



MMC/SD卡

multimedia card (MMC)/secure digital (SD) card controller

广州创龙电子科技有限公司

Guangzhou Tronlong Electronic
Technology Co., Ltd



SD/MMC卡控制器 简介

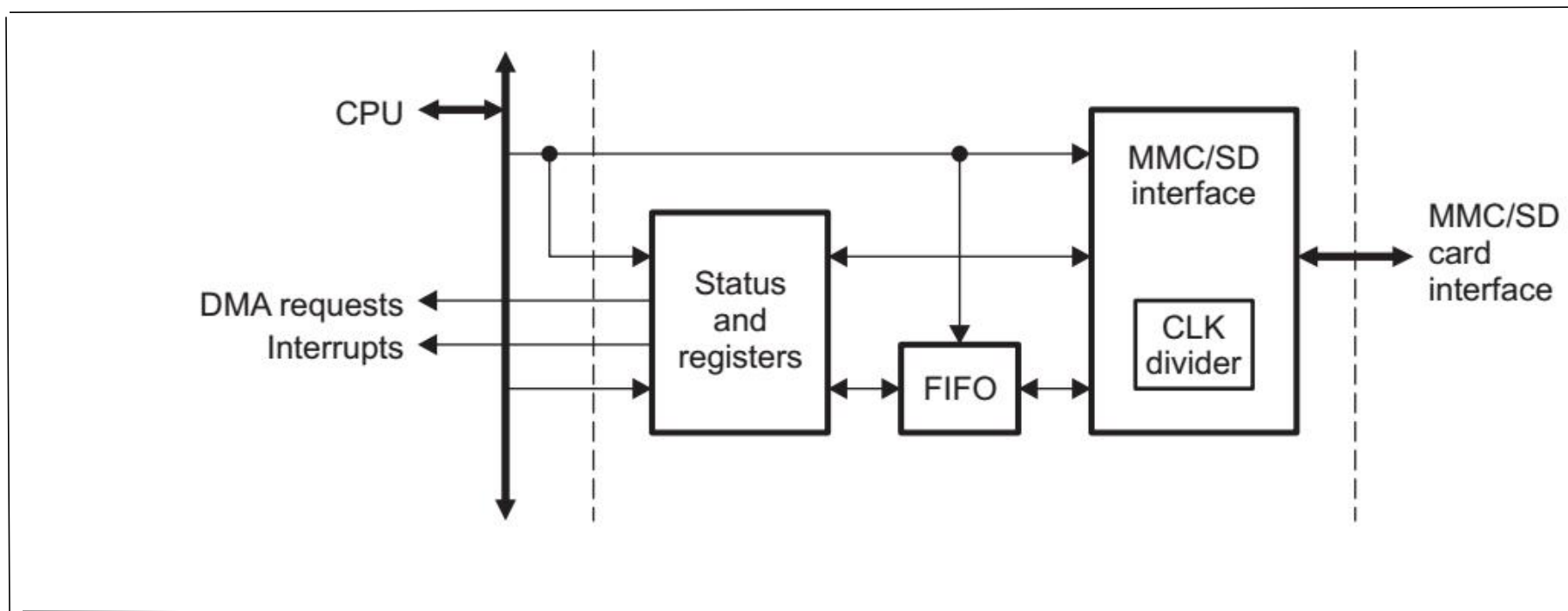
- 应用程序需要用MMC/SD卡来提供可移动数据的存储。该外设提供一个接口用来控制MMC卡和SD卡。
- 他们之间的通信协议是基于MMC/SD协议。支持MMC和SD接口以及SDIO协议。
- 有可读写的512位的FIFO以降低系统开销。
- 可以使用增强型直接内存访问（EDMA）来跟从机通信
- 支持MMC4.0、SD物理层规范1.1、SDIO2.0工业标准。不支持SPI模式。



SD/MMC控制器原理框图

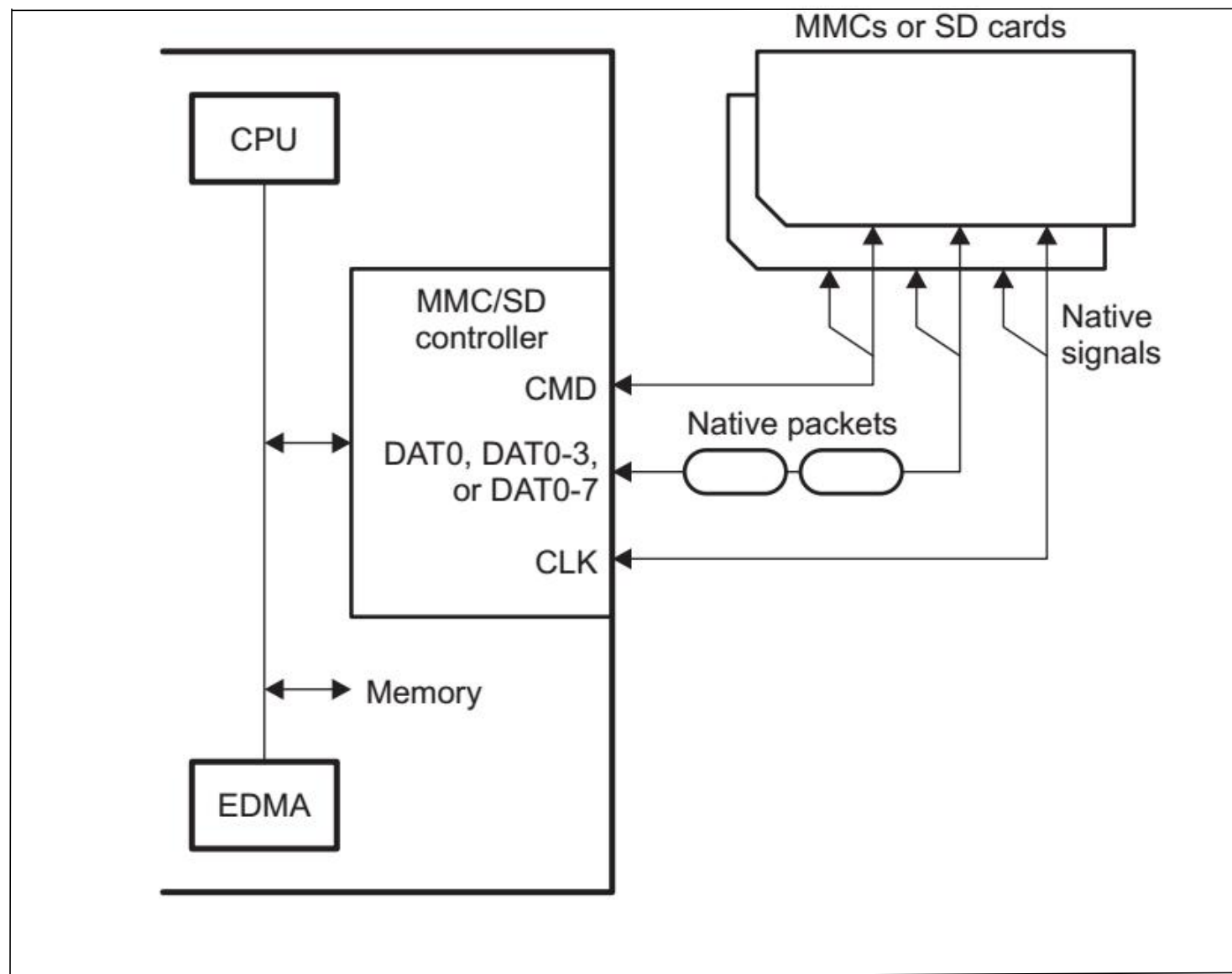
MMC/SD控制有两种传输方式：支持单块和多块读写。

1.通过CPU与卡进行通信。2.通过EDMA与卡进行通信。





MMC/SD控制器原理框图





MMC/SD控制器引脚

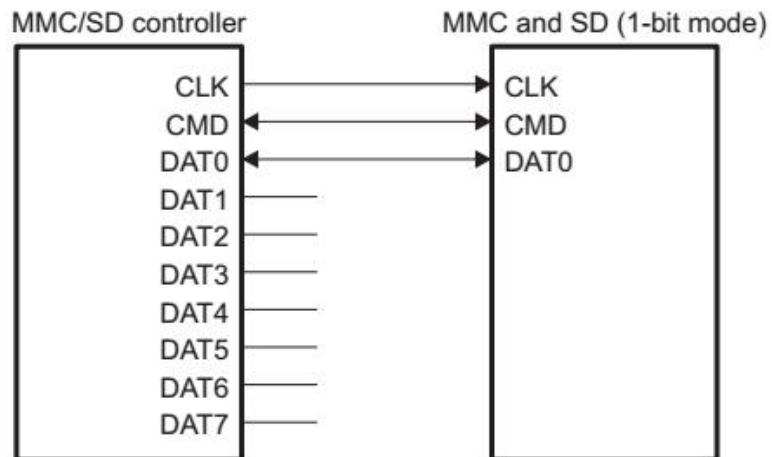
控制器相关引脚定义：
通过通信设备的不同，可以配置成MMC模式或者SD模式。

管脚	类型	描述
CMD	I/O	主机发送命令，从机响应命令
CLK	O	控制器通过该引脚给存储卡提供时钟
DAT0	I/O	MMC/SD卡一数据线制
DTA0-3	I/O	MMC/SD卡三数据线制
DTA0-7	I/O	MMC卡八数据线制

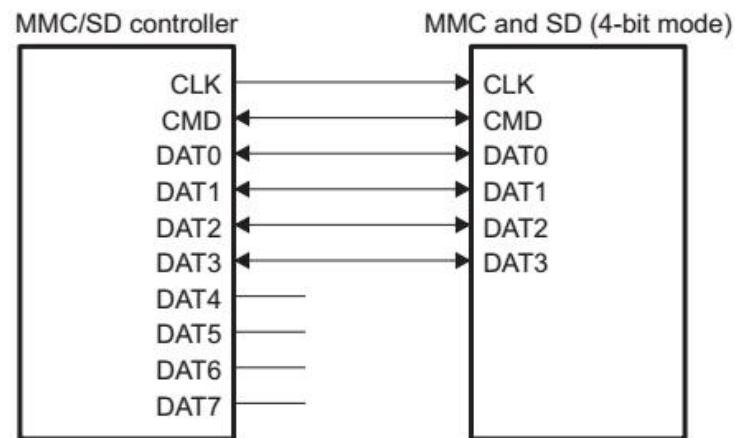


MMC/SD控制器引脚

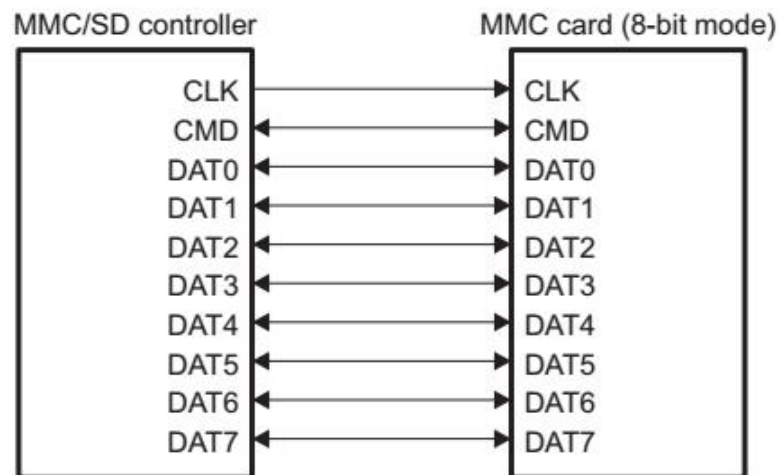
MMC/SD configuration



MMC/SD configuration



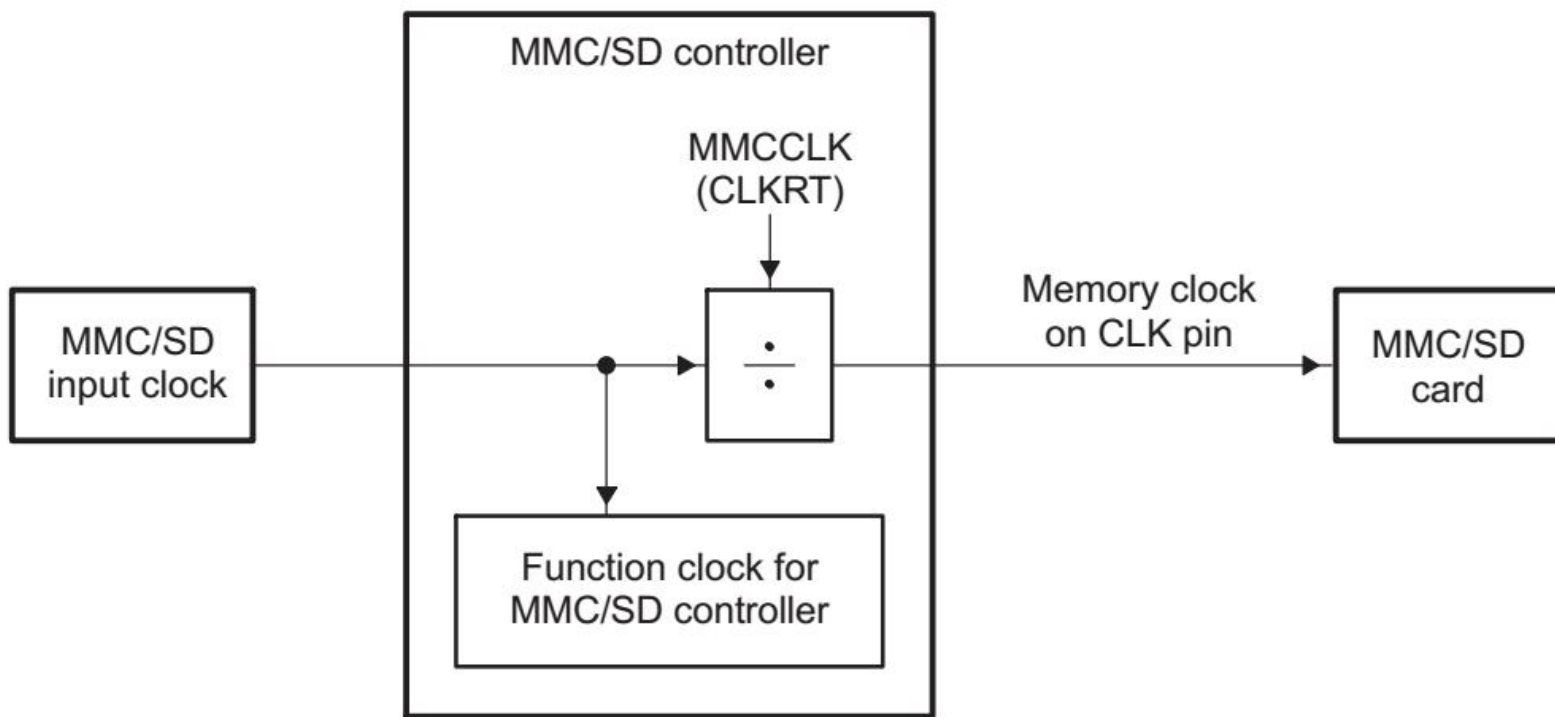
MMC configuration





MMC/SD 时钟

控制器有两个时钟，分别是功能时钟和存储器时钟。存储器时钟由功能时钟分频而来，输出到CLK引脚，是双方通信的时钟。而功能时钟决定了控制器的工作频率。

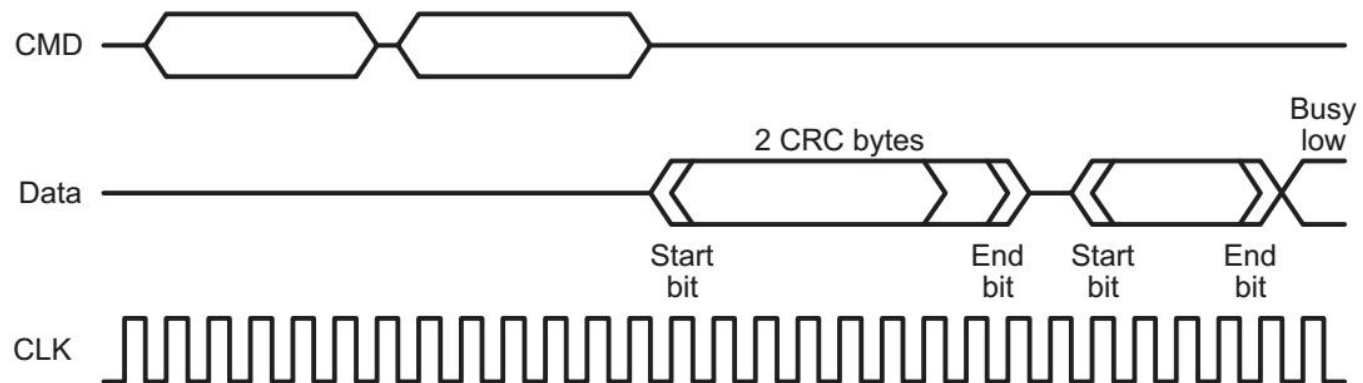




MMC/SD时序

MMC/SD卡写时序:

序列部分	描述
WR CMD	写命令: 一个6-byte WRITE_BLOCK 命令表示是从CPU发送到卡中。
CMD RSP	命令响应: 此卡在模式R1并识别CPU的WRITE_BLOCK下发送一个6-byte响应
DAT BLK	数据模块: 该CPU向此卡写入一个数据模块。 数据内容在开始位之前同时紧随两个CRC位和一个结束位之后。
CRC STAT	CRC状态: 此卡发送一位CRC状态信息, 此位指示由于CRC的错误, 从卡向CPU发送的数据是否已经被接收或者被拒绝。CRC状态信息是在开始为之前一位和跟在结束位之后一位。
BSY	BUSY位: CRC状态信息是紧随在一个低忙位的连续流, 直到所有的数据已经被编程到在卡上的flash内存器中。





MMC/SD时序

一个成功的写协议序列应该执行以下步骤：

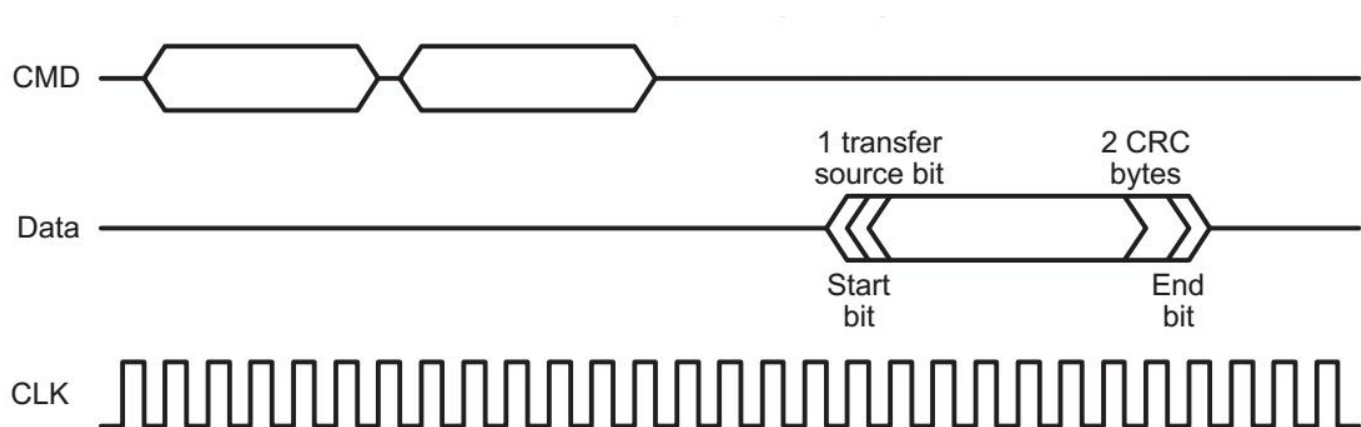
- MMC / SD控制器请求查询CSD内容。
- 存储卡收到命令并发送CSD寄存器的内容作为其响应信号。
- 如果需要的块长度，即WRITE_BL_LEN的值，与由响应确定的默认值不同，MMC / SD控制器会发出块长度命令。
- 存储卡收到此命令并对该命令做出响应。
- MMC / SD控制器请求存储卡把状态从待机更改为转移。存储卡接受命令并响应。
- MMC / SD控制器向存储卡发送写命令。存储卡接受命令并做出响应。
- MMC / SD控制器将数据块发送到存储卡。
- 存储卡向MMC / SD控制器发出CRC状态。
- 存储卡发送一个低电平的BUSY信号，直到所有的数据被写入到存储卡里面的闪存。



MMC/SD时序

MMC/SD卡读序列

序列部分	描述
RD CMD	读命令: 一个6-位READ_SINGLE_BLOCK命令代表从CPU到卡。
CMD RSP	命令响应: 此卡向识别 READ_SINGLE_BLOCK命令类型的CPU发送响应模式R1。
DAT BLK	数据模块:此卡向CPU发送一个数据块。数据内容在开始位之前同时紧随两个CRC位和一个结束位之后。





MMC/SD时序

一个成功的读协议序列应该执行以下步骤：

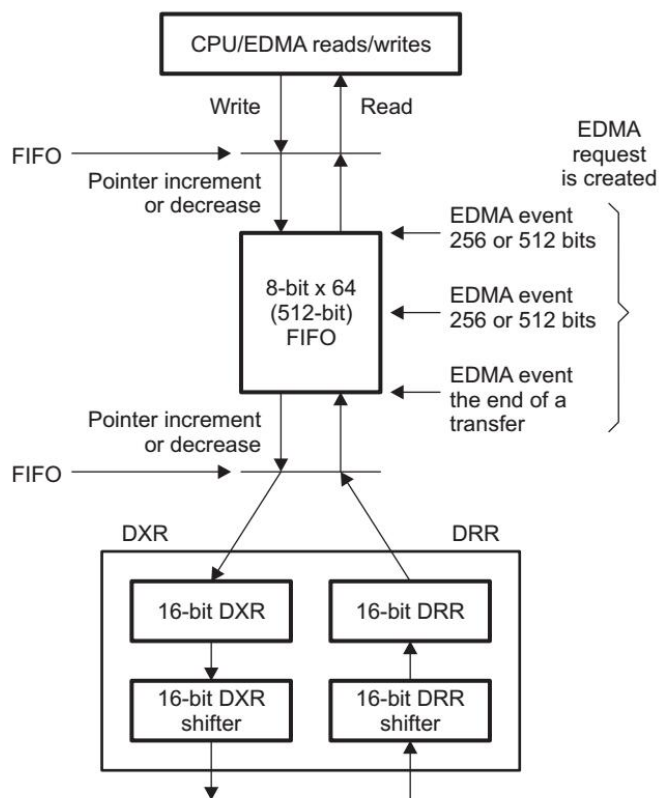
- MMC/ SD控制器请求CSD内容。
- 该卡接收命令，并发送CSD寄存器作为其响应的内容。
- 如果所需的块长度， READ_BL_LEN值，不同于从响应中得到的默认值， MMC / SD控制器发送块长度命令。
- 该卡接收该命令，并发送响应命令。
- MMC / SD控制器请求存储卡把状态从待机更改为转移。存储卡接受命令并响应。
- MMC / SD控制器发送一个读命令到卡上。
- 该卡驱动响应该命令。
- 该卡发送一个数据块到CPU。



MMC/SD的FIFO

MMC / SD控制器包含一个512位的FIFO。有8位×64个分区。FIFO被用于从存储卡读数据和写数据到存储卡。FIFO作为存储通过CPU或EDMA暂时从/向卡传送数据的临时位置（缓冲）。FIFO包括形成EDMA事件的逻辑和基于在FIFO中的数据量以及可编程的接收/发送字节数的中断的逻辑。

➤ 右图为FIFO的操作图标表



数据发送

- Step 1: 设置FIFO复位
- Step 2: 设置FIFO方向
- Step 3: EDMA驱动的事务
- Step 4: CPU驱动的事务:
通过写MMCDXR（仅第一次）或每次发送256或512位数据到FIFO，就可以填满FIFO产生DXR DYINT中断
- Step 5: EDMA发送XMIT数据
- Step 6: 如果DXR准备好有效，FIFO -> 16位DXR

数据接收

- Step 1: 设置FIFO复位
- Step 2: 设置FIFO方向
- Step 3: 如果DRR准备好有效，16位DRR -> FIFO
- Step 4: 当FIFO一个一个地接收
- Step 5: 到256或512位的数据时发生DRRDYINT中断
- Step 6: EDMA读取接收数据



MMC/SD的FIFO

在读存储卡操作期间 **FIFO**操作:

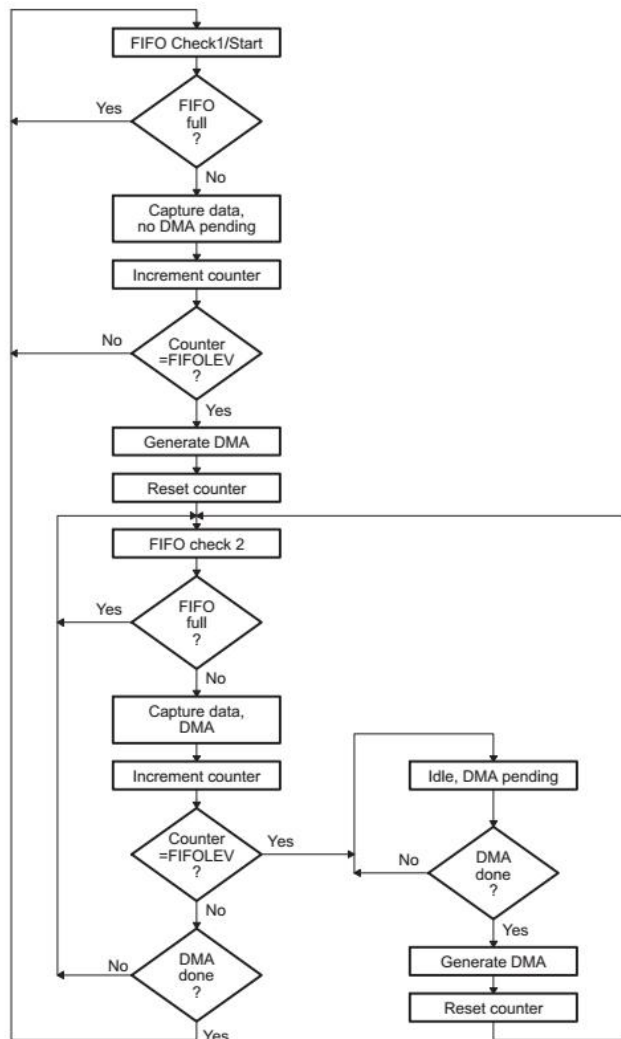
EDMA读: FIFO控制器负责从存储卡读取数据和发起EDMA读取事件的活动的管理。每发发出一次EDMA读取事件，一个EDMA读请求中断产生。当从存储卡接收到数据，则读入FIFO中。当接收的数据的字节数等于由MMCFIFOCTL中的FIFOLEV位设定的水平时，一个EDMA读事件发出且新的EDMA事件禁用直到EDMA发出当前事件的传送完成时。数据由MMCDRR从FIFO中读取。在检查EDMA事件发生或FIFO变满期间，FIFO控制器一直从存储卡中读取数据。一旦EDMA事件结束后，新的EDMA事件是有效的。如果FIFO填满时，FIFO控制器停止MMC / SD控制器读取任何数据，直到FIFO不再是满的。那么数据就会由此引脚输出。

CPU读: 系统CPU也可以直接通过读取MMC数据接收寄存器（MMCDRR）来读取存储卡内的数据。当从存储卡接收到数据，则读入FIFO中。当接收数据的字节数等于由MMCFIFOCTL中的FIFOLEV位设定的水平，一次DRRDYINT中断将发生并且MMC状态寄存器0（MMCST0）的DRRDY位被置位。在接收到中断时，CPU迅速读出接收的字节（等于由FIFOLEV位设置的水平）。

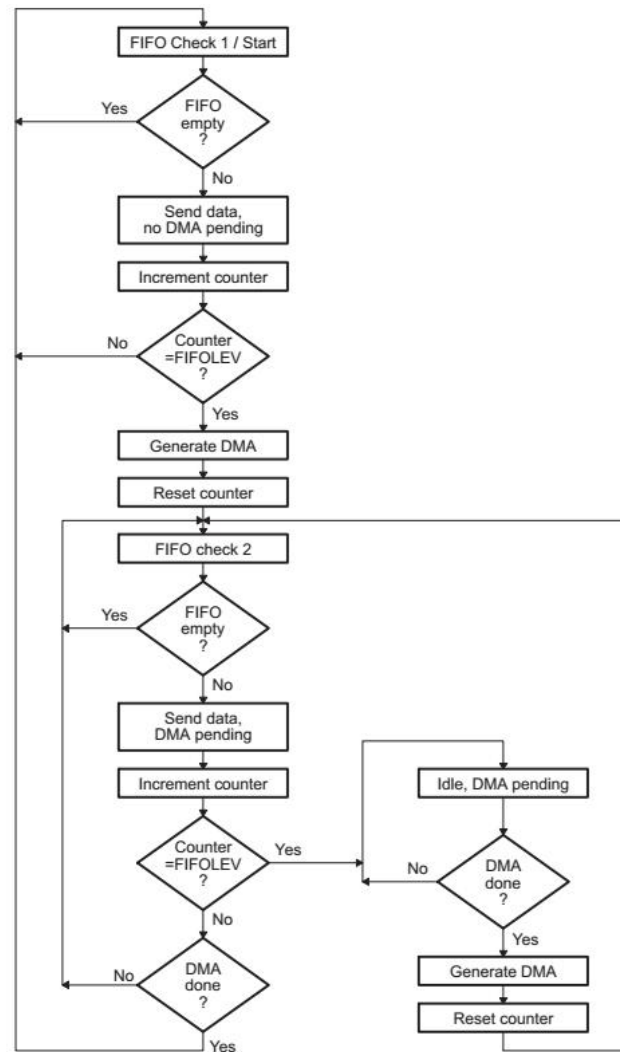


MMC/SD的FIFO

读操作流程图



写操作流程图





MMC/SD的FIFO

写存储卡时的FIFO操作

EDMA写： FIFO控制器管理从CPU或EDMA接受数据并将数据传递到MMC / SD控制器的活动。FIFO控制器适当地发起EDMA写事件。发出的EDMA写事件时，一个EDMA写请求中断产生。数据通过MMCDXR写入FIFO。需要注意的是，EDMA访问MMCDXR是透明的。CPU或EDMA控制器将数据写入到FIFO。FIFO将数据传递给负责将数据写入到存储卡中的MMC / SD控制器。当在FIFO中的数据字节数小于由MMCFIFOCTL中的FIFOLEV位设定的水平，一个EDMA写事件发起，同时新的EDMA事件被禁止。在检查该EDMA事件完成或FIFO成为空的同时FIFO控制器持续给MMC / SD控制器发送数据。一旦EDMA事件结束后，新的EDMA事件使能。如果FIFO变空，FIFO控制器会通知MMC / SD控制器。

CPU写： 系统CPU还可以通过写MMC数据发送寄存器（MMCDXR）直接写入存储卡数据。MMC / SD外设支持1， 2， 3， 或4字节宽的数据写操作。CPU利用FIFO通过MMC / SD控制器传送数据到存储卡上。CPU写被传输到MMCDXR的数据。与EDMA驱动一样，当FIFO中的数据数目小于MMCFIFOCTL中的FIFOLEV位设定的值时，一个DXRDYINT中断产生且MMC状态寄存器0（MMCST0）的DXRDY位被设置以通知CPU，表示该空间可用于新的数据的存储。



MMC / SD控制器初始化

用于初始化的MMC / SD控制器的一般方法，如下面的步骤。

- 将通过设置MMC控制寄存器（MMCCTL）的CMDRST位和DATRST位使MMC / SD控制器处于复位状态。复位后，可以设置MMCCTL其他位。
- 给其它寄存器写入所需的值来完成MMC / SD控制器配置I 或 I2C。
- 清除MMCCTL的CMDRST位和DATRST位，使MMC / SD控制器脱离复位状态。建议不要重写已经写入MMCCTL其他位。
- 通过设置MMC内存时钟控制寄存器（MMCCLK）的CLKEN位来使能MMCSA_CLK引脚，使内存时钟发送到存储卡中。



MMC/SD的中断

支持的中断事件如下表

当中断事件发生时，其在MMC状态寄存器0（MMCST0）对应的标志位将被置位。如果对应于各标志的使用位在MMC中断屏蔽寄存器（MMCIM）被置位，将产生一个中断请求。所有这样的请求通过被多路复用单一的MMC / SD中断请求来实现到从MMC / SD外设到CPU的传递。MMC / SD中断是CPU可屏蔽中断的一部分。一个CPU中断与MMC/SD的功能（详见设备的数据手册）相关。MMC / SD中断的中断服务例程序（ISR）可以通过检查MMCST0位来确定造成中断的事件类型。MMCST0被读取，所有的寄存器位将被自动清除。在数据传输期间的中间，取决于在MMC FIFO控制寄存器（MMCFIFOCTL）的设定，每传输256位或512位数据DXRDY和DRRDY位将被设置。对FIFO产生的中断响应写和读操作，会自动清除相应的中断位/标识位。

中断请求	中断事件
TRNDNEINT	对于读操作：MMC / SD控制器已接收到最后一个字节（CRC校验之前）。 对于写操作：MMC / SD控制器已发送最后一个字到输出移位寄存器。
DATEDINT	MMCS0_DAT3引脚上检测到跳变沿。
DRRDYINT	MMCDDR准备被读取（在FIFO中的数据是阈值以上）。
DXRDYINT	MMCDDR准备就绪发送新数据（FIFO中的数据小于阈值）。
CRCRSINT	在从存储卡的响应，检测到CRC错误。
CRCRDINT	在从存储卡读出的数据中检测到CRC错误。
CRCWRINT	在写入到存储卡中的数据中检测到CRC错误。
TOUTRSINT	而MMC控制器在等待一个命令的响应时发生超时。
TOUTRDINT	MMC控制器，等待从存储卡中发出的数据时发生超时。
RSPDNEINT	对于一个需要响应的命令：MMC控制器收到响应且无CRC错误。 对于不需要的响应的命令：MMC控制器已经发送完完成命令。
BSYDNEINT	MMC控制器等待忙信号的时候，存储卡已停止或不再发送忙信号
DATDNEINT	对于读操作：在MMC控制器接收到的数据没有CRC错误。 对于写操作：在MMC控制器已发送完数据。



MMC/SD卡的程序常规操作

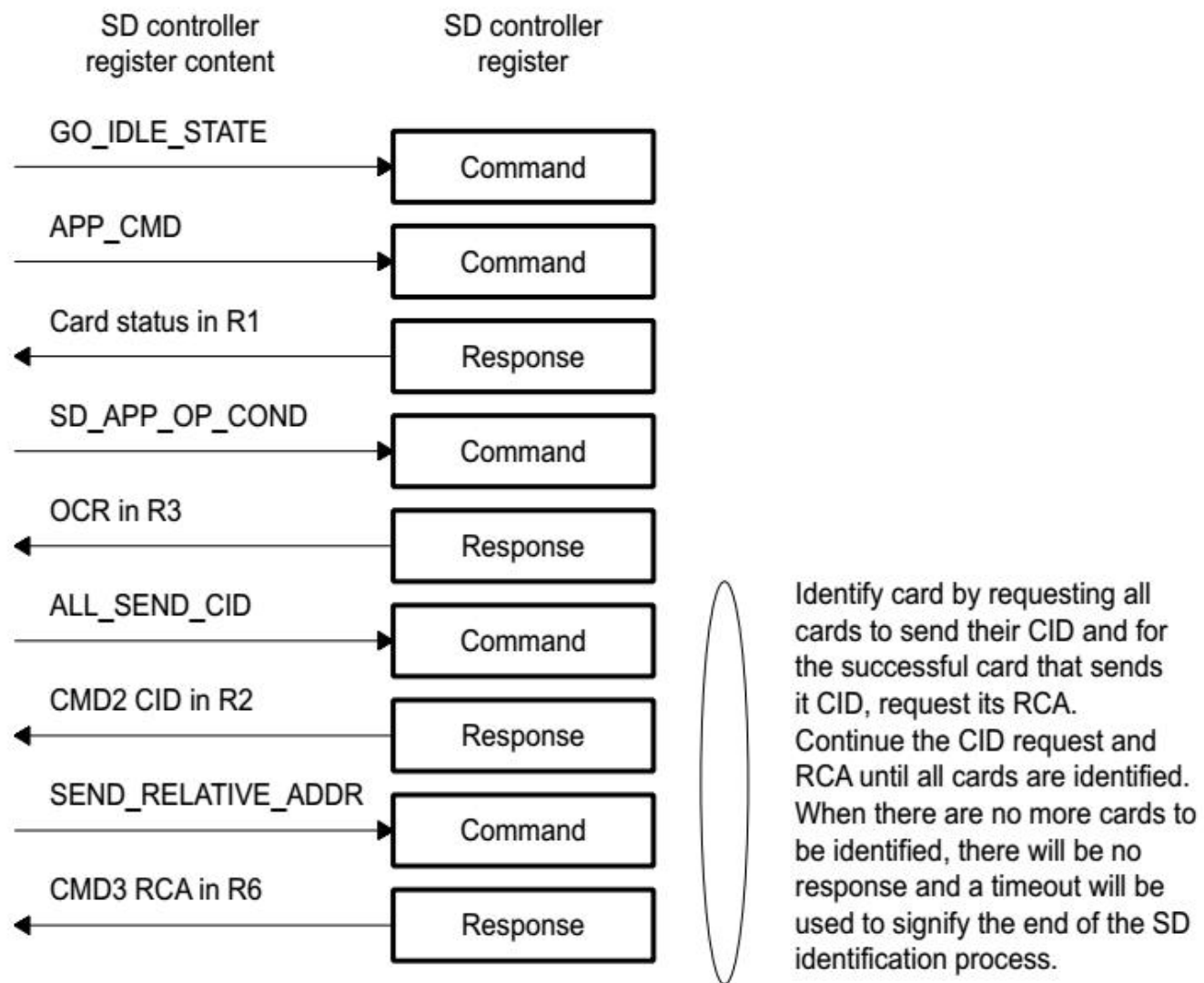
SD卡的识别步骤

MMC / SD控制器在MMC / SD本机模式下开始数据传输或从存储卡获取数据前，它必须先确定总线上有多少卡出现，并对其进行配置。对于响应ALL_SEND_CID卡广播命令的每一个卡时，控制器将读出卡的唯一标识地址（CID），然后给它分配相对地址（RCA）。

- 使用MMC命令寄存器（MMCCMD）发出GO_IDLE_STATE（CMD0）命令到MMC卡。所有空闲状态卡（MMC和SD）使用MMCCMD发出CMD0命令，不需要卡的无响应。
- 使用MMCCMD发出APP_CMD（CMD55）命令（R1响应预期），以表明后面的命令是一个应用程序的命令。
- 使用MMCCMD发送以SD_SEND_OP_COND（ACMD41）命令所支持的电压范围发送该命令到SD卡的（需要R3响应预期）。使用MMCCMD发送ACMD41命令允许主机识别，并拒绝那些不符合主机支持的VDD范围的卡。
- 使用MMCCMD发送ALL_SEND_CID（CMD2）命令（需要R2响应）到MMC卡。使用MMCCMD发送CMD2命令通知所有卡发送其独特的卡标识（CID）号。应该只有一张卡能成功地发送其完整的CID号到主机。发送成功的卡进入识别状态，不再响应此命令。CMD2命令通知所有卡发送其独特的卡识别（CID）号码。应该只有一张卡能成功地发送其完整的CID号到主机。该发送成功的卡进入识别状态，不再响应这个命令。
- 使用MMCCMD发出SEND_RELATIVE_ADDR（CMD3）命令（需要R1响应）使该卡发布一个新的相对地址以供将来使用，以便在数据传输模式寻找该卡。



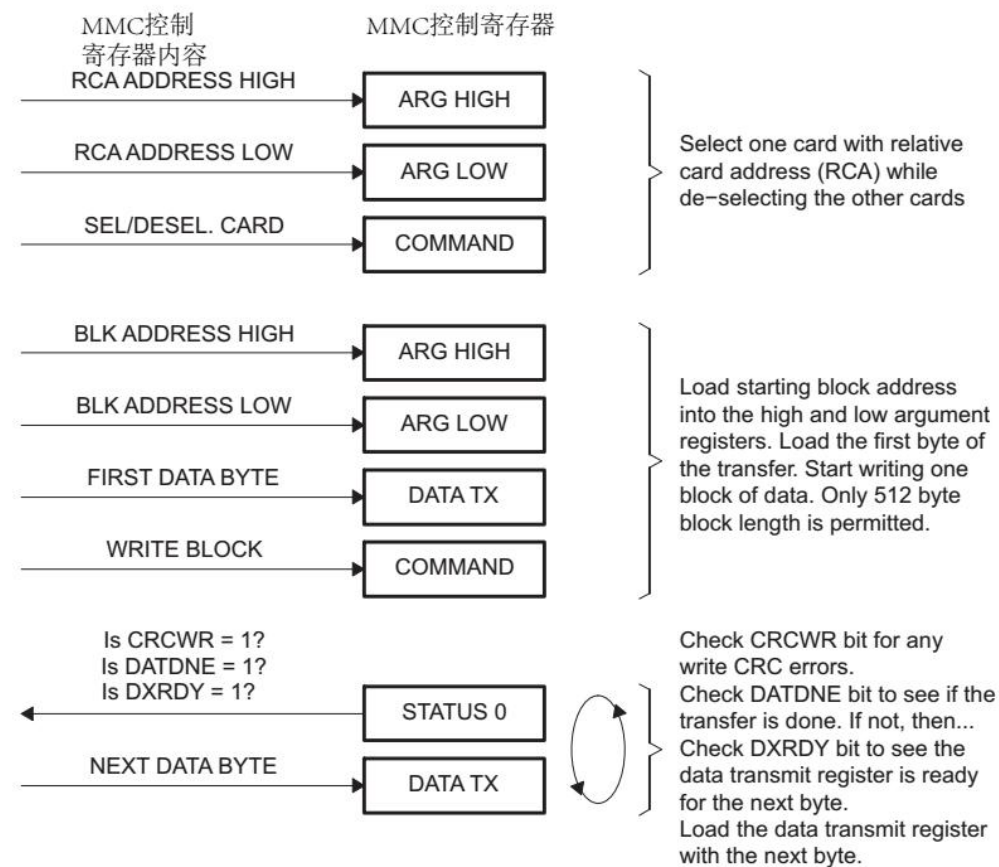
SD卡识别流程图





SD卡写操作（单块）

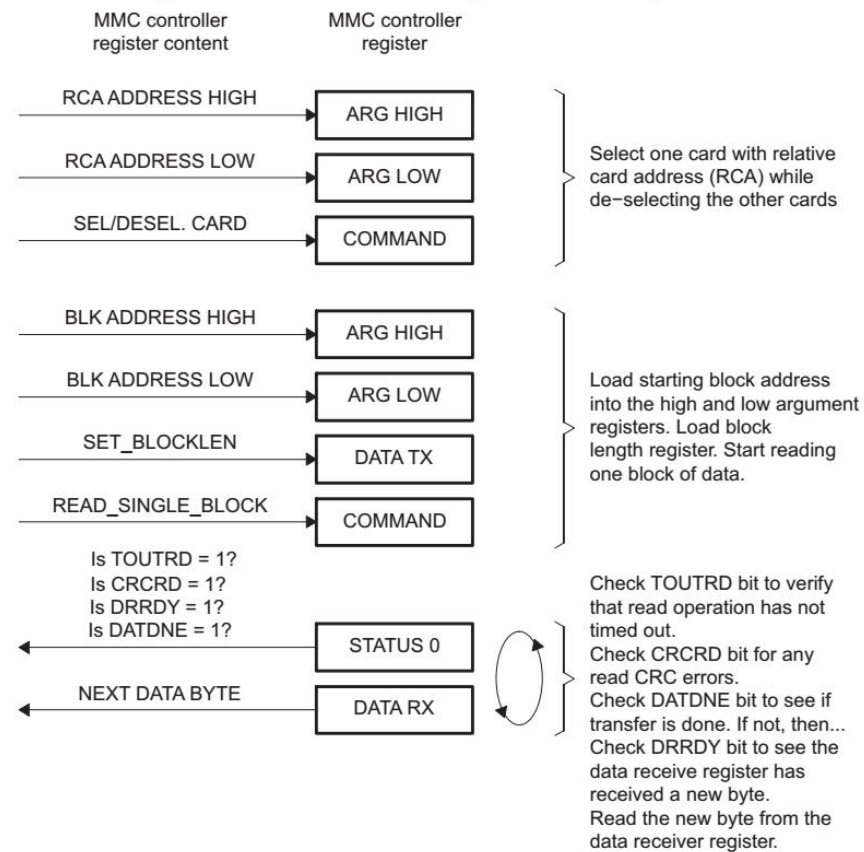
- 写卡的相对地址到MMC参数寄存器（MMCARGH和MMCARGL）。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 使用MMC命令寄存器（MMCCMD）发送SELECT / DESELECT_CARD广播命令。这将选择已编寻址的卡并不选择其他卡。
- 编写目标起始地址到MMC参数寄存器。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 读取CSD卡以确定该卡的最大块长度。
- 使用MMCCMD发送SET_BLOCKLEN命令（如果当前操作的块长度是和以前操作中使用的长度不同的情况）。块长度必须为512个字节的整数倍且小于在CSD中规定的最大块长度。
- 复位FIFO（设置MMCFIFOCTL的FIFORST位）。
- 将FIFO方向设为发送（设置MMCFIFOCTL的FIFODIR位）。
- 设置访问宽度（设置MMCFIFOCTL的ACCWD位）。
- 使能MMC中断。
- 使能DXRDYINT中断。
- 向数据发送寄存器（MMCDXR）写数据块的前32字节。
- 使用MMCCMD发送WRITE_BLOCK命令到卡上。
- 设置MMCCMD的DMATRIG位触发第一个数据传输。
- 等待MMC中断。
- 使用MMC状态寄存器0（MMCST0）检查错误和FIFO的状态。如果所有的数据还没有被写入，如果FIFO不满，则转到步骤16。如果所有的数据已被写入，则停止。
- 写数据块的下一个n个字节（字节数依赖于FIFOLEV位的设置，该位在MMCFIFOCTL寄存器中：0 = 32字节，1 = 64个字节）到MMC数据发送寄存器（MMCDXR）





SD卡读操作（单块）

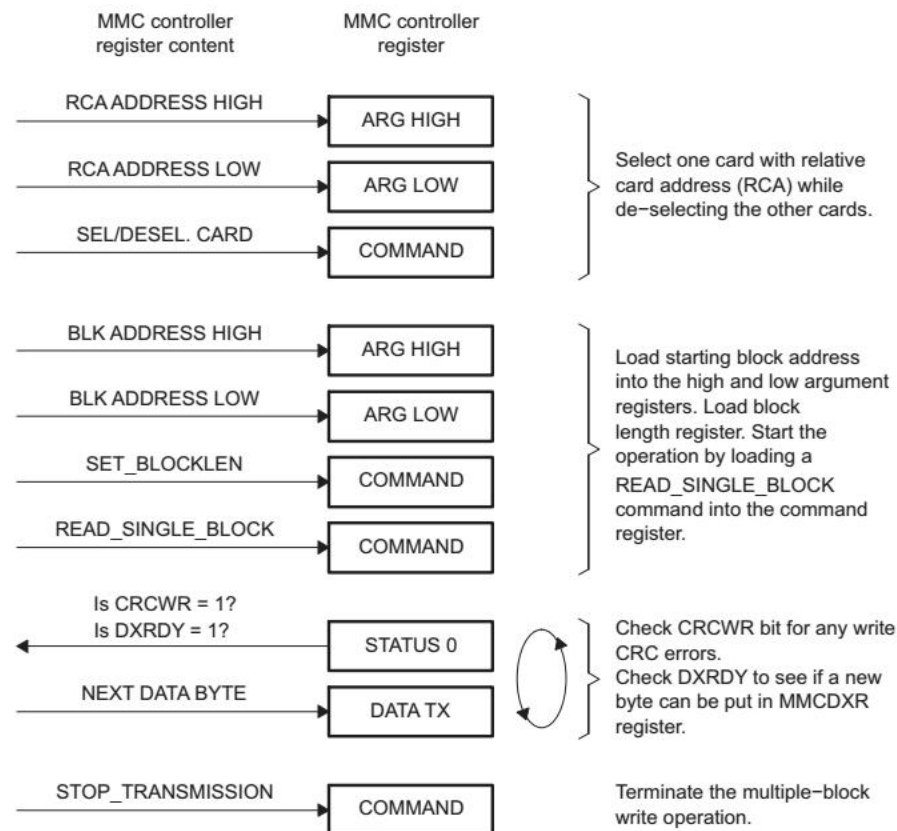
- 写卡的相对地址到MMC参数寄存器（MMCARGH和MMCARGL）。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 使用MMC命令寄存器（MMCCMD）发送SELECT / DESELECT_CARD广播命令。这将选择已编寻址的卡并不选择其他卡。
- 编写目标起始地址到MMC参数寄存器。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 读取CSD卡以确定该卡的最大块长度。
- 使用MMCCMD发送SET_BLOCKLEN命令（如果当前操作的块长度是和以前操作中使用的长度不同的情况）。块长度必须为512个字节的整数倍且小于在CSD中规定的最大块长度。
- 复位FIFO（设置MMCFIFOCTL的FIFORST位）。
- 将FIFO方向设为接受（设置MMCFIFOCTL的FIFODIR位）
- 设置访问宽度（设置MMCFIFOCTL的ACCWD位）
- 使能MMC中断。
- 使能DXRDYINT中断。
- 使用MMCCMD发送READ_SINGLE_BLOCK命令到卡上。
- 设置MMCCMD的DMATRIG位触发第一个数据传输。
- 等待MMC中断。
- 使用MMC状态寄存器0（MMCST0）来检查错误和FIFO的状态。如果FIFO不为空，则转到步骤16。如果所有已读取的数据，则停止。
- 读MMC数据接收寄存器（MMCDRR）中下一个n个字节（取决于MMCFIFOCTL中FIFOLEV的设置：0 = 32字节，1 = 64字节）的数据块，并返回步骤14。





SD卡写操作（多块）

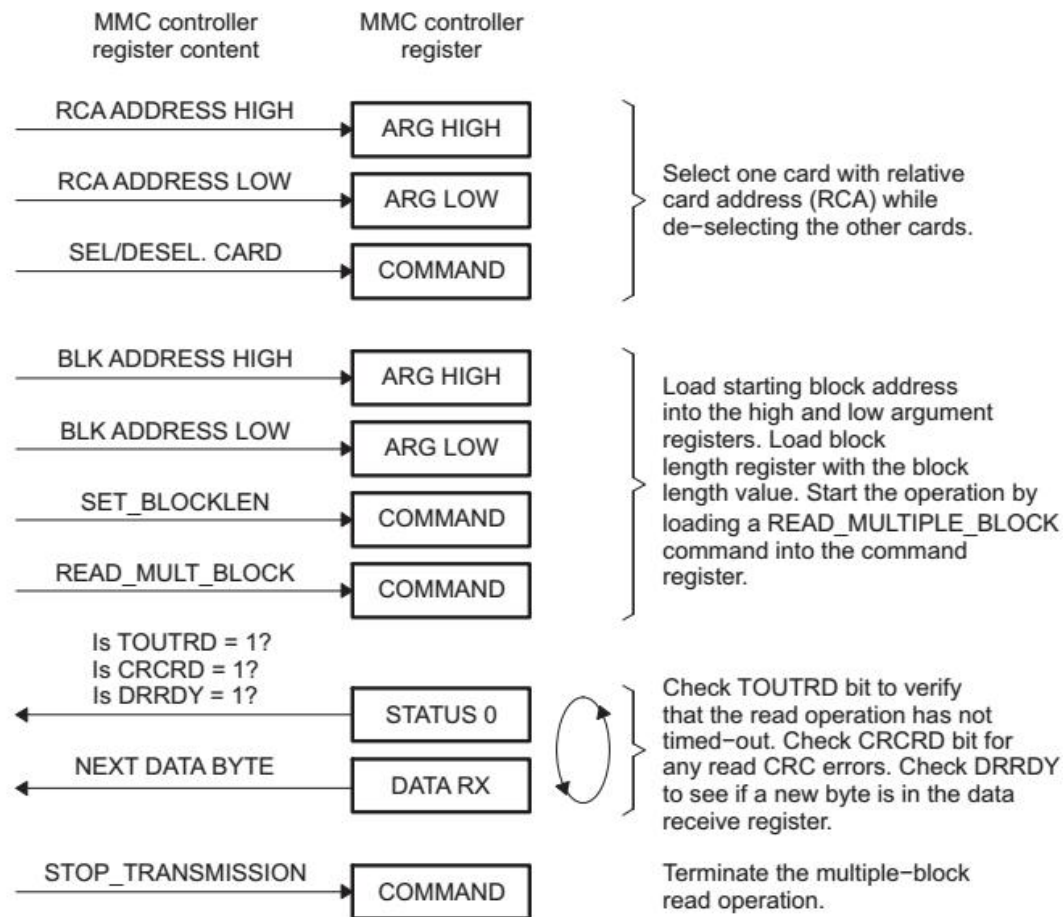
- 为了执行多块写入，需要在MMC / SD控制器和卡设置同一个块长度。
- 写卡的相对地址到MMC参数寄存器（MMCARGH和MMCARGL）。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 读取CSD卡以确定该卡的最大块长度。
- 使用MMCCMD发送SET_BLOCKLEN命令（如果当前操作的块长度是和以前操作中使用的长度不同的情况）。块长度必须为512个字节的整数倍且小于在CSD中规定的最大块长度。
- 复位FIFO（设置MMCFIFOCTL的FIFORST位）。
- 将FIFO方向设为发送（设置MMCFIFOCTL的FIFODIR位）。
- 设置访问宽度（设置MMCFIFOCTL的ACCWD位）。
- 设置FIFO阈值（设置MMCFIFOCTL的FIFOLEV位）。
- 使能MMC中断。
- 使能DXRDYINT中断。
- 向数据发送寄存器（MMCDXR）写数据块的前32字节。
- 使用MMCCMD发送WRITE_BLOCK命令到卡上。
- 设置MMCCMD的DMATRIG位触发第一个数据传输。
- 等待MMC中断。
- 使用MMC状态寄存器0（MMCST0）检查错误和FIFO的状态。如果所有的数据还没有被写入，如果FIFO不满，则转到步骤16。如果所有的数据已被写入，则停止。
- 写数据块的下一个n个字节（字节数依赖于FIFOLEV位的设置，该位在MMCFIFOCTL寄存器中：0 = 32字节，1 = 64个字节）到MMC数据发送寄存器（MMCDXR）。
- 使用MMCCMD发送STOP_TRANSMISSION命令。





SD卡读操作（多块）

- 为了执行多块读取，需要在MMC / SD控制器和卡设置同一个块长度。
- 写卡的相对地址到MMC参数寄存器（MMCARGH和MMCARGL）。加载地址高部分到MMCARGH和地址低部分到MMCARGL。
- 读取CSD卡以确定该卡的最大块长度。
- 使用MMCCMD发送SET_BLOCKLEN命令（如果当前操作的块长度是和以前操作中使用的长度不同的情况）。块长度必须为512个字节的整数倍且小于在CSD中规定的最大块长度。
- 复位FIFO（设置MMCFIFOCTL的FIFORST位）。
- 将FIFO方向设为接受（设置MMCFIFOCTL的FIFODIR位）。
- 设置访问宽度（设置MMCFIFOCTL的ACCWD位）。
- 设置FIFO阈值（设置MMCFIFOCTL的FIFOLEV位）。
- 使能MMC中断。
- 使能DXRDYINT中断。
- 使用MMCCMD发送WRITE_BLOCK命令到卡上。
- 设置MMCCMD的DMATRIG位触发第一个数据传输。
- 等待MMC中断。
- 使用MMC状态寄存器0（MMCST0）检查错误和FIFO的状态。如果所有的数据还没有被写入，如果FIFO不满，则转到步骤15。如果所有的数据已被写入，则停止。
- 读取MMC数据接收寄存器（MMDRR）中n个字节的数据（取决于MMCFIFOCTL中FIFOLEV的设置：0 = 32字节，1 = 64字节），并返回步骤10。
- 使用MMCCMD发送STOP_TRANSMISSION命令。





SDIO功能

为了支持SDIO卡，可利用MMC / SD控制器以下功能：

- 读取等待操作请求。
- 在读取等待操作开始时中断CPU。
- 在SDIO中断检测时中断CPU。

当1位模式和传输时钟（存储器时钟）是关闭的，该外设不能识别SD_DATA1线上的SDIO中断。有两种方法可用来处理这种情况：

- 不要关闭在1位模式下的存储器时钟。时钟是由MMC存储时钟控制寄存器（MMCCLK）的CLKEN位使能的。
- 如果存储器时钟需要被关闭，硬件上连接GPIO信号和SD_DATA1，并使用GPIO作为外部中断输入。当存储器时钟使能，禁用GPIO中断并使能SDIO中断。当存储器时钟被禁用，通过软件使能GPIO中断并禁用SDIO中断。

广州创龙电子科技有限公司

电话:020-89986280

传真:020-89986280 - 803

官网:www.tronlong.com

论坛:51ele.net

微信公众号:广州创龙

谢谢



扫描关注创龙公众号