电冲击，损坏芯片。将芯片插入电路板上的底座时以及将芯片从电路板上的底座取出时，应注意施力方向以确保芯片引脚均匀受力。不要因为用力过猛，损坏芯片引脚，导致无法使用。

推荐下列操作措施：

a）器件应在防静电的工作台上操作，或带指套操作；

b）试验设备和器具应接地；

c）此不能触摸器件引线；

d）器件应存放在导电材料制成的容器中（如：集成电路专用盒）；

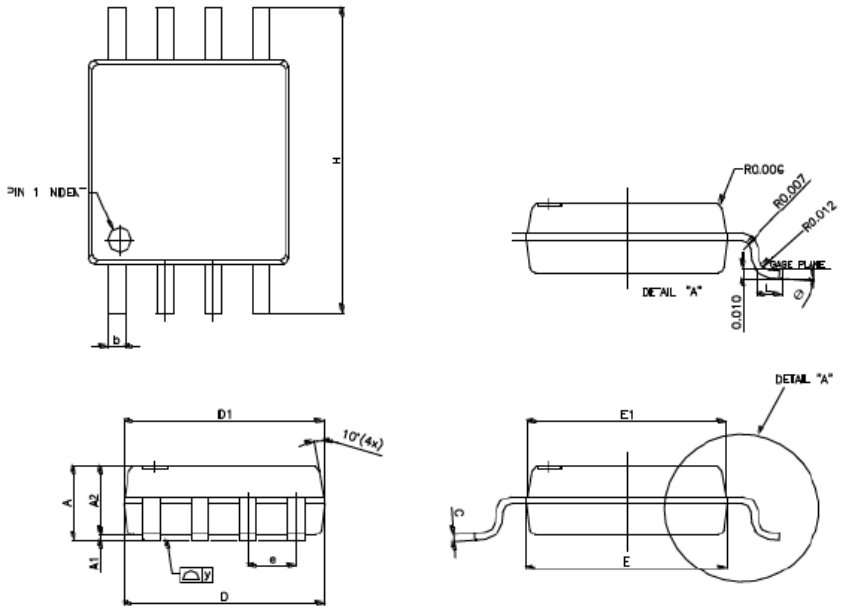
e）生产、测试、使用以及转运过程中应避免使用引起静电的塑料、橡胶或丝织物；

4.4 质量保证

公司质量管理体系根据国军标 GJB9001 要求制定了完善的质量管理工作流程，对产品的设计、生产和销售进行日常质量管理。产品制定依据 GJB7400《合格制造厂认证用半导体集成电路通用规范》裁剪后的标准进行设计和生产，并按照 GJB548B-2005《微电子器件试验方法和程序》的要求进行试验和检验。产品兼容性好、可靠高。

5 封装尺寸

5.1 SOP8 封装尺寸



单位为毫米

尺寸符号	最小值	公称值	最大值
A	1.75	1.95	2.16
A_1	0.05	0.15	0.25
A_2	1.70	1.90	1.91
b	0.35	0.42	0.48
c	0.19	0.20	0.25
D	5.18	5.28	5.38
D_1	5.13	5.23	5.33
E	5.18	5.28	5.38
E_1	5.13	5.23	5.33
e	--	1.27	--
H	7.70	7.90	8.10
L	0.50	0.65	0.80
y	--	--	0.10
θ	0°	--	8°

图 5-1SOP8 封装尺寸图

6 订货信息

6.1 选型列表

表 6-1 选型列表

型号	封装	引脚数
AST25QW256S	SOP (208mil)	8