

## SD 卡规格书第五章 SPI 总线协议

### 译者注:

在本章节内, 详细的描述了主控制器和 SD 卡之间, 用 SPI 总线相互握手及通讯的标准和方法。关于 SPI 总线拓扑的信息已经在第三章描述过了。

本章节的重点在于描述 SD 卡加电后的复位工作情况, 如何进入 SPI 模式, 如何用软件操作 SD 卡工作。以及 SD 卡是如何接收并响应主控制器的命令。它们之间的通讯产生错误的原因和处理的相关办法。

其中还介绍了 SD 卡的命令集及其可实现的功能 (SPI 模式下的命令集)。包括读取数据、写入数据、擦除数据、保护数据等等, 和这些命令所使用的方法和限制条件。

为了方便查阅, 涉及到本章节内容的其它章节的一点资料被附加在本文的最后, 如果没包括您需要的信息, 请您去查阅原文, 谢谢!

**注意:** 这篇译文的内容来自 SD 产品手册 2.2 版本。SD 组织对自己的技术资料是保密的, 除了他自己的客户外很难拿到。保护自己的技术产权无可厚非, 但这样做不利于 SD 卡产品在中国国内市场上的推广与使用, 国内的电子工程师们还没有富裕的金钱去购买 SD 的技术, 体制上受限又多, 语言也不通, 很少能有渠道获得支持, 想在自己设计的东西上加入对 SD 卡的支持都不容易。

这个版本已经很陈旧了, 没有容量 2G 以上的标准。如果哪位朋友有新版的手册, 请发布出来, 支持一下国内的电子爱好者, 咱们又不想去另搞一个类似 SD 的组织同他抢饭碗, 只不过想让自己设计的东西上能顺利地用上 SD 卡而已。

## SanDisk Secure Digital Card

---

### Product Manual

Version 2.2

Document No. 80-13-00169

September 2004

## SPI 总线协议目录

- 5.1. **SPI 总线协议**
  - 5.1.1. 模式选择
  - 5.1.2. 总线传送保护
  - 5.1.3. 数据的读取
  - 5.1.4. 数据写入
  - 5.1.5. 擦除和写保护处理
  - 5.1.6. 读取CID/CSD寄存器
  - 5.1.7. 复位顺序
  - 5.1.8. 时钟控制
  - 5.1.9. 错误条件
    - 5.1.9.1. CRC和无效指令
    - 5.1.9.2. 读取、写入和擦除的超时情形
  - 5.1.10. 存储器阵列分布
  - 5.1.11. 卡的锁定/解锁
  - 5.1.12. 特殊用途的命令集
  - 5.1.13. 版权保护命令集
- 5.2. **SPI 命令设置**
  - 5.2.1. 命令格式
  - 5.2.2. 命令类
    - 5.2.2.1. 详细的命令描述
  - 5.2.3. 应答
    - 5.2.3.1. R1 的格式
    - 5.2.3.2. R1b的格式
    - 5.2.3.3. R2 的格式
    - 5.2.3.4. R3 的格式
    - 5.2.3.5. 数据应答
  - 5.2.4. 数据标记
  - 5.2.5. 数据错误标记
  - 5.2.6. 清除状态位
- 5.3. **卡寄存器**
- 5.4. **SPI 总线时序图解**
  - 5.4.1. 命令 / 应答
  - 5.4.2. 数据读取
  - 5.4.3. 数据写入
  - 5.4.4. 时钟数值
- 5.5. **SPI 电气接口**
- 5.6. **SPI 总线操作条件 (环境)**
- 5.7. **总线时钟**

### 附录:

- 英文特殊命令表
- 英文通用命令表
- 存储器阵列图
- 产品规格表
- 主控制器设计参考 (简介, 无实际内容)

## 5.1. SPI Bus Protocol

### 5.1. SPI 总线协议

While the SD Card channel is based on command and data bit-streams, which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of eight bit bytes and is byte aligned (multiples of eight clocks) to the CS signal.

当 SD 卡通道是基于命令和数据位流模式上时, 通常是开始于一个“起始位”和终止于一个“停止位”, 这个 SPI 通道是用字节作为单位的。CS 信号上的每个命令或数据块都是 8bit 的字节和字节队列 (就是 8 个时钟的倍数)。

Similar to the SD Bus protocol, the SPI messages are built from command, response and data-block tokens. All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.

相似 SD 总线模式, 这个 SPI 信息是建立在命令、应答和数据块标记上的。所有的主控制器和 SD 卡之间的通讯都是受控于主控制器。而主控制器的每次总线处理都是从拉低 CS 的信号电平开始的。

The response behavior in SPI Bus mode differs from the SD Bus mode in the following three ways:

- \* The selected card always responds to the command.
- \* An eight or 16-bit response structure is used.
- \* When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than time-out as in the SD Bus mode.

SPI 总线协议和 SD 总线协议的应答行为有下列三处的不同点:

1. 被选中的卡总是对命令做出应答;
2. 使用的应答结构是 8 或 16 位的;
3. 当这个卡碰到一个数据检索问题, 它将用一个错误应答来回答 (取代了原先期望的数据块) 而不是 SD 总线协议模式下的超时。

In addition to the command response, every data block sent to the card during write operations will be responded with a special data response token. A data block may be as big as one card write block (WRITE\_BL\_LEN) and as small as a single byte (NOTE1).

除了命令的应答之外, 每个数据块发送到 SD 卡后, 在写操作的期间将用一个特别的数据应答标记来应答。一个数据块可以大到卡的写入块长度 (WRITE\_BL\_LEN) 或小到一个单字节的大小 (注意 1)。

NOTE1: The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.

注意 1: 缺省的块长度定义在 CSD 寄存器内 (512 字节)。设定块的长度小于 512 字节将会导致写入错误。唯一合法的块写入长度只能定为 512 字节。若此缺省值可接受, 则无需使用 CMD16 命令, 它不是必要的。

#### 5.1.1. Mode Selection

##### 5.1.1. 模式选择

The SD Card wakes up in the SD Bus mode. It will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0). If the card recognizes that the SD Bus mode is required it will not respond to the command and remain in the SD Bus mode. If SPI mode is required, the card will switch to SPI mode and respond with the SPI mode R1 response.

SD 卡启动时处在 SD 总线模式下。它将在 CS 信号有效 (低电平) 时接收一个复位命令 (CMD0) 来进入 SPI 总线模式。如果这个卡认为必须停留在 SD 总线模式, 那么它将不应答这个命令并继续保持在 SD 总线模式。如果可以转换到 SPI 总线模式, 则它会转换到 SPI 总线模式并用 SPI 模式的 R1 应答来作回应。

The only way to return to the SD Bus mode is by power cycling the card. In SPI mode, the SD Card protocol state machine is not observed. All the SD Card commands supported in SPI mode are always available.

而唯一让卡返回 SD 总线模式的办法是用电源周期 (即电源开关一次)。在 SPI 总线模式下, SD 卡的保护状态机将不会被遵守了。SD 卡所有能支持的命令在 SPI 模式下总是可以使用的。

The default command structure/protocol for SPI mode is that CRC checking is disabled. Since the card powers up in SD Bus mode, CMD0 must be followed by a valid CRC byte (even though the command is sent using the SPI structure). Once in SPI mode, CRCs are disabled by default.

SPI 总线模式下, 默认的命令结构/协议 CRC 检查是失效的。由于卡在电源开启时总是处于 SD 总线模式, CMD0 命令必须跟随一个合法的 CRC 字节 (即使这个命令使用了 SPI 结构来发送)。一旦进入 SPI 总线模式, CRC 就被默认为失效的了。

CMD0 is a static command and always generates the same 7-bit CRC of 4Ah. Adding the "1," end bit (bit 0) to the CRC creates a CRC byte of 95h. The following hexadecimal sequence can be used to send CMD0 in all situations for SPI mode, since the CRC byte (although required) is ignored once in SPI mode. The entire CMD0 sequence appears as 40 00 00 00 00 95 (hexadecimal).

CMD0 是一个静态的命令, 总是用 0A4h 来产生同一个 7 位的 CRC, 附加一个 "1" 在最低位 (bit0) 来产生了 CRC 检查字节 095h。随后的 CMD0 命令的十六进制数序列可以在 SPI 总线模式的任何情况下发送, 而此后的 CRC 字节 ("尽管还是必须的")。译者注: 原文的意思是指这个 CRC 字节所占的位置还是不可省略掉) 会在 SPI 模式里被忽略。这个完整的 CMD0 命令是 40 00 00 00 00 95 (十六进制系列)。

### 5.1.2. Bus Transfer Protection

#### 5.1.2. 总线传送保护

Every SD Card token transferred on the bus is protected by CRC bits. In SPI mode, the SD Card offers a non protected mode which enables systems built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.

在总线上传送的每个 SD 卡标志都是用 CRC 的校验位来保护的。在 SPI 模式里, SD 卡提供了一个可靠的非保护模式的连接方式, 它排除了用硬件或固件来产生 CRC 作为校验功能的必要性。

In the non-protected mode the CRC bits of the command, response and data tokens are still required in the tokens however, they are defined as "don't care" for the transmitters and ignored by the receivers.

在这个非保护模式的命令、应答、和数据标记中的 CRC 特征位仍然是需要的, 然而, 它们的含意为: 发送方 "无需关注" 和由接收方忽略掉。

The SPI interface is initialized in the non-protected mode. The host can turn this option on and off using CRC\_ON\_OFF command (CMD59).

这个 SPI 接口是初始化成非保护模式的, 而主控制器可以通过使用 CRC\_ON\_OFF 命令 (CMD59) 来切换它的开、关状态。(译者注: 根据本人的实际使用情况来看, SPI 传输非常可靠, 没必要使用 CRC 来自找麻烦。)

The CRC7/CRC16 polynomials are identical to that used in SD Bus mode. Refer to this section in the SD Bus mode chapter.

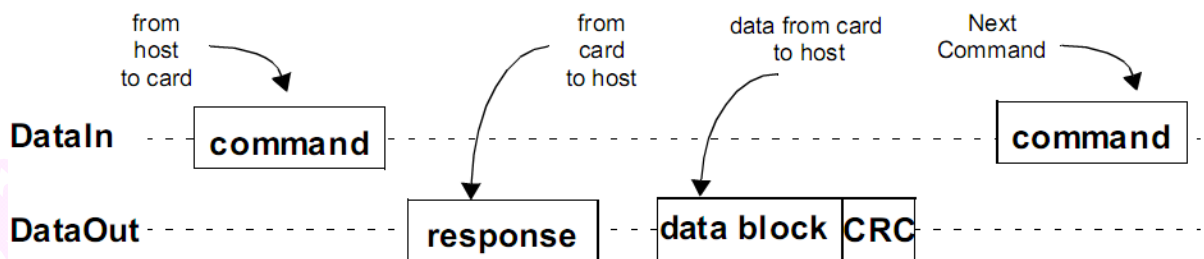
此 CRC7/CRC16 多项式是和 SD 总线模式相同的。请参考 SD 总线模式的这些章节描述。

### 5.1.3. Data Read

#### 5.1.3. 数据的读取

SPI mode supports single block and multiple block read operations (SD Card CMD17 or CMD18). Upon reception of a valid read command the card will respond with a response token followed by a data token in the length defined in a previous SET\_BLOCK\_LENGTH (CMD16) command (see Figure 5-1).

SPI 模式可支持单个块和多个块的读操作 (SD 卡的 CMD17 或 CMD18)。在接收到一个合法的读取命令后, 这个卡将用一个应答标志来应答, 随后的就是一个数据块, 它的长度是先前定义在 SET\_BLOCK\_LENGTH (CMD16) 命令中的 (请看图表 5-1)。



图表 5-1 单个数据块的读取操作

A valid data block is suffixed with a 16-bit CRC generated by the standard CCITT polynomial:

一个有效的数据块的尾缀是 16bit 的 CRC, 它是用符合 CCITT 标准的多项式产生的:

$$X^{16} + X^{12} + X^5 + 1.$$

The maximum block length is 512 bytes as defined by READ\_BL\_LEN (CSD parameter). Block lengths can be any number between 1 and READ\_BL\_LEN.

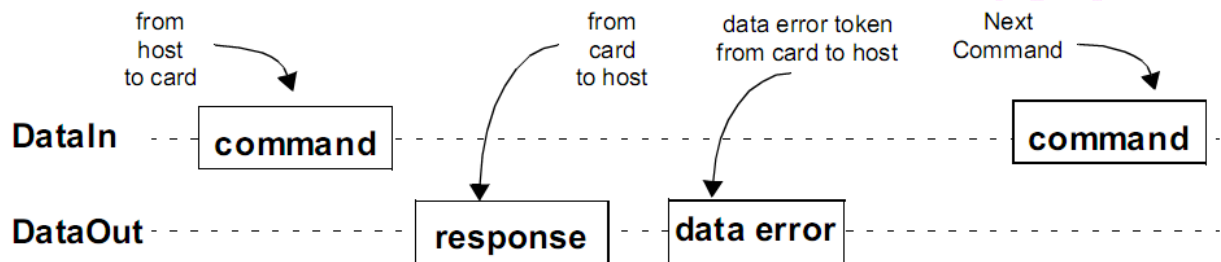
这个最大的块长度是 512 字节, 由 READ\_BL\_LEN (CSD 参数) 定义的。块的长度可以是 1 到 READ\_BL\_LEN 之间的任何一个数值。

The start address can be any byte address in the valid address range of the card. Every block, however, must be contained in a single physical card sector.

这个开始地址可以是包含在卡的合法地址范围内的任何字节地址。但是, 每个块必须是包含在卡的一个单独物理扇区内。

In case of data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 5-2 shows a data read operation, which terminated with an error token rather than a data block.

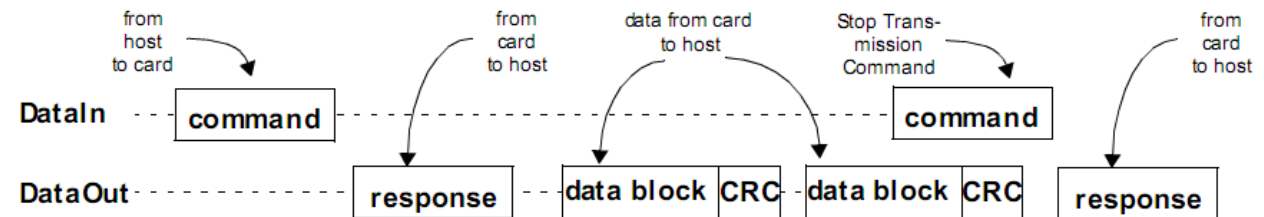
在数据接收错误时 (译者注: 这里应该是指 SD 卡接收到的读取命令中包含了错误的参数, 例如“块长度”或“地址”参数使用了 SD 卡规定之外的格式), 这个卡将不传送任何数据。替代的将是一个报告数据错误的标记被发送到主控制器。图表 5-2 显示了一个数据的读操作, 结果由于一个错误标记而终止, 而不是期待的一个数据块。



图表 5-2. 读取操作 - 数据错误

In the case of a Multiple Block Read operation, every transferred block has a 16-bit CRC suffix. The Stop Transmission command (CMD12) will actually stop the data transfer operation (the same as in SD Bus mode).

在多个数据的读取操作情况下, 每个被传送的块都有一个 16bit 的 CRC 后缀。而停止发送命令 (CMD12) 将会终止这个数据传送操作 (和 SD 总线模式是相同的)。



图表 5-3. 多个块的读取操作

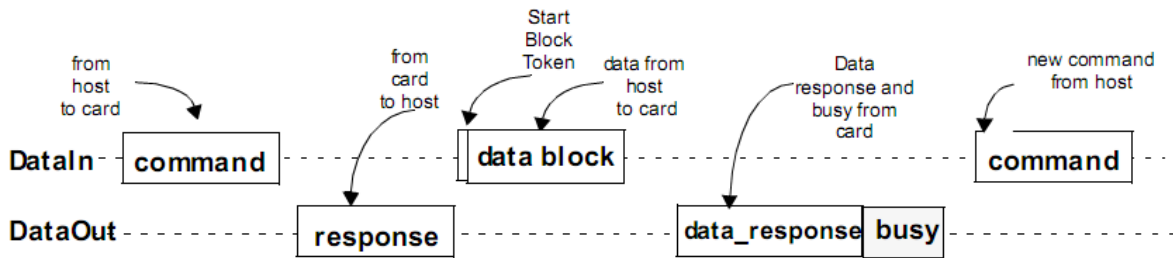
#### 5.1.4. Data Write

#### 5.1.4. 数据写入

In SPI mode, the SD Card supports single block or multiple block write operations. Upon reception of a valid write command (SD Card CMD24 or CMD25), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix and start address restrictions are identical to the read operation (see Figure 5-4). The only valid block length, however, is 512 bytes. Setting a smaller block length will cause a write error on the next write command.

在 SPI 模式下, SD 卡支持单块的或多块的写操作。在接收到一个合法的写命令时 (SD 卡命令 CMD24 或 CMD25), SD 卡将响应一个应答标记和等待主控制器发送这个数据块。CRC 后缀和起始地址的限制和读操作是一样的 (参看图表 5-4)。唯一合法的块长度是 512 字节。如果设置了一个小于此值的块长度, 将会导致下一条写入命令产生错误。





图表 5-4. 单个块的写入操作

Every data block has a prefix or 'start block' token (one byte). After a data block is received the card will respond with a data-response token, and if the data block is received with no errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the dataOut line low).

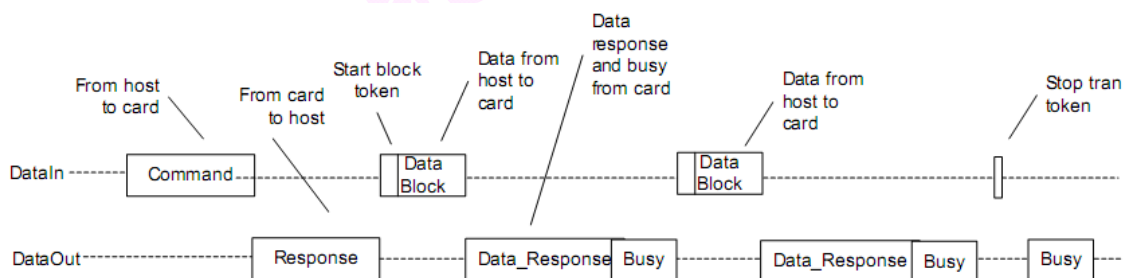
每个数据块都有一个前缀或“开始快”标记（一个字节）。如果在一个数据块的接收被确认无误后，SD 卡将用一个数据应答来回应，然后它将被烧录（编程）。卡在烧录（编程）期间是长时间处于忙状态的，一个持续不断的忙标记将被发送到主控制器（即始终保持 DATAOUT 信号线为低电平）。

Once the programming operation is completed, the host must check the results of the programming using the SEND\_STATUS command (CMD13). Some errors (e.g., address out of range, write protect violation, etc.) are detected during programming only. The only validation check performed on the data block and communicated to the host via the data-response token is CRC and general Write Error indication.

一旦这个烧录（编程）操作完成，主控制器必须使用 SEND\_STATUS 命令(CMD13)来检查这次编程烧录的结果。一些错误（例如：地址超出范围、违反写保护等等）的检查只会在烧录（编程）期间才能进行。唯一在此数据块和主控制器通讯上执行的检查校验，只能是通过数据应答标记的 CRC 和产生写入错误的指示标记。

In Multiple Block write operation the stop transmission will be done by sending 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block. In case of Write Error indication (on the data response) the host shall use SEND\_NUM\_WR\_BLOCKS (ACMD22) in order to get the number of well written write blocks. The data token's description is given in Section 5.2.4.

如果需要在多个块的写入操作中停止传送过程，则是在开始传送下一块的时候，通过发送“停止传送”标记来代替“起始块”标记。如果发生写错误指示（在此数据应答上），主控制器可以使用 SEND\_NUM\_WR\_BLOCKS 命令（ACMD22）来获得已经写入的块的数量。此数据标记的描述在章节 5.2.4 中。



图表 5-5. 多个块的写入操作

Resetting the CS signal while the card is busy does not terminate the programming process. The card releases the dataOut line (tristate) and continue to program. If the card is reselected before the programming is done, the dataOut line will be forced back to low and all commands will be rejected.

在卡处于“忙”状态的时候，重新设置 CS 信号并不能终止烧录（编程）的处理过程。卡将会释放 DATAOUT 信号线（变成三态）和继续进行烧录（编程）的处理。如果在烧录（编程）完成之前重新选中该卡，DATAOUT 信号线将会被强制置为低电平，并且拒绝所有的命令。

Resetting a card (using CMD0) will terminate any pending or active programming operation. This may destroy the data formats on the card. It is the host's responsibility to prevent it.

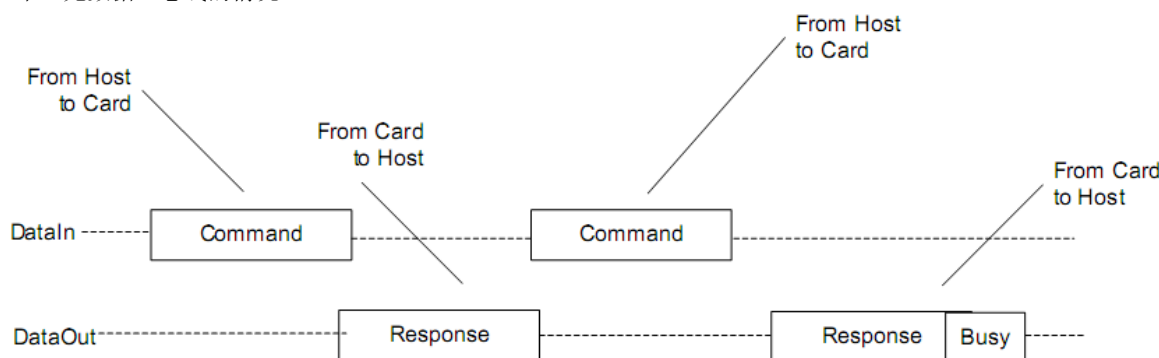
而复位一个卡（使用 CMD0 命令）将终止任何等待执行或正在进行的编程烧录操作动作。这样会破坏卡上的数据格式。主控制器必须负责避免出现这种状况（译者注：这种状况可能会导致你要重新格式化该卡。）。

### 5.1.5. Erase and Write Protect Management

#### 5.1.5. 擦除和写保护处理

The erase and write protect management procedures in the SPI mode are identical to the SD Bus mode. While the card is erasing or changing the write protection bits of the predefined sector list it will be in a busy state and will hold the dataOut line low. Figure 5-6 illustrates a “no data” bus transaction with and without busy signaling.

在 SPI 模式下, 擦除和写保护管理程序和 SD 模式下是一样的。当该卡进行擦除、或对指定的扇区目录的写保护位进行修改时, 它将处于忙状态并保持 DATAOUT 信号线为低电平。图表 5-6 举例说明了用外部的忙信号来处理一个“无数据”总线的情况。



图表 5-6. “无数据”操作

### 5.1.6. Read CID/CSD Registers

#### 5.1.6. 读取 CID/CSD 寄存器

Unlike the SD Bus protocol (where the register contents are sent as a command response), reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token followed by a data block of 16 bytes suffixed with a 16-bit CRC.

与 SD 总线协议不同 (那些寄存器内容是包含在发送的命令应答内), 在 SPI 总线模式里是通过一个简单的“读块”处理来获取 CSD 和 CID 寄存器的内容。这个卡将用一个标准的响应标记、和跟随其后的 16 字节的数据块加上它 16bit CRC 的后缀来响应。

The data time out for the CSD command cannot be set to the card TAAC since this value is stored in the CSD. Therefore, the standard response time-out value (Ncr) is used for read latency of the CSD register.

由于此数值储被储存在 CSD 中, 不能用 CSD 命令中的 TimeOut 数值来设置 TAAC。因此, 这个标准的响应 TimeOut 值 (Ncr) 是用来表示 CSD 寄存器的读取等待时间。

### 5.1.7. Reset Sequence

#### 5.1.7. 复位顺序

The SD Card requires a defined reset sequence. After power on reset or CMD0 (software reset), the card enters an idle state. At this state, the only legal host commands are CMD1 (SEND\_OP\_COND), ACMD41 (SD\_SEND\_OP\_COND), CMD59 (CRC\_ON\_OFF) and CMD58 (READ\_OCR).

SD 卡有设定好的复位顺序。在上电复位或用软复位 (用 CMD0 命令) 之后, 这个卡将进入静止状态。在这个状态里, 只能接收 4 条合法的主控命令: CMD1 (SEND\_OP\_COND), ACMD41 (SD\_SEND\_OP\_COND), CMD59 (CRC\_ON\_OFF) 和 CMD58 (READ\_OCR)。

The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card completed its initialization processes and is ready for the next command.

主控制器必须连续检测这个卡 (通过不断地发送 CMD1), 直到 in-idle-state 位在卡的应答中指示 (即被设置成 0) 出该卡已经完成了初始化处理, 并且可以接收下一条命令。

In SPI mode, however, CMD1 has no operands and does not return the contents of the OCR register. Instead, the host can use CMD58 (SPI Mode Only) to read the OCR register. It is the responsibility of the host to refrain from accessing cards that do not support its voltage range.

在 SPI 模式里, 无论什么情况下 CMD1 都没有操作数, 和它不会返回 OCR 寄存器的内容。作为一个替代品, 命令 CMD58 (仅仅为 SPI 模式命令) 可以读取 OCR 寄存器。它的职责是防止主控制器去访问一个不在规定电压支持范围内的卡。

The use of CMD58 is not restricted to the initialization phase only, but can be issued at any time. The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card has completed its initialization process and is ready for the next command.

而这个 CMD58 也并不是只限制在初始化阶段使用, 它可以在任何时候使用。只是主控制器必须连续检测这个卡 (通过不断地发送 CMD1), 直到 in-idle-state 位在卡的应答中指示 (即被设置成 0) 出该卡已经完成了初始化处理并且可以接收下一条命令 (译者注: 这里的描述很含糊, 意思应该是说在 SPI 协议模式下, CMD58 命令可以替代 SD 协议模式下的 CMD1 命令来获取 OCR 寄存器的内容, 但是必须要等 SD 卡的初始化处理过程完毕, 而 SD 卡的初始化处理过程是否已经完毕还是要用 CMD1 来查询)。

#### 5.1.8. Clock Control

##### 5.1.8. 时钟控制

The SPI bus clock signal can be used by the SPI host to set the cards to energy-saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to change the clock frequency or shut it down.

SPI 总线时钟可以通过 SPI 主控来设置这个卡到“省电模式”, 或者用来控制数据在总线上的流动率 (用来避免“使用不足”或“过度占用”情况)。主控可以修改时钟频率或者关掉它 (译者注: 应该是指关闭 SPI 总线和相关控制电路)。

There are a few restrictions the SPI host must follow:

这里对 SPI 主控有几条必须遵守的限制情况:

- \* The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the SD Cards).

- \* 总线频率可以在任何时间修改 (不允许超过数据的最大传送频率, 它在 SD 卡中是有定义的)。

- \* It is an obvious requirement that the clock must be running for the SD Card to output data or response tokens. After the last SPI bus transaction, the host is required to provide 8 (eight) clock cycles for the card to complete the operation before shutting down the clock. Throughout this 8-clock period, the state of the CS signal is irrelevant. It can be asserted or de-asserted. Following is a list of the various SPI bus transactions:

- \* 显而易见, SD 卡要依靠这个时钟的运作来输出数据或应答标记。在当前的 SPI 总线传送结束之后, 和在停止时钟运作之前, 主控必须提供 8 个时钟周期给卡来完成这个操作。在这整个的 8 个时钟期间, CS 信号的状态是无所谓的, 它可以有效 (低电平) 也可以无效 (高电平)。下面是 SPI 总线的各种状况列表:

- A command/response sequence. Eight clocks after the card response end bit. The CS signal can be asserted or de-asserted during these 8 clocks.

一个命令/应答顺序。在卡的应答的结束 bit 之后跟随 8 个时钟。CS 信号在此 8 个时钟期间的状态是无所谓的。

- A read data transaction. Eight clocks after the end bit of the last data block.

读数据的处理。在数据块最后结束位之后跟随 8 个时钟。

- A write data transaction. Eight clocks after the CRC status token.

写数据的处理。在 CRC 的状态标记之后跟随 8 个时钟。

- \* The host is allowed to shut down the clock of a "busy" card. The SD Card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge, the SD Card (unless previously disconnected by de-asserting the CS signal) will force the dataOut line down, permanently.

主控制器可以在卡的“忙”状态时关闭时钟信号。SD 卡将去完成这个烧录 (编程) 操作而不理睬主控制器的时钟信号。然而, 主控制器必须要提供一个时钟沿给 SD 卡来关掉它的“忙”信号。没有这个时钟沿, SD 卡将永久性的将 DATAOUT 信号线强制为低 (除非先前通过将 CS 信号声明为无效来断开连接)。

#### 5.1.9. Error Conditions

##### 5.1.9. 错误条件

The following sections provide valuable information on error conditions.

以下章节提供了关于错误情况方面的重要信息。

##### 5.1.9.1. CRC and Illegal Commands

##### 5.1.9.1. CRC 和无效指令

Unlike the SD Card protocol, in SPI mode the card will always respond to a command. The response indicates acceptance or rejection of the command. A command may be rejected in any one of the following cases:

不同于 SD 协议模式, 在 SPI 模式里卡总是对命令做出应答。这个应答指出了对该命令进行接受或拒绝。命令被拒绝是属于下列的情况之一:



- \* It is sent while the card is in read operation (except CMD12 which is legal).
- \* 它发送的时候卡正处于读模式操作中 (合法的 CMD12 是个例外)。
- \* It is sent while the card is in Busy.
- \* 它发送的时候卡正处于忙状态。
- \* Card is locked and it is other than Class 0 or 7 commands.
- \* 卡锁定的时候, 它是除了 0 或 7 类之外的命令。
- \* It is not supported (illegal opcode).
- \* 它是不被支持的 (非法操作码)。
- \* CRC check failed.
- \* CRC 检查失败。
- \* It contains an illegal operand.
- \* 它包含了一个非法操作。
- \* It was out of sequence during an erase sequence.
- \* 它在擦除序列期间脱离了该顺序。

Note that in case the host sends command while the card sends data in read operation then the response with an illegal command indication may disturb the data transfer.

要注意这种情况: 当主控制器发送了命令, 而卡正在读操作下发送数据的时候, 一个非法命令指示的应答可能会干扰数据的传送。(译者注: 必须要杜绝这种情况, 否则被打乱的读操作传送可能会引起后续的时序混乱, 导致你编写的软件读取不到预期的数据也产生混乱状态。)

#### 5.1.9.2. Read, Write and Erase Time-out Conditions

##### 5.1.9.2. 读取、写入和擦除的超时情形

The times after which a time-out condition for read operations occur are (card independent) either 100 times longer than the typical access times for these operations given below or 100ms.

读取操作超时的情况 (卡不受控制) 有两种。这些操作超过了: 1) 约定的典型访问时间的 100 倍; 2) 100 毫秒。

The times after which a time-out condition for Write/Erase operations occur are (card independent) either 100 times longer than the typical program times for these operations given below or 250ms.

写入/擦除超时的情况 (卡不受控制) 有两种。这些操作超过了: 1) 约定的典型访问时间的 100 倍; 2) 250 毫秒。

A card shall complete the command within this time period, or give up and return an error message. If the host does not get any response with the given time out it should assume the card is not going to respond anymore and try to recover (for example; reset the card, power cycle, reject). The typical access and program times are defined in the following sections.

一个卡会在命令的规定时间段内完成它, 或者放弃并返回一个错误信息。如果主控制器在给定的时限没有获得任何的应答, 它应该假设该卡不能再应答了, 并要设法来恢复它 (例如: 复位这个卡、重新开关电源、弹出)。这种典型的访问和编程烧录时间是定义在下面的章节中。

For more information, refer to Table 4-17 in Section 4.0, Table 5-5 in Section 5.0 and the applications note in Appendix A, "Host Design Considerations: NAND MMC and SD-based Products."

更多信息请参考章节 4.0 中的表 4-17, 章节 5.0 中的表 5-5 和附录 A 中的应用注意事项, “主控制器设计的注意事项: NAND MMC 和基于 SD 的产品。”

Table 4-17. Timing Values

|     | Min. | Max.      | Unit         |
|-----|------|-----------|--------------|
| NCR | 2    | 64        | Clock Cycles |
| NID | 5    | 5         | Clock Cycles |
| NAC | 2    | See note. | Clock Cycles |
| NRC | 8    | -         | Clock Cycles |
| NCC | 8    | -         | Clock Cycles |
| NWR | 2    | -         | Clock Cycles |

Table 4-17 NOTE:  $\min \{[(TAAC * f) + (NSAC * 100)], \{(100ms * f)\}$  where units = (clocks) and "f" is the clock frequency.

表 4-17 注:  $\min \{[(TAAC * f) + (NSAC * 100)], \{(100ms * f)\}$  (单位: clocks), "f" 是时钟频率。

**Table 5-5. Timing Constants Definitions**

|     | Min | Max      | Unit           |
|-----|-----|----------|----------------|
| NCS | 0   | -        | 8 Clock Cycles |
| NCR | 0   | 8        | 8 Clock Cycles |
| NRC | 1   | -        | 8 Clock Cycles |
| NAC | 1   | See Note | 8 Clock Cycles |
| NWR | 1   | -        | 8 Clock Cycles |
| NEC | 0   | -        | 8 Clock Cycles |
| NDS | 0   | -        | 8 Clock Cycles |
| NBR | 0   | 1        | 8 Clock Cycles |

Table 5-5 NOTE:  $\min \{ \{ (TAAC * f) + (NSAC * 100) \} * 1/8, \{ (100ms * f) * 1/8 \}$  where units = (8 clocks) and “f” is the clock frequency.

表 5-5 注:  $\min \{ \{ (TAAC * f) + (NSAC * 100) \} * 1/8, \{ (100ms * f) * 1/8 \}$  (单位: 8 个 clocks), “f” 是时钟频率。

#### Read

The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block.

#### 读取

读取访问时间是定义为 CSD 的参数中的 TAAC 和 NSAC 两个之和。这些卡的参数定义了在**读取命令的结束位**和**数据块的起始位**之间的典型延时。

#### Write

The R2W\_FACTOR field in the CSD is used to calculate the typical block program time obtained by multiplying the read access time by this factor. It applies to all write/erase commands (e.g., SET (CLEAR)\_WRITE\_PROTECT, PROGRAM\_CSD (CID) and the block write commands).

#### 写入

CSD 中的 R2W\_FACTOR 区域用于计算典型的块烧录(编程)时间, 是通过这个因素乘以读取访问时间来获得的。它将用于所有的写入/擦除命令(例如: 设置(CLEAR)\_WRITE\_PROTECT、PROGRAM\_CSD (CID)和块写入命令)。

#### Erase

The duration of an erase command will be (order of magnitude) the number of write blocks (WRITE\_BL) to be erased multiplied by the block write delay.

#### 擦除

这个擦除命令的持续时间将是(数量级)被擦除的写入块数量(WRITE\_BL)乘以块的写入延迟。

### 5.1.10. Memory Array Partitioning

#### 5.1.10. 存储器阵列分布

Same as for SD Card mode.

存储器的阵列分布和 SD 总线模式相同(译者注: 实际内容在原文第一章, 而不是第四章的“SD 协议模式”, 请翻到本文的最后看存储器的阵列图)。

### 5.1.11. Card Lock/Unlock

#### 5.1.11. 卡的锁定/解锁

The Card Lock/Unlock feature is currently in the SanDisk SD Card.

卡的锁定和解锁的特性已经普遍使用在 SanDisk 的 SD 卡中。

### 5.1.12. Application Specific Commands

#### 5.1.12. 特殊用途的命令集

The Application Specific commands are identical to SD mode with the exception of the APP\_CMD status bit (see Section 3.5.5), which is not available in SPI.

特殊用途的命令集除了在 APP\_CMD 状态位上以外和 SD 总线模式是一样的(参考章节 3.5.5 “Card Status (卡身份) 寄存器 - 这个状态区域是兼容 MMC 协议的”), 只有它不能在 SPI 协议模式下使用。

| 位号 | 标识符     | 类型  | 参数                          | 说明                           | 清除条件 |
|----|---------|-----|-----------------------------|------------------------------|------|
| 5  | APP_CMD | S R | '0'=Disabled<br>'1'=Enabled | 该卡期望着ACMD, 或者表示命令可以当作ACMD一样。 | C    |

- S - 状态位。
- R - 由实际命令的应答来侦测和设置。
- C - 通过读取清除。

### 5.1.13. Copyright Protection Commands

#### 5.1.13. 版权保护命令集

All the special Copyright Protection ACMDs and security functionality are the same as for SD mode.

所有的版权保护命令集 (ACMDs) 和安全功能同 SD 总线协议模式是一样的。(译者注: 原文中, 在 SD 协议模式下的描述中并未提到这个命令集, 所以本人也无法在附录中提供相关内容了, 还有的可能就是, 在指令表: “表 5-3. 用于或保留在 SPI 模式下的特殊用途命令集” 中的相关描述内容。)

### 5.2. SPI Command Set

#### 5.2. SPI 命令设置

The following sections provide valuable information on the SPI Command Set.

下面的章节提供了关于 SPI 命令规定的重要信息。

#### 5.2.1. Command Format

##### 5.2.1. 命令格式

All the SD Card commands are 6 bytes long and transmitted MSB first.

所有的命令格式都是 6 字节长度, 首先发送 MSB (译者注: 此命令系列的第一个字节 Byte1 的最高位 bit7, 也就是下表中的第一个二进制数值 “0” )。

| Byte 1 |   |                |       | Bytes 2—5 |                           |       |   | Byte 6      |   |
|--------|---|----------------|-------|-----------|---------------------------|-------|---|-------------|---|
| 7      | 6 | 5              | ..... | 0         | 31                        | ..... | 0 | 7           | 0 |
| 0      | 1 | Command ( 命令 ) |       |           | Command Argument ( 命令参数 ) |       |   | CRC ( 校验值 ) |   |
|        |   |                |       |           |                           |       |   |             | 1 |

Commands and arguments are listed in Table 5-2.

命令集和它们的参数都罗列在表格 5-2 中。(译者注: arguments 也有资料称作 “变元”, 尤其是港台出版的中文资料, 而在国内基本称作 “参数” )

7-bit CRC Calculation ( 7 位 CRC 的计算方法 ) :  $G(x) = x^7 + X^3 + 1$

$M(x) = (\text{start bit}) * x^{39} + (\text{host bit}) * x^{38} + \dots + (\text{last bit before CRC}) * x^0$

$CRC[6...0] = \text{Remainder ( 余数 ) } [(M(x) * x^7)/G(x)]$

#### 5.2.2. Command Classes

##### 5.2.2. 命令类

As in SD mode, the SPI commands are divided into several classes (See Table 5-1). Each class supports a set of card functions. An SD Card will support the same set of optional command classes in both communication modes (there is only one command class table in the CSD register). The available command classes, and the supported commands for a specific class, however, are different in the SD Card and the SPI communication mode.

在 SD 卡里 (译者注: 原文的意思是 “在 SD 模式下”, 这可能是原文的笔误), SPI 命令集被分为几个单独的类别 (参看表 5-1)。每个类别支持一组在卡中所规定的功能。SD 卡有选择地支持在两种通讯协议模式下有相同规定的命令类 (这唯一的一张命令类表在 CSD 寄存器内)。然而, 有一些命令类和一些特殊类的支持命令集在 SD 卡里的 SPI 总线模式下是不同的 (译者注: 这段原文的意思很含糊。它的意思应该是说: 在 CSD 寄存器里面储存着当前这张 SD 卡所能支持的命令类, 这些类中包含的命令集在 SD 协议模式和 SPI 协议模式下, 其中一部分的使用方法和它所代表的功能是相同的, 但有一些命令集和特殊功能的命令集是不同的, 请看后面的详细叙述)。

Note that except the classes that are not supported in SPI mode (class 1, 3 and 9), the mandatory required classes for the SD mode are the same for the SPI mode.

注意例外的情况: 有些类是不支持 SPI 模式的 (类别 1, 3 和 9), 和 SPI 模式下的专用命令一样, 它们是被强制在 SD 模式下使用的。

表 5-1. SPI 模式下的命令类

| Card CMD Class<br>(CCC) | Class Description              | Supported Commands |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
|-------------------------|--------------------------------|--------------------|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
|                         |                                | 0                  | 1 | 9 | 10 | 12 | 13 | 16 | 17 | 18 | 24 | 25 | 27 | 28 | 29 | 30 | 32 | 33 | 38 | 42 | 55 | 56 | 58 | 59 |  |
| class 0<br>(基本指令)       | Basic                          | +                  | + | + | +  | +  | +  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | +  | +  |  |
| class 1                 | Not supported in SPI           |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| class 2<br>(数据块读取)      | Block read                     |                    |   |   |    |    |    | +  | +  | +  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| class 3                 | Not supported in SPI           |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| class 4<br>(数据块写入)      | Block write                    |                    |   |   |    |    |    |    |    |    | +  | +  | +  |    |    |    |    |    |    |    |    |    |    |    |  |
| class 5<br>(擦除类)        | Erase                          |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    | +  | +  | +  |    |    |    |    |    |  |
| class 6<br>(写保护) 可选     | Write-protection<br>(Optional) |                    |   |   |    |    |    |    |    |    |    |    |    | +  | +  | +  |    |    |    |    |    |    |    |    |  |
| class 7<br>(锁定卡) 可选     | Lock Card (Optional)           |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | +  |    |    |    |    |  |
| class 8<br>(特殊用途)       | Application specific           |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | +  | +  |    |    |  |
| class 9                 | Not supported in SPI           |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| class 10-11             | Reserved                       |                    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |

\* The Lock Card command is supported in the SD Card.

\* 锁定卡的命令是在 SD 卡里支持的 (译者注: 联系本手册的上下文说明, 每个 SD 卡产品实际上是必需要支持锁定类命令, 而不是本文此处说的“它是可选项”)。

5.2.2.1. Detailed Command Description

5.2.2.1.详细的命令描述

The following table provides a detailed description of the SPI bus commands. The responses are defined in Section 5.2.3. Table 5-2 lists all SD Card commands. A “yes” in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are reserved in SD Card mode as well.

下面这张表提供了 SPI 总线协议下命令的详细描述, 而应答的定义在章节 5.2.3 中。表 5-2 列出了所有的 SD 卡命令。在“SPI 模式”栏内“yes”代表该命令支持 SPI 模式。然而有一些限制, 这些命令类在 CSD 寄存器中必须被描述为有效的。如果一个命令不需要参数, 则这个命令的参数区域应该全部设置为 0。那些保留的命令在 SD 协议模式里也同样是保留的。

The binary code of a command is defined by the mnemonic symbol. As an example, the content of the Command field for CMD0 is (binary) ‘000000’ and for CMD39 is (binary) ‘100111.’

这些命令的二进制代码被定义为助记符的形式。例如: 这个命令代码区域对于 CMD0 是(二进制)“000000”和 CMD39 是“100111”。

译者注: 为了排版方便, 表格 5-2、5-3 的英文原文将被附加在译文的最后面, 如有需要, 请翻到后面查阅。

表 5-2. SPI 总线协议模式下的命令集描述

| 命令索引               | SPI 模式 | 参数                                      | 应答方式            | 简写                      | 命令描述   |
|--------------------|--------|---|-----------------|-------------------------|--|
| CMD0               | Yes    | None                                    | R1              | GO_IDLE_STATE           | 复位这个SD卡。   |
| CMD1               | Yes    | None                                    | R1              | SEND_OP_COND            | 激活该卡的初始化处理。  |
| CMD2...4, 7        | No     |   |                 |                         |  |
| CMD5...6, 8        |        |   |                 | Reserved                |  |
| CMD9               | Yes    | None                                    | R1              | SEND_CSD                | 让被选中的卡发送它的详细数据 (CSD寄存器)。   |
| CMD10              | Yes    | None                                    | R1              | SEND_CID                | 让被选中的卡发送它的卡ID (CID寄存器)。  |
| CMD11              | No     |   |                 |                         |  |
| CMD12              | Yes    | None                                    | R1b             | STOP_TRANSMISSION       | 强制该卡在读取多个数据块的操作期间中止传送。   |
| CMD13              | Yes    | None                                    | R2              | SEND_STATUS             | 让被选中的卡发送它的 status register 身份寄存器。                                |
| CMD14...15         | No     |   |                 |                         |  |
| CMD16              | Yes    | [31:0] 块长度                              | R1              | SET_BLOCKLEN            | 设定后面所有数据块 (读/写) 的长度 (字节)。 <sup>注1</sup>                          |
| CMD17              | Yes    | [31:0] 数据地址                             | R1              | READ_SINGLE_BLOCK       | 读取一个大小定义为 SET_BLOCKLEN 的数据块 (字节)。 <sup>注2</sup>                  |
| CMD18              | Yes    | [31:0] 数据地址                             | R1              | READ_MULTIPLE_BLOCK     | 从卡中向主控制器持续不断地传送数据块, 直到被 STOP_TRANSMISSION 命令终止。                  |
| CMD19, 21...23     |        |   |                 | Reserved                |  |
| CMD20              | No     |   |                 |                         |  |
| CMD24              | Yes    | [31:0] 数据地址                             | R1 <sup>3</sup> | WRITE_BLOCK             | 写入大小为 SET_BLOCKLEN 的数据块 (字节)。 <sup>注4</sup>                      |
| CMD25              | Yes    | [31:0] 数据地址                             | R1              | WRITE_MULTIPLE_BLOCK    | 从主控向SD卡内持续不断地写入数据块, 直到停止传送标志被发送 (“开始块” 标记被替代)。                   |
| CMD26              | No     |   |                 |                         |  |
| CMD27              | Yes    | None                                    | R1              | PROGRAM_CSD             | 对CSD中可编程改写的位进行烧录 (编程) 操作。  |
| CMD28 <sup>5</sup> | Yes    | [31:0] 数据地址                             | R1b             | SET_WRITE_PROT          | 如果该卡有写入保护功能, 则此命令可以设置这个地址组的写保护位。写保护部分的编码在卡的特殊数据 (WP_GRP_SIZE) 里。 |
| CMD29 <sup>5</sup> | Yes    | [31:0] 数据地址                             | R1b             | CLR_WRITE_PROT          | 如果此卡有写保护功能, 则此命令可以清除这个地址组的写保护位。                                  |
| CMD30              | Yes    | [31:0] 写保护数据地址                          | R1              | SEND_WRITE_PROT         | 如果此卡有写保护功能, 这个命令可以让卡发送它的写保护位来查询这些位的状态。 <sup>注6</sup>             |
| CMD31              |        |   |                 | Reserved                |  |
| CMD32              | Yes    | [31:0] 数据地址                             | R1              | ERASE_WR_BLK_START_ADDR | 设置需要擦除的首个可写块的地址。   |
| CMD33              | Yes    | [31:0] 数据地址                             | R1              | ERASE_WR_BLK_END_ADDR   | 设置需要擦除的一个连续范围内的最后那个可写块的地址。                                       |
| CMD34...37         |        |   |                 | Reserved                |  |
| CMD38              | Yes    | [31:0] 无所谓 *                            | R1b             | ERASE                   | 擦除所有先前设定的可写块 (指CMD32、33的设定)                                      |
| CMD39...40         | No     |   |                 |                         |  |
| CMD41...54         |        |   |                 | Reserved                |  |
| CMD55              | Yes    | [31:0] 填充位                              | R1              | APP_CMD                 | 通知该卡下个命令是条特殊命令, 而不是标准命令。   |
| CMD56              | Yes    | [31:0] 填充位<br>[0:0]: RD/WR <sup>7</sup> | R1              | GEN_CMD                 | 向卡发送或从卡接收数据块来获得通用/特殊的命令集。此数据块的大小是 SET_BLOCK_LEN 定义的。             |
| CMD57              |        |   |                 | Reserved                |  |
| CMD58              | Yes    | None                                    | R3              | READ_OCR                | 从此卡中读取 OCR 寄存器的内容。   |
| CMD59              | Yes    | [31:0] 无所谓 *<br>[0:0]: CRC选项            | R1              | CRC_ON_OFF              | 切换 CRC 可选项的开和关状态。“1” 在它的可选位上表示 CRC 将被打开, “0” 表示将被关上。             |
| CMD60...63         | No     |   |                 |                         |  |

\* The bit places must be filled but the values are irrelevant.

\* 这些位必须被填满, 但它填入的数值是无关紧要的。



1) The only valid block length for write is 512 bytes. The valid block length for read is 1 to 512 bytes. A set block length of less than 512 bytes will cause a write error. The card has a default block length of 512 bytes. CMD16 is not mandatory if the default is accepted.

**注 1)** 对于写入操作, 合法的块长度只能是 512 字节。而对于读取, 它的长度可以是 1 到 512 个字节。如果设置块长度小于 512 字节, 将会导致写入发生错误。卡对块长度的缺省定义是 512 字节。如果这个设定操作本身是合法的, CMD16 对此并不会作出强行限制。

2) The start address and block length must be set so that the data transferred will not cross a physical block boundary.

**注 2)** 块的起始地址和长度必须符合物理块的边界设定, 不能出现交叉的现象。

3) Data followed by data response plus busy.

**注 3)** 发送的数据块后面跟随的是数据应答和附加的忙状态。

4) The start address must be aligned on a sector boundary. The block length is always 512 bytes.

**注 4)** 起始地址必须和扇区的边界对齐。该块的长度总是 512 字节。

5) These features are not currently supported in the SanDisk SD Card.

**注 5)** 这些特性在当前的 SD 卡中还没有被全部支持。

6) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line.

**注 6)** 32 个写保护位和跟随的 16 位 CRC 用有效的格式通过数据信号线来传送 (体现这 32 个写保护的组是起始于这个指定的地址)。

7) RD/WR: "1"=the host will get a block of data from the card. "0"=the host sends a block of data to the card.

**注 7)** RD/WR: "1" 表示主控制器将从卡获得一个数据块。"0" 表示这个主控制器发送一个数据块到卡中。

译者注: 在 CMD29 的说明中, 原文有个“注 4”, 但是却又找不到相关说明, 联系上下文之后, 估计是原著者笔误, 实际上应该为“注 1”, 在本译文中注解序号被调整为“注 5”, 上面表格中的注解号已经被本人修改正确了。

Table 5-3 describes all the application specific commands supported or reserved by the SD Card. All the following commands should be preceded with APP\_CMD (CMD55).

表 5-3 描述了 SD 卡所有支持和保留的特殊用途的命令集。使用下列所有的命令之前应该先用 CMD55 来声明。

表 5-3. 用于或保留在 SPI 模式下的特殊用途命令集

| 命令索引           | SPI 模式 | 参数                         | 应答方式 | 简写                     | 命令描述   |
|----------------|--------|----------------------------|------|------------------------|--|
| ACMD6          | No     |                            |      |                        |  |
| ACMD13         | Yes    | [31:0] 填充位                 | R2   | SD_STATUS              | 发送SD卡的状态。此状态区域的描述在表4-21中。                                      |
| ACMD17         |        |                            |      | Reserved               |  |
| ACMD18         | Yes    | --                         | --   | --                     | 为SD卡的安全用途而保留。 <b>注1</b>  |
| ACMD19...21    |        |                            |      | Reserved               |  |
| ACMD22         | Yes    | [31:0] 填充位                 | R1   | SEND_NUM_WR_BLOCKS     | 发送已经写好块的数量 (如果没有错误)。用 32bit+CRC的数据块来应答。                        |
| ACMD23         | Yes    | [31:23] 填充位<br>[22:0] 块的数量 | R1   | SET_WR_BLK_ERASE_COUNT | 写入前设定写入块的数量来预擦除 (用于快速的多块写命令)。1=缺省 (一个写入块)。 <b>注2</b>           |
| ACMD24         |        |                            |      | Reserved               |  |
| ACMD25, 26, 38 | Yes    | --                         | --   | --                     | 为SD卡的安全用途而保留。 <b>注1</b>  |
| ACMD39, 40     |        |                            |      | Reserved               |  |
| ACMD41         | Yes    | None                       | R1   | SEND_OP_COND           | 启动此卡的初始化过程。  |
| ACMD42         | Yes    | [31:1] 填充位<br>[0] set_cd   | R1   | SET_CLR_CARD_DETECT    | 连接 [1] / 断开[0] 该卡上 CD/DAT3 (pin 1)上的 50K Ohm 上拉电阻。而这个上拉用于卡的侦测。 |
| ACMD43...49    | Yes    | --                         | --   | --                     | 为SD卡的安全用途而保留。 <b>注1</b>  |
| ACMD51         | Yes    | [31:0] 填充位                 | R1   | SEND_SCR               | 读取该卡的配置寄存器 (SCR)。  |

(1) Refer to "SD Card Security Specification" for detailed explanation about the SD Security Features.

(1) 请参考 SD 卡安全章节中更详细的关于“SD 卡安全说明”的描述。

(2) Command STOP\_TRAN (CMD12) shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

(2) 这个 STOP\_TRAN (CMD12) 命令可用来终止多块写操作期间的传送来决定预擦除 (ACMD23) 功能是否启用。

### 5.2.3. Responses

#### 5.2.3. 应答

There are several types of response tokens. As in the SD Card mode, all are transmitted MSB first.

这儿是应答标记的几种类型。和在 SD 卡模式下一样, 所有的发送都是 MSB 在前。

#### 5.2.3.1. Format R1

##### 5.2.3.1 R1 的格式

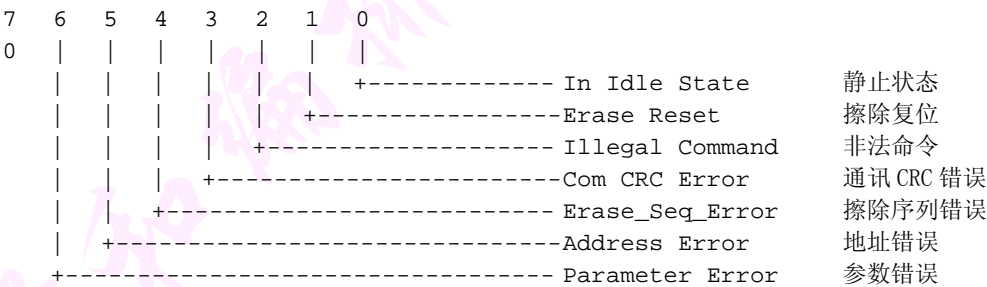
This response token is sent by the card after every command with the exception of SEND\_STATUS commands. It is 1 byte long, the MSB is always set to zero and the other bits are error indications. A '1' signals error.

这个应答标记是在卡的每个命令 (除了 SEND\_STATUS 命令外) 之后发送的。它是一个字节长度, MSB 总是设定为 “0”, 其它位包含了错误信息。当那些位为 “1” 时, 则代表一个错误信号。

- \* In idle state—The card is in idle state and running initializing process.  
处于静止状态 - 这个卡正在执行初始化处理, 处于静止状态。
- \* Erase reset—An erase sequence was cleared before executing because an out of erase sequence command was received.  
擦除复位 - 由于一条擦除序列之外的命令被确认, 导致擦除序列在执行操作之前被清除了。
- \* Illegal command—An illegal command code was detected.  
非法命令 - 发现了非法的命令代码。
- \* Communication CRC error—The CRC check of the last command failed.  
通信 CRC 错误 - 最后一条命令的 CRC 检查发现错误。
- \* Erase sequence error—An error in the sequence of erase commands occurred.  
擦除序列错误 - 在一个擦除的序列里出现错误。
- \* Address error—A misaligned address, which did not match the block length was used in the command.  
地址错误 - 不正确的地址, 命令里的地址和块长度无法相匹配。
- \* Parameter error—The command's argument (e.g., address, block length) was out of the allowed range for this card.  
参数错误 - 命令里的参数 (例如: 地址、块长度) 不在该卡所允许的范围内。

The structure of the R1 format is shown in Figure 5-7.

这个 R1 格式的结构显示在下面图片 5-7 中



图表 5-7. R1 应答格式

#### 5.2.3.2. Format R1b

##### 5.2.3.2. R1b 的格式

This response token is identical to R1 format with the optional addition of the busy signal. The busy signal token can be any number of bytes. A zero value indicates card is busy. A non-zero value indicates card is ready for the next command.

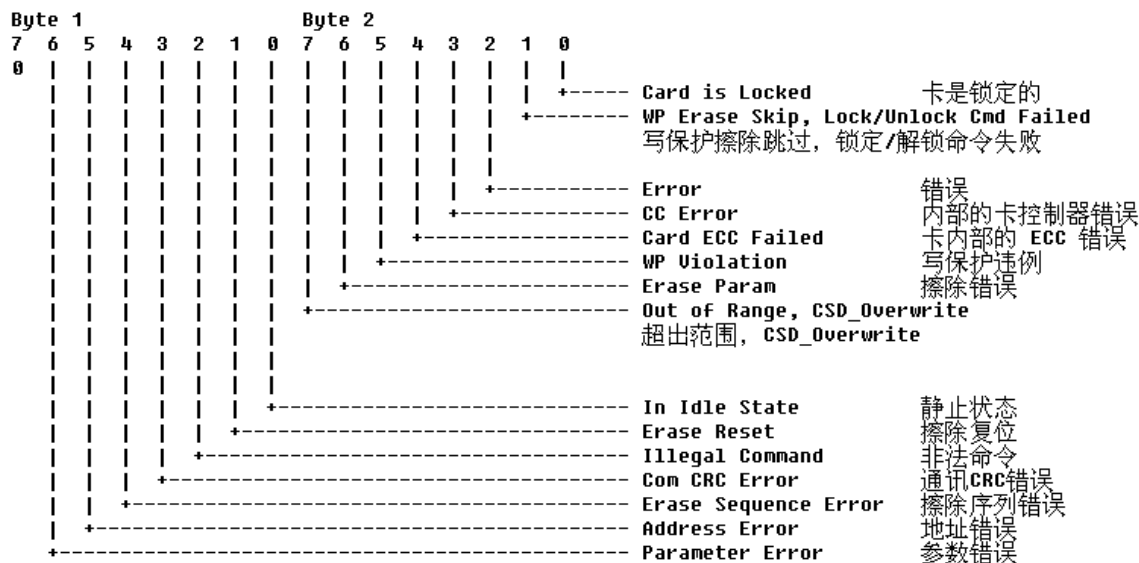
这个应答标记是和 R1 格式相同的, 只是附加了一个可选的忙信号。这个忙信号可以是任意数量的字节。一个 “0” 值指出这个卡处于忙状态。而非 “0” 值指出该卡已经准备好可以接收下一个命令了。(译者注: 这里的 “忙信号可以是任意数量的字节” 表示, 查询时必须要用 8 个时钟为单位来读取 “忙” 状态, 而不是用单片机的读取 I/O 电平的常用方法来获取 SD 卡是否已经结束了 “忙” 状态, 否则有可能会打乱 SD 卡的时序。)

### 5.2.3.3. Format R2

### 5.2.3.3. R2 的格式

This 2-bytes long response token is sent by the card as a response to the SEND\_STATUS command. The format of the R2 status is shown in Figure 5-8.

这是 2 字节长的应答标记是由 SD 卡来应答 SEND STATUS 命令的。它的格式如下表 5-8 所示:



图表 5-8. R2 应答格式

首先第一个字节是和应答 R1 相同的。后面跟随的第二个字节则如下所述：

- \* CSD Overwrite—This status bit is set if the host is trying to change the ROM section, or reverse the copy bit (set as original) or the permanent WP bit (un-protect) of the CSD register.

## CSD 覆写

- 如果主控制器试图修改 ROM（只读）区域，或反转 CSD 寄存器的 copy 位（恢复到原始状态）和永久保护位（无保护的），则此状态位被设置（译者注：这段内容在原文中是没有的，本人在三星的产品说明书中找到了，如果内容错误，请来信告知，谢谢！）。

- \* Erase param—An invalid selection, sectors for erase.

擦除 param (错误)

- 选择的擦除扇区有问题。

- \* Write protect violation—The command tried to write a write-protected block.

违背写保护

- 这个命令试图写入一个有写保护的块。

- \* Card ECC failed—Card internal ECC was applied but failed to correct the data.

### 卡的 ECC 校验失败

- 卡内部的 ECC 被使用了, 但是未能校正当前的数据。

- \* CC error—Internal card controller error.

CC 错误

- 内部的卡控制器错误。

- \* Error—A general or an unknown error occurred during the operation.

错误

- 在操作期间出现了一个不明错误或常规错误。

## 擦除跳过写保护

- 由于存在写保护的块，只有部分地址空间被擦除。

- \* Card is locked—Supported by the SanDisk SD Card.

卡是锁住的

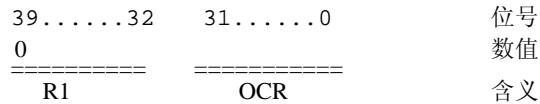
- 由 SanDisk SD 卡所支持的。

#### 5.2.3.4. Format R3

#### 5.2.3.4. R3 的格式

This response token is sent by the card when a READ\_OCR command is received. The response length is 5 bytes. The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

这个应答标记是 SD 卡用来回应 READ\_OCR 命令的。这个应答长达 5 个字节。它结构中的第一个字节和 R1 相同的。剩下的四个字节是 OCR 寄存器的内容。请看下面的图表 5-9:



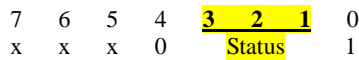
图表 5-9. R3 的应答格式

5.2.3.5. Data Response

5.2.3.5. 数据应答

Every data block written to the card is acknowledged by a data response token. It is one byte long and has the following format:

每个数据块是否成功写入卡中是通过数据应答标记来确认的。它只有一个字节长度, 格式如下:



The meaning of the status bits is defined as follows:

这些状态位的含义如下所示:

- \* '010' — Data accepted. (数据确认)
- \* '101' — Data rejected due to a CRC error. (数据由于 CRC 错误被抛弃)
- \* '110' — Data Rejected due to a Write Error (数据由于写入错误被抛弃)

In case of any error (CRC or Write Error) during Write Multiple Block operation, the host shall stop the data transmission using CMD12. In case of Write Error (response '110') the host may send CMD13 (SEND\_STATUS) in order to get the cause of the write problem. ACMD22 can be used to find the number of well written write blocks.

假如任何一个错误 (CRC 或写入错误) 出现在写入多个块操作期间的, 主控制器应该通过 CMD12 命令来停止数据的发送。假如出现写入错误 (应答 "110"), 主控制器可以通过发送 CMD13 命令 (SEND\_STATUS) 来获得导致写入出现问题的原因。而 ACMD22 可以用于查询已经写入的块的数量。

5.2.4. Data Tokens

5.2.4. 数据标记

Read and write commands have data transfers associated with them. Data is being transmitted or received via data tokens. All data bytes are transmitted MSB.

读取和写入命令的数据传送是利用它们来联系的。数据开始发送或接收都是通过数据标记进行的。所有的数据字节是 (首先) 发送 MSB。

Data tokens are 4 to 515 bytes long and have the following format:

数据标记是 4 到 515 字节长度, 它使用了下面的格式:

For Single Block Read, Single Block Write and Multiple Block Read:

对于单个块的读取、单个块的写入和多个块的读取:

- \* First byte: Start Block
  - \* 第一个字节: 开始一个块。
- \* Bytes 2-513 (depends on the data block length): User data
  - \* 字节 2-513 (由数据块长度决定): 用户数据。
- \* Last two bytes: 16-bit CRC
  - \* 最后两个字节: 16 位的 CRC。

For Multiple Block Write operation:

对于多个块的写入操作:

- \* First byte of each block
  - \* 每个块的首个字节。
- If data is to be transferred then—Start Block
  - 如果数据在传送时 - 起始块

If Stop transmission is requested—Stop Tran

如果被请求停止传送 - 停止传送

7 0  
11111101

Note that this format is used only for Multiple Block Write. In case of Multiple Block Read the stop transmission is done using STOP\_TRAN Command (CMD12).

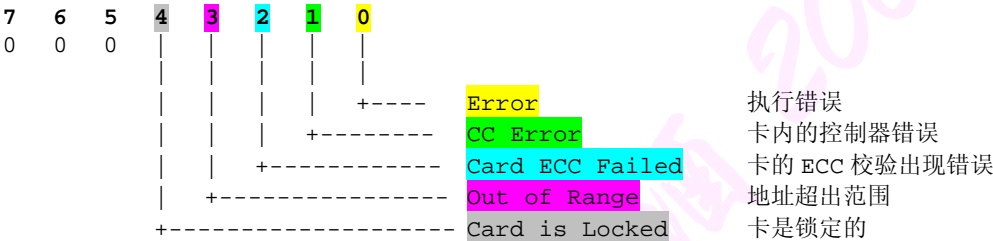
注意: 这个格式仅仅用于多个块的写入操作。要停止多个块的读取传送操作应该使用 CMD12 (STOP\_TRAN 命令)。

### 5.2.5. Data Error Token

#### 5.2.5. 数据错误标记

If a read operation fails and the card cannot provide the required data it will send a data error token, instead. This token is one byte long and has the format shown in Figure 5-10.

如果读取操作失败和该卡不能提供必需的数据, 它将发送一个数据错误标记来替代数据。这个标记是一个字节长度, 它的格式如下图表 5-10 所示:



图表 5-10. 数据错误标记

The four least significant bits (LSB) are the same error bits as in response format R2.

这四个最低有效位(LSB)和应答格式 R2 中的相同错误位含义是一样的。

### 5.2.6. Clearing Status Bits

#### 5.2.6. 清除状态位

As described in the previous paragraphs, in SPI mode, status bits are reported to the host in three different formats: response R1, response R2 and data error token (the same bits may exist in multiple response types—e.g., Card ECC failed). As in the SD mode, error bits are cleared when read by the host, regardless of the response format.

在前面的章节描述中, SPI 模式里, 那些状态位是用三种不同的格式来报告给主控制器: 应答 R1、应答 R2 和数据错误标记 (在多种应答类型中是处于相同的位置, 例如 卡的 ECC 失败)。而在 SD 协议模式下, 错误位是通过主控制器的读操作来清除的, 而不关应答格式的事。(译者注: 这里的意思应该是 SPI 协议模式下, 各命令的操作状态和结果报告是由随后的应答来提供并自动消除来处理的, 但实际上, SPI 协议模式下的清除动作是有条件的, 这在第三章的译文中有很详细的描述。SD 协议模式在这里和 SPI 协议模式有着很大的不同, 对 SD 协议模式来说, 需要主控制器主动去消除前条命令产生的这些状态位。)

### 5.3. Card Registers

#### 5.3. 卡寄存器

In SPI Mode, only the OCR, CSD and CID registers are accessible. Their format is identical to their format in the SD Card mode. However, a few fields are irrelevant in SPI mode.

在 SPI 模式, 只有 OCR、CSD、CID 这三个寄存器可以访问。它们的格式和 SD 协议模式下的格式是相同的。然而, 其中有一小部分区域的内容和 SPI 协议模式是无关的。

### 5.4. SPI Bus Timing Diagrams

#### 5.4. SPI 总线时序图解



All timing diagrams use the schematics and abbreviations listed in Table 5-5.  
用图表和缩写表来图解所有的时序, 见图表 5-5。

表 5-4. SPI 总线时段缩写

|            |                               |            |                 |
|------------|-------------------------------|------------|-----------------|
| H          | Signal is high (logical '1')  | H          | 信号为高 (逻辑 "1")   |
| L          | Signal is low (logical '0')   | L          | 信号为低 (逻辑 "0")   |
| X          | Don't care                    | X          | 不必关心            |
| Z          | High impedance state (-> = 1) | Z          | 高阻抗状态 (也等同 "1") |
| *          | Repeater                      | *          | 转发              |
| Busy       | Busy Token                    | Busy       | 忙标记             |
| Command    | Command token                 | Command    | 命令标记            |
| Response   | Response token                | Response   | 应答标记            |
| Data block | Data token                    | Data block | 数据标记            |

All timing values are defined in Table 5-5. The host must keep the clock running for at least Ncr clock cycles after the card response is received. This restriction applied to command and data response tokens.

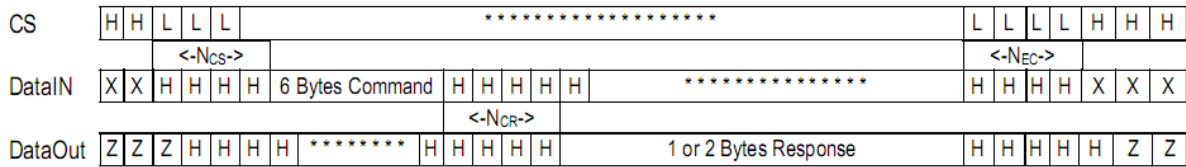
所有的时序值别定义在图表 5-5 种。主控制器必须在卡的应答被确认之后, 保持时钟最少运行 Ncr 个时钟周期。这个限制将被用于命令和数据的应答标记。

5.4.1. Command/Response

5.4.1. 命令 / 应答

Host Command to Card Response-Card is Ready

从主控制器命令到卡的应答 - 卡是准备好的



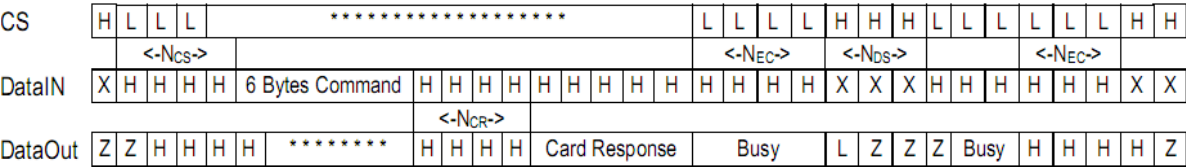
图表 5-11. 从主控制器命令到卡的应答 - 卡是准备好的

Host Command to Card Response-Card is Busy

从主控制器命令到卡的应答 - 卡是处于忙状态的

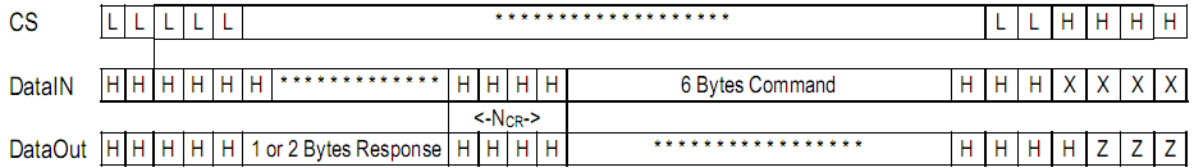
The following timing diagram describes the command response transaction for commands when the card responses which the R1b response type (e.g., SET\_WRITE\_PROT and ERASE). When the card is signaling busy, the host may deselect it (by raising the CS) at any time. The card will release the DataOut line one clock after the CS going high. To check if the card is still busy it needs to be reselected by asserting (set to low) the CS signal. The card will resume busy signal (pulling DataOut low) one clock after the falling edge of CS.

下列时序图描述了那些命令在卡回应 R1b 类型时的命令应答处理过程 (例如: SET\_WRITE\_PROT 和 ERASE 等命令)。当这个卡正处于忙状态, 主控制器可以重新在任何时间检测它 (通过提高这个 CS 信号)。一旦 CS 信号变高, 该卡将在一个时钟信号 (最小 8 个 CLOCK) 之后释放 DATAOUT 信号线。如需要检查该卡的是否还在忙状态, 主控制器必须重新将 CS 信号拉低来设置有效。该卡将在 CS 信号下降边沿开始在一个时钟后将重新恢复忙信号 (拉 DATAOUT 信号线到低)。(译者注: 前面也提到过, 即使将卡的 CS 信号拉高, 你仍然需要提供最少 8 个时钟来给卡来处理它的任务。所以, 对操纵 SD 卡来说, 保险的做法是在当前操作结束时, 将 CS 拉高之后, 继续在时钟线上送出 8 个时钟, 这样有利于 SD 卡的同步。)



图表 5-12. 从主控制器命令到卡的应答 - 卡是处于忙状态的

Card Response to Host Command  
从卡的应答到主控制器的命令

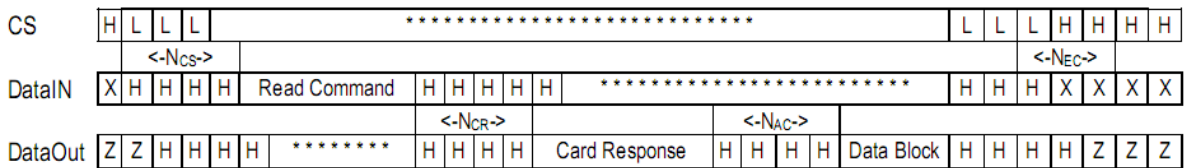


图表 5-13. 从卡的应答到主控制器的命令

5.4.2. Data Read  
5.4.2. 数据读取

The following timing diagram describes all single block read operations with the exception of SEND\_CSD command.

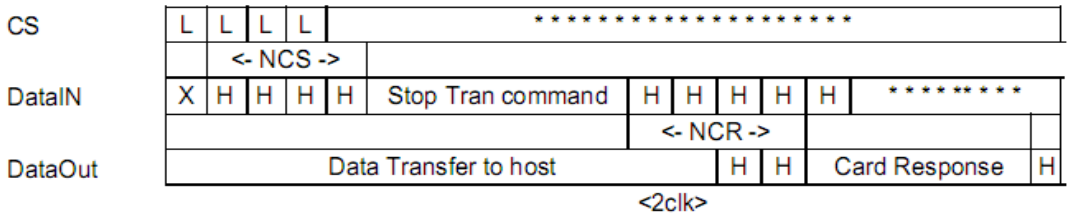
下面的时序图描述了除了 SEND\_CSD 命令之外的所有单个块读取操作。



图表 5-14. 单个数据块读取时序

The following table describes Stop transmission operation in case of Multiple Block Read.

下面的表描述了在多块读取操作的情况下停止发送操作。

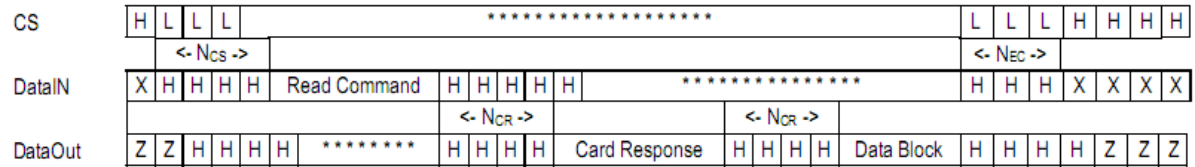


图表 5-15. 多数据块的读取时序

Reading the CSD Register  
读取 CSD 寄存器

The following timing diagram describes the SEND\_CSD command bus transaction. The timeout values for the response and the data block are Ncr (Since the Nac is still unknown).

下列的时序图描述了 SEND\_CSD 命令的总线处理。此应答和数据块的的超时值是 Ncr (因为 Nac 是不明确的)。

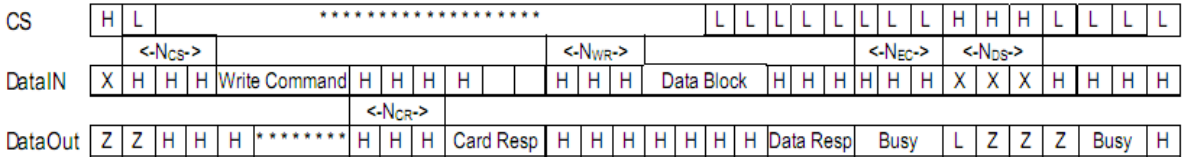


图表 5-16. 读取 CSD 寄存器

5.4.3. Data Write  
5.4.3. 数据写入

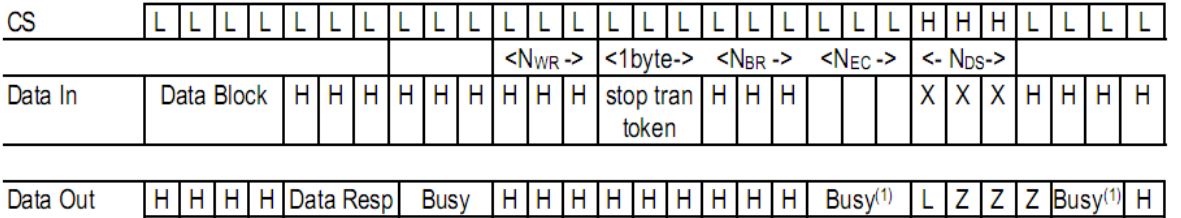
The host may deselect a card (by raising the CS) at any time during the card busy period (refer to the given timing diagram). The card will release the DataOut line one clock after the CS going high. To check if the card is still busy it needs to be re-selected by asserting (set to low) the CS signal. The card will resume busy signal (pulling DataOut low) one clock after the falling edge of CS.

主控制器可以在 SD 卡忙碌期间 (将 CS 信号拉高) 的任何时候来检查卡的忙状态 (请参考特定的时序图)。而卡将在 CS 信号变高一个时钟周期之后释放 DATAOUT 信号线。如果要检查可是否继续处于忙状态, 必须将 CS 信号激活 (将 CS 拉低)。该卡将在 CS 信号下降边沿开始在一个时钟后将重新恢复忙信号 (拉 DATAOUT 信号线到低)。(译者注: 这里的另一个意思是: 在 SD 卡忙碌的时候, 可以将它的 CS 信号拉高来释放 DATAOUT 信号线。这样可避免忙碌中的 SD 卡霸占 SPI 总线, 其它外设就可以利用这段时间在 SPI 总线上发送信息。)



图表 5-17. 设备写时序

The following figure describes stop transmission operation in Multiple Block Write transfer. 下面的表描述了多个块在整组写入传送操作的情况下停止发送操作。



图表 5-18. 停止传送的时序 - 多块写入

(1)The Busy may appear within Nbr clocks after Stop Tran Token. If there is no Busy, the host may continue to the next command.

注 1: 这个忙状态将出现在停止传送标记 Nbr 的时钟之后。如果没有出现忙状态, 则主控制器可继续发送下条命令。

#### 5.4.4. Timing Values

##### 5.4.4. 时钟数值

Table 5-5 shows the timing values and definitions. For more information, refer to Table 4-17 in Section 4.0, Section 5.1.9.2, and the applications note in Appendix A, "Host Design Considerations: NAND MMC and SD-based Products."

表 5-5 显示了时钟数值和它的定义。更多信息请参考表 4-17, 它在章节 4.0、5.1.9.2 和附录 A 中, 标题是 "Host Design Considerations: NAND MMC and SD-based Products."

Table 5-5. 时钟常量定义

|     | Min | Max | Unit           |
|-----|-----|-----|----------------|
| Ncs | 0   | -   | 8 Clock Cycles |
| Ncr | 0   | 8   | 8 Clock Cycles |
| Nrc | 1   | -   | 8 Clock Cycles |
| Nac | 1   | 注   | 8 Clock Cycles |
| Nwr | 1   | -   | 8 Clock Cycles |
| Nec | 0   | -   | 8 Clock Cycles |
| Nds | 0   | -   | 8 Clock Cycles |
| Nbr | 0   | 1   | 8 Clock Cycles |

NOTE: min  $\{[(TAAC * f) + (NSAC * 100)] * 1/8\}$ ,  $\{(100ms * f) * 1/8\}$  where units = (8 clocks) and "f" is the clock frequency.

注: 最小  $\{[(TAAC * f) + (NSAC * 100)] * 1/8\}$ ,  $\{(100ms * f) * 1/8\}$  这些单位 = (8 个时钟) 和 "f" 是时钟频率。

## 5.5. SPI Electrical Interface

### 5.5. SPI 电气接口

The SPI Mode electrical interface is identical to that of the SD Card mode.

SPI 模式下的电气接口是和 SD 模式相同的。

## 5.6. SPI Bus Operating Conditions

### 5.6. SPI 总线操作条件 (环境)

Identical to SD Card mode.

和 SD 模式相同

## 5.7. Bus Timing

### 5.7. 总线时钟

Identical to SD Card mode. The timing of the CS signal is the same as any other card input.

和 SD 模式相同。这个 CS 信号的时间和其它任何卡的输入是一样的。

**附注:**

这里附加的两张表是原文中的两张指令表, 在前面的译文中已经翻译成中文, 此处将原文添加在此处, 供所需者查阅。

**Table 5-3. Application Specific Commands Used or Reserved by the SD Card-SPI Mode**

| CMD INDEX         | SPI Mode | Argument                                     | Resp | Abbreviation           | Command Description  |
|-------------------|----------|--|------|------------------------|--|
| ACMD6             | No       |  |      |                        |  |
| ACMD13            | Yes      | [31:0] stuff bits                            | R2   | SD_STATUS              | Send the SD Card status. The status fields are given in Table 4-21   |
| ACMD17            | Reserved |  |      |                        |  |
| ACMD18            | Yes      | --   | --   | --                     | Reserved for SD security applications1   |
| ACMD19 to ACMD21  | Reserved |  |      |                        |  |
| ACMD22            | Yes      | [31:0] stuff bits                            | R1   | SEND_NUM_WR_BLOCKS     | Send the numbers of the well-written (without errors) blocks. Responds with 32bit+CRC data block.  |
| ACMD23            | Yes      | [31:23] stuff bits<br>[22:0]Number of blocks | R1   | SET_WR_BLK_ERASE_COUNT | Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"=default (one wr block)(2). |
| ACMD24            | Reserved |  |      |                        |  |
| ACMD25            | Yes      | --   | --   | --                     | Reserved for SD security applications1   |
| ACMD26            | Yes      | --   | --   | --                     | Reserved for SD security applications1   |
| ACMD38            | Yes      | --   | --   | --                     | Reserved for SD security applications1   |
| ACMD39 to ACMD40  | Reserved |  |      |                        |  |
| ACMD41            | Yes      | None   | R1   | SEND_OP_COND           | Activates the card's initialization process.   |
| ACMD42            | Yes      | [31:1] stuff bits<br>[0]set_cd               | R1   | SET_CLR_CARD_DETECT    | Connect[1]/Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin 1) of the card. The pull-up may be used for card detection.                 |
| ACMD43 ... ACMD49 | Yes      | --   | --   | --                     | Reserved for SD security applications.1  |
| ACMD51            | Yes      | [31:0] stuff bits                            | R1   | SEND_SCR               | Reads the SD Configuration Register (SCR).   |



**Table 5-2. Description of SPI Bus Commands**

| CMD INDEX             | SPI Mode | Argument                          | Resp            | Abbreviation            | Command Description   |
|-----------------------|----------|-----------------------------------|-----------------|-------------------------|---|
| CMD0                  | Yes      | None                              | R1              | GO_IDLE_STATE           | Resets the SD Card  |
| CMD1                  | Yes      | None                              | R1              | SEND_OP_COND            | Activates the card's initialization process.  |
| CMD2                  | No       |                                   |                 |                         |   |
| CMD3                  | No       |                                   |                 |                         |   |
| CMD4                  | No       |                                   |                 |                         |   |
| CMD5                  | Reserved |                                   |                 |                         |   |
| CMD6                  | Reserved |                                   |                 |                         |   |
| CMD7                  | No       |                                   |                 |                         |   |
| CMD8                  | Reserved |                                   |                 |                         |   |
| CMD9                  | Yes      | None                              | R1              | SEND_CSD                | Asks the selected card to send its card-specific data (CSD).  |
| CMD10                 | Yes      | None                              | R1              | SEND_CID                | Asks the selected card to send its card identification (CID).   |
| CMD11                 | No       |                                   |                 |                         |   |
| CMD12                 | Yes      | None                              | R1b             | STOP_TRANSMISSION       | Forces the card to stop transmission during a multiple block read operation.  |
| CMD13                 | Yes      | None                              | R2              | SEND_STATUS             | Asks the selected card to send its status register.   |
| CMD14                 | No       |                                   |                 |                         |   |
| CMD15                 | No       |                                   |                 |                         |   |
| CMD16                 | Yes      | [31:0] block length               | R1              | SET_BLOCKLEN            | Selects a block length (in bytes) for all following block commands (read & write). <sup>1</sup>   |
| CMD17                 | Yes      | [31:0] data address               | R1              | READ_SINGLE_BLOCK       | Reads a block of the size selected by the SET_BLOCKLEN command. <sup>2</sup>  |
| CMD18                 | Yes      | [31:0] data address               | R1              | READ_MULTIPLE_BLOCK     | Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.  |
| CMD19                 | Reserved |                                   |                 |                         |   |
| CMD20                 | No       |                                   |                 |                         |   |
| CMD21<br>...<br>CMD23 | Reserved |                                   |                 |                         |   |
| CMD24                 | Yes      | [31:0] data address               | R1 <sup>3</sup> | WRITE_BLOCK             | Writes a block of the size selected by the SET_BLOCKLEN command. <sup>4</sup>   |
| CMD25                 | Yes      | [31:0] data address               | R1              | WRITE_MULTIPLE_BLOCK    | Continuously writes blocks of data until a stop transmission token is sent (instead of 'start block').  |
| CMD26                 | No       |                                   |                 |                         |   |
| CMD27                 | Yes      | None                              | R1              | PROGRAM_CSD             | Programming of the programmable bits of the CSD.  |
| CMD28 <sup>1</sup>    | Yes      | [31:0] data address               | R1b             | SET_WRITE_PROT          | If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). |
| CMD29 <sup>4</sup>    | Yes      | [31:0] data address               | R1b             | CLR_WRITE_PROT          | If the card has write protection features, this command clears the write protection bit of the addressed group.   |
| CMD30                 | Yes      | [31:0] write protect data address | R1              | SEND_WRITE_PROT         | If the card has write protection features, this command asks the card to send the status of the write protection bits. <sup>2</sup>   |
| CMD31                 | Reserved |                                   |                 |                         |   |
| CMD32                 | Yes      | [31:0] data address               | R1              | ERASE_WR_BLK_START_ADDR | Sets the address of the first write block to be erased.   |

|                        |          |  |     |                       |  |
|------------------------|----------|--|-----|-----------------------|--|
| CMD33                  | Yes      | [31:0] data address                    | R1  | ERASE_WR_BLK_END_ADDR | Sets the address of the last write block in a continuous range to be erased.   |
| CMD34<br>....<br>CMD37 | Reserved |  |     |                       |  |
| CMD38                  | Yes      | [31:0] don't care*                     | R1b | ERASE                 | Erases all previously selected write blocks.   |
| CMD39                  | No       |  |     |                       |  |
| CMD40                  | No       |  |     |                       |  |
| CMD41<br>...<br>CMD54  | Reserved |  |     |                       |  |
| CMD55                  | Yes      | [31:0] stuff bits                      | R1  | APP_CMD               | Notifies the card that the next command is an application specific command rather than a standard command.   |
| CMD56                  | Yes      | [31:0] stuff bits<br>[0]: RD/WR.3      | R1  | GEN_CMD               | Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. The size of the Data Block is defined with SET_BLOCK_LEN command. |
| CMD57                  | Reserved |  |     |                       |  |
| CMD58                  | Yes      | None                                   | R3  | READ_OCR              | Reads the OCR register of a card.  |
| CMD59                  | Yes      | [31:1] don't care*<br>[0:0] CRC option | R1  | CRC_ON_OFF            | Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off.   |
| CMD60<br>...<br>CMD63  | No       |  |     |                       |  |

译者注: 这里是和本章节有点关联的内容, 虽然属于其它章节, 但放在这里方便查阅。

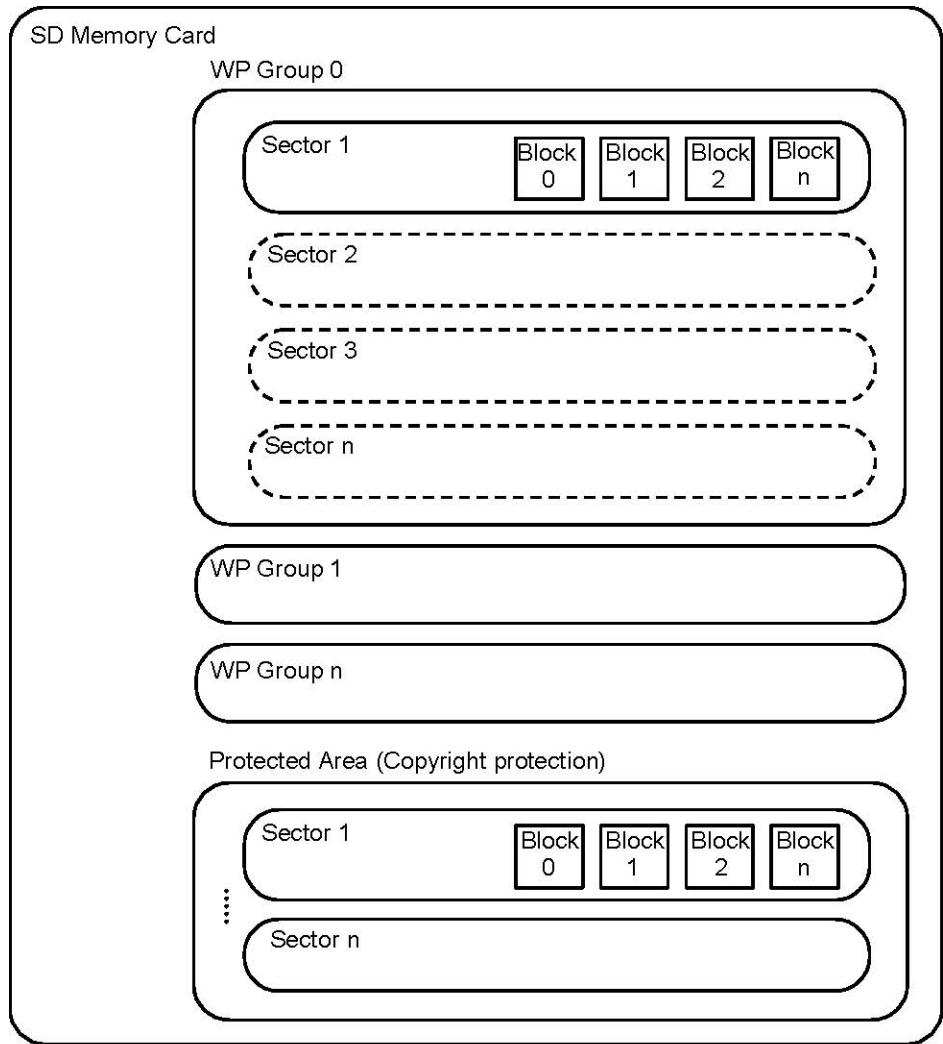


Figure 1-2. Memory Array Partitioning

Table 1-3. Memory Array Structures Summary

| Part Number | Block Size<br>(Byte) | Data Area + Protected<br>size (Blocks) | Protected Area<br>size (Blocks) | User Area<br>(Blocks) |
|-------------|----------------------|--|---------------------------------|-----------------------|
| SDSDJ-1024  | 512                  | 2,004,224                              | 20,480                          | 1,983,744             |
| SDSDJ-512   | 512                  | 1,001,104                              | 10,240                          | 940,864               |
| SDSDJ-256   | 512                  | 499,456                                | 5,376                           | 494,080               |
| SDSDJ-128   | 512                  | 248,640                                | 2,624                           | 246,016               |
| SDSDJ-64    | 512                  | 123,232                                | 1,376                           | 121,856               |
| SDSDB-32    | 512                  | 60,512                                 | 736                             | 59,776                |
| SDSDB-16    | 512                  | 29,152                                 | 352                             | 28,800                |

## Host Design Considerations: NAND MMC and SD-based Products

### Introduction

SanDisk's MultiMediaCard (MMC) and Secure Digital (SD) Card have been designed into a wide variety of consumer electronic products: MP3 players, cell phones, PDAs, digital still and video cameras, data loggers, and more. Although these cards were designed to support this wide range of products, there are many options an engineer needs to consider before designing a card slot into a product. Design considerations include how the end product handles timeout delays, bus type selection, block mode selection, and other options. These can have a major impact on the performance and compatibility of the product. This Application Note will review these options and provide recommendations on the optimum way to manage them.

## 主控制器设计参考: NAND MMC 和 SD 为基础的产品

### 介绍

SanDisk 的 MMC 和 SD 卡可以用在很宽的产品线中: MP3 播放器、手机、PDA、数字和视频类相机等等。由于这些卡可以支持很宽的产品线, 一些工程师会在选择在他们设计的产品上留出 SD 的卡槽。

设计中需要考虑的事情有: 如何在产品的时限内完成设计、总线类型的选择、块模式的选择和一些其它选项。主要的冲突在于性能和兼容性上。这些附加信息就是提供一些建议来处理这些可选的项目。

译者注: 这条信息说明, 如果在设计中碰到一些不明之处, 附录提供的信息可能对设计者有帮助。实际上, 附录中的信息在各章节都有详细的描述了, 这里只是作为重点把它们集中在一起了。随后的附录信息就不翻译了, 请您查阅原文。