## eMMC4.4协议

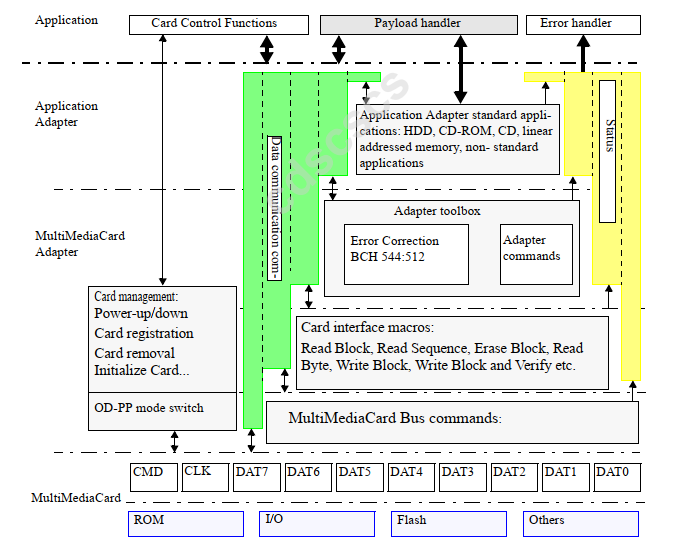
### 多媒体卡系统

多媒体卡系统的主要设计目标是提供一个低成本的大容量存储产品，具有简单的控制单元以及易于实现的接口。这些要求导致每个卡的功能减少到最低限度。然而，由于完整的多媒体卡系统必须具有执行任务的功能，例如错误纠正和标准总线连接。

**四种典型的多媒体卡系统**：

* 软件协议仿真：降低数据速率，通常为每秒100-300Kb，受主机限制；
* 点对点链接：全数据速率；
* 简单总线：全数据速率，一组可寻址单元的一部分；
* PC总线：全数据速率，可寻址，扩展功能，如DMA功能；

在第一种变体中，多媒体卡总线协议在软件中使用微控制器的多达10个端口引脚进行仿真。此解决方案不需要额外的硬件并且是列表中最便宜的系统。不同的系统虽然在特性集上有所不同，但具有基本的公共功能。



多媒体卡系统至少包含两个组件：多媒体卡和控制器。控制器分为两个主要模块。在一些实现中控制器可以实现整个应用程序，而在其他实现中，控制器可能会分为几个物理组件。

* 应用适配器：应用特定的块，例如，微处理器或适配器到标准总线，如USB；通常作为标准总线的从设备连接；
* 多媒体卡适配器：通用块，包含所有卡特定的功能，如初始化和纠错；作为多媒体卡总线的主机；实现卡的标准接口；

#### 1.1 高于2GB

最低的公共指定值将设置限制。实现高于2GB的内存将无法向后兼容较低容量的内存。首先，高于2GB内存的地址参数被更改为扇区地址(512B)而不是字节地址。其次，从EXT\_CSD寄存器而不是CSD寄存器读取卡的容量。最后，系统实现需要包括一个能够处理扇区类型地址的文件系统。

该标准进一步定义了两种卡类型MMC加强版和MMC便携版，以描述具有明确定义的强制性功能和属性的R/W或ROM卡。只有符合MMC加强或MMC便携版要求的卡片才有资格携带MMC加强版或MMC便携名称和徽标。

* MMC加强版定义为普通尺寸的R/W或ROM卡，支持2.7-3.6V操作，x1/x4/x8总线宽度；
* MMC便携版定义为缩小尺寸的R/W或ROM卡，支持1.70-1.95V和2.7-3.6V操作，x1/x4/x8总线宽度，至少2.4MB/s的读写性能；

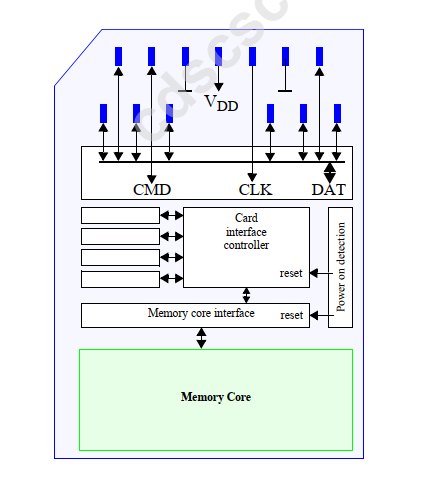
#### 1.2 引脚信号

* CLK：该信号的每个周期指示命令上的一位传输并且在所有数据线上进行一位(1x)或两位传输(2x)；频率可以在0到最大时钟频率之间变化；
* CMD：该信号是用于卡初始化和命令传输的双向命令通道。CMD信号有两种操作模式：用于初始化模式的开漏输出和用于快速命令传输的上拉输出。命令从总线主机发送到卡，响应从卡发送到主机；
* DAT0-DAT7：双向数据通道。数据信号工作在推拉模式。一次只有卡或主机驱动这些信号。默认情况下，上电或复位后，只使用DAT0进行数据传输。一个更宽的数据总线可以配置数据传输，使用DAT0-DAT3或DAT0-DAT7。多媒体卡包括数据线DAT1-DAT7的内部上拉。进入4位模式后，卡立即断开内部拔线DAT1, DAT2和DAT3。相应地，在进入8位模式后，卡立即断开内部上拉线DAT1-DAT7；

**多媒体卡分类**：

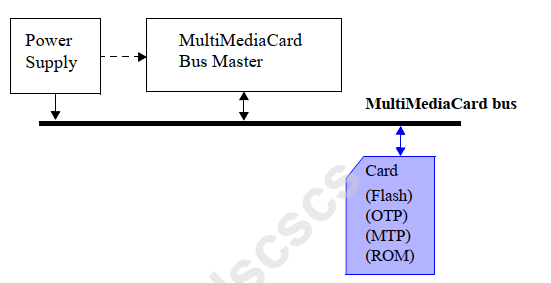
* 只读存储器(ROM)卡：这些卡片是具有固定的数据内容，不可被修改。通常被用作软件、音频、视频等的分发媒介；
* 读/写(RW)卡：这些卡通常作为空白(空)媒体出售，用于大量数据存储，最终用户录制视频、音频或数字图像；
* I/O卡：这些卡用于通信(例如调制解调器)，通常会有一个额外的接口链接；

主机可以通过关闭电源和重新打开电源来复位卡。卡应具有自己的上电检测电路，该电路在上电后将卡置于规定的状态。对于MMC卡，不需要显式复位信号。但是，对于eMMC有一个复位信号，主机可以使用该信号复位eMMC设备。MMC卡和eMMC也可以通过特殊的命令复位。



#### 1.3 总线

多媒体卡总线设计用于连接固态大容量存储器或卡格式的IO设备到多媒体应用程序。总线实现允许覆盖应用领域，从低成本系统到具有快速数据传输速率的系统。主总线是总线控制器，从总线是单个大容量存储卡或具有自己的控制单元(卡上)的IO卡来执行数据传输。



多媒体卡总线还包括为这些卡供电的电源连接。总线通信采用一种特殊的协议(多媒体卡总线协议)。主机和卡之间的有效数据传输可以是双向的。

总线可分为三组：

* 电源:VSS1和VSS2用于卡供电；
* 数据传输：CMD和DAT0-DAT7用于双向通信；
* 时钟：CLK用于在总线上同步数据传输；

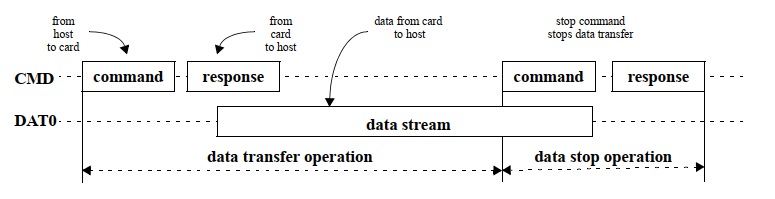
上电复位后，主机必须通过一个特殊的基于消息的多媒体卡总线协议初始化卡。

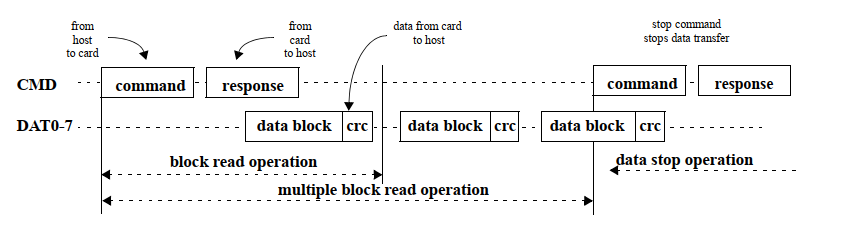
* 命令：启动操作的令牌。命令从主机发送到卡。命令在CMD行上串行传输；
* 响应：响应是一个令牌，作为对先前收到的命令的回答从卡发送到主机。响应在CMD行上串行传输；
* 数据：数据可以从卡传输到主机，反之亦然。数据通过数据线传输。用于数据传输的数据线的数量可以是1(DAT0)， 4(DAT0- dat3)或8(DAT0- DAT7)。对于每条数据线，每个时钟周期可以传输1位(单数据速率)或2位(双数据速率)的数据；

卡寻址是使用会话地址来实现的，会话地址在初始化阶段由总线控制器分配给所连接的卡。一张卡是由它的CID号来识别的。这种方法需要卡有一个唯一的CID号。为了确保CID的唯一性，CID寄存器包含24位，由MMCA/JEDEC定义。每个卡制造商都需要申请一个唯一的MID(和可选的OID)号。

多媒体卡总线数据传输由这些令牌组成。一种数据传输是总线操作。有不同类型的操作。寻址操作总是包含一个命令和一个响应令牌。此外，一些操作具有数据令牌，其他操作直接在命令或响应结构中传输其信息。在这种情况下，操作中不存在数据令牌。DAT0-DAT7和CMD行的位同步传输到主机时钟。

* 顺序命令：这些命令启动一个连续的数据流，只有当CMD行上跟随一个停止命令时，才会终止。这种模式将命令开销减少到最小；
* 面向块的命令：这些命令发送数据块后加上CRC位。读和写操作都允许单块或多块传输。当在CMD行上跟随一个停止命令时，与顺序读取类似，多块传输被终止；





每个命令令牌之前都有一个开始位，之后是一个结束位)。总长度为48位。每个令牌都由CRC位保护，因此可以检测到传输错误，并且可以重复操作。响应令牌根据其内容有五种编码方案。令牌长度为48或136位。

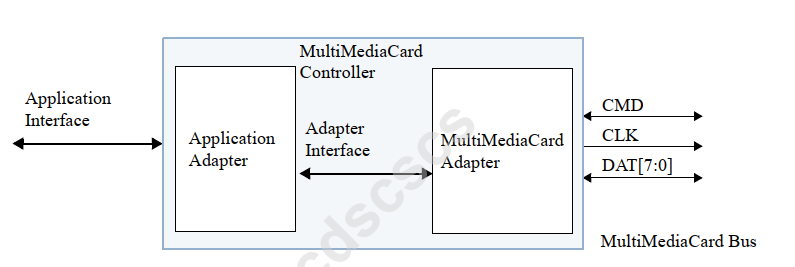
由于顺序数据传输没有预定义的结束，因此不包含CRC保护。块数据的CRC保护算法是一个16位的CCITT多项式。

#### 1.4 控制器

多媒体卡被定义为低成本的大容量存储产品。公共功能必须在多媒体卡系统中实现。包含这些功能的单元称为多媒体卡控制器。

* 从标准多媒体卡总线到应用总线的协议转换；
* 数据缓冲以实现最小的数据访问延迟；
* 用于常见复杂命令序列的宏；

多媒体卡控制器是应用程序和多媒体卡总线及其卡之间的链接。将标准多媒体卡总线的协议转换为应用总线。分为两个主要部分：应用程序适配器，面向应用程序的部分；多媒体卡适配器，面向多媒体卡的部分。



应用程序适配器至少由总线从机和连接到多媒体卡系统的桥接组成。可以通过扩展成为应用程序总线上的主节点并支持DMA等功能满足特定于应用程序的需求。更高的集成将把多媒体卡控制器与应用程序结合起来。

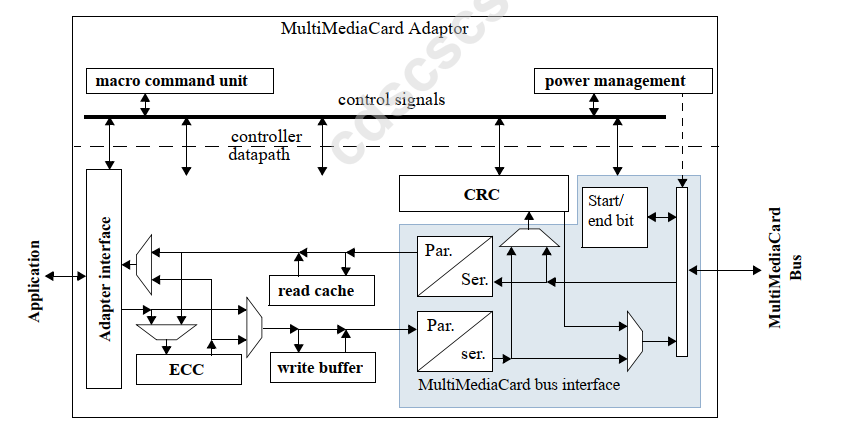
无论应用程序的类型和要求如何，多媒体卡总线都需要一个主机。这个主机可能是多媒体卡适配器。在多媒体卡总线端，是唯一的总线主机并控制该总线上的所有活动。另一方面，是应用程序适配器或应用程序的从设备。这里不支持特定于应用程序的功能，除了大多数多媒体卡系统中常见的功能。支持所有多媒体卡总线命令并提供额外的一组宏命令。适配器包括非无错误卡的纠错功能。

##### 1.4.1 应用适配器

应用适配器增强了多媒体卡系统，使其在每个标准总线环境中都成为即插即用。每个环境都需要其独特的应用程序适配器。对于某些总线系统标准，现成的应用程序适配器存在，并且可以与多媒体卡适配器进行接口。为了减少面积，建议将现有的应用程序适配器与多媒体卡适配器模块集成以形成多媒体卡控制器。应用程序适配器扩展是应用程序适配器的功能增强，从总线从到标准应用程序总线上的总线主。

##### 1.4.2 多媒体卡适配器

该适配器分为两大部分：控制器，宏单元和电源管理；数据路径，适配器接口、ECC单元、读缓存、写缓存、CRC单元和多媒体卡总线接口。



数据路径单元应在硬件中实现，以保证多媒体卡系统的全部功能。适配器的控制器部分可以在硬件或软件中实现，具体取决于应用程序体系结构。数据路径的宽度应该是一个字节；处理数据的单元应该处理字节或块。

多媒体卡总线的命令遵循严格的协议。每个命令都封装在一个语法框架中。每个帧包含一些特殊的控制信息，如开始结束位和CRC保护。一些命令包括填充位，使简单的解释器能够使用固定的帧长度。这个传输管理信息应该在多媒体卡适配器中生成。

多媒体卡系统的响应延迟可能会有所不同；取决于卡片的类型。因此适配器接口必须通过握手信号(STB,ACK)处理异步模式，或者如果不需要握手信号(同步模式)，主机必须轮询状态(忙/不忙)。该接口可以是支持大多数应用程序协议的通用单元，也可以针对一个应用程序进行定制。

建议为多媒体卡适配器配备数据缓冲区，以进行写入和读取操作。在大多数情况下，将提高应用程序端的系统级性能。多媒体卡总线以高达832Mb/秒的数据速率传输数据。这可能比典型的应用程序CPU总线慢。启用CPU将数据卸载到缓冲区将为系统级任务释放CPU时间，而多媒体卡适配器将处理数据传输到卡。

可以通过在读缓存中缓存数据块来改进从卡随机访问读操作的访问时间。在将一个完整的块读入适配器缓存后，对该块的重复访问可以非常快地完成。特别是在SRAM交换器的帮助下，读-修改-写操作可以在块缓冲区上以非常有效的方式执行。

### 2 功能描述

#### 2.1 概述

主机和卡之间的所有通信都由主机控制。主机发送的命令有两种类型：广播命令和寻址(点对点)命令。广播命令适用于多媒体卡系统中的所有卡片。其中一些命令需要响应。寻址(点对点)命令被发送到寻址卡并引起该卡的响应。

**操作模式**：

* 启动模式：在电源周期后，卡将处于启动模式，接收CMD0参数为0xF0F0F0F0或(仅eMMC)断言硬件复位信号；
* 卡识别模式：在启动模式完成后，或者主机或卡不支持启动模式时，卡将处于卡识别模式。卡将处于这种模式，直到接收到SET\_RCA命令(CMD3)；
* 中断模式：主机和卡同时进入和退出中断模式。在中断模式下，没有数据传输。唯一允许的消息是来自卡或主机的中断服务请求；
* 数据传输模式：一旦为卡分配了RCA，卡将进入数据传输模式。主机将在总线上识别卡后输入数据传输命令；
* 非激活模式：卡将进入非激活模式，卡的工作电压范围或访问模式无效。卡也可以使用进入非激活命令(CMD15)进入非活动模式。卡将复位到预空闲状态与电源周期；

#### 2.2 分区管理

##### 2.2.1 概述

内存设备的默认区域包括一个用户数据区用于存储数据，两个可用于引导的引导区和重放保护内存块区域分区(以认证和重放保护的方式管理数据)。内存配置初始化包括(在任何分区操作之前)用户数据区和RPMB区域分区和引导区域分区。

嵌入式设备还提供了由主机配置额外的内存分区，这些分区具有从逻辑地址0x00000000开始的独立可寻址空间，适用于不同的使用情况。

内存块区域扫描分为：

* 两个启动区分区，大小为128kb的倍数，可以从eMMC启动；
* 通过可信机制访问的RPMB分区，其大小定义为128kb的倍数；
* 4个通用分区，用于存放敏感数据或其他主机使用模式，大小为写保护组的倍数；

每个通用分区都可以通过增强的技术特性来实现，从而与默认存储介质区分开来。如果设备支持增强存储介质特性，则启动分区和RPMB区域分区默认实现为增强存储介质。

启动分区和RPMB区域分区的大小和属性由内存制造商定义，而通用区域分区的大小和属性只能由主机在设备生命周期中编程一次(一次性可编程)。此外，主机可以自由配置用户数据区中的一个段作为增强型存储介质，并根据写保护组指定其起始位置和大小。此增强用户数据区的属性在设备生命周期中只能编程一次(一次性可编程)。

主机的通用分区和增强型用户数据区配置可能会影响先前存储的数据和设备初始化时间。特别是，在配置之后的第一个电源周期后的初始化时间可以超过规格定义的最大初始化时间，因为内部控制器可以执行操作设置主机所述的配置。

##### 2.2.2 命令限制

不同分区命令限制：

* 引导分区：命令等级6(写保护)和等级7(锁卡)不被接受；
* RPMB分区：只接受Class0、Class2和Class4类的命令。命令CMD0、CMD6、CMD8、CMD12、CMD13、CMD15以外的命令将会被认为非法；
* 通用分区：允许类别0、2、4、5、6的命令；每个分区下的每个写保护组可以单独设置写保护。因此，主机可以在每个写保护组中设置不同的写保护类型；

##### 2.2.3 配置分区

扩展CSD寄存器中PARTITIONING\_SUPPORT字段的第0位表示该设备是否支持分区特性。同一字段中的第1位表示该设备是否支持通用分区和增强用户数据区属性的增强特性属性。通用分区和增强用户数据区的属性可以通过主机在设备生命周期中只设置一次扩展CSD寄存器中的相应值来编程。

增强用户数据区起始地址(扩展CSD中的ENH\_START\_ADDR)应写入保护组对齐。对于密度高达2GB的组地址，是以字节为单位，对于密度大于2GB的，是以扇区为单位。设备将忽略写组大小以下的低比特位，并将增强用户数据区起始地址与所属的写保护组地址(以字节或扇区为单位)对齐。增强用户数据区域的地址空间与用户数据区域其余部分的地址空间是连续的(增强用户数据区域与其余用户数据区域之间没有地址间隔)。

通用分区和增强用户数据区的粒度以高容量写保护组大小为单位。当设置分区参数为时，扩展 CSD寄存器中的ERASE\_GROUP\_DEF需要被设置。如果在设置ERASE\_GROUP\_DEF位之前，CMD6将分区参数发送给设备，则从设备显示错误。

一旦对设备进行了分区，并且配置稳定后，所有的第5和6类命令都会被引用到高容量擦除组和写保护组。除了前面提到的分区参数字段外，主机还需要在模式字段的PARTITIONING\_SETTING\_COMPLETED中设置0位，这样主机就会通知设备设置过程已经成功完成。这个位设置是为了保护分区序列免受意外断电事件的影响。如果在分区过程只是部分执行之后突然断电，那么在下次上电时，设备可以检测并使之前不完整的分区过程失效(如果这个位没有设置)，从而使主机有可能重复并正确地完成分区过程。

主机应发出CMD13确保所有参数设置正确。如果任何一个分区参数不正确，设备将抛出SWITCH\_ERROR。因为在设置PARTITIONING\_SETTING\_COMPLETED位之前，设备将不知道已配置分区和用户区域的总大小，所以当主机设置PARTITIONING\_SETTING\_COMPLETED位时，如果已配置分区和用户数据区域的总大小不适合设备的可用空间，设备可能会显示SWITCH\_ERROR。在下一个上电周期后，所有设置将被清除。因此主机需要在每个分区配置寄存器字节中设置适当的值。

只有在电源周期之后设备将根据扩展CSD中的分区参数进行实际配置。在PARTITIONING\_SETTING\_COMPLETED位之后但在上电周期发生之前发出的任何命令都将会正常执行。在设置此位之前的任何先前不完整的分区配置序列将在电源周期时取消。

在完成分区配置后的电源周期后，对于不超过2GB的设备，C-SIZE值将被更改，对于大于2GB的设备，SEC\_COUNT值将被更改，以指示配置后用户数据区域的大小。与2GB相比的大小应该是配置分区之前的用户数据区域大小(例如，对于配置分区之前超过2GB的设备，SEC\_COUNT应该保持指示配置分区后的用户数据区域大小，即使大小减小到低于或等于2GB)。用户数据区域大小包括“用户”区域中“增强用户数据”区域大小。因此主机可能需要在电源周期后读取这些值来计算用户数据区的大小。配置分区后应保持访问模式。

如果主机尝试通过使用CMD6在按照配置过程上电后更改通用分区和增强用户数据区特性，设备将在CMD6响应的状态寄存器中断言SWITCH\_ERROR位，而不执行任何内部操作。分区配置参数存储在扩展CSD寄存器的一次性可编程字段中。即使还没有设置PARTITIONING\_SETTING\_COMPLETED，主机也可以通过CMD8读取，但是只有在上电之后才会执行分区。

##### 2.2.4 访问分区

每次上电后，当主机使用配置了分区的设备时，在下发读、写、擦除和写保护命令之前，必须将ERASE\_GROUP\_DEF位设置为高，因为这个位在上电后会复位。否则，这些可能无法正常工作，并且可能使存储的数据处于未知状态。

访问一个分区流程：

1. 在扩展CSD寄存器的PARTITION\_CONFIG字段中设置PARTITION\_ACCESS位，以便对其中一个分区进行寻址；
2. 发出指向所选分区的命令；
3. 恢复对用户数据区的默认访问或将访问重定向到另一个分区；

所有复位事件(CMD0或硬件复位)将默认恢复对用户数据区的访问。如果发生意外掉电，默认情况下将恢复对用户数据区的访问。当主机试图访问一个以前没有创建过的分区时，设备会在状态寄存器中设置 SWITCH\_ERROR位，而不会更改PARTITION\_ACCESS位。

#### 2.3 引导模式

在引导操作模式下，主机可以在发出CMD1之前，通过保持CMD行低或发送带有+ 0xFFFFFFFA参数的CMD0，从(MMC设备)读取引导数据。根据寄存器设置，可以从引导区或用户区读取数据。

##### 2.3.1 复位到预空闲状态

进入预空闲状态：

* 主机上电后，卡(即使一直处于非活动状态)处于多媒体卡模式且处于预空闲状态；
* GO\_PRE\_IDLE\_STATE命令(CMD0带0xF0F0F0F0参数)是软件复位命令，将卡置于预空闲状态；
* 硬件复位可用于主机复位卡，将卡移动到预空闲状态，并在复位之前设置为上电写保护的块上电期间取消写保护。当卡在睡眠状态下收到GO\_PRE\_IDLE\_STATE命令(CMD0带0xF0F0F0F0参数)或断言硬件复位信号时，卡进入到预空闲状态；

在上电，GO\_PREIDLE\_STATE命令或硬件复位断言之后，卡的输出总线驱动处于高阻抗状态，并且卡被初始化为默认的相对卡地址(0x0001) 和默认的驱动级寄存器设置。

如果在VCCQ完全上电之前RST\_n信号下降，则将VCCQ上升沿视为RST\_n信号的下降沿。在这种情况下，RST\_n信号的脉宽应该在RST\_n信号上升沿到VCCQ上电时间之间进行测量。

在上电后的卡内部初始化序列中，卡可能无法检测到RST\_n信号，因为卡可能还没有完成将扩展CSD寄存器的RST\_n\_ENABLE位加载到控制器中。然而，由于上电，卡已经开始内部初始化序列，本质上包括RST\_n信号断言的复位序列。卡可能不必再次执行重置序列，但应该在1秒内完成内部初始化序列。

##### 2.3.2 引导操作

如果在上电或复位操作(通过CMD0，参数为0xF0F0F0F0或断言硬件复位为eMMC)之后，CMD行保持LOW 74个时钟周期或更长时间，如果在发出第一个命令之前，在扩展CSD寄存器162字节位[1:0]中启用，则从属识别启动模式正在启动并开始在内部准备启动数据。可以使用EXT\_CSD179字节位[5:3]选择主机读取引导数据的分区。主机在引导过程中可以读取的数据大小可以计算128KB ×BOOT\_SIZE\_MULT (EXT\_CSD byte[226])。在CMD行变为LOW后的1秒内，从设备开始将第一个引导数据发送到DAT线上的主机。主机必须保持CMD低电平以读取所有启动数据。主机必须使用推拉模式，直到启动操作终止。

主机可以选择使用具有向后兼容接口定时的单数据速率模式，具有高速接口定时的单数据速率或双数据速率定时(如果支持)，通过在EXT\_CSD寄存器177字节位[4:3]中设置适当的值。EXT\_CSD寄存器228字节第2位告诉主机设备是否支持引导期间的高速定时。通过在EXT\_CSD寄存器177字节位[4:3]中设置为2，主机也可以选择使用双数据速率模式。EXT\_CSD寄存器228字节位1告诉主机设备在引导过程中是否支持双数据速率模式。通过在EXT\_CSD寄存器179字节第6位设置为高电平，主服务器可以选择从服务器接收引导确认，这样主服务器就可以识别从服务器在引导模式下运行。

如果启动确认被启用，从机必须在CMD变为低电平后的50ms内将确认模式“010”发送给主机。如果启动确认被禁用，从设备将不发送确认模式“0-1-0”。主机可以用命令线设置为高电平终止引导模式。如果主机在数据传输的中间将CMD设置为高电平，从设备必须在NST时钟周期(一个数据周期和结束位周期)内终止数据传输或确认模式。如果主机在连续的块之间终止引导模式，从服务器必须在NST时钟周期内释放数据线。当启用的引导数据的所有内容被发送到主机时，引导操作将被终止。启动操作完成后，从机准备进行CMD1操作，主机需要通过发送CMD1启动一个正常的MMC初始化序列。

如果在CMD1发出之前，上电后CMD行保持LOW状态少于74个时钟周期，或者主机在初始化引导模式之前发送了除CMD0以外的任何带0xFFFFFFFA参数的正常MMC命令，则从设备将不响应并且将被锁定在启动模式之外，直到下一个上电周期或硬件复位，并进入空闲状态。当设置BOOT\_PARTITION\_ENABLE位，主发送CMD1 (SEND\_OP\_COND)时，从必须进入卡识别模式并响应命令。如果从设备不支持v4.2及之前版本的引导操作方式，或者清除BOOT\_PARTITION\_ENABLE位，则从设备上电后自动进入空闲状态。

##### 2.3.4 可选引导操作

这个引导功能对于4.4标准版本的设备是必需的。遵循4.4标准的设备必须在扩展CSD寄存器228字节中显示第1位为0。

在上电或重置操作之后(CMD0的断言参数为0xF0F0F0F0或硬件复位)，如果主机在74个时钟周期后发出CMD0，参数为0xFFFFFFFA，在CMD1发出或CMD行变低之前，从机识别到引导模式正在启动并开始在内部准备启动数据。可以使用EXT\_CSD寄存器179字节位[5:3]提前选择主机将从中读取引导数据的分区。主机在引导过程中可以读取的数据大小可以计算为128KB × BOOT\_SIZE\_MULT (EXT\_CSD byte[226])。在发出参数为0xFFFFFFFA的CMD0后的1秒内，从服务器开始在DAT行上向主服务器发送第一个引导数据。主机必须使用推拉模式，直到启动操作终止。

通过在EXT\_CSD寄存器177字节位[4:3]中设置适当的值，主机可以选择使用具有向后兼容接口定时的单数据速率模式，具有高速接口定时的单数据速率模式或双数据速率定时。EXT\_CSD寄存器228字节，第2位告诉主机设备是否支持引导期间的高速定时。

主机可以通过发出CMD0 (复位)来终止引导模式。如果主站在数据传输的中间发出CMD0 (复位)，从机必须在NST时钟周期(一个数据周期和结束位周期)内终止数据传输或确认模式。如果主机在连续的块之间终止引导模式，从设备必须在NST时钟周期内释放数据线。

当启用的引导数据的所有内容被发送到主机时，引导操作将被终止。启动操作完成后，从机需要准备好CMD1操作，主机需要发送CMD1启动一个正常的MMC初始化序列。

如果在CMD1发出之前，上电后CMD处于低电平状态的时间少于74个时钟周期，或者在初始化引导模式之前，主机发送了除CMD1和CMD0以外的任何正常MMC命令，参数为0xFFFFFFFA，则从设备不响应，并且将被锁定在启动模式之外，直到下一个上电周期并进入空闲状态。

当设置BOOT\_PARTITION\_ENABLE位，主发送CMD1 (SEND\_OP\_COND)时，从必须进入卡识别模式并响应命令。

如果从设备不支持4.2及之前版本的引导操作方式，或者清除BOOT\_PARTITION \_ENABLE位，上电后从设备会自动进入空闲状态。

##### 2.3.5 访问引导分区

当从机进入传输状态后，主机发送CMD6 (SWITCH)来设置EXT\_CSD寄存器中的PARTITION\_ACCESS位。之后，主机可以使用普通的MMC命令访问引导分区。

主机可以使用CMD24 (WRITE\_BLOCK)或CMD25 (WRITE\_MULTIPLE\_BLOCK)在数据线上编程引导数据，从支持寻址模式，即字节寻址或扇区寻址。如果主机使用CMD25 (WRITE\_MULTIPLE\_BLOCK)并且写操作超过了所选的分区边界，那么从设备将报告一个地址超出范围错误。分区边界内的数据将被写入所选的引导分区。

主机可以使用CMD17 (READ\_SINGLE\_BLOCK)或CMD18 (READ\_MULTIPLE\_BLOCK)读取数据线上的引导数据，从支持寻址模式，即字节寻址或扇区寻址。如果主机使用CMD18 (READ\_MULTIPLE\_BLOCK)，然后读取超出所选分区边界，则从设备将报告地址超出范围错误。

在完成对引导分区的数据访问之后，应该清除PARTITION\_ACCESS位。然后，应该设置EXT\_CSD寄存器中的非易失性BOOT\_PARTITION\_ENABLE位，以指示启用哪个分区进行引导。这将允许从服务器在引导操作期间从引导分区读取数据。

主机也可以通过使用普通命令清除EXT\_CSD寄存器中的PARTITION\_ACCESS位到0来访问用户区。如果用户区被锁定并启动，则在启动操作模式下数据不会发送到主机。但是，如果用户区被锁定并且两个分区中的一个被启用，那么数据将在引导操作模式期间发送到主机。

##### 2.3.6 引导模式总线配置

在引导操作期间，总线宽度可以通过扩展CSD寄存器177字节位[0:1]中的非易失配置位来配置。寄存器177字节中的第2位决定从机在启动操作后是否返回到x1总线宽度和单数据速率模式，并具有向后兼容的定时，或者在正常操作期间是否保持在配置的引导总线宽度。位[4:3]寄存器字节决定数据线是否配置为单数据速率使用向后兼容或高速定时或双数据速率模式在启动操作。如果启动操作不执行，从机将初始化在正常的x1总线宽度，单数据速率操作和向后兼容的时间，不管寄存器设置。

##### 2.3.7 引导分区写保护

为了允许主机保护引导区域不被擦除或写入，多媒体卡应该支持引导区域的两级写保护。

* 通过设置B\_PERM\_WP\_EN引导区可以永久写保护；
* 启动区可以通过设置B\_PWR\_WP\_EN (EXT\_CSD[173] bit 0)上电写保护；

主机必须注意以下几点：

* 在设备掉电导致重启或硬件复位后，如果需要，必须重新应用上电写保护，因为启动区会恢复到未保护状态；
* 开启开机保护后，启动区仍然可以进行永久写保护。因此，如果不需要永久写保护，则应该设置B\_PERM\_WP\_DIS (EXT\_CSD[173] bit 4)，以防止恶意或无意地设置永久保护；

通过设置B\_PERM\_WP\_DIS (EXT\_CSD[173]位4)和B\_PWR\_WP\_DIS (EXT\_CSD[173]位6)，主机可以关闭引导区永久写保护和开机写保护。如果不需要设置引导区保护，建议设置这两个位，以确保引导区不会被无意或恶意保护。

#### 2.4 卡识别模式

在卡识别模式下，主机重置卡，验证操作电压范围和访问模式，识别卡并在总线上为卡分配一个相对卡地址(RCA)。卡识别模式下的所有数据通信仅使用命令线(CMD)。

##### 2.4.1 卡复位

收到GO\_IDLE\_STATE命令(CMD0参数为0x00000000)后，卡进入空闲状态。

进入空闲状态：

* 在完成启动操作后；
* 在接收CMD1处于预空闲状态后；
* 如果设备未启动，则在上电后在此状态下，卡的输出总线驱动处于高阻抗状态，并且卡被初始化为默认相对卡地址(0x0001)和默认驱动级寄存器设置；

参数为0x00000000的CMD0在除非活跃外的所有状态下都有效。当处于非活动状态时，卡不接受参数为0x00000000的CMD0。出于向后兼容的原因，如果设备接收到的CMD0参数不是0xFFFFFFFA或0xF0F0F0F0，则设备将其视为卡复位命令，并进入空闲状态。带0xFFFFFFFA参数的CMD0是一个处于预启动状态的启动命令，但是如果主机在除非活跃状态和预启动状态外的任何状态下发出该命令，则设备将其视为复位命令并进入空闲状态。

##### 2.4.2 检测工作电压

SEND\_OP\_COND (CMD1)命令旨在为多媒体卡主机提供机制，以识别和拒绝与主机所需的VDD范围不匹配的卡。这是通过主机发送所需的VDD电压窗口作为该命令的操作数来完成的。如果卡不能在指定范围内执行数据传输，必须从进一步的总线操作中丢弃本身，并进入非活动状态。否则，卡将响应发送回其VDD范围，eMMC设备将根据容量以固定模式响应0x00FF8080或0x40FF8080。

对于eMMC器件，CMD1中的电压范围不再有效。无论主机指示的电压范围 如何，如果设备繁忙，eMMC设备将以固定模式响应0x00FF 8080 (容量小于或等于2GB)或0x40FF 8080(容量大于2GB)并且不会进入非活跃状态。

如果主机要在1.70V ~ 1.95V范围内工作双电压多媒体卡，建议主机先验证工作电压在2.7V ~ 3.6V范围内，然后将卡完全下电，再将卡重新上电到1.70V ~ 1.95V范围内工作。最初使用2.7V到3.6V的范围，这是常见的高压和双电压多媒体卡，将允许可靠的筛选主机和卡电压不兼容。如果使用VDD 小于 2.0V建立通信，高压卡可能无法正常工作。如果使用1.95 ~ 2.7V，双电压卡可能会失效。

##### 2.4.3 访问模式检测

SEND\_OP\_COND (CMD1)命令和OCR寄存器也包括两个位，用于指示支持的内存访问模式。CMD1命令参数中特别设置的位指示主机能够处理扇区类型寻址的内存。在OCR寄存器中相应地设置的位表示该卡需要使用寻址的扇区类型。OCR寄存器的这些特定位仅在CMD1(卡进入就绪状态)的卡的最后响应中有效。这种双向握手是必要的。

在多卡系统中，字节访问模式卡(小于2GB)以这样的方式阻塞OCR响应，使得扇区访问模式卡(大于2GB)在初始化期间不一定被识别为扇区访问模式卡。因此，这需要通过从EXT\_CSD寄存器中读取SEC\_COUNT信息来重新确认。

##### 2.4.4 从忙进入就绪状态

CMD1响应中的忙位可以被卡用来告诉主机仍然在上电复位过程中工作，并且还没有准备好进行通信。在这种情况下，主机必须重复CMD1，直到忙位被清除。在初始化过程中，主机不允许更改工作电压范围或访问模式设置。如果的运行条件真的发生了变化，主机必须重置卡(使用参数为0x00000000的CMD0)并重新启动初始化过程。但是，对于访问已经处于非活动状态的卡，必须通过关闭电源并重新打开电源来进行硬复位。命令GO\_INACTIVE\_STATE (CMD15)可以用来将地址卡发送非活跃状态。当主机明确地想要停用一张卡时，使用这个命令(例如，主机正在将VDD更改为已知该卡不支持的范围)。

##### 2.4.5 卡识别过程

在总线被激活后，主机将请求卡发送其有效的操作条件(CMD1)。对CMD1的响应是对系统中所有卡的条件限制进行与操作。不兼容的卡被发送到非活动状态。然后主机发出广播命令ALL\_SEND\_CID (CMD2)，向所有卡请求其唯一的卡标识(CID)号。所有未识别卡(即那些处于就绪状态的卡)同时开始按顺序发送其CID号码，同时按位监控其传出的比特流。那些卡的输出CID位在任何一个位周期内都与命令行上的相应位不匹配，因此立即停止发送CID，必须等待下一个识别周期(保持在就绪状态)。由于每张卡的CID号码都是唯一的，因此应该只有一张卡成功地将其完整的CID号码发送到主机。然后这张卡进入身份识别状态。此后，主机发出CMD3 (SET\_RELATIVE\_ADDR)来为该卡分配一个相对卡地址(RCA)，该地址比CID短，并将用于在未来的数据传输模式(通常具有比fOD更高的时钟速率)中对该卡进行寻址。一旦收到RCA，卡的状态就会变为就绪状态，并且卡不会对进一步的识别周期做出反应。此外，卡切换其输出驱动器从开漏到推挽。

主机只要收到对其识别命令(CMD2)的响应(CID)，就重复识别过程，即CMD2和CMD3的循环。如果没有更多的卡响应此命令，则表示所有的卡已被识别。识别过程完成的超时条件是在发送CMD2后超过NID时钟周期没有起始位

#### 2.5 中断模式

多媒体卡系统上的中断模式允许多媒体卡主机同时向从设备授予传输许可。这种模式减少了主机的轮询负载，从而减少了系统的功耗，同时保持主机对卡请求服务的足够响应。

中断模式：

* 主机在发出GO\_IRQ\_STATE (CMD40)命令之前，必须确保卡处于备用状态。当等待卡的中断响应时，主机必须保持时钟信号活动。时钟速率可能会根据需要的响应时间而改变；
* 主机使用GO\_IRQ\_STATE (CMD40)命令将卡设置为中断模式；
* 处于等待中断状态的卡正在等待内部中断触发事件。一旦事件发生，卡片就开始向主机发送响应。此响应以开漏模式发送；
* 当等待内部中断事件时，卡也在等待命令行上的起始位。一旦检测到启动位，卡将中止中断模式并切换到待机状态；
* 无论在CMD40响应期间赢得或失去总线控制，卡都切换到备用状态(与CMD2相反)；
* 在主机接收到中断响应后，主机返回到标准数据通信过程；
* 如果主机希望在收到中断响应之前终止中断模式，可以使用预留的RCA地址0x0000自行生成CMD40响应；这将使卡从等待中断状态回到备用状态。现在主机可以恢复标准的通信过程；

#### 2.6 数据传输模式

当卡处于待机状态时，通过CMD和DAT线的通信将以推拉模式进行。在主机知道CSD寄存器的内容之前，fPP时钟速率必须保持fOD。主机发出SEND\_CSD (CMD9)来获取卡特定的数据(CSD寄存器)，例如块长度，卡存储容量，最大时钟速率等。

广播命令SET\_DSR (CMD4)配置卡的驱动阶段。根据应用总线布局(长度)和数据传输频率对DSR寄存器进行编程。时钟速率也从fOD切换到fPP。

当卡处于待机状态时，使用CMD7通过在参数中包含卡的相对地址来选择卡并将其置于转移状态。如果卡之前被选中并且处于传输状态，则与主机的连接将被释放，并且当CMD7在参数中使用任何不等于卡自己的相对地址的地址取消选择时，将进入备用状态。当CMD7获得预留的相对卡地址0x0000时，卡将恢复到待机状态。当卡处于传输状态时，卡忽略接收带有卡自身相对地址的CMD7，可能被视为非法命令。在为卡分配RCA之后，将不会响应识别命令CMD1、CMD2或CMD3。

当卡处于断开状态时，CMD7用于选择卡并通过在参数中包含卡的相对地址将其置于编程状态。如果卡之前被选中并且处于编程状态，与主机的连接将被释放，并且当CMD7在参数中使用任何不等于卡自己的相对地址的地址取消选择时，将进入断开状态。当卡处于编程状态时，接收带有卡自身相对地址的CMD7会被卡忽略，并可能被视为非法命令

##### 2.6.1 命令集合和设置

卡在给定的命令集中运行，默认情况下，在电源周期后，由CMD0复位，参数为0x00000000或在启动操作后；这是多媒体卡标准命令集，使用单个数据线DAT0。主机可以通过发出SWITCH命令(CMD6)来更改活动命令集，并使用选择命令集访问模式。支持的命令集以及当前选择的命令集在EXT\_CSD寄存器中定义。EXT\_CSD寄存器分为属性段和模式段。属性段包含有关卡片功能的信息。模式段部分反映了当前卡的选定模式。主机通过发出SEND\_EXT\_CSD命令来读取EXT\_CSD寄存器。卡将EXT\_CSD寄存器作为一个512字节长的数据块发送。任何保留的或只写的字段读为低电平。主机可以通过发出SWITCH命令和设置一种访问模式来写入EXT\_CSD寄存器的模式段。

SWITCH命令既可以用于写入EXT\_CSD寄存器，也可以用于更改命令集。如果使用SWITCH命令更改命令集，则忽略索引和值字段，并且不写入EXT\_CSD。如果使用SWITCH命令写EXT\_CSD寄存器，命令集字段将被忽略，命令集保持不变。

SWITCH命令响应类型为R1b，因此，主机应该在忙信号解除后，使用SEND\_STATUS命令读取卡状态，以检查SWITCH操作的结果。

##### 2.6.2 选择高速模式

主机确认卡符合4.0及以上版本标准后，需先使能卡内的高速模式定时，再将时钟频率修改为高于20MHz的频率。为了使主机切换到更高的时钟频率，必须使能高速接口定时。主机使用SWITCH命令将0x01写入HS\_TIMING字节，在EXT\_CSD寄存器的模式段中。如果主机尝试写无效值，则不修改HS\_TIMING字节，不启用高速接口定时，并且设置SWITCH\_ERROR位。

##### 2.6.3 电源选择

主机验证卡符合4.0或更高版本的标准后，可能会更改卡的功率类型。上电或软件复位后，卡的电源类别为0类，这是卡类型(高压或双电压卡)的默认最小电流消耗类别。在EXT\_CSD寄存器中的PWR\_CL\_ff\_vvv字节反映了卡在支持的时钟频率(26MHZ或52MHz)下的4位总线和8位总线的功耗水平。主机使用SEND\_EXT\_CSD命令读取此信息，并确定是否允许卡使用更高的功率类。如果需要更改电源类，主机使用SWITCH命令在EXT\_CSD寄存器的模式段中写入POWER\_CLASS字节。

##### 2.6.4 总线测试

通过在单数据速率模式下发出CMD19和CMD14命令，主机可以检测总线上的功能引脚。在双数据速率模式下，CMD19和CMD14被认为是非法命令。在第一个步骤中，主机将CMD19发送到卡，然后在每个选定的数据线上发送特定的数据模式。每条数据线要发送的数据模式在下表中定义。作为第二步，主机发送CMD14请求卡片发送回反转的数据模式。使用主机发送的数据模式和使用卡返回的反向模式，可以检测总线上的功能引脚。

卡片忽略数据模式的前两位以外的所有内容。因此，卡缓冲区大小不限制数据模式的最大长度。数据模式的最小长度是两个字节，其中每条数据线的前两位被卡反向发送回来。主机发送的数据模式可以可选地包括一个CRC16校验和。

该卡检测到DAT0上的起始位，并相应地同步其所有数据输入的读取。主机忽略反向数据模式的前两位以外的所有内容。反向数据模式的长度是8个字节，并且总是使用卡的所有DAT行发送卡发送的反向数据模式可以选择性地包括一个CRC16校验和，被主机忽略。

##### 2.6.5 总线位宽选择

在主机验证了总线上的功能引脚之后，应该使用SWITCH命令相应地更改总线宽度配置。总线宽度配置通过写入EXT\_CSD寄存器的模式段中的BUS\_WIDTH字节来改变(使用SWITCH命令这样做)。上电或软件复位后，BUS\_WIDTH字节的内容为0x00。如果主机试图写入无效值，则不会更改BUS\_WIDTH字节，并设置SWITCH\_ERROR位。这个寄存器只能写。

##### 2.6.6 数据读取

##### 2.6.7 写入数据

##### 2.6.8 擦除

多媒体卡除了作为写操作的一部分由卡片执行的隐式擦除之外，还提供了主机显式擦除功能。多媒体卡的可擦除单元是擦除组；擦除组是用写块来衡量的，写块是卡的基本可写单位。擦除组的大小是一个特定于卡的参数，当ERASE\_GROUP\_DEF被禁用时在CSD中定义，当ERASE\_GROUP\_DEF被启用时在EXT\_CSD中定义。显式擦除的内存范围的内容应根据不同的内存技术为0或1。这个值在EXT\_CSD中定义

主机可以擦除连续的擦除组。开始擦除过程有三个步骤。首先，主机使用ERASE\_GROUP\_START (CMD35)命令定义范围的起始地址，然后使用ERASE\_GROUP\_END (CMD36)命令定义范围的最后一个地址，最后通过发出擦除 (CMD38)命令启动擦除过程，参数位设置为零。擦除命令中的地址字段是擦除组地址，密度为2GB时以字节为单位，密度大于2GB时以扇区为单位。该卡将忽略所有低于擦除组p大小的LSB，有效地将地址四舍五入到擦除组边界。

如果从定义的擦除序列中接收到擦除命令(CMD35、CMD36、CMD38)，则卡将在状态寄存器中设置ERASE\_SEQ\_ERROR位，并重置整个序列。如果主机提供超出范围的地址作为CMD35或CMD36的参数，卡将拒绝该命令，以ADDRESS\_OUT\_OF\_RANGE位设置响应并重置整个擦除序列。

如果收到非擦除命令(CMD35、CMD36、CMD38或CMD13均非)，则卡响应设置ERASE\_RESET位，重置擦除顺序，执行最后一个命令。未指定所选卡地址的命令不会中止擦除序列。

如果擦除范围包括写保护块，则保留写保护块，只擦除非保护块。应该设置状态寄存器中的WP\_ERASE\_SKIP状态位。

对于块写入，卡将通过保持DAT0低来指示擦除正在进行中。实际擦除时间可能相当长，主机可能发出CMD7来取消卡的选择。

当执行擦除命令时，主机应该注意到一个擦除组包含多个写块，每个写块可以包含不同的信息片段。当擦除被执行时，将应用于擦除组内的所有写块。在主机执行擦除命令之前，应该确保各个写块中的信息不再需要。因此，为了避免意外删除有效数据，最好使用擦除命令擦除整个设备或分区。如果主机只希望清除单个写块，则使用裁剪命令可能更合适。

##### 2.6.9 安全擦除

除了标准的擦除命令之外，还有一个可选的安全擦除命令。安全擦除命令不同于基本的擦除命令，要求卡片在发出命令时对内存阵列执行擦除操作，并且要求卡片和主机等待操作完成后再进行下一个卡片操作。此外，安全擦除命令要求卡对擦除组以及这些擦除组中标识为要擦除的项目的任何副本执行安全清除操作。

此命令允许具有高安全性要求的应用程序请求设备执行安全操作，同时接受可能的擦除时间性能影响。

安全擦除命令的执行方式与擦除命令相同，只是擦除 (CMD38)命令执行时将参数位31设置为1，其他参数位设置为0。请谨慎执行安全擦除命令，避免意外数据丢失。重置卡(使用CMD0, CMD15或硬件复位eMMC)或电源故障将终止任何挂起或活动的安全擦除命令。这可能使操作中涉及的数据处于未知状态。

##### 2.6.10 安全裁剪

#### 2.7 时钟控制

主机可以使用多媒体卡总线时钟信号使卡进入节能模式，或控制总线上的数据流(以避免欠运行或过运行的情况)。允许主机降低时钟频率或关闭主机。

时钟控制限制：

* 总线频率可以随时更改(在卡定义的最大数据传输频率和标准文件定义的识别频率的限制下)；
* 很明显，时钟必须运行才能使卡输出数据或响应令牌。在最后一次多媒体卡总线事务之后，主机需要在关闭时钟之前为卡提供8个时钟周期以完成操作；

#### 2.8 错误检测

所有命令都由CRC(循环冗余校验)位保护。如果地址卡的CRC检查失败，则该卡不响应，命令不执行;卡不改变状态，状态寄存器中设置COM\_CRC\_ERROR位。同样，如果收到非法命令，卡将不改变其状态，不响应，并在状态寄存器中设置ILLEGAL\_COMMAND错误位。

非法命令：

* 属于卡不支持的类的命令(例如只读卡中的写命令)；
* 不允许在当前状态执行命令(例如CMD2在传输状态)；
* 未定义的命令(例如CMD44)。

#### 2.9 命令

命令分类：

* 无响应的广播命令(bc)；
* 带响应的广播命令(bcr)；
* 寻址(点对点)命令(ac)，在数据线上无数据传输；
* 寻址(点对点)数据传输命令(adtc)，在数据线上具有数据传输，所有命令和响应都通过多媒体卡总线的CMD线发送；

命令总是以起始位(总是'0 )开始，后面跟着指示传输方向的位。接下来的6位表示命令的索引，该值被解释为一个二进制编码数(介于0和63之间)。一些命令需要一个参数(例如地址)，是由32位编码的。所有的命令都受到CRC的保护。每个命令码字以结束位(总是'1 )结束。

多媒体卡系统的命令集分为几类。每个类支持卡功能的一个子集。第0类是强制性的，所有卡都必须支持。其他类要么是特定卡类型的强制类，要么是可选类。通过使用不同的类，可以选择几种配置(例如块可写卡或流可读卡)。支持的卡命令类(CCC)被编码为每个卡的特定卡数据(CSD)寄存器中的参数，为主机提供如何访问卡的信息。

#### 2.10 回应

所有响应都通过命令行CMD发送。响应传输总是从响应码字对应的位串的左边位开始。代码长度取决于响应类型。响应总是以一个起始位(总是0 )开始，后面跟着指示传输方向的位。除了R3类型(见下文)之外，所有响应都受CRC保护。每个命令码字以结束位(总是' 1 ')结束。

响应分类：

* R1：正常响应命令，码长48位。第45到40表示要响应的命令的索引，该值被解释为二进制编码数(介于0和63之间)。卡的状态用32位编码；
* R1b与R1相同，在数据线DAT0上传输一个可选的忙信号。根据卡在接收命令之前的状态，在接收到这些命令后，卡可能会变得忙碌；
* R2：CID以及CSD寄存器；码长136位。CID寄存器的内容作为对CMD2和CMD10命令的响应发送。CSD的内容作为对CMD9的响应发送。只有比特[127…]CID和CSD寄存器的保留位[0]被传输，这两个寄存器的保留位[0]为响应的结束位；
* R3：OCR寄存器；码长48位。OCR寄存器的内容作为响应发送给CMD1；
* R4：快速IO；码长48位。参数字段包含地址卡的RCA、要读出或写入的寄存器地址及其内容。如果操作成功，则设置参数中的状态位；
* R5：中断请求；代码长度48位。如果响应是由主机生成的，则参数中的RCA字段为0x0；

#### 2.11 卡状态

响应格式R1包含一个名为卡状态的32位字段。这个字段用于传输卡的状态信息。

两个不同的属性与每个卡状态位相关联：

* 位类型：定义了错误位和状态位两种类型的卡状态位。错误位表示卡检测到的错误情况。一旦发送响应(报告错误)，这些位就会被清除；状态位。这些位仅作为信息字段，不改变被响应的命令的执行。这些位是持久的，根据卡状态进行设置和清除。
* 错误位的检测方式：卡在命令解释和验证阶段(响应模式)或命令执行阶段(执行模式)期间检测异常。响应模式异常将在引发异常的命令的响应中报告。执行模式异常会在终止操作的STOP\_TRANSMISSION命令的响应中报告，或者在执行操作时或操作完成后发出的GET\_STATUS命令的响应中报告；

#### 2.12 内存分区

进出多媒体卡的数据传输的基本单位是一个字节。所有需要块大小的数据传输操作总是将块长度定义为字节的整数倍。一些特殊功能需要其他分区粒度。

块是与面向块的读写命令相关的单元。大小是主机发送一个块命令时要传输的字节数。块的大小要么是可编程的，要么是固定的。有关允许的块大小和可编程性的信息存储在CSD中。

对于读写卡，定义了特殊的擦除和写保护命令。可擦除单元的粒度为擦除组，可寻址用于擦除的连续写块的最小数量。擦除组的大小是卡特定的，当ERASE\_GROUP\_DEF被禁用时存储在CSD中，当ERASE\_GROUP\_DEF被启用时存储在EXT\_CSD中。写保护单元的粒度为写保护组，可以单独写保护的最小单元。其大小以擦除组为单位定义。当禁用ERASE\_GROUP\_DEF时，存储在CSD中；当启用ERASE\_GROUP\_DEF时，存储在EXT\_CSD中。

### 3 错误保护

CRC旨在保护多媒体卡命令、响应和数据传输，防止多媒体总线上的传输错误。为每个命令生成一个CRC，并检查CMD线上的每个响应。对于数据块，每个传输块生成一个CRC。

#### 3.1 错误校正码

为了检测卡上的数据缺陷，主机可以在有效载荷数据中包括错误纠正码。对于无错误设备，不需要此功能。通过在卡外实现纠错，可以实现最佳的硬件共享。另一方面，系统中的代码种类必须限制，否则将需要可编程ECC控制器，这超出了多媒体卡适配器。如果多媒体卡需要外部纠错(外部意味着卡的外部)，则必须在多媒体卡主机中实现ECC算法。CSD寄存器中的DEFAULT\_ECC字段定义了该卡的推荐ECC算法。

#### 3.2 CRC校验

CRC旨在保护多媒体卡命令、响应和数据传输，防止多媒体卡总线上的传输错误。为每个命令生成一个CRC，并检查CMD线上的每个响应。对于数据块，每传输一个数据块，每条数据线生成一个CRC。生成并检查CRC。