SD 卡协议学习点滴（一）

首先 SD 卡有所谓操作模式（operation mode）的概念，每种操作模式又具体对应一种或多种状态，主机通过发送命令可以使 SD 卡在不同的状态间转换，SD 卡则接受命令，并根据自己现在所处状态做出不同的响应。

Card state	Operation mode
Inactive State	inactive
Idle State	card identification mode
Ready State	
Identification State	
Stand-by State	data transfer mode
Transfer State	
Sending-data State	
Receive-data State	
Programming State	
Disconnect State	

系统上电时刻或者搜寻 SD 卡时，SD 卡控制器应该处于 SD 卡识别模式；SD 卡在刚接入系统时刻也处于这种模式，并且处于此模式下的 Idle 状态。

SD卡识别模式：在这种模式下，控制器会检验SD卡的工作电压范围，识别SD卡类型，并要求它们发送各自的相对地址（Relative Card Address）；这些操作在SD卡各自的CMD线上进行。所有的操作均使用默认的 SD卡识别时钟频率（identification clock rate）

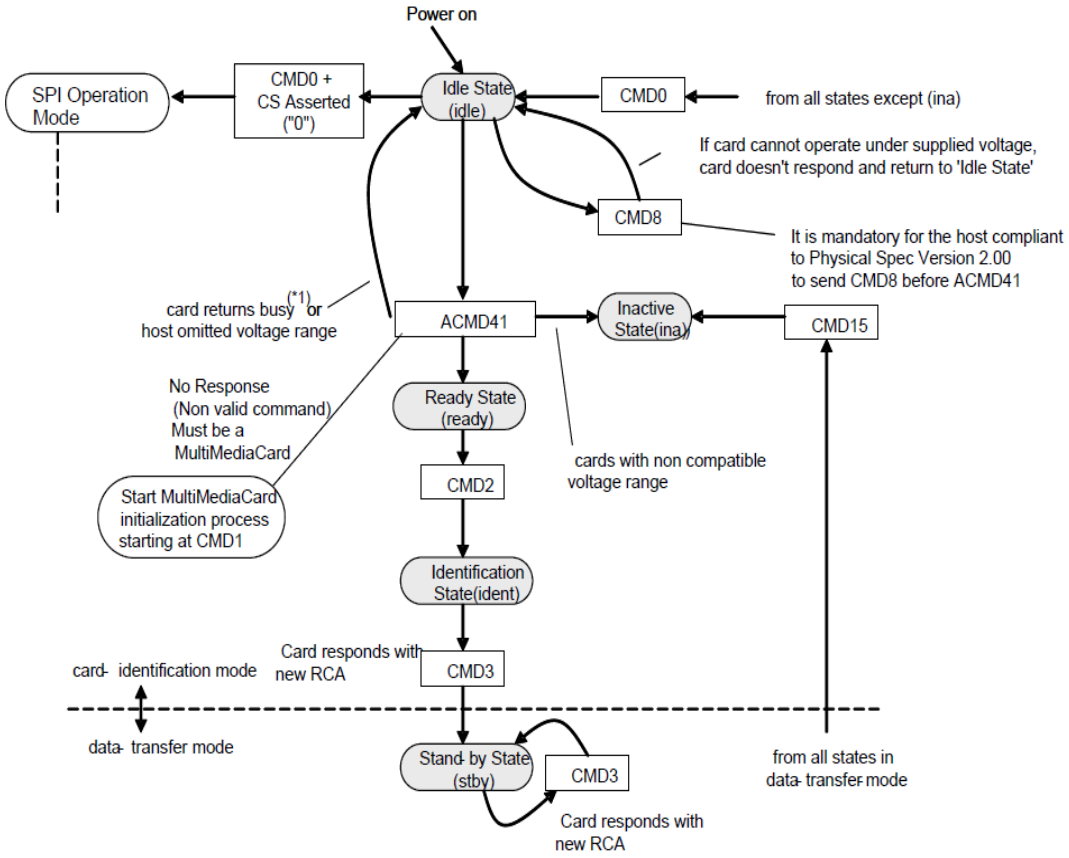
SD卡复位：发送GO_IDLE_STATE(CMD0)到SD卡后，除处于非活动状态（Inactive state）之外的SD卡都会进入空闲状态（Idle state）；在Idle状态，SD卡的CMD线处于输入模式，默认相对地址为 0x0000，默认驱动寄存器设定为最低速度，最大驱动电流能力。

工作条件检测：

在控制器和 SD 卡进行任何通信之前，控制器不清楚 SD 卡支持的工作电压范围，故而控制器首先使用默认的电压发送一条 reset 指令（CMD0），紧跟着的 CMD8 指令，用于取得 SD 卡支持工作电压范围数据。SD 卡通过检测 CMD8 的参数部分来检查控制器使用的工作电压，控制器通过分析回传的 CMD8 参数部分来校验 SD 卡是否可以在所给电压下工作。如果 SD 卡可以在指定电压下工作，则它回送 CMD8 的命令响应字，其中包含 check voltage, check pattern。如果 SD 卡不支持所给电压，则 SD 卡不会给出任何响应信息，并继续处于 Idle 状态。在 PLV2.0(physical layer version2.0)下，在首次执行 ACMD41 之前，必须执行 CMD8 指令，用以初始化 SDHC 卡，SDHC 卡根据是否接收到 CMD8 指令来鉴别控制器是否支持 PLV2.0 协议。使用低电压的控制器也必须在 ACMD41 命令之前发送 CMD8，避免可以工作在两种电压模式下的 SD 卡因为没有接收到 CMD8， 而默认工作在高电压环境下，被误认为是只支持高电压工作模式。

SD_SEND_OP_COND(ACMD41)命令的目的是给予 SD 卡控制器一个识别 SD 卡是否可以在所给 Vdd 范围下工作的机制，如果 SD 卡无法在指定 Vdd 范围内工作，则它会进入非活动状态（Inactive state）。要注意的是，ACMD41 是应用相关型命令，因而，每次发出的 ACMD41 命令都必须紧跟在一条 APP_CMD（CMD55）命令之后。在空闲态（Idle State）下使用的 CMD55 命令使用默认的卡相对地址（RCA）0x0000。

每次控制器发送 CMD0 复位 SD 卡后，都要重新进行系列初始化操作（CMD8,ACMD41...）。如果 ACMD41 指令的 OCR 比特位为 0，控制器可以查询各个 SD 卡，并决定它们共同的工作电压范围。在作为查询的 ACMD41 指令发送之后，SD 卡并不会开始初始化过程，直到控制器重新发送一条 ACMD41 指令。



SD卡初始化和识别过程：

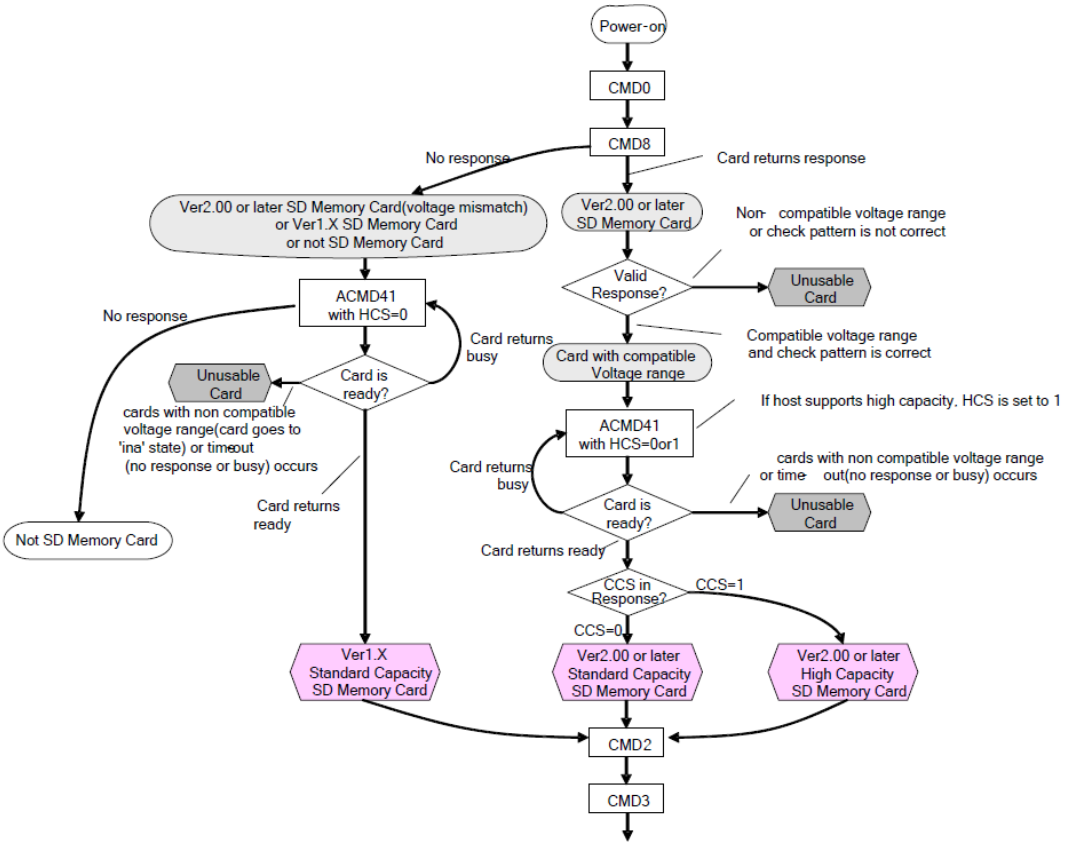
SD 卡的初始化开始于接收到 ACMD41 指令之后，ACMD 指令的 HCS（Host Capacity Support）位如果设定为 1 的话，表明控制器支持 SDHC 卡，否则表示不支持。

在 CMD8 命令发送之后的 ACMD41 指令其功能有所扩展，在参数里多了 HCS 部分，在响应里面多了 CCS（Card Capacity Status）部分。HCS 参数会被不响应 CMD8 命令的 SD 卡所抛弃。控制器向不响应 CMD8 的卡发送 ACMD41 指令时，HCS 位应该设置为零 0。如果向 SDHC 卡发送 HCS 位为 0 的 ACMD41 命令，SDHC 卡返回的响应，其 busy 标识位永远为 0，代表忙状态。HCS 标识位用来表明 SD 卡是否已经完成初始化，如果未完成，HCS 为零，否则为 1，如果 HCS 为 0，控制器会重复发送 ACMD41 指令，SD 卡只检查首次接收到的 ACMD41 指令的 HCS 位。

响应 CMD8 的 SD 卡发送的对于 ACMD41 指令响应会包含 CCS 部分，控制器只检查 HCS 标志位为 1 的响应所包含的 CCS 位。CCS=1 表明其为 SDHC 卡，否则为标准 SD 卡。

控制器随后发送 ALL_SEND_CID(CMD2)命令，查询各个卡的 CID（unique card identification）值，还没有被识别的 SD 卡（处于 Ready 状态）会发送 CID 值作为响应，发送完 CID 值之后，SD 卡进入识别状态（Identification state），然后控制器发送 CMD3（SEND_RELATIVE_ADDR）命令，要求各个 SD 卡发送一个新的相对地址（RCA），RCA 在之后

的数据传输模式中用于寻址。RCA 发送完之后。SD 卡进入 Stand-by 状态，在这个状态，如果控制器想要给 SD 卡分配一个新的 RCA，它可以发送另一条 CMD3 命令给 SD 卡。最后发布的 RCA 为 SD 卡的真实 RCA。



数据传输模式：

在 SD 卡识别模式结束之前，控制器使用的时钟频率均为 Fod。在数据传输模式，控制器可能会使用 Fpp 频率。控制器发送一条 SSEND_CSD（CMD9）命令来获取 SD 卡 CSD 寄存器（Card Specific Data）里面的描述值，譬如，块长度，卡容量信息等。广播命令 SET_DSR（CMD4）为各个已识别的 SD 卡配置驱动阶段（？？）。它会向 SD 卡的 DSR 寄存器写入相关的信息。控制器的时钟频率也在这个时刻从 Fod 转到 Fpp。SET_DSR 命令是可选的。

CMD7 命令用来选择某个 SD 卡，使其进入 Transfer 状态，在指定时间段内，只有一个卡能处于 Transfer 状态。当某个先前被选中的处于 Transfer 状态的 SD 卡接收到 CMD7 之后，会释放与控制器的连接，并进入 Stand-by 状态。当 CMD7 使用保留地址 0x0000 时，所有的 SD 卡都会进入 Stand-by 状态。

数据传输模式下各个状态的转换关系总结如下：

所有的数据读命令都可以被停止命令（CMD12）在任意时刻终止。数据传输会终止，SD 卡返回 Transfer 状态。读命令有：块读操作（CMD17）、多块读操作（CMD18）、发送写保护（CMD30）、发送 scr（ACMD51）以及读模式下的普通命令（CMD56）

所有的数据写命令都可以被停止命令（CMD12）在任意时刻终止。写命令也会在取消选择命令（CMD7）之前停止。写命令有：块写操作（CMD24，CMD25）、编程命令（CMD27）、锁定/解锁命令（CMD42）以及写模式下的普通命令（CMD56）

数据传输一旦完成，SD 卡会退出数据写状态，进入 Programming 状态（传输成功）或者 Transfer 状态（传输失败）

如果块写操作被叫停，但是写操作包含的最终块其长度和 CRC 校验是正确的话，数据会被编程到 SD 卡（从缓存写入到 Flash？）

SD 卡可能会提供缓存模式，意思是前次写入块在编程到 Flash 的时刻，控制器可以接着发送下一块的数据

当写缓存为满时刻，并且 SD 卡处于 Programming 状态，DAT0 会保持为低电平（BUSY），表明其为忙状态

写 CSD，写保护，擦除这些操作没有缓存的功能，当 SD 卡正在处理这些命令的时候，其余的数据传输命令会被忽略。当 SD 卡为忙，并且处于 Programming 状态的时候，DAT0 也会被 SD 卡拉低，

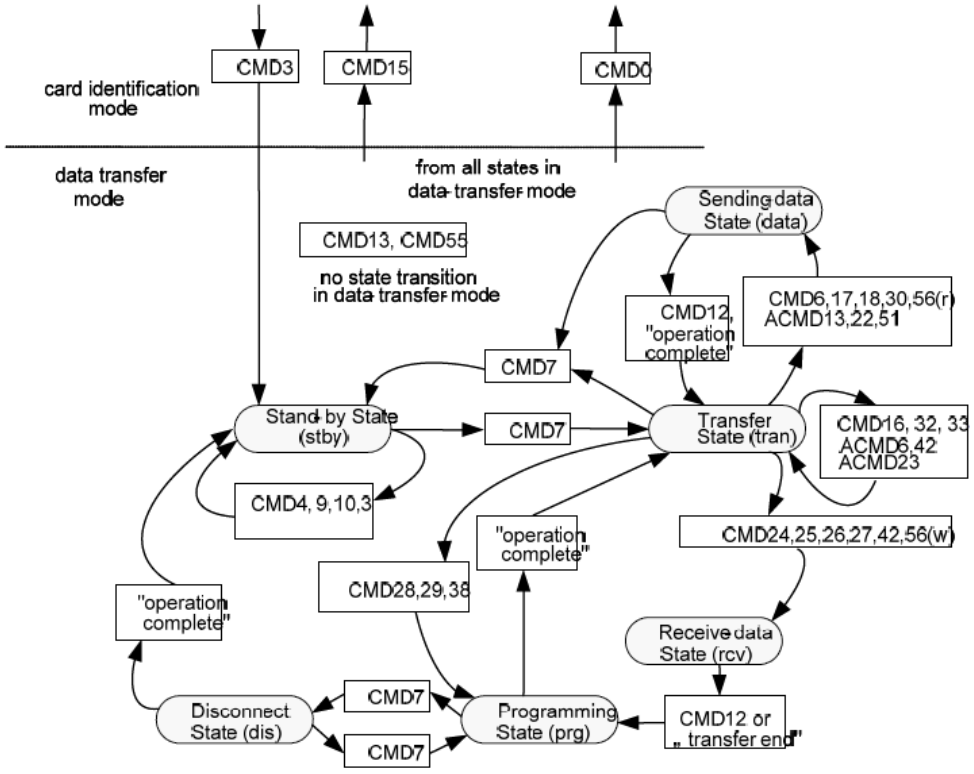
在 SD 卡处于 Programming 状态时候，不允许控制器发送设置参数命令。设置参数命令有：设置块长度（CMD16）、擦除块开始（CMD32）以及擦除块结束（CMD33）在 SD 卡编程时刻，读命令也是不允许的

当把另一个卡从 Stand-by 状态转换为 Transfer 状态的时候，正处于 erase 和 Programming 状态的卡其操作不会终止，它会自动进入 Disconnect 状态，释放数据线。

处于 Disconnect 状态的卡可以通过发送 CMD7 命令使其脱离此状态，并进入 Programming 状态，并重新激活忙标识符

复位 SD 卡（使用 CMD0 或者 CMD15）会终止任何等待中或正在进行的 Programming 操作。这可能会损毁 SD 卡的数据

CMD34-37 CMD50，CMD57 保留



宽总线选择/取消选择：

宽总线（4bit）模式可以通过 ACMD6 选择/取消选择，默认工作模式为 1bit 模式。为了改变总线宽度，以下两个条件须满足：**1. SD 卡处于 Transfer 状态 2.SD 卡未被锁定**

2Gbyte SD卡：

为使用 2Gbyte 的 SD 卡，Maximum Block Length（READ_BL_LEN=WRITE_BL_LEN）参数须设置为 1024byte。但是，由 CMD6 设置的 Block Length 须为 512byte，从而可与最大仅可使用 512byte Block Length 的 SD 卡相兼容

读数据：

DAT 线上没有数据传输时，由外部上拉电阻保持为高电平。传输数据块由一个起始比特（1 或者 4 个比特的低电平），以及紧跟着的连续数据流组成。数据流包含有效载荷数据（以及错误校验比特，如果有使用到 ECC）。数据流以数据结束标识符结束（1 或者 4 比特的高电平）。数据传输和时钟信号是同步的。有效载荷数据后紧跟着 1 比特或者 4 比特的 CRC 校验码

读命令在出现 BLOCK_LEN_ERROR 或者 ADDRESS_ERROR 的情况下可以被否决，这种情况下，不会发生数据传输过程；

块读操作

块传输下，基本的数据传输单位是块，它最大为 512byte，小一些的块传输也是可以的，如果它指向的内容其地址范围处于以 512 字节为边界的某块内（譬如 地址范围为 200~300 的传输是可以的，但是地址范围为 500~600 因为地址横跨了两个块，则不可行）

由 CMD16 设置的 Block Length 参数最大可为 512byte，而与 READ_BL_LEN 的大小无关块数据的结尾都附加有 CRC 校验码，用以保证传输数据的完整性。CMD17（READ_SINGLE_BLOCK）发起单次的块传输，传输结束后，SD 卡返回 Transfer 状态，CMD18（READ_MULTIPLE_BLOCK）开始多个连续块的数据传输。块传输会一直持续下去，除非收到一条 STOP_TRANSMISSION 命令（CMD12）。停止传输命令会有执行上的延迟，因为它是串行传输的，数据传输会在 stop 命令的停止位之后结束

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first misaligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command. If the misaligned block is the first data block of the command (i.e. ADDRESS_ERROR was reported in the actual response to the command), then no data is transferred and the card remains in the TRAN state.

CSD value			Current Blocklen *1	Read CMD Start Address
Max block size READ_BL_LEN	Misalign	Partial		
512Bytes	0 (Disable)	1 (Enable)	1- 512 bytes	Any address is accepted. *2
1kBytes	0 (Disable)	1 (Enable)	1- 512 bytes	Any address is accepted. *2
2kBytes	0 (Disable)	1 (Enable)	1- 512 bytes	Any address is accepted. *2

*1: "Current Blocklen" size is set or changed by CMD16. If value is less than or equal 512 bytes (There are no relations with Misalign and Partial option), it is set with no error.
*2: When the Blocklen size data range crosses 512 bytes block boundary, card outputs the data until the 512 bytes block boundary" and then the data becomes invalid and CRC error also may occur. The card will send "ADDRESS_ERROR" on the next command response. Host should issue CMD12 to recover.

上面一段讲的是 misalignment，理解不够，直接贴英文了

写数据：

数据写传输类似读传输，块数据传输的话，在每块数据后面附加有 CRC 校验码

当出现 BLOCK_LEN_ERROR 或者 ADDRESS_ERROR 时，写命令会被拒绝，并不会发生实际的数据传输

写数据块

写数据块命令（CMD24-27, 42, 56(w)）会将一块或多块数据写入 SD 卡，支持块写入操作的SD卡要求由命令CMD16 设定的块长度(Block Lenghth)为 512byte，而不管WRITE_BL_LEN 是 1K 还是 2K

下表给出了部分写入禁止（WRITE_BL_PARTIAL=0）情况下 SD 卡的写入行为

CSD value			Current Blocklen *1	Write CMD Start Address
Max block size WRITE_BL_LEN	Misalign	Partial		
512Bytes	0 (Disable)	0 (Disable)	512 bytes *2	n * 512 bytes *3 (n: Integer)
1kBytes	0 (Disable)	0 (Disable)	512 bytes *2	n * 512 bytes *3 (n: Integer)
2kBytes	0 (Disable)	0 (Disable)	512 bytes *2	n * 512 bytes *3 (n: Integer)

*1: "Current Blocklen" size is set or changed by CMD16. If value is less than 512 bytes (there are no relations with Misalign and Partial option), it is set with no error. And then "Current Blocklen" size is tested when write command execution.
*2: If the current Blocklen is other than this value, the card indicates "BLOCK_LEN_ERROR" on the Write command response.
*3: If start address is other than this value, the card will send "ADDRESS_ERROR" on the Write command response.

如果 WRITE_BL_PARTIAL=1，则更小块数据的写入--精度可以为 1byte---是可以进行的。如果 CRC 校验失败，SD 卡会在 DAT 线上给出相应的指示，已传输数据会被丢弃，并且这其后所传输的数据块（在多块写入模式下）也会被忽略

编程 CSD 寄存器并不需要先前设定好的块长度。传输数据同样附加有 CRC 校验，如果 CSD 的某部分存储在 ROM 里面，这个不可改变的部分同样会映射到数据缓存的相应部分（但是这部分数据不会写入），如果这个匹配条件不满足，SD 卡会回报一个错误信息，并且此次写入无效

某些 SD 卡会需要比较长并且不可预期的写入时间。如果 SD 卡的缓存为满，则其在接收完块数据并完成 CRC 校验后，SD 卡会开始数据的写入，并拉低 DAT0。任何时刻，控制器可以通过命令 SEND_STATUS（CMD13）主动查询 SD 卡的状态，SD 卡对应的响应其标志位 READY_FOR_DATA 表明其是否可以接收新数据。

多块写操作之前的预擦除指令

使用预擦除命令（ACMD23）设定多个写入块的预擦除会让随后的多块写入更快完成。控制器一般会使用这个值（擦除多少块）来决定随后的多块数据写入的块数。如果控制器在写入多块数据之前使用 stop 名林终止数据传输，那么，余下的数据块里面的值是未知的，可能是旧的值，也可能是擦除后的空白值（全 1）。如果控制器传输的数据块数多于预擦除指定的块数，，SD 卡会以一次擦除一块的操作方式来擦除数据（当收到新数据的时候）。ACMD32 所定义的擦除块数会在多块数据写入完成后复位为默认值（1）

建议在每次使用 CMD25 之前使用此命令。注意 ACMD23 必须在刚好要使用写入命令之前使用，否则，如果中间穿插了别的命令，可能或把 ACMD23 所设定的值自动清除

发送可写入块数

在使用流水线机制的系统里面，控制器可能在某些情况下不清楚哪个数据块是最后的可正确写入的块。如果在多块写入中间出现写入错误，SD 卡会返回一个 ACMD22 命令，其间包含了可正确写入块的数值

擦除：

同时擦除多块数据在提高数据传输速率方面是很有帮助的，标识这些写入块可以通过命令 ERASE_WR_BLK_START(CMD32) ERASE_WR_BLK_END(CMD33)完成

控制器必须使用如下的命令序列来完成所需的操作：ERASE_WR_BLK_START ERASE_WR_BLK_END ， ERASE
如果这三个命令的某一个没有按照此顺序来执行，SD 卡会置位其 ERASE_SEQ_ERROR 位，并复位整个执行序列
If an out of sequence command (except SEND_STATUS) is received, the card shall set the
ERASE_RESET status bit in the status register, reset the erase sequence and execute the last
command. ---这段话啥子意思嘛...

如果擦除区域包含了写保护扇区，则这些区域不会被擦除，仅其余部分被擦除。WP_ERASE_SKIP 标志位也会置 1

在地址设定命令里面包含的地址以字节为单位，SD 卡会自动忽略小于 WRITE_BL_LEN 大小的低位地址信号

如同块写入，在擦除过程中，SD 卡也会拉低 DAT0。