

SD 卡读写硬件逻辑

MMC 的典型结构图如下，主要包含卡接口，卡接口控制器，寄存器堆，存储单元，存储单元接口，上电检测模块这几部分。

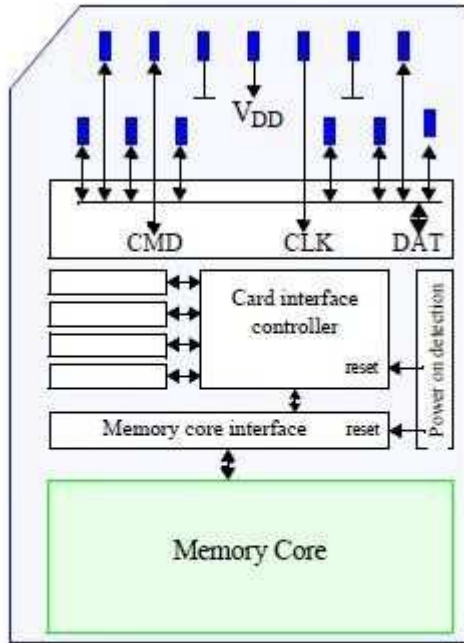


Figure 4 — MultiMediaCard architecture

<http://www.myspace.cn/1310160207>

MMC 卡状态图，识别模式

主要包含以下三个状态：

idle: Idle State

ready: Ready State

ident: Identification State

可以用 CMD3 进入数据传输模式的 stby 状态，Stand-by State。

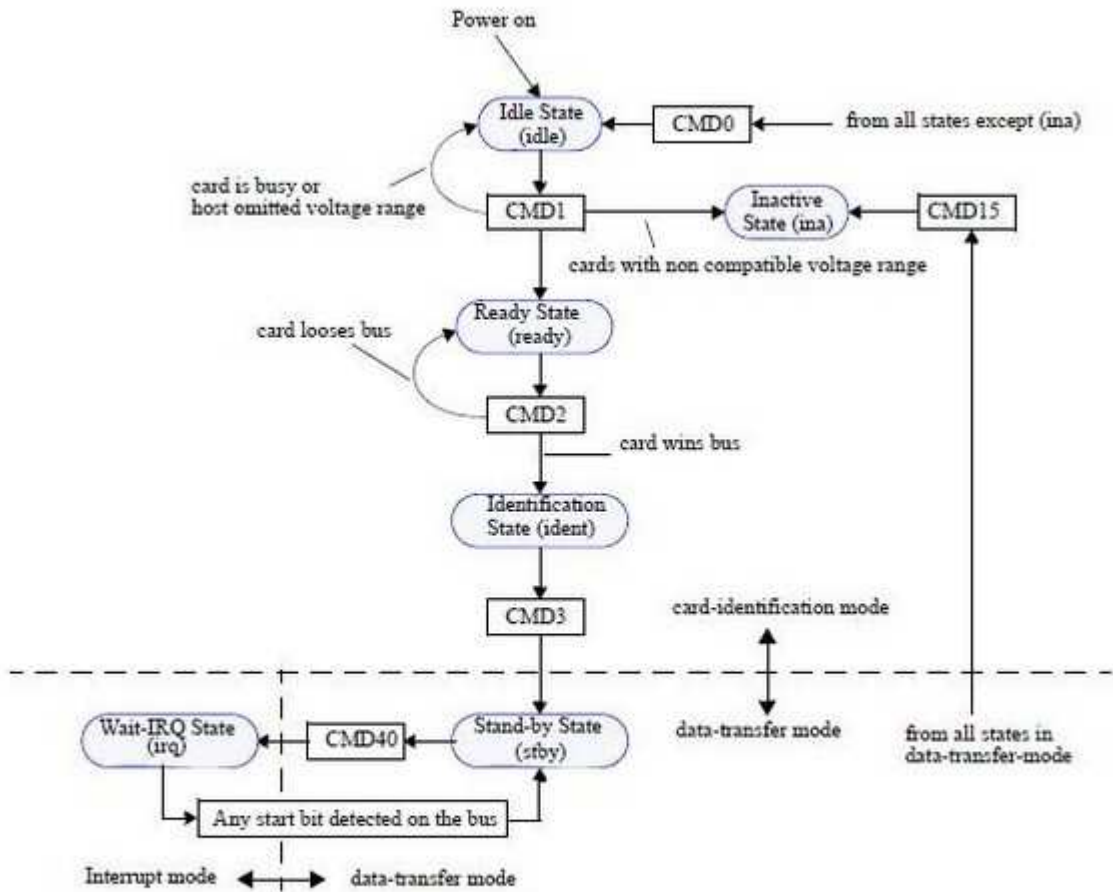


Figure 16 — MultiMediaCard state diagram (card identification mode)

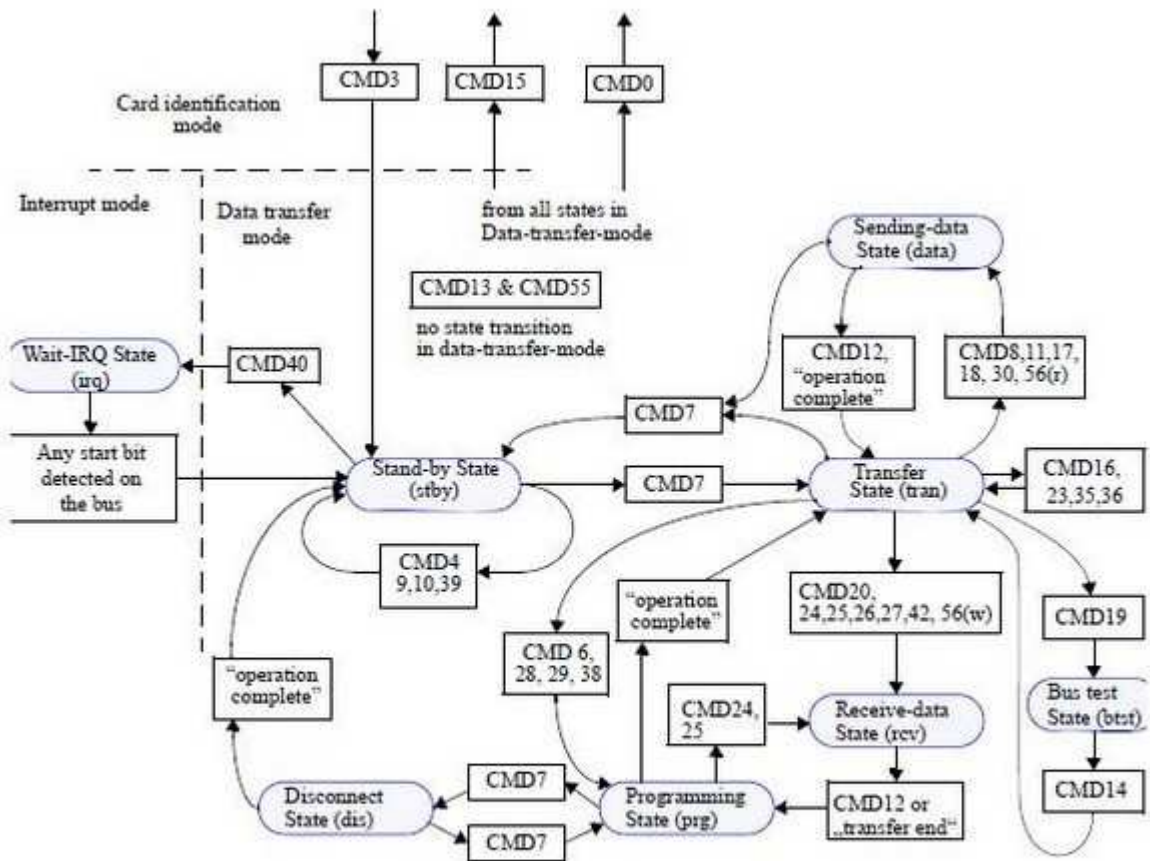


Figure 18 — MultiMediaCard state diagram (data transfer mode)

<http://www.myspace.cn/1310160207>

MMC 卡状态图，中断模式

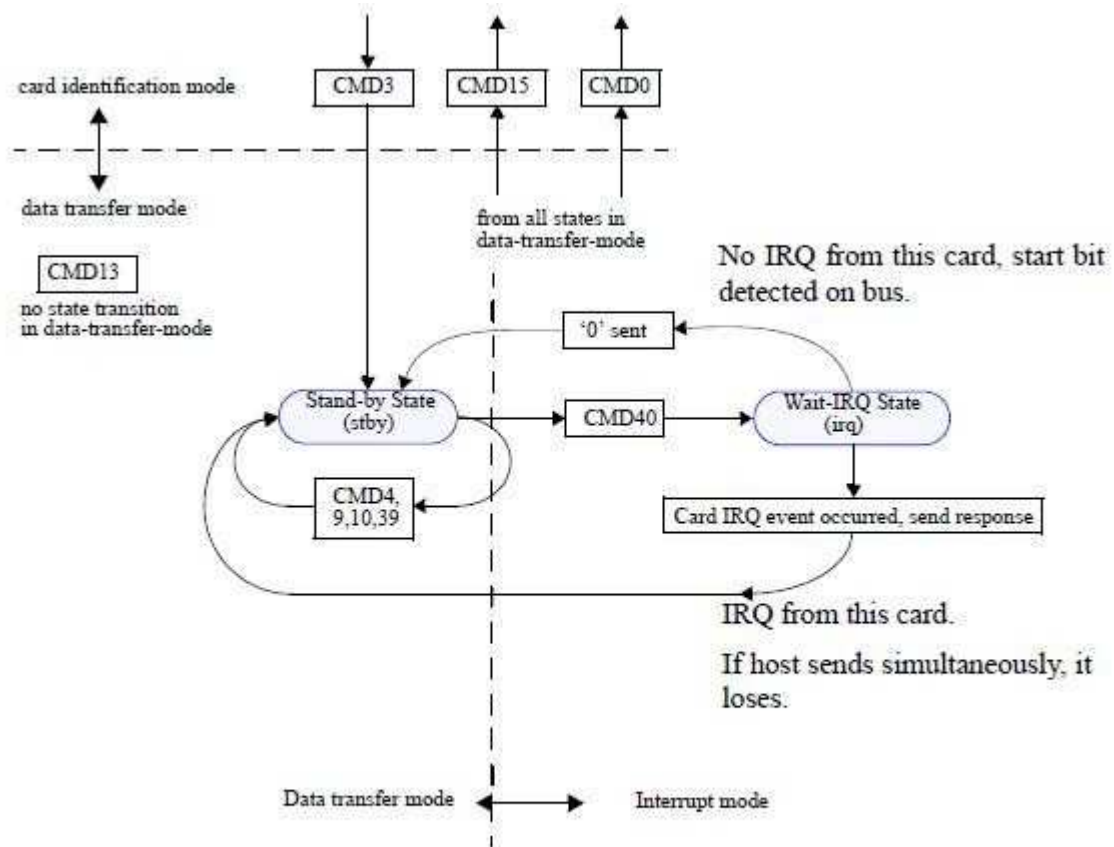


Figure 17 — MultiMediaCard state transition diagram, interrupt mode

<http://www.myspace.cn/1310160207>

MMC Identification Mode

MMC 识别模式

简介:

MMC 识别模式只用来识别卡。但是识别卡并不等同于初始化卡，这只是 MMC 卡整个初始化的一部分。

状态图详见 MMC 卡状态图，识别模式。

状态:

MMC 识别模式只包含三种状态，如下所示:

1. idle: Idle State 空闲状态

空闲状态是卡复位后进入的状态，匹配完 MMC 卡电压范围之后，MMC 卡转变到 Ready State。

2. ready: Ready State 准备状态

准备状态是 MMC 卡匹配完电压范围之后的状态，获取完 CID 值之后，MMC 卡进入 Identification State。

3. ident: Identification State 识别状态

识别状态是 MMC 卡响应 CID 之后进入的状态，如果分配 RCA 成功，MMC 卡进入 Stand-by State。

功能流程:

MMC 识别模式下只有一种功能流程，即 Identification Process 识别流程。详细介绍如下:

1. Identification Process: 识别流程

->idle->ready->ident->stby

这是 MMC 识别模式唯一支持的功能流程，涉及到 idle, ready, ident, stby 这四个状态，CMD0, CMD1, CMD2, CMD3 这四种指令，并对 OCR, CID 和 RCA 这三个寄存器进行了存取。下面列出详细的过程以及过程中的要点：

->idle:

a. 进入 Idle State 的方式只有两种，一种是通过 Power On 即上电进入，另外一种是通过 CMD0 切换。不过需要注意的是，CMD0 无法让卡从 Inactive State 进入 Idle State。

b. 进入 Idle 后，所有的寄存器都复位成默认值，RCA 也变成了 0x0001。而此时，CLK 需要设置成 fOD 的范围，即 0-400kHz，以进行下面的流程。

c. 虽然进入到了 Idle State，但是上电复位过程并不一定完成了，这主要靠读取 OCR 的 busy 位来判断，而流程归结为下一步。

详见 CMD0 详细介绍。

idle->ready:

a. 在 Idle State 只有 CMD1 和 CMD58 是合法指令。所以在进入 Idle State 之后，除了让卡从 Idle State 转变成 Ready State，别无他途。

b. Idle State 进入 Ready State 是通过 CMD1 完成的，只有发送带有 OCR 格式电压参数的 CMD1 给 MMC 卡后，busy 位为 1 且电压匹配，才进入 Ready State。

c. 如果回复的 OCR 中的 busy 位一直为 0，则说明上电复位仍没有完成，则需要一直发送 CMD1，直到 busy 位为 1。

d. 如果 MMC 卡收到 CMD1 的电压参数不匹配，则进入 Inactive State，不会对后续过程进行任何响应。

e. 如果 Host 的 IO 电压可调整，那调整前需要读取 OCR。为了不使卡误进入 Inactive State，可以给 MMC 卡发送不带参数的 CMD1，这样可以仅获取 OCR 寄存器，而不会改变卡的状态。

f. 在识别过程中，Host 不能改变电压范围，如果需要修改，则修改之后必须重新启动识别流程。

详见 CMD1 详细介绍。

详见 OCR 详细介绍。

ready->ident:

a. 在 Ready State 下，发送 CMD2 指令，可以使 MMC 卡则进入 Identification State。

b. CMD2 如果发送成功，Host 得到 CID 寄存器的值，而 MMC 卡则进入 Identification State。若不成功，则卡状态不变。

c. 在单卡识别的情况下，没有图示的 card looses bus 的可能，所以只要 fOD 在范围内，且上拉正确，应该可以准确无误的收到 CID 的值，并进入 Identification State。

d. 这一节最复杂之处在于多卡识别，单卡则很简单，后面有个 MMC 多卡识别的特别内容会有详细介绍。

详见 CMD2 详细介绍。

详见 CID 详细介绍。

ident->stby:

a. 在 Identification State 下，仅有 CMD3 才可以使 MMC 卡结束识别流程，进入 Stand-by State。

b. CMD3 如果发送成功，MMC 卡的 RCA 寄存器被写入分配的地址，卡进入 Stand-by State，以后均可以用这个分配的地址来寻址。若不成功，则卡状态不变。

c. MMC 卡进入 Stand-by State 之后，就不会响应 CMD1, CMD2, CMD3 这几条指令了。这其实给多卡识别创造了条件，后续有详细介绍。

详见 CMD3 详细介绍。

详见 RCA 详细介绍。

2. 特别内容：MMC 多卡识别

看过 SD 卡的多卡连接拓扑结构图之后，一直以来以为在 SD 接口模式下，多卡的 CMD，DATA 都是独立的，每个卡都有自己的链路。但是仔细阅读 MMC 协议后，发现 MMC 卡在多卡识别上有其独到之处。MMC 卡的多卡 CMD 和 DATA 是共用的，它们采用一种特殊的方式来完成识别和操作。过程如下：

a. 在 idle->ready 过程中，Host 用电压范围这一条件，首先让不符合电压条件的卡进入 Inactive State。

b. 在 ready->ident 过程中，CMD2 指令可以让所有的卡发送 CID 值，所有卡的 CMD 都接在一起，这样回传 CID 的时候，相当于做了一个或操作，与 CMD 上或出来的结果不一致的卡就不会继续发送（这里是需要卡的硬件有识别支持），而停留在 ready。这里 CID 的唯一性发挥了作用，这样每次 CMD2 操作之后，仅有一张卡进入 ident，其余的都停留在 ready 而不能响应 CMD3。

c. 在 ident->stby 过程中，上述过程中筛选出的唯一卡独占了 CMD 线，完成了 RCA 分配，进入 stby。进入 stby 的卡不再响应 CMD2 和 CMD3。

d. 识别完一张卡之后，因为其他的卡还停留在 ready，这样就可以继续 c, d 的步骤来一张一张的识别卡，最后直到 CMD2 再没有回应，说明卡都识别完了。

e. 完成上面的过程后，每张识别好的卡都进入了 stby，而且每张都有自己的 RCA，这样之后的操作只需要在命令中带入相应的 RCA，就可以对任意的卡进行独立操作了，别的卡是不会响应的。

MMC Interrupt Mode

MMC 中断模式

简介：

MMC 中断模式是一个可选模式，此模式的存在主要是为了省电。

状态图详见 MMC 卡状态图，中断模式。

状态：

MMC 中断模式只包含两种状态，如下所示：

1. stby: Stand-by State 等待状态

等待状态是 MMC 卡中断模式和数据传输模式共用的状态。

2. irq: Wait-IRQ State 等中断状态

等中断状态是 MMC 卡中断模式的唯一独立状态，用来等待卡的中断事件。

功能流程：

MMC 识别模式下只有一种功能流程，即 IRQ 流程。详细介绍如下：

1. IRQ Process: 中断流程

->irq->stby

这是 MMC 识别模式唯一支持的功能流程，涉及到 stby, irq 这两个状态，CMD40 指令。

下面列出详细的过程以及过程中的要点：

->irq:

a. 进入 Wait-IRQ State 只有一种办法，就是发送 CMD40。不过与其他指令不同的是，卡接收到 CMD40 之后就进入 irq，而回复却是在返回 stby 的时候才由卡发送。

b. 发送完 CMD40 之后，Host 需要继续发时钟，等待 CMD40 的回复。

详见 CMD40 详细介绍。

irq->stby:

- a. 从 Wait-IRQ State 返回 Stand-by State 有三种方式，但成功实现 IRQ 的只有一种。
 - b. MMC 的 IRQ 事件发生之后，卡发送 CMD40 的 RSP，之后进入 stby。
 - c. 卡需要强制从 irq 返回，则发送修正后的 CMD40 的 RSP 给卡，RCA 设定为 0，则卡跳出等待，进入 stby。
 - d. 卡检测到了 CMD 线上的起始位，直接跳回 stby。
- 详见 RCA 详细介绍。

Note:

1. MMC 卡的驱动很少涉及到中断模式，这部分验证也比较困难，这里只是理论分析。
2. 返回 stby 的后两种模式有些类似，暂时还不清楚区别在哪里。

CMD 命令详解:

class 0:

CMD0:

CMD1:

CMD2:

CMD3:

CMD4:

CMD7:

CMD8:

CMD9:

CMD10:

CMD12:

CMD13:

CMD14:

CMD15:

CMD19:

class1:

CMD11:

class2:

CMD16:

CMD17:

CMD18:

CMD23:

class3:

CMD20:

class4:

CMD24:

CMD25:

CMD26:

CMD27:

class5:

CMD35:

CMD36:

能从 Inactive State 复位。

2. 虽然没有回复，但是因为下一步是发送 CMD1，需要判断复位是否完成，所以最好发完 CMD0 之后延时一段时间。

MMC Command CMD1 (class 0)

CMD1: SEND_OP_COND (bcr)

在 idle 状态时，向卡传送 Host 支持的电压范围，卡回复 OCR 的值以及上电复位的状态。如果发送的电压参数为 0，则卡仅传回 OCR 的值，并不进行判断。如果发送的电压参数存在，则和卡本身的 OCR 对比，若不符合，则卡进入 Inactive State，符合，则返回 OCR 寄存器的值。

CMD:

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          OCR-          |
+++++
| -OCR  |  CRC7  |E|
+++++
```

CMD: 000001

OCR: OCR 格式的电压范围，忽略 OCR 的 busy 位。

CMD 典型值:

1. Host 传送的是高电压范围 2.7-3.6:

```
0_1_000001_000000001111111100000000 == 0x4100FF800099
00000000_1001100_1
```

2. Host 传送的是低电压范围 1.65-1.95:

```
0_1_000001_000000000000000000000000 == 0x41000000807B
10000000_0111101_1
```

3. Host 忽略电压参数，仅用来获取卡的 OCR 值:

```
0_1_000001_000000000000000000000000 == 0x4100000000F9
00000000_1111100_1
```

RSP: R3

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CHK1 |          OCR-          |
+++++
| -OCR  |  CHK2  |E|
+++++
```

CHK1: 111111

OCR: OCR 寄存器的值。详见 OCR 详细介绍。

CHK2: 111111

RSP 典型值:

1. 卡为高电压卡，上电复位过程已完成：
0_0_111111_100000001111111110000000 == 0x3F80FF8000FF
00000000_1111111_1
2. 卡为高电压卡，上电复位过程未完成：
0_0_111111_000000001111111110000000 == 0x3F00FF8000FF
00000000_1111111_1
3. 卡为双电压卡，上电复位过程已完成：
0_0_111111_100000001111111110000000 == 0x3F80FF8080FF
10000000_1111111_1
4. 卡为双电压卡，上电复位过程未完成：
0_0_111111_000000001111111110000000 == 0x3F00FF8080FF
10000000_XXXXXXX_1

Note:

1. CMD1 只用在在上电复位的时候，主要功能就是判断上电复位是否完成，以及卡是否和当前的 Host 支持电压匹配。
2. 如果 Host 支持 IO 电压调节，则可以在发送 CMD1 的时候，先用参数 0 获取卡的电压范围，然后根据 OCR 调整自身的电压，再把电压参数带上是发送一次 CMD1。
3. 如果 Host 获取到的 OCR 中的 busy 始终为 0，则 Host 需要不断的发送 CMD1，直到 busy 为 1。
4. Host 和卡通过 CMD1 实现电压匹配，如果匹配成功，则进入 Ready Status，如果匹配不成功，则进入 Inactive State。

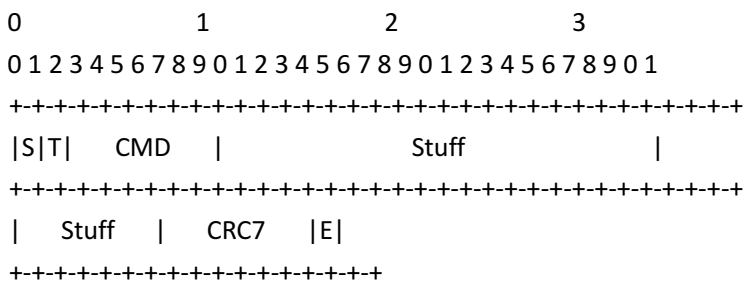
MMC Command CMD2 (class 0)

CMD2: ALL_SEND_CID (bcr)

使所有的卡回复 CID 寄存器的值。

CMD:

传送序列:



CMD: 000010

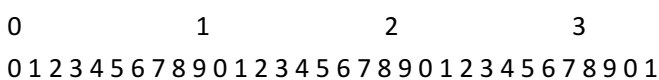
Stuff: 填充位，为 0。

CMD 典型值:

1. 因为参数为填充位，所以只有一种典型值：
0_1_000010_000000000000000000000000 == 0x42000000004D
00000000_0100110_1

RSP: R2

传送序列:



```

+++++
|S|T|  CHK  |          CID-          |
+++++
|          -CID-          |
+++++
|          -CID-          |
+++++
|          -CID-          |
+++++
|  -CID  |E|
+++++

```

CHK: 111111

CID: CID 寄存器的值，最后一位不传送。详见 CID 详细介绍。

RSP 典型值:

1. CID 会因为卡的不同而不同，所以没有具体的典型值:

```

0_0_111111_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_1

```

Note:

1. 另外还有一个命令 CMD10 和 CMD2 指令功能一致，区别只在于 CMD10 指定了 RCA，也就是指定为选定的卡。
2. CMD2 读取的时候，CID 的最后一位不传送，这必定是为了让整个命令长度满足字节对齐。
3. CID 寄存器的最后一个位“1”，在读取的时候，被 End 位取代，所以或许看到 MMC 控制器读回来的 CID 寄存器值最后一位为 0，这只是由于 MMC 控制器的处理方式问题。

MMC Command CMD3 (class 0)

CMD3: SET_RELATIVE_ADDR (ac)

给卡分配关联地址。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          RCA          |  Stuff-  |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 000011

RCA: 关联地址。详见 RCA 寄存器详细介绍。

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为 RCA 不确定，所以典型值也不确定：

0_1_000011_bbbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1

2. 假定设定的 RCA 为 0x02，则值为：

0_1_000011_000000000000010_00000000 == 0x43000200009D
00000000_1001110_1

RSP: R1

传送序列：

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++
```

CMD: 000011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值：

1. 典型值：

0_0_000011_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. 此命令是 MMC 卡初始化的最后一个命令，发送成功后，卡就完成了初始化，进入了数据传输模式。

2. 配置的 RCA 不能为 0x0000，因为 0x0000 有特殊用途。

3. 配置的 RCA 不能为 0x0001，因为 0x0001 是 RCA 寄存器的默认值，所有的卡都是 0x0001。

MMC Command CMD4 (class 0)

CMD4: SET_DSR (bc)

设置卡的 DSR 寄存器。

CMD:

传送序列：

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           DSR           |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++
```

CMD: 000100

DSR: DSR 寄存器的设定值。详见 DSR 寄存器详细介绍。

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为 DSR 的值不确定, 所以典型值不确定:

```
0_1_000100_bbbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1
```

2. 假定 DSR 的值为 0x0404, 则典型值为:

```
0_1_000100_00000100000000100_00000000 == 0x440404000045
00000000_0100010_1
```

RSP: None

没有任何回复。

Note:

1. 因为没有任何回复, 卡的 Status 也没有任何表示, 所以这条命令发送后无法得知是否正确。

2. MMC 卡的 DSR 寄存器未必存在, 设置前, 最好判断下 CSD 寄存器中的 DSR_IMP 是否为 1。

3. 通常在总线切换频率的时候, 调用 CMD4 来调整 MMC 端总线的驱动能力。

MMC Command CMD7 (class 0)

CMD3: SELECT/DESELECT CARD (ac)

选定或断开指定的卡。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          RCA          |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++
```

CMD: 000111

RCA: MMC 关联地址。详见 RCA 寄存器详细介绍。

00: 断开卡。

BB: 选定卡的 RCA 的值。

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为 RCA 不确定, 所以典型值也不确定:

```
0_1_000111_bbbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1
```

2. 假定选定的 MMC 其 RCA 为 0x02, 则值为:

```
0_1_000111_0000000000000010_00000000 == 0x47000200003F
00000000_0011111_1
```

2. 假定断开 MMC, 则值为:

```
0_1_000111_0000000000000000_00000000 == 0x470000000083
00000000_1000001_1
```

RSP: R1b

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 000111

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 典型值:

```

0_0_000111_bbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD7 通常在 stby 状态下发送, 选定后卡进入 tran 状态。
2. 使用 CMD7 选定 MMC 进入 tran 状态, 表示着整个 MMC 识别, 初始化过程已经完成。
3. 在 MMC 处于 prg 时, 也可以使用 CMD7 使当前 MMC 进入 dis 状态, 再用 CMD7 选定其他的 MMC 卡。

MMC Command CMD8 (class 0)

CMD8: SEND_EXT_CSD (adtc)

发送该命令, 使选定的卡以 Block 的方式回复 EXT_CSD 寄存器的值。

详见 EXT_CSD 寄存器详细介绍。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 001000

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为参数为填充位, 所以只有一种典型值:

```

0_1_001000_000000000000000000000000 == 0x4800000000C3
00000000_1100001_1

```

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```

```

+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 001000

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 典型值:

```

0_0_001000_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_bbbbbbb_1

```

Note:

1. CMD8 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 data 状态。
2. 需要注意的是，此命令回复中并不带有 EXT_CSD，而是以 Block 的方式发送。
3. 使用 CMD8 读取 EXT_CSD，使卡进入 data 状态，和一般的读数据走的流程一样。

MMC Command CMD9 (class 0)

CMD10: SEND_CID (ac)

发送该命令，使选定的卡回复 CID 寄存器的值。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           RCA           |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 001001

RCA: 卡的关联地址。详见 RCA 寄存器详解。

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为 RCA 的值不确定，所以典型值也不确定:

```

0_1_001001_bbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1

```

RSP: R2

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CHK  |           CSD-           |
+++++
|           -CSD-           |

```

```

+++++
|           -CSD-           |
+++++
|           -CSD-           |
+++++
| -CSD |E|
+++++

```

CHK: 111111

CSD: CSD 寄存器的值，最后一位不传送。详见 CSD 详细介绍。

RSP 典型值:

1. CSD 会因为卡的不同而不同，所以没有具体的典型值:

```

0_0_111111_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_1

```

Note:

1. CMD9 一般在 MMC 处于 stby 状态下发送，成功后 MMC 状态不变。
2. CMD9 读取的时候，CSD 的最后一位不传送，这必定是为了让整个命令长度满足字节对齐。
3. CSD 寄存器的最后一个位“1”，在读取的时候，被 End 位取代，所以或许看到 MMC 控制器读回来的 CSD 寄存器值最后一位为 0，这只是由于 MMC 控制器的处理方式问题。

MMC Command CMD10 (class 0)

CMD10: SEND_CID (ac)

发送该命令，使选定的卡回复 CID 寄存器的值。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           RCA           |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 001010

RCA: 卡的关联地址。详见 RCA 寄存器详解。

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为 RCA 的值不确定，所以典型值也不确定:

```

0_1_001010_bbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1

```

RSP: R2

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CHK  |          CID-          |
+++++
|          -CID-          |
+++++
|          -CID-          |
+++++
|          -CID-          |
+++++
|  -CID  |E|
+++++

```

CHK: 111111

CID: CID 寄存器的值，最后一位不传送。详见 CID 详细介绍。

RSP 典型值:

1. CID 会因为卡的不同而不同，所以没有具体的典型值:

```

0_0_111111_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_1

```

Note:

1. CMD10 一般在 stby 状态下发送，成功后状态不变。
2. 另外还有一个命令 CMD2 和 CMD10 指令功能一致，区别只在于 CMD2 未指定卡。
3. CMD10 读取的时候，CID 的最后一位不传送，这必定是为了让整个命令长度满足字节对齐。
4. CID 寄存器的最后一个位“1”，在读取的时候，被 End 位取代，所以或许看到 MMC 控制器读回来的 CID 寄存器值最后一位为 0，这只是由于 MMC 控制器的处理方式问题。

MMC Commond CMD13 (class 0)

CMD13: SEND_STATUS (ac)

发送该命令，使选定的卡回复状态寄存器的值。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          RCA          |  Stuff-  |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 001101

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 000011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 典型值:

```

0_0_111111_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_bbbbbbb_1

```

Note:

1. CMD14 一般在 MMC 处于 btst 状态下发送，成功后 MMC 进入 tran 状态。

MMC Command CMD15 (class 0)

CMD15: GO_INACTIVE_STATE (ac)

发送该命令，使选定的卡进入 Inactive State，即非激活状态。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           RCA           |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 001111

RCA: 卡的关联地址。详见 RCA 寄存器详解。

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为 RCA 的值不确定，所以典型值也不确定:

```

0_1_001111_bbbbbbbbbbbbbbb_00000000
00000000_bbbbbbb_1

```

RSP: None

没有任何回复。

RSP 典型值:

没有。

Note:

1. CMD15 一般在 MMC 处于数据传输模式下发送，成功后 MMC 进入 ina 状态。

2。

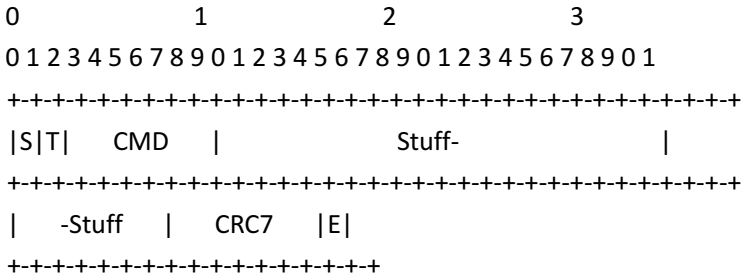
MMC Command CMD19 (class 0)

CMD19: BUSTEST_W (adtc)

发送该命令，使卡准备接收。

CMD:

传送序列:



CMD: 010011

Stuff: 填充位，为 0。

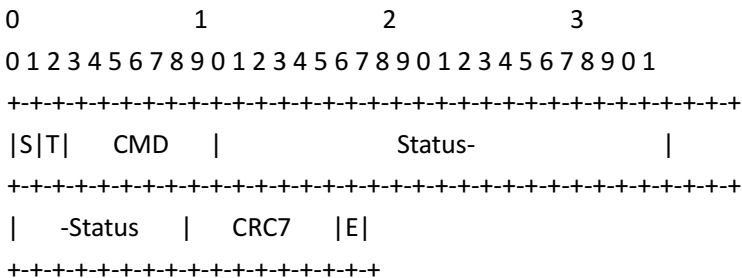
CMD 典型值:

1. 因为参数为填充位，所以只有一种典型值:

```
0_1_010011_000000000000000000000000 == 0x53000000008D
00000000_1000110_1
```

RSP: R1

传送序列:



CMD: 010011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 典型值:

```
0_0_111111_bbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1
```

Note:

1. CMD19 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 btst 状态。
2. 如果 CMD19 没有回复，则卡可能不支持总线测试。

MMC Command CMD11 (class 1)

CMD19: READ_DATA_UNTIL_STOP (adtc)

发送该命令，使卡从指定的地址开始，以流的方式传送数据，直到收到 CMD12。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Data Address-           |
+++++
| -Data Address |  CRC7  |E|
+++++

```

CMD: 001011

Data Address: 起始地址, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

```

0_1_001011_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 001011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

```

0_0_010000_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD11 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 进入 data 状态。
2. CMD11 的起始地址因为参数长度有限制, 所以最大为 4GB。

MMC Command CMD16 (class 2/4/7)

CMD16: SET_BLOCK_LEN (ac)

发送该命令, 设定 Block 的长度, 此长度作用于所有的 Block 操作。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Block Length-           |
+++++

```

| -Block Length | CRC7 |E|

+++++

CMD: 010000

Block Length: 设定的 Block 长度。

CMD 典型值:

1. 因为 Block 长度不确定, 所以典型值不确定:

0_1_010000_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

RSP: R1

传送序列:

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+++++

|S|T| CMD | Status- |

+++++

| -Status | CRC7 |E|

+++++

CMD: 010000

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_010000_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

- 1. CMD16 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 状态不变。
- 2. 如果不使用 CMD16 设置 Block 长度, 则 Block 长度默认为 CSD 寄存器中的值。

MMC Command CMD17 (class 2)

CMD17: READ_SINGLE_BLOCK (adtc)

发送该命令, 使卡从指定的地址开始, 发送一个 Block 的数据。

CMD:

传送序列:

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+++++

|S|T| CMD | Data Address- |

+++++

| -Data Address | CRC7 |E|

+++++

CMD: 010001

Data Address: 起始地址, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

0_1_001011_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

RSP: R1

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |          Status-          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Status  |  CRC7  |E|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

CMD: 010001

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_010001_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. CMD17 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 进入 data 状态。
2. 此命令中所指的一个 Block 长度可由 CMD16 指定, 也可以使用 CSD 中的默认值。

MMC Commond CMD18 (class 2)

CMD18: READ_MULTIPLE_BLOCK (adtc)

发送该命令, 使卡从指定的地址开始, 按 Block 为单位持续发送数据, 直到收到 CMD12 或达到 CMD23 设定的数量。

CMD:

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |          Data Address-          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Data Address |  CRC7  |E|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

CMD: 010010

Data Address: 起始地址, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

0_1_010010_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

RSP: R1

传送序列:

```
0          1          2          3
```

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |                Status-                |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 010010

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_010010_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb_bbbbbbb_1

```

Note:

1. CMD18 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 data 状态。
2. 此命令中所指的一个 Block 长度可由 CMD16 指定，也可以使用 CSD 中的默认值。

MMC Command CMD23 (class 2/4)

CMD23: SET_BLOCK_COUNT (ac)

发送该命令，设定 Block 的数量，此数量作用于接着的 CMD18 和 CMD25 命令。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          0          | Block Num- |
+++++
| -Block Num |  CRC7  |E|
+++++

```

CMD: 010111

Number of Blocks: Block 的数量，如果值为 0，则表明不限定 Block 数量。

CMD 典型值:

1. 因为 Block 数量不确定，所以典型值不确定:

```

0_1_010111_0000000000000000bbbbbbbb
bbbbbbb_bbbbbbb_1

```

RSP: R1

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |                Status-                |
+++++
| -Status  |  CRC7  |E|
+++++

```


CMD: 010111

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 典型值:

0_0_010111_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. CMD23 一般在 MMC 处于 tran 状态下发送，成功后 MMC 状态不变。
- 2.

MMC Command CMD20 (class 3)

CMD18: WRITE_DATA_UNTIL_STOP (adtc)

发送该命令，使卡从指定的地址开始，以流的方式接收数据，直到收到 CMD12。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Data Address-          |
+++++
| -Data Address |  CRC7  |E|
+++++

```

CMD: 010010

Data Address: 起始地址，以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定，所以典型值不确定:

0_1_010010_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

RSP: R1

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Status-          |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 010010

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

0_0_010010_bbbbbbbbbbbbbbbbbbbbbbb

bbbbbbbb_bbbbbbb_1

Note:

1. CMD20 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 rcv 状态。
2. CMD20 的起始地址因为参数长度有限制，所以最大为 4GB。

MMC Command CMD24 (class 4)

CMD24: WRITE_BLOCK (adtc)

发送该命令，使卡从指定的地址开始，发送一个 Block 的数据。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |           Data Address-           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Data Address |   CRC7   |E|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

CMD: 011000

Data Address: 起始地址，以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定，所以典型值不确定:

```

0_1_011000_bbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |           Status-           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Status  |   CRC7   |E|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

CMD: 011000

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_011000_bbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD24 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 rcv 状态。
- 2.

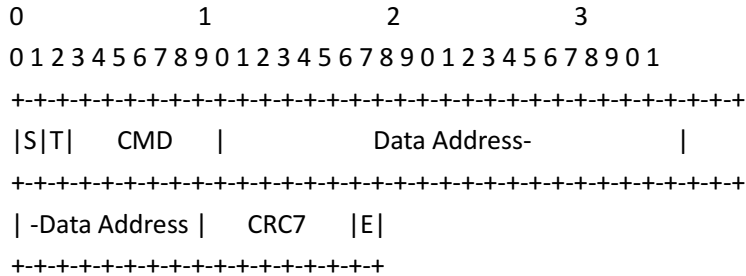
MMC Command CMD25 (class 4)

CMD25: WRITE_MULTIPLE_BLOCK (adtc)

发送该命令，使卡从指定的地址开始，按 Block 为单位持续接收数据，直到收到 CMD12 或达到 CMD23 设定的数量。

CMD:

传送序列:



CMD: 011001

Data Address: 起始地址，以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定，所以典型值不确定:

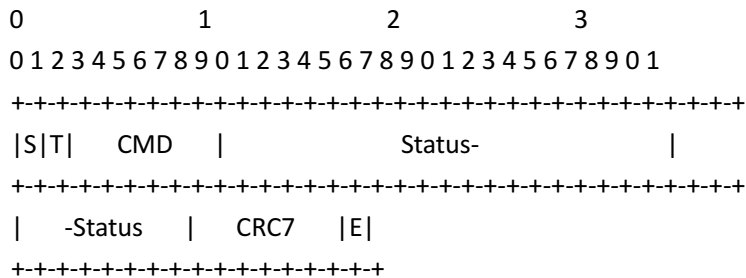
```

0_1_011001_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

RSP: R1

传送序列:



CMD: 011001

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_011001_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD25 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 rcv 状态。
- 2.

MMC Command CMD26 (class 4)

CMD26: PROGRAM_CID (adtc)

发送该命令，编写 CID 寄存器。

CMD:

传送序列:



```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 011010
 Stuff: 填充位，为 0。

CMD 典型值:

1. 因为参数为填充位，所以只有一种典型值：
 0_1_011010_000000000000000000000000 == 0x5A00000000B7
 00000000_1011011_1

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
|  -Status  |  CRC7  |E|
+++++

```

CMD: 011010
 Status: 卡的状态值。详见 Status 详细介绍。
 CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定：
 0_0_011010_bbbbbbbbbbbbbbbbbbbbbbb
 bbbbbbbb_bbbbbbb_1

Note:

1. CMD26 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 rcv 状态。
- 2.

MMC Command CMD27 (class 4)

CMD27: PROGRAM_CSD (adtc)

发送该命令，编写 CSD 寄存器的可编程部分。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 011011

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为参数为填充位, 所以只有一种典型值:

```
0_1_011011_000000000000000000000000 == 0x5B00000000DB
0000000_1101101_1
```

RSP: R1

传送序列:

0	1	2	3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+																						
S T	CMD		Status-																			
+-----+																						
	-Status		CRC7	E																		
+-----+																						

CMD: 011011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

```
0_0_011011_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1
```

Note:

- 1. CMD27 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 进入 rcv 状态。
- 2.

MMC Commond CMD35 (class 5)

CMD35: ERASE_GROUP_START (ac)

发送该命令, 设定擦除范围的起始地址, 与 CMD36 配合使用。

CMD:

传送序列:

0	1	2	3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+																						
S T	CMD		Data Address-																			
+-----+																						
	-Data Address		CRC7	E																		
+-----+																						

CMD: 100011

Data Address: 地址值, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

```
0_1_011001_bbbbbbbbbbbbbbbbbbbbbbb
```

bbbbbbbb_bbbbbbb_1

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 100011

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

```

0_0_100011_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD35 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 状态不变。
- 2.

MMC Command CMD36 (class 5)

CMD36: ERASE_GROUP_END (ac)

发送该命令, 设定擦除范围的结束地址, 与 CMD35 配合使用。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Data Address-           |
+++++
| -Data Address |  CRC7  |E|
+++++

```

CMD: 100100

Data Address: 地址值, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

```

0_1_100100_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```

```

+++++
|S|T|  CMD  |           Status-           |
+++++
|  -Status  |  CRC7  |E|
+++++

```

CMD: 100100

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_100100_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD36 一般在 MMC 处于 tran 状态下发送，成功后 MMC 状态不变。
- 2.

MMC Command CMD38 (class 5)

CMD38: ERASE (ac)

发送该命令，擦除 CMD35 和 CMD36 设定的擦除范围。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 100110

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为参数为填充位，所以只有一种典型值:

```

0_1_100110_000000000000000000000000 == 0x6600000000A5
00000000_1010010_1

```

RSP: R1b

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
|  -Status  |  CRC7  |E|
+++++

```

CMD: 100110

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_100110_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. CMD38 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 进入 prg 状态。
- 2.

MMC Command CMD28 (class 6)

CMD28: SET_WRITE_PROT (ac)

发送该命令, 设定指定写保护区域的保护。

CMD:

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |          Data Address-          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Data Address |  CRC7  |E|
+--+--+--+--+--+--+--+--+--+--+
```

CMD: 011100

Data Address: 地址值, 以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定, 所以典型值不确定:

0_1_011100_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

RSP: R1b

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|S|T|  CMD  |          Status-          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| -Status  |  CRC7  |E|
+--+--+--+--+--+--+--+--+--+--+
```

CMD: 011100

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_011100_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. CMD28 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 prg 状态。
- 2.

MMC Command CMD29 (class 6)

CMD29: CLR_WRITE_PROT (ac)

发送该命令，取消指定写保护区域的保护。

CMD:

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Data Address-          |
+++++
| -Data Address |  CRC7  |E|
+++++

```

CMD: 011101

Data Address: 地址值，以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定，所以典型值不确定:

```

0_1_011101_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

RSP: R1b

传送序列:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Status-          |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 011101

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_011101_bbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD29 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 prg 状态。
- 2.

MMC Command CMD30 (class 6)

CMD30: SEND_WRITE_PORT (adtc)

发送该命令，使卡发送从指定的写保护块开始的块保护状态。

CMD:

传送序列:

```
0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Data Address-           |
+++++
| -Data Address |  CRC7  |E|
+++++
```

CMD: 011110

Data Address: 地址值，以 Byte 为单位。

CMD 典型值:

1. 因为 Address 不确定，所以典型值不确定:

```
0_1_011110_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1
```

RSP: R1

传送序列:

```
0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++
```

CMD: 011110

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```
0_0_011110_bbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1
```

Note:

1. CMD30 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 data 状态。
- 2.

MMC Command CMD42 (class 7)

CMD42: LOCK_UNLOCK (adtc)

发送该命令，擦除 CMD35 和 CMD36 设定的擦除范围。

CMD:

传送序列:

```
0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

```

+++++
|S|T|  CMD  |           Stuff-           |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 101010

Stuff: 填充位，为 0。

CMD 典型值:

1. 因为参数为填充位，所以只有一种典型值:

```

0_1_100110_000000000000000000000000 == 0x6600000000A5
00000000_1010010_1

```

RSP: R1b

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 100110

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定，所以典型值不确定:

```

0_0_100110_bbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD38 一般在 MMC 处于 tran 状态下发送，成功后 MMC 进入 prg 状态。
- 2.

MMC Commond CMD55 (class 8)

CMD55: APP_CMD (ac)

发送该命令，指示 MMC 下一个命令是 ACMD。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           RCA           |  Stuff-  |
+++++
| -Stuff  |  CRC7  |E|
+++++

```

CMD: 110111

RCA: MMC 的关联地址。详见 RCA 寄存器详解。

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为参数为填充位, 所以只有一种典型值:

0_1_110111_bbbbbbbbbbbbbbb00000000
00000000_bbbbbbb_1

RSP: R1

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Status-           |
+++++
| -Status  |  CRC7  |E|
+++++

```

CMD: 110111

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_110111_bbbbbbbbbbbbbbb
bbbbbbb_bbbbbbb_1

Note:

1. CMD55 一般在 MMC 处于数据传输模式下发送, 成功后 MMC 状态不变。
- 2.

MMC Command CMD56 (class 8)

CMD56: GEN_CMD (adtc)

发送该命令,。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
| -Stuff  |D|  CRC7  |E|
+++++

```

CMD: 111000

Stuff: 填充位, 为 0。

D: RD/WR。表示读写的方向。1 为 Host 从 MMC 读取, 0 为 Host 向 MMC 写入。

CRC7: 7 位循环冗余校验。

CMD 典型值:

1. 当 Host 从 MMC 读取时, RD/WR 为 1, 典型值为:

0_1_111000_000000000000000000000000 == 0x780000000137
00000001_0011011_1

2. 当 Host 向 MMC 写入时, RD/WR 为 0, 典型值为:

0_1_111000_000000000000000000000000 == 0x780000000025
00000000_0010010_1

RSP: R1b

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Status-          |
+++++
|  -Status  |  CRC7  |E|
+++++
```

CMD: 111000

Status: 卡的状态值。详见 Status 详细介绍。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 Status 不确定, 所以典型值不确定:

0_0_111000_bbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbb_bbbbbbb_1

Note:

1. CMD56 一般在 MMC 处于 tran 状态下发送, 成功后 MMC 进入 rcv 或 data 状态。
- 2.

MMC Command CMD39 (class 9)

CMD39: FAST_IO (ac)

发送该命令,。

CMD:

传送序列:

```
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |          Stuff-          |
+++++
|  -Stuff  |  CRC7  |E|
+++++
```

CMD: 101010

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为参数为填充位, 所以只有一种典型值:

0_1_100110_000000000000000000000000 == 0x6600000000A5
00000000_1010010_1

RSP: R4

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           RCA           |S|  Reg Addr  |
+++++
|  Reg Data  |  CRC7  |E|
+++++

```

CMD: 101010

RCA: MMC 关联地址。详见 RCA 寄存器详解。

Status: 状态值。为 1 表示 CMD39 操作成功, 反之不成功。

Register addr.: 寄存器的地址。

Read register contents: 寄存器的值。

CRC7: 7 位循环冗余校验。

RSP 典型值:

1. 因为 RSP 的参数均不确定, 所以典型值不确定:

```

0_0_101010_bbbbbbbbbbbbbbbb_b_bbbbbbb
bbbbbbbb_bbbbbbb_1

```

Note:

1. CMD39 一般在 MMC 处于 stby 状态下发送, 成功后 MMC 进入 stby 状态。
- 2.

MMC Command CMD40 (class 9)

CMD40: GO_IRQ_STATE (bcr)

发送该命令, 使所有的卡进入中断模式。

CMD:

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|S|T|  CMD  |           Stuff-           |
+++++
|  -Stuff  |  CRC7  |E|
+++++

```

CMD: 101000

Stuff: 填充位, 为 0。

CMD 典型值:

1. 因为参数为填充位, 所以只有一种典型值:

```

0_1_101000_000000000000000000000000 == 0x680000000089
0000000_1000100_1

```

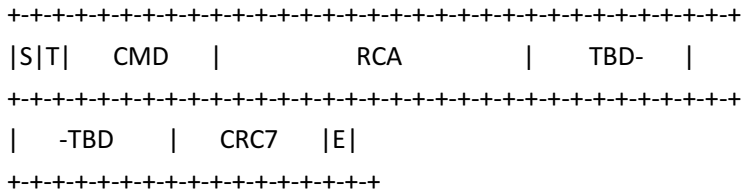
RSP: R5

传送序列:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```



CMD: 101000

RCA: RCA 寄存器的值。详见 RCA 详细介绍。

TBD: 还没有定义，可能会用作 IRQ 的附带数据。

RSP 典型值:

1. 每张卡 的 RCA 都不一样，且 TBD 可能会用来传递一些数据，所以没有具体的典型值:

```

0_0_111111_XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXX_XXXXXX_1

```

Note:

1. 这是一条特殊的指令，只用在 MMC 的中断模式，而且并不是所有卡都支持。
2. 这条指令比较奇特，发送完 CMD 之后，RSP 要等到卡的 IRQ 事件发生之后才能返回，其特定的参数可以携带终端类型等数据。
3. 该指令的 RSP 将 RCA 参数设置为 0x0000 后，还可以由 Host 发给卡，使卡返回 stby 状态。

Status 并没有归结到寄存器，但是在卡中应该也是以寄存器的方式存储的，所以列在寄存器系列。

MMC Register OCR

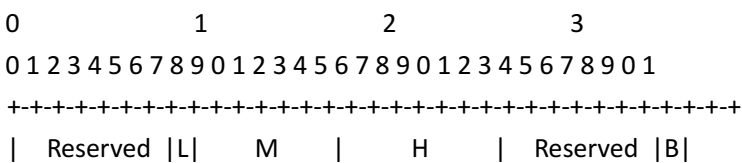
OCR: Operation Condition Register 工作条件寄存器

OCR 寄存器保存了卡所支持的 VDD 范围以及指示了上电过程的状态。OCR 主要用在上电复位中，主要有两个作用，一是在程序进入 Idle State 之后，读 busy 位来判断上电过程是否完成；二是用来判断 CMD1 发送来的电压范围卡是否支持，如果不支持，则进入 Inactive State，初始化失败。

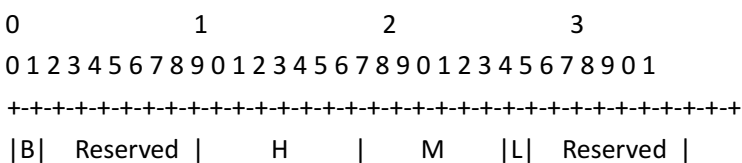
寄存器位域:

OCR 一共有 32 个 bit，总共分为 4 个位域，与电压相关的有 3 个，另外还有一个特殊的位域表示上电过程的状态。位域如下所示:

存储序列:



传送序列:



B: [30:24] 7bit card power up status bit(busy)

0: MMC 处于 busy 状态, 即上电过程未完成。

1: MMC 的上电过程已经完成。

LV: [7] 1bit 1.65-1.95

0: 不支持 1.65-1.95V, High Voltage MMC 为 0。

1: 支持 1.65-1.95V, Dual Voltage MMC 为 1。

MV: [14:8] 7bit 2.0-2.6

0000000: 该位域固定为此值, 任何 MMC 均不支持 2.0-2.6V。

HV: [23:15] 9bit 2.7-3.6

111111111: 该位域固定为此值, 任何 MMC 卡均支持 2.7-3.6V。

访问命令:

OCR 的访问命令仅有 CMD1。命令如下所示:

CMD1: SEND_OP_COND 详见 CMD1 详细介绍。

CMD1 将所支持的电压以 OCR 的格式发送到 MMC 卡, 并从 RSP 中获取 OCR 的值。发送 OCR 格式电压值忽略 busy 位。

Note:

1.MMC 卡只有 High Voltage 和 Dual Voltage 两种, 所以在上电完成后 OCR 的值只有两种:

高电压卡: 1000_0000_1111_1111_1000_0000_0000_0000 : 0x80FF8000

双电压卡: 1000_0000_1111_1111_1000_0000_1000_0000 : 0x80FF8080

2.OCR 表示一个电压范围用了很多位, 用一位岂不是更简便? 其实是有深意的, 这是为了和 SD 卡兼容, 因为 SD 把电压分的更细, 有兴趣的对照 SD 卡的 OCR 分析下, 就看出来了。

MMC Register CID

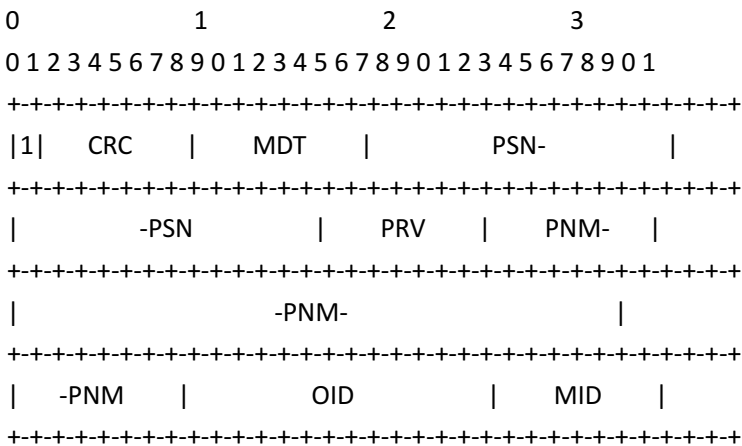
CID: Card Identification Register 卡识别寄存器

CID 寄存器保存了卡的识别信息, 主要应用于卡的识别过程。每张卡都应该有一个独一无二的寄存器值。

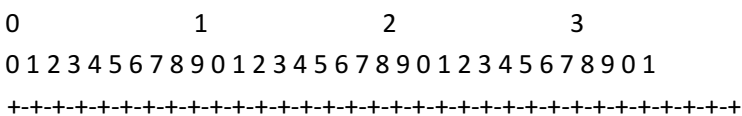
寄存器位域:

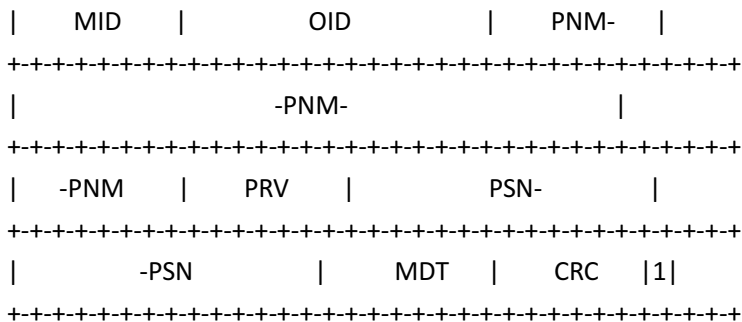
OCR 一共有 128 个 bit, 总共分为 4 个位域, 与电压相关的有 3 个, 另外还有一个特殊的位域表示上电过程的状态。位域如下所示:

存储序列:



传送序列:





MID: [127:120] 8bit Manufacturer ID

BB: 制造商 ID 号。由 MMCA 分配。典型值有 Sandisk 0x02, Kingston 0x37, Samsung 0x15。

OID: [119:104] 16bit OEN/Application ID

BB: OEM/应用 ID 号。由 MMCA 分配。

PNM: [103:56] 48bit Product Name

BBBBBB: 产品名称。由 6 个 ASCII 码组成。

PRV: [55:48] 8bit Product revision

BB: 产品版本。由 2 个 BCD 码组成。4bit 为一个 BCD 码，形成 x.x 这样的版本格式。

PSN: [47:16] 32bit Product serial number

BBBBB: 产品序列号。这个序列号很随意，由工厂定义。

MDT: [15:8] 8bit Manufacturing date

BB: 生产日期。分为 2 个 0-16 的值，前一个是月份，1 对应为 1 跃，依次类推，后一个为年份，0 对应 1997。

CRC: [7:1] 7bit CRC7 checksum

XXXXXXX: 循环冗余校验。详见 CRC7。

访问命令:

OCR 的访问命令有 CMD2, CMD10 和 CMD26。命令如下所示:

CMD2: ALL_SNED_CID 详见 CMD2 详细介绍。

CMD2 用于读取 CID 寄存器的值。

CMD10: SEND_CID 详见 CMD10 详细介绍。

CMD10 也是用于读取 CID 寄存器的值，不同的是，它指定了 RCA，只能读取选定的卡。

CMD26: PROGRAM_CID 详见 CMD26 详细介绍。

CMD16 用于修改 CID 寄存器的值，但是如果 CID 不可写入，则该命令无作用或者返回一个错误 (Status 的 CID/CSD_OVERWRITE 位)。

Note:

1. 通常 CID 寄存器由硬件保证为仅允许一次写入，保留给工厂用来设置 CID 值。
2. CID 寄存器的最后一个位“1”，在读取的时候，被 End 位取代，所以或许看到 MMC 控制器读回来的 CID 寄存器值最后一位为 0，这只是由于 MMC 控制器的处理方式问题。
3. MDT 中，年份的值为 0-16，基数为 1997，这样范围只有 1997-2013，看来下个 MMC 协议版本，这里肯定会修改。
4. 标准中规定每张卡都有唯一的 CID 值，但是实际并非如此，CID 号重复是常有的事情。

MMC Register RCA

RCA: Relative Card Address Register 关联卡地址寄存器

RCA 寄存器保存了 Host 在识别过程中给卡配置的地址。此地址的主要作用是在识别过程之后，用于卡的识别。

寄存器位域:

RCA 一共有 16 个 bit, 只有一个位域, 表示卡被分配的地址。位域如下所示:

存储序列:

```
0          1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+++++
|          RCA          |
+++++
```

传送序列:

```
0          1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+++++
|          RCA          |
+++++
```

RCA: [15:0] 16bit Relative card address

BB: 关联卡地址, 默认值为 0x0001。

访问命令:

RCA 的访问命令有 CMD3。命令如下所示:

CMD3: SET_RELATIVE_ADDR 详见 CMD3 详细介绍。

CMD3 用于设定 RCA 寄存器的值, 参数中的 RCA 值不能使用 0x0000 和 0x0001。

Note:

1. 0x0000 这个值保留, 用来给 CMD7 作为特殊参数。
2. 按照标准来说, 这个地址只是为了替代 CID 的值, 因为标准认为 CID 的值足够用来识别卡, 只是较复杂, 可实际来看, CID 的重复很普遍。

MMC Register DSR

DSR: Driver Stage Register

DSR 寄存器是一个可选寄存器, 用来配置卡的总线驱动参数。

寄存器位域:

DSR 一共有 16 个 bit, 分为两个位域。位域如下所示:

存储序列:

```
0          1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+++++
|  Tswitch  |  lpeak  |
+++++
```

传送序列:

```
0          1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+++++
|  lpeak    |  Tswitch |
+++++
```

lPeak: [31:18] 16bit lpeak max & lpeak min

BB: 选择 lpeak 的最大值和最小值。这是一个选项, 每个 bit 对应一组最大值和最小值。

Tswitch:[17:0] 16bit Tswitch-on max & Tswitch-on min

BB: 选择 Tswitch-on 的最大值和最小值。这是一个选项，每个 bit 对应一组最大值和最小值。

访问命令：

RCA 的访问命令有 CMD4。命令如下所示：

CMD4: SET_DSR 详见 CMD4 详细介绍。

CMD4 用于设定 DSR 寄存器的值，没有返回值。

Note:

1. DSR 寄存器是一个可选的寄存器，由 CSD 寄存器中的 DSR_IMP 标示，为 1 则存在。
2. DSR 寄存器的默认值是 0x0404。
3. 之所以用 DSR 寄存器来设定总线的驱动能力，是因为总线走线的不一和拓扑结构的多样，造成总线的容值特性多样。