

## SD 操作学习笔记

标签： 笔记 学习 SD 卡驱动 2009-11-11 21:51

### SD 卡操作 一、 概述 1、 简介

SD 卡是基于 flash 的存储卡。

SD 卡和 MMC 卡的区别在于初始化过程不同。

SD 卡的通信协议包括 SD 和 SPI 两类。

SD 卡使用卡内智能控制模块进行 FLASH 操作控制，包括协议、安全算法、数据存取、ECC 算法、缺陷处理和分析、电源管理、时钟管理。

### 2、功能介绍 2.1 特点

#### 1) 主机无关的 FLASH 内存擦除和编程

读或写数据，主机只要发送一个带地址的命令，然后等待命令完成，主机无需关心具体操作的完成。当采用新型的 FLASH 时，主机代码无需更新。

#### 2) 缺陷管理

#### 3) 错误恢复

#### 4) 电源管理

Flash 每个扇区有大约 10 万次的写寿命，读没有限制。

擦除操作可以加速写操作，因为在写之前会进行擦除。

### 3 SD 总线模式 3.1 Negotiating Operation Conditions

当主机定义了 SD 卡不支持的电压范围时，SD 卡将处于非活动状态，将忽略所有的总线传输。要退出非活动状态唯一的方法就是重新上电。

### 3.2 SD 卡获取和识别

SD 卡总线采用的是单主多从结构，**总线上所有卡共用时钟和电源线**。主机依次分别访问每个卡，每个卡的 CID 寄存器中已预编程了一个唯一的卡标识号，用来区分不同的卡。

主机通过 READ\_CID 命令读取 CID 寄存器。CID 寄存器在 SD 卡生产过程中的测试和格式化时被编程，主机只能读取该号。

DAT3 线上内置的上拉电阻用来侦测卡。在数据传输时电阻断开(使用 ACMD42)。

### 3.3 卡状态

卡状态分别存放在下面两个区域：

卡状态 (Card Status)，存放在一个 32 位状态寄存器，在卡响应主机命令时作为数据传送给主机。

SD 状态 (SD\_Status)，当主机使用 SD\_STATUS (ACMD13) 命令时，512 位以一个数据块的方式发送给主机。SD\_STATUS 还包括了和 BUS\_WIDTH、安全相关位和扩展位等的扩展状态位。

### 3.4 内存组织

数据读写的基本单元是一个字节，可以按要求组织成不同的块。

**Block:**块大小可以固定，也可以改变，**允许的块大小是实际大小等信息存储在 CSD 寄存器。**

**Sector:**和擦除命令相关，由几个块组成。**Sector** 的大小对每个设备是固定的，大小信息存储在 CSD 寄存器。

**WP Group:**写保护单位。大小包括几个 group，写保护由一位决定，对每个设备大小是固定的，存储在 CSD 寄存器。

### 3.5 读写操作

**Single Block Mode:**主机根据事先定义的长度读写一个数据块。由发送模块产生一个 16 位的 CRC 校验码，接受端根据校验码进行检验。**读操作的块长度受设备 sector 大小 (512 bytes)的限制，但是可以最小为一个字节。**不对齐的访问是不允许的，每个数据块必须位于单个物理 sector 内。写操作的大小必须为 sector 大小，起始地址必须与 sector 边界对齐。

**Multiple Block Mode:**主机可以读写多个数据块（相同长度），根据命令中的地址读取或写入连续的内存地址。操作通过一个停止传输命令结束。写操作必须地址对齐。

## 3.6 数据传输速率

SD 卡可以通过单数据线（DAT0）或四根数据线（DAT0-DAT3）进行数据传输。单根数据线传输最大传输速率为 25 Mbit/s，四根数据线最大传输速率为 100 Mbit/s。

## 3.7 数据保护

每个 sector 的数据通过 Error Correction Code (ECC)进行保护。在写 sector 时生成 ECC，在读 sector 时检验 ECC。如果发现错误，在传输前进行纠正。

## 3.8 数据擦除

**SD 卡数据擦除的最小单位是 sector。**为了加速擦除操作，多个 sector 可以同时擦除。为了方便选择，第一个指令包含起始地址，第二个指令包含结束地址，在地址范围内的所有 sector 将被擦除。

## 3.9 写保护

两种写保护方式可供选择，永久保护和临时保护，两种方式都可以通过 PROGRAM\_CSD 指令进行设置。永久保护位一旦设置将无法清除。

## 3.10 拷贝位

通过 CSD 寄存器中的拷贝位（copy bit）设置 SD 卡中的数据是原始数据还是拷贝数据。拷贝位一旦设置，将无法清除，在测试和格式化时使用。

## 3.11 CSD 寄存器

**所有 SD 卡的配置信息存储在 CSD 寄存器。通过 SEND\_CSD 读取，PROGRAM\_CSD 修改。**

主机通过 9 个引脚和 SD 卡相连

## 1.1 SD 模式引脚

扩展数据线(DAT1-DAT3)上电后为输入，SET\_BUS\_WIDTH 命令执行后作为数据线。即使只有 DAT0 使用，所有数据线都和外部上拉电阻连接，否则 DAT1 & DAT2（如果未被使用）的振荡输入将引起非期望的高电流损耗。

上电后，数据线输入 50K(+/-20K)欧姆的上拉（用来进行卡侦测和 SPI 模式选择）。用户可以在常规数据传输时，通过 SET\_CLR\_CARD\_DETECT (ACMD42) 命令分离上拉。

## 1.2 SPI 模式引脚

## 1.3 寄存器

名称	宽度	描述
CID	128	卡标识号
RCA	16	相对卡地址（Relative card address）:本地系统中卡的地址，动态变化，在主机初始化的时候确定 *SPI 模式中没有
CSD	128	卡描述数据:卡操作条件相关的信息数据
SCR	64	SD 配置寄存器:SD 卡特定信息数据
OCR	32	操作条件寄存器

主机通过重新上电来重置（reset）卡。卡有它自身检测上电的电路，当上电后卡状态切换到 idle 状态。也可以通过 GO\_IDLE (CMD0)指令来重置。

## 2 SD 卡总线拓扑

SD 总线有 6 根通信线和三根电源供应线：

- ✧ **CMD**——命令线是双向信号线。主机和卡通过 push pull 模式工作。
- ✧ **DAT0-3**——数据线是双向信号线。主机和卡通过 push pull 模式工作。
- ✧ **CLK**——时钟是从主机到卡的信号。CLK 通过 push pull 模式操作。
- ✧ **VDD**—VDD 是所有卡的电源供应线。
- ✧ **VSS[1:2]**—VSS 是 2 根地线。

在初始化的时候，向每个卡分别发送命令，允许应用检测卡并给物理槽（physical slot）分配逻辑地址。数据通常分别传输给每个卡。然后，为了方便处理卡堆栈，初始化后所有命令同时发送给所有卡，在命令数据包中包含了操作地址。

SD 总线允许动态配置数据线数目。上电后默认 SD 卡只用 DAT0 作为数据传输线。初始化后，主机可以改变总线宽度。这个特性使得在硬件开销和系统性能间取得平衡。

### 3 SPI 总线拓扑 4 电气接口 4.1 上电

上电后，包括热插入，卡进入 idle 状态。在该状态 SD 卡忽略所有总线操作直到接收到 ACMD41 命令。**ACMD41 命令是一个特殊的同步命令，用来协商操作电压范围，并轮询所有的卡。**除了操作电压信息，ACMD41 的响应还包括一个忙标志，表明卡还在 power-up 过程工作，还没有准备好识别操作，即告诉主机卡还没有就绪。主机等待(继续轮询)直到忙标志清除。单个卡的最大上电时间不能操作 1 秒。

上电后，主机开始时钟并在 CMD 线上发送初始化序列，初始化序列由连续的逻辑“1”组成。序列长度为最大 1 毫秒，74 个时钟或 supply-ramp-up 时间。额外的 10 个时钟(64 个时钟后卡已准备就绪)用来实现同步。

每个总线控制器必须能执行 ACMD41 和 CMD1。CMD1 要求 MMC 卡发送操作条件。在任何情况下，ACMD41 或 CMD1 必须通过各自的 CMD 线分别发送给每个卡。

## 5 寄存器 5.1 OCR(Operating Conditions Register)

32 位的操作条件寄存器存储了 VDD 电压范围。SD 卡操作电压范围为 2~3.6V。然而从内存中访问数据的电压是 2.7~3.6V。OCR 显示了卡数据访问电压范围，结构如下表所示。

表 3-8 OCR 寄存器定义

OCR 位	VDD 电压范围
-------	----------

0-3	保留
4	1.6~1.7
5	1.7~1.8
6	1.8~1.9
7	1.9~2.0
8	2.0~2.1
9	2.1~2.2
10	2.2~2.3
11	2.3~2.4
12	2.4~2.5
13	2.5~2.6
14	2.6~2.7
15	2.7~1.8
16	2.8~2.9
17	2.9~3.0
18	3.0~3.1
19	3.1~3.2
20	3.2~3.3
21	3.3~3.4
22	3.4~3.5
23	3.5~3.6
24-30	保留
31	卡上电状态位(忙)

OCR 结构如下图所示。如果第 32 位 (busy bit) 置位, 表明卡上电过程已结束。

## 5.2 CID(Card Identification)

CID 寄存器长度为 16 个字节的卡唯一标识号, 该号在卡生产厂家编程后无法修改。SD 和 MMC 卡的 CID 寄存器结构不一样。

名称	类型	宽度	CID 位	内容	CID 值
厂商 ID	Binary	8	[127:120]	SD 卡协会管理和分配	0x03
OEM/Application ID(OID)	ASCII	16	[119:104]	识别卡的 OEM 或卡内容, 由制造商分配	0x53,0x44
产品名 (PNM)	ASCII	40	[103:64]	5 个 ASCII 字符	SD128
产品版本 (PRV)	BCD	8	[65:56]	2 个二进制编码的十进制数	产品版本 (30) 1
序列号 (PSN)	Binary	32	[55:24]	32 位无符号整数	产品序列号
保留		4	[23:20]		
生成日期 (MDT)	BCD	12	[19:8]	yym (从 2000 年的偏移量)	如:Apr 2001=0x014
CRC7 校验和 (CRC)	Binary	7	[7:1]	CRC Calculation: $G(x)=x^7+3+1$ $M(x)=(MID-MSB)*x^{119}+...+(CIN-LSB)*x^0$ $CRC[6...0]=\text{Remainder}[(M(x)*x^7)/G(x)]$	CRC7
未用		1	[0:0]		

1、格式为“n.m”，如“6.2”表示为 0110 0010

### 5.3 CSD(Card Specific Data)

CSD 寄存器包含访问卡数据所需的配置信息。SD 卡和 MMC 卡的 CSD 不同。

## 6 数据交互格式和卡容量

通常，SD 卡分为 2 个区：

✧ 用户区—用户通过读写命令存储安全和非安全数据。

- ✧ 安全保护区（Security Protected Area）—版权保护应用程序用来保存安全相关数据，通过 SD 安全规范中定义的条件验证后，由主机使用安全的读写指令完成操作。安全保护区的大小大概是总大小的 1%。

### 三、 SD 卡协议 1 SD 总线协议

SD 总线通信是基于命令和数据位流方式的，由一个起始位开始，以一个停止位结束：

命令——命令是开始操作的标记。命令从主机发送一个卡（寻址命令）或所有连接的卡（广播命令）。命令在 CMD 线上串行传送。

响应——响应是从寻址卡或所有连接的卡（同步）发送给主机用来响应接受到的命令的标记。命令在 CMD 线上串行传送。

数据——数据可以通过数据线在卡和主机间双向传送。

卡寻址通过会话地址方式实现，地址在初始化的时候分配给卡。SD 总线上的基本操作是 command/response。

数据传送采用块方式，数据块后接 CRC 校验位，操作包括单数据块和多数据块。多数据块更适合快速写操作，多数据块传输当在 CMD 线出现停止命令时结束。数据传输可以在主机端设置采用单数据线或多数据线方式。

块写操作在 DAT0 数据线写操作期间使用忙信号，无论用来传输的信号线数目是多少。

命令格式如下所示：

响应标记（token）根据内容不同具有四种格式，标记长度。长度为 48 位或 136 位。数据块的 CRC 算法采用 16 位的 CCITT 多项式。

在命令行中，MSB 位首先传送，LSB 位最后传送。

当使用宽总线模式时，数据同时在 4 根数据线上传输。开始位、结束位和 CRC 在每根数据线上传输。CRC 对每根数据线单独计算。CRC 状态响应和 Busy 信号只通过 DAT0 由卡发送给主机。

## 2 协议功能描述

所有主机和 SD 卡间的通信由主机控制。主机发送下述两类命令：

- **广播命令**——广播命令发送给所有 SD 卡，有些命令需要响应。
- **寻址（点对点）命令**——寻址命令只发送给具有相应地址的卡，并需要从卡返回一个响应。

对卡而言也有两类操作：

- **卡识别模式**——在重置（reset）后当主机查找总线上的新卡时，处于卡识别模式。重置后 SD 卡将始终处于该模式，直到收到 SEND\_RCA 命令（CMD3）。
- **数据传输模式**——一旦卡的 REC 发布后，将进入数据传输模式。主机一旦识别了所有总线上的卡后，将进入数据传输模式。

操作模式与卡状态关系：

## 3 卡识别模式

在卡识别模式，主机重置所有处于卡识别模式的 SD 卡，检验操作电压范围，识别卡并请求卡发送相对卡地址 RCA。操作对每个卡在各自的 CMD 线上单独进行，所有的数据传送只使用 CMD 线。

### 3.1 重置

GO\_IDLE\_STATE(CMD0)是软件重置命令，设置每个 SD 卡进入 Idle 状态。处于 Inactive 状态的卡不受此命令影响。

主机上电后，所有 SD 卡进入 Idle 状态，包括处于 Inactive 状态的卡。至少 74 个时钟周期后才能开始总线传输。

上电或 CMD0(**重置**)后，所有 SD 卡的命令线处于输入模式，等待下一个命令的起始位。卡通过一个默认的相对卡地址 RCA（RCA=0x0000）和默认驱动寄存器设置（最低速，最高驱动电流）初始化。

### 3.2 操作电压范围验证

SD 的物理规范标准要求所有 SD 卡能通过最小和最大供电电压间的任何电压和主机建立通信。然而，数据传输时的最小和最大电压值在操作条件寄存器 OCR 中定义，可能并不能覆盖所有的电压范围。SD 卡主机希望通过读取卡的 OCR 寄存器获取合适的电压值或弹出卡。

SD 卡

### 3.3 卡识别过程

在识别时钟速率  $f_{OD}$  下主机开始卡识别过程。SD 卡的 CMD 线输出驱动是 push-pull 驱动。

总线激活后，主机要求卡发送它们的有效操作条件（ACMD41 preceding with APP\_CMD—CMD55 with RCA=0x0000）。ACMD41 命令的响应是卡的操作条件寄存器。相同的命令将发送给系统中所有的卡。不兼容的卡将进入 Inactive 状态。主机然后发送命令 ALL\_SEND\_CID（CMD2）到每个卡以获取每个卡的唯一标识 CID 号。未识别的卡通过 CMD 线发送 CID 号作为响应。当卡发送 CID 号后，进入识别状态（Identification State）。此后，主机发送 CMD3

（SEND\_RELATIVE\_ADDR）要求卡发布一个新的相对卡地址 RCA，地址比 CID 短，在以后的数据传输模式中用来寻址卡。一旦获得 RCA 后，卡状态变成就绪状态（Stand-by state）。此时，如果主机要求卡换成其他的 RCA 号，可以通过发送另一个 SEND\_RELATIVE\_ADDR 命令给卡，要求发布一个新的 RCA，最后发布的 RCA 是实际使用的 RCA。主机对系统中的每个卡重复识别过程。

所有的 SD 卡初始化完以后，系统将开始初始化 MMC 卡（如果有的话），使用 MMC 卡的 CMD2 和 CMD3。

## 4 数据传输模式

直到主机知道所有 CSD 寄存器的内容， $f_{pp}$  时钟速率必须保持在  $f_{OD}$ ，因为一些卡有操作频率限制。主机发送 SEND\_CSD（CMD9）获取卡定义数据（Card Specific Data，CSD 寄存器），如块大小、卡存储容量、最大时钟速率等。

CMD7 用来选择一个卡并将它置于传输状态（Transfer state），在任何时间只能有一个卡处于传输状态。如果已有一个卡处于传输状态，它和主机的连接将释放，并返回到 Stand-by 状态。当 CMD7 以保留相对地址“0x0000”发送时，

所有卡将返回到 Stand-by 状态。这可以用来识别新的卡而不重置其他已注册的卡。在这种状态下已有一个 RCA 地址的卡不响应识别命令 (ACMD41,CMD2,CMD3)。

注意：当卡接收到一个带有不匹配 RCA 的 CMD7 时，卡将取消选中。在公用 CMD 线时，选中一个卡时将自动不选中其他卡。因此，在 SD 卡系统中，主机具有如下功能：

- 初始化完成后，在公用 CMD 线时，不选中卡是自动完成的。
- 如果使用单独的 CMD 线，需要关注不选中卡的操作

在主机和选择的 SD 卡之间的所有数据通信是点对点的方式。所有寻址命令都需要响应。

不同数据传输模式的关系如图 4-8 所示，使用如下步骤：

- 所有读数据命令可以在任何时候通过停止命令 (stop command,CMD12) 中止。数据传输将中止，卡回到传输状态 (Transfer State)。读命令有：块读命令 (CMD17)，多块读命令 (CMD18)，发送读保护 (CMD30)，发送 scr(ACMD51)，以及读模式的通用命令(CMD56)。
- 所有写数据命令可以在任何时候通过停止命令 (stop command,CMD12) 中止。在不选中卡命令 CMD7 前写命令必须停止。写命令有：块写命令 (CMD24 and CMD25)，写 CID (CMD26)，写 CSD(CMD27)，lock/unlock 命令(CMD42) 以及写模式通用命令(CMD56)。
- 一旦数据传输完成，卡将退出数据写状态并进入 Programming State(传输成功)或 Transfer State (传输失败)。
- 如果一个快写操作停止，而且最后一块长度和 CRC 是有效的，那么数据可以被操作 (programmed)。
- 卡可能提供块写缓冲。这意味着在前一块数据被操作时，下一块数据可以传送给卡。如果所有卡写缓冲已满，只要卡在 Programming State，DAT0 将保持低电平 (BUSY)。
- 写 CSD、CID、写保护和擦除时没有缓冲。这表明在卡因这些命令而处于忙时，不再接收其他数据传输命令。在卡忙时 DAT0 保持低电平，并处于 Programming State。实际上如果 CMD 和 DAT0 线分离，而且主机

占有的忙 DAT0 线和其他 DAT0 线分开，那么在卡忙时，主机可以访问其他卡。

- 在卡被编程（programming）时，禁止参数设置命令。参数设置命令包括：设置块长度（CMD16），擦除块开始（CMD32）和擦除块结束（CMD33）。
- 卡在操作时不允许读命令。
- 使用 CMD7 指令把另一个卡从 Stand-by 状态转移到 Transfer 状态不会中止擦除和编程（programming）操作。卡将切换到 Disconnect 状态并释放 DAT 线。
- 使用 CMD7 指令可以不选中处于 Disconnect 状态的卡。卡将进入 Programming 状态，重新激活忙指示。
- 使用 CMD0 或 CMD15 重置卡将中止所有挂起和活动的编程（programming）操作。这可能会破坏卡上的数据内容，需要主机保证避免这样的操作。

## 4.1 宽总线选择/不选择

宽总线（4 位总线宽度）操作模式通过 ACMD6 选择和不选择。在上电后或 GO\_IDLE（CMD0）命令后默认的总线宽度是 1 位。ACMD6 命令只在“tran state”有效，即只有在卡选中后（CMD7）总线宽度才能修改。

## 4.2 读数据格式

DAT 总线在没有数据传输时处于高电平。一个传输数据块包含一个起始位（LOW），接着连续的数据流。数据流包含有效数据（如果使用了 ECC 了还包括错误纠正位）。数据流以一个结束位（HIGH）结束。数据传输和时钟信号同步。

以块传输的有效数据包含 CRC 校验和。产生多项式是标准 CCITT 多项式。

采用了缩短的 BCH 码，d=4，有效数据长度最长为 2048 字节。CRC 校验和对每个 DAT 线单独计算并附加在每个数据块后。在宽总线模式操作（DAT0-DAT3）中，16 位的 CRC 校验对每个 DAT 分别计算。

**数据块读**

传输的基本单位是数据块，最大尺寸在 CSD 中定义（READ\_BL\_LEN）。开始和结束地址完全包含在一个物理数据块（如 READ\_BL\_LEN 定义）中的较小的块也可以传递。**CRC 附加在每个数据块的尾部用来保证数据传输的完整性。**CMD17（READ\_SINGLE\_BLOCK）开始一个块读操作，然后传输完成后进入 Transfer 状态。CMD18(READ\_MULTIPLE\_BLOCK)开始连续的块传输，直到停止命令。停止命令有一个执行延迟。在停止命令最后一位发送完以后数据传输停止。

如果主机使用累计长度不是块对齐的部分块，在第一个不对齐块的开始，卡会发现一个块未对齐错误，在状态寄存器中设置 ADDRESS\_ERR 错误，中止传输并等待（在 Data 状态）停止命令。

### 4.3 数据写格式

数据写传输格式类似于读格式。对于以块为单位的写数据传输，CRC 检验位附加到每个数据块。卡的每根数据线在接收到数据并在写操作前，执行 CRC 校验。

#### 数据块写

数据块写（CMD24-27,42,56(W)），一个或多个数据块从主机发送给卡，主机在每个数据块后附加 CRC 校验。数据块长度 WRITE\_BL\_LEN（512B）。如果 CRC 校验失败，卡将在 DAT 数据线上指示错误。传输的数据将被抛弃，而且后续传输的数据块（在多数据块写模式）也都会被忽略。

多数据块写命令比连续的单数据块写命令速度快。不允许部分块写（小于 512B）。

当主机试图在写保护区域写数据时写操作将中止。在这种情况下，卡在状态寄存器设置 WP\_VIOLATION 位，并忽略所有后续数据传输，并在 Receive-data 状态下等待停止命令。

对 CID 和 CSD 寄存器进行编程操作不需要实现设置块长度，传输的数据也是 CRC 保护的。如果 CSD 或 CID 寄存器一部分存储于 ROM 中，那么不可改变部分必须和接收缓冲中的相应部分内容保持一致。如果匹配失败，卡将报告一个错误，而且不改变任何寄存器内容。

接收到一个数据块并完成 CRC 校验后，卡将开始写，如果写缓冲满而且不能从一个新的 WRITE\_BLOCK 命令接收新数据时，DAT0 线保持为低电平。任何时候主机都可以通过 SEND\_STATUS(CMD13)命令获取卡的状态。状态位 READY\_FOR\_DATA 指示卡是否可以接收新数据或写操作还在进行中。主机通过 CMD7（选中另一个卡）不选中卡，这个操作可以把卡的状态编程 Disconnect 并释放 DAT 线而不中断写操作。当不选中卡时，如果编程还在进行而且写缓冲不可用时，将通过下拉 DAT 为低电平来重新激活忙信号。实际上，主机通过 interleaving 可以实现多个卡同时写操作，interleaving 过程可以通过在卡忙时访问其他卡实现。

### **预擦除设置优先于多数据块写操作**

设置多个写数据块的预擦除 (ACMD23) 可以使得接下来的多个数据块写操作比没有预先执行 ACMD23 的相同操作更快。主机可以通过该命令设置多少个数据块将在接下来的写操作中发送。如果在所有数据块发送给卡时中止了写操作（使用停止传输命令），残余写数据块的内容(指要写入新内容的数据块?)将变得不确定（可能已擦除或还是原来的数据）。如果主机发送了超过 ACMD23 中定义的数据块数目的数据，卡将逐个擦除数据块（在收到新数据时）。多数据块写操作完成后值将重新设置为默认值 1。

建议在 CMD25 命令前使用该命令以加速写操作。如果需要预擦除主机在写命令前发送 ACMD23。如果不发送 ACMD23 命令，设置的预擦除数将在其他指令执行时自动清除。

### **发送写数据块数目**

系统使用管道机制进行数据缓冲管理，有时候在多数据块写操作过程中发生错误，使得无法确定哪一个数据块是最后成功写入的数据块。卡可以把正常写入的数据块数 (the number of well-written blocks) 作为对命令 ACMD22 的响应。

### **擦除**

同时擦除多个写数据块可以提高数据吞吐量。通过 ERASE\_WR\_BLK\_START(CMD32)和 ERASE\_WR\_BLK\_END(CMD33)实现写数据块的识别。

主机必须严格按照下列的命令操作顺序：ERASE\_WR\_BLK\_START, ERASE\_WR\_BLK\_END, and ERASE (CMD38)。

如果不按顺序接收到擦除指令 (CMD38) 或地址设置指令 (CMD32,33), 卡将在状态寄存器中设置 ERASE\_SEQ\_ERROR 位, 并重置整个顺序(sequence)。

如果接收到一个不顺序的命令 (除了 SEND\_STATUS), 卡将在状态寄存器设置 ERASE\_RESET 状态位, 重置擦除顺序和执行最后的命令。

如果擦除范围包括写保护扇区, 将不被擦除, 擦除命令只擦除无保护的扇区。状态寄存器的 WP\_ERASE\_SKIP 位将设置。

地址设置命令中的地址是以字节为单位的块写地址。卡将忽略所有小于 WRITE\_BLK\_LEN (CSD) LSB (最低有效位)。

如上所述的块写操作, 卡通过保持 DAT0 为低电平指示擦除操作正在进行中。实际的擦除操作时间可能会很长, 主机可以通过 CMD7 不选中卡或执行卡断开操作。

卡上擦除操作后的数据为“0”或“1”, 由卡制造商确定。SCR 寄存器的 DATA\_STAT\_AFTER\_ERASE(bit55)定义了是“0”或“1”。

## 4.4 写保护管理

写保护方法如下:

- 机械写保护开关 (由主机负责)
- 卡内部写保护 (由卡负责)
- 密码保护锁操作

### 4.4.1 机械写保护开关 4.4.2 卡内部写保护 4.4.3 密码保护锁操作

## 5 时钟控制

SD 卡主机可以使用 SD 卡总线时钟信号设置卡进入节能模式或控制总线上的数据流。主机可以降低时钟频率或直接关闭。

SD 卡主机必须遵循下列约束:

- 总线频率可以在任何时候改变 (满足最大和最小值的约束)。

- ACMD41(SD\_APP\_OP\_COND)是一个例外。发送 ACMD41 命令后，主机将执行下面步骤 1 和步骤 2 直到卡进入就绪状态：
  - 1) 持续发送 100KHZ-400KHZ 之间的时钟频率。
  - 2) 如果主机要停止时钟，通过 ACMD41 命令以小于 50ms 的间隔设置 busy 位。

## 6 CRC 7 错误条件 7.1 CRC 和非法命令

## 8 命令 8.1 卡类型

共有四类用来控制 SD 卡的命令：

- 广播命令 (bc)，无响应——广播命令只有在所有 CMD 线一起连接到主机时才能使用。如果分开连接，那么每个卡将单独接收命令。
- 带响应的广播命令 (bcr) ——所有卡同时响应。因为 SD 卡没有开漏模式，这个命令只有在所有的 CMD 线分开时采用使用。该命令将被每个卡分别接收和响应。（OPEN DRAIN 输出只能做输出口，当外部无上拉电阻时，该口为高阻状态。只有外部有上拉电阻时，才有可能输出高或低的电平。）
- 寻址（点对点）命令 (ac) ——DAT 上没有数据传输。
- 寻址（点对点）数据传输命令 (adtc) ——DAT 上传输数据。

所有的命令和响应通过 CMD 线传输。

## 8.2 命令格式

命令长度 48 位，1.92us@25MHZ

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
起始位	host	命令	参数	CRC7	end bit

7-bit CRC 计算:  $G(x) = x^7 + x^3 + 1$

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

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

## 8.3 命令分类

SD 卡命令集分为几个类，每个类支持一个卡功能集合操作。

	0	1	2	3	4	5	6	7	8	9-11
支持命令	基本	保留	读块	保留	写块	擦除	写保护	锁	应用	保留
CMD0	+									
CMD2	+									
CMD3	+									
CMD4	+									
CMD7	+									
CMD9	+									
CMD10	+									
CMD12	+									
CMD13	+									
CMD15	+									
CMD16			+		+					
CMD17			+							
CMD18			+							
CMD24					+					
CMD25					+					
CMD27					+					
CMD28							+			
CMD29							+			
CMD30							+			
CMD32						+				
CMD33						+				
CMD38						+				

CMD42								+		
CMD55									+	
CMD56									+	
ACMD6									+	
ACMD13									+	
ACMD22									+	
ACMD23									+	
ACMD41									+	
ACMD42									+	
ACMD51									+	

## 8.4 详细命令描述

\*所有无用位必须填入，但是值 irrelevant

表 4-3 基本命令（类 0 和类 1）

命令索引	类型	参数	响应	缩写	描述
CMD0	bc	[31:0]无用	—	GO_IDLE_STATE	重置所有卡到 Idle 状态
CMD1	保留				
CMD2	bcr	[31:0]无用	R2	ALL_SEND_CID	要求所有卡发送 CID 号
CMD3	Bcr	[31:0]无用	R6	SEND_RELATIVE_ADDR	要求所有卡发布一个新的相对地址 RCA
CMD4	不支持				
CMD5	保留				
CMD6	保留				
CMD7	ac	[31:16]RCA	R1(只	选中/不选中卡	Command toggles a card between the

		[15:0]无用	来自 选中 的卡)		Stand-by and Transfer states or between the Programming and Disconnect state. In both cases the card is selected by its own relative address and deselected by any other address; address 0 deselects all. When the RCA equals 0, the host may do one of the following: —use other RCA number to perform card deselection Or —re-send CMD3 to change its RCA number to other than 0 and then use CMD7 with RCA=0 for card de-selection.
CMD8	保留				
CMD9	ac	[31:16] RCA [15:0]无用	R2	SEND_CSD	寻址卡并让其发送卡定义数据 CSD
CMD10	ac	[31:16] RCA [15:0]无用	R2	SEND_CID	寻址卡并让其发送卡识别号 CID
CMD11	adtc	[31:0]数据 地址	R1	READ_DAT_UNTIL_STOP	从卡读取数据流, 从给定地址开始, 知

					道停止传输 命令结束
CMD12	ac	[31:0]无用	R1b	STOP	中止多个块的读/写操作
CMD13	ac	[31:16] RCA [15:0]无用	R1	SEND_STATUS	寻址卡并发 送卡状态寄 存器
CMD14	保留				
CMD15	ac	[31:16] RCA [15:0]无用	_	GO_INACTIVE_STATE	设置卡到 inactive 状态

表 4-4 块读操作命令（类 2）

命令索引	类型	参数	响应	缩写	描述
CMD16	ac	[31:0]块 长度	R1	SET_BLOCKLEN	为接下来的 块操作指令 设置块长度
CMD17	adtc	[31:0]数 据地址	R1	READ_SINGLE_BLOCK	读取一个块
CMD18	adtc	[31:0]数 据地址	R1	READ_MULTIPLE_BLOCK	连续读取多 个块，直到停 止命令
CMD19-23	保留				

表 4-5 块写操作命令（类 4）

命令索引	类型	参数	响应	缩写	描述
CMD24	adtc	[31:0]	R1	WRITE_BLOCK	写一个长度由

		数据地址			SET_BLOCKLEN 指定的块
CMD25	adtc	[31:0] 数据地址	R1	WRITE_MULTIPLE_BLOCK	连续写多个块直到 STOP_TRANSMISSION 命令
CMD26	不支持				
CMD27	adtc	[31:0] 无用	R1	PROGRAM_CSD	编辑 CSD 位

表 4-6 写保护 (类 6)

命令索引	类型	参数	响应	缩写	描述
CMD28	ac	[31:0] 数据地址	R1b	SET_WRITE_PROT	设置地址组保护位。 写保护由卡配置数据的 WP_GRP_SIZE 指定
CMD29	ac	[31:0] 数据地址	R1b	CLR_WRITE_PROT	清除保护位
CMD30	adtc	[31:0] 写保护数据地址	R1	SEND_WRITE_PROT	要求卡发送写保护位状态
CMD31	保留				

表 4-7 擦除命令 (类 5)

命令索引	类型	参数	响应	缩写	描述
CMD32	ac	[31:0] 数据	R1	ERASE_WR_BLK_START	设置要擦除的第一个写数据块地址

		地址			址
CMD32	ac	[31:0] 数据 地址	R1	ERASE_WR_BLK_END	设置要擦除的最后一个写数据块地址
CMD34 ... CMD37	保留				
CMD38	ac	[31:0] 无用	R1b	ERASE	擦除所有选中的写数据块
CMD39 ... CMD41	保留				

表 4-7 擦除命令（类 5）

命令索引	类型	参数	响应	缩写	描述
CMD32	ac	[31:0] 数据 地址	R1	ERASE_WR_BLK_START	设置要擦除的第一个写数据块地址
CMD32	ac	[31:0] 数据 地址	R1	ERASE_WR_BLK_END	设置要擦除的最后一个写数据块地址
CMD34 ... CMD37	保留				
CMD38	ac	[31:0] 无用	R1b	ERASE	擦除所有选中的写数据块
CMD39 ...	保留				

CMD41	
-------	--

表 4-8 卡锁命令 (类 7)

命令索引	类型	参数	响应	缩写	描述
CMD42 .... CMD54	SDA 可选命令				

表 4-9 应用相关 (Application Specific) 命令 (类 8)

命令索引	类型	参数	响应	缩写	描述
CMD55	ac	[31:16] RCA [15:0]填充位	R1	APP_CMD	告诉卡接下来的命令是应用相关命令,而非标准命令。
CMD56	adtc	[31:1] 填充位 [0]:RD/WR, 1 读, 0 写	R1	GEN_CMD	应用相关(通用目的)的数据块读写命令
CMD57 ... CMD59	保留				
CMD60 ... CMD63	厂商保留				

\*命令相关命令, 可能指 SD 卡专用命令

所有应用相关命令之前必须先执行 APP\_CMD(CMD55)。

表 4-10 SD 卡使用/保留的应用相关命令

ACMD 索引	类 型	参 数	响 应	缩 写	描 述
ACMD6	ac	[31:2] 填充位 [1:0]总线宽度	R 1	SET_BUS_WIDTH	00:1bit 10:4bit
ACMD1 3	adt c	[31:0] 填充位	R 1	SD_STATUS	设置 SD 卡状态
ACMD1 7	保留				
ACMD1 8	—	—	—	—	保留作为 SD 安全应用
ACMD1 9 ... ACMD2 1	保留				
ACMD2 2	- adt c	[31:0]填充位	R 1	SEND_NUM_WR_BLOCKS	发送写数据块的数目。响应为 32 位 +CRC
ACMD2 3	ac	[31:23]填充位 [22:0]数据块数目	R 1	SET_WR_BLK_ERASE_COUNT	设置写前预擦除的数据块数目(用来加速多数据块写操作)。 “1” =默认(一个块)(1)
ACMD2 4	保留				

ACMD2 5	—	—	—	—	保留作为 SD 安全应用
ACMD2 6	—	—	—	—	保留作为 SD 安全应用
ACMD3 8	—	—	—	—	保留作为 SD 安全应用
ACMD3 9 ... ACMD4 0	保留				
ACMD4 1	bcr	[31:0]OC R without busy	R 3	SD_APP_OP_COND	要求访问的 卡发送它的 操作条件寄 存器(OCR) 内容
ACMD4 2	ac	[31:1]填 充位 [0]set_cd	R 1	SET_CLR_CARD_DETECT	连接[1]/断开 [0]卡上 CD/DAT3(pi n 1)的 50K 欧 姆上拉电阻。 上拉电阻可 用来检测卡
ACMD4 3 ACMD4 9	—	—	—	—	保留作为 SD 安全应用
ACMD5 1	- adt	[31:0]填 充位	R 1	SEND_SCR	读取 SD 配置 寄存器 SCR

	c				
--	---	--	--	--	--

(1)不管是否使用 ACMD23，在多数据块写操作中都需要 STOP\_TRAN(CMD12) 命令

## 9 卡状态迁移表 10 响应

所有响应通过 CMD 线传输，响应以 MSB 开始，不同类型的响应长度根据类型不同而不同。

响应以起始位开始(通常为“0”)，接着这是传输方向的位(卡为0)。除了 R3 外其他响应都有 CRC。每个响应都以结束位(通常为“1”)结束。

共有四类响应，格式分别为：

**R1(标准响应)**：长度 48 位

Bits45:40 指示被响应的命令索引号。如果有到卡的数据传输，每个数据块传输后数据线上都会出现忙信号。主机在数据块传输后检测忙信号。

表 4-12 响应 R1

起始位	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
值	0	0	x	x	x	1
描述	起始位	传输位	命令索引	卡状态	CRC7	结束位

**Rb1** 和 R1 相同，带有一个可选的忙信号传输。根据接收到命令前的状态和接收到的命令可能变成忙。主机可以在响应时检测忙信号。

**R2(CID,CSD)**：响应长度为 136 位

CID 寄存器内容作为 CMD2 和 CMD10 的响应发送。CSD 寄存器内容作为 CMD9 的响应发送。只传输 CID 和 CSD 的[127...1]位，寄存器的[0]位被响应的结束位取代。

表 4-13 响应 R1

起始位	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
值	0	0	‘111111’	x	1
描述	起始位	传输位	保留	CID 或 CSD，包括内容 CRC7	结束位

**R1(OCR 寄存器)**：长度 48 位

OCR 寄存器作为 ACMD41 的响应发送。





#### 四、 在 S3C2410 中的使用 1 基本操作 1.1 SDI 操作

根据传输频率设置 SDIPRE 寄存器，改变频率可以调整波特率。

操作步骤：

- 1) 设置 SDICON，设置时钟和中断
- 2) 设置 SDIPRE
- 3) 等待 74SDCLK，初始化卡

### 1.2 CMD Path Programming

- 1) 写命令参数（32-bit）到 SDICARG 寄存器
- 2) 通过设置 SDICCON[8]确定命令类型和起始命令
- 3) 当 SDICSTA 的特殊标记设置时，确定 SDI 命令操作的结束方式：
  - a) 如果是无响应（no-response）命令，标记是 SDICSTA[11]
  - b) 如果有响应（with-response）命令，标记是 SDICSTA [9]
- 4) 清除 SDICSTA 寄存器相应标记

### 1.3 DAT Path Programming

- 1) 在 SDIDTIMER 寄存器中设置超时
- 2) 在 SDIBSIZE 中设置块大小，通常为 0x200
- 3) 在 SDIDCON 寄存器中设置块模式（mode of block）、总线宽度(bus width)、DMA 等以及开始数据传送
- 4) 通过检查 SDIFSTA 判断 Tx FIFO 的可用性（available,half or empty），当可用时(available)写 Tx-data 到 SDIDAT 寄存器

- 5) 通过检查 SDIFSTA 判断 Rx FIFO 的可用性 (available, half or empty), 当可用时(available)从 SDIDAT 寄存器读数据到 Rx-data
- 6) 当传输结束标记 SDIDSTA[4]置位时, 确定 SDI 数据操作的数据
- 7) 清除 SDIDSTA 寄存器相应位

## 2 SDIO 操作

两类 SDIO 操作: SDIO 中断接收和读等待请求, 分别通过 SDICON 寄存器的 RcvIOInt 位和 RwaitEn 位标记。

### 2.1 SDIO 中断

在 SD1 位模式, 中断从 SDDAT1 脚接收所有范围的中断。

在 SD4 位模式, SDDAT1 脚共享为接收数据和中断。中断检测范围 (Interrupt Period) :

- 1) 单数据块: A 和 B 之间的间隔
  - A: 数据组装 (packet) 结束后 2 个时钟周期
  - B: 下一条带数据命令 (with-data command) 最后一位发送结束
- 2) 多数据块, SDIDCON[21]=0: A 和 B 之间的时间, 在范围 C 检测重启 (restart) 中断
  - A: 数据组装结束后 2 个时钟周期
  - B: A 后 2 个时钟周期
  - C: 中止命令响应最后一位后 2 个时钟
- 3) 多数据快, SDIDCON[21]=1: A 和 B 之间, 在 A 重启
  - A: 数据组装结束后 2 个时钟周期
  - A 后 2 个时钟周期

## 2.2 读等待请求 (Read Wait Request)

无论 1 位还是 4 位模式，读等待请求信号在下述情况下通过 SDDAT2 脚传送：

- 1) 在读多块操作，请求信号在数据块结束后 2 个时钟开始传输
- 2) 传输在用户向 SDIDSTA[10]中写 1 后结束

## 3 寄存器 3.1 SDICON

寄存器名	地址	R/W	描述	重置值
SDICON	0x5A000000	R/W	SDI 控制寄存器	0x0

SDICON	位	描述	初始值
字节顺序类型 (ByteOrder)	[4]	确定读写时的字节顺序类型 0=Type A, 1=Type B - Type A: D[7:0] → D[15:8] → D[23:16] → D[31:24] - Type B: D[31:24] → D[23:16] → D[15:8] → D[7:0]	0
从卡接收 SDIO 中断 (RcvIOInt)	[3]	确定 SD 主机是否接收卡 (或 SDIO) 的 SDIO 中断 0:不接收                    1:接收	0
读等待使能 (RWaitEn)	[2]	在多数数据块读模式中当 SD 主机等待下一个数据块时是否产生读等待请求信号。这一位需要延迟从卡传输下一数据块。 0: 不产生                    1:产生	0
FIFO 重置 (FRST)	[1]	重置 FIFO 值，该位自动清除 0: 普通模式                1:FIFO 重置	0
Clock out Enable (ENCLK)	[0]	是否使能 SDCLK Out 0: 禁止(不分频)        1:使能	0

## 3.2 SDIPRE(SDI 模特率分频寄存器)

寄存器名	地址	R/W	描述	重置值
SDIPRE	0x5A000004	R/W	SDI 模特率分频寄存器	0x0

SDIPRE 名	位	描述	初始值
分频值	[7:0]	确定 SDI 时钟率： Baud rate=PCLK/2/(prescaler value+1)	0x00

### 3.3 SDICARG(SDI 命令参数寄存器)

寄存器名	地址	R/W	描述	重置值
SDICARG	0x5A000008	R/W	SDI 命令参数寄存器	0x0

SDICARG	位	描述	初始值
CmdArg	[31:0]	命令参数	0x00000000

### 3.4 SDICCON(SDI 命令控制寄存器)

寄存器名	地址	R/W	描述	重置值
SDICCON	0x5A00000C	R/W	SDI 命令控制寄存器	0x0

SDICCON	位	描述	初始值
中止命令 (AbortCmd)	[12]	确定是否是中止命令类型 0: 普通命令      1:中止命令 (CMD12,CMD52)	0
带数据命令 (WithData)	[11]	确定命令类型是否是带数据命令 0: 否, 无数据      1:带数据	0
LongRsp	[10]	确定主机是否接收 136 位长的响应 0: 短响应格式      1:长响应格式	0
WaitRsp	[9]	确定主机是否等待响应 0: 不等待      1:等待	0
命令开始 (CMST)	[8]	确定命令是否开始操作 0: 命令就绪      1:命令开始	0
CmdIndex	[7:0]	带 2 个开始位的命令索引	0x00

### 3.5 SDICSTA(SDI 命令状态寄存器)

寄存器名	地址	R/W	描述	重置值
SDICSTA	0x5A000010	R/W	SDI 命令状态寄存器	0x0

SDICSTA	位	描述	初始
---------	---	----	----

			值
响应 CRC 失败 (RspCrc)	[12] R/W	当接收到命令响应时 CRC 校验是否失败, 该位通过写 1 清 0 0: 无          1:失败	0
命令发送 (CmdSent)	[11] R/W	命令发送 (不管是否响应), 该位通过写 1 清 0 0: 无          1:命令结束	0
命令超时 (CmdTout)	[10] R/W	命令响应超时 (64clk), 该位通过写 1 清 0 0: 无          1:超时	0
响应接收结束 (RspFin)	[9] R/W	响应接收结束, 该位通过写 1 清 0 0: 无          1:响应结束	0
CMD 线忙 (CmdOn)	[8] R	命令传输操作中 0: 无          1:操作中	0
RspIndex	[7:0] R	带 2 个开始位的响应命令索引	0x00

### 3.6 SDIRSP0(SDI 响应寄存器 0)

寄存器名	地址	R/W	描述	重置值
SDIRSP0	0x5A000014	R	SDI 响应寄存器 0	0x0

SDIRSP0	位	描述	初始值
Response0	[31:0]	卡状态[31:0](短格式); 卡状态[127:96](长格式)	0x00000000

### 3.7 SDIRSP1(SDI 响应寄存器 1)

寄存器名	地址	R/W	描述	重置值
SDIRSP1	0x5A000018	R	SDI 响应寄存器 1	0x0

SDIRSP1	位	描述	初始值
RCRC7	[31:24]	CRC7(带结束位, 短格式); 卡状态[95:88](长格式)	0x00
Response1	[23:0]	未用(带结束位, 短格式); 卡状态[87:64](长格式)	0x000000

### 3.8 SDIRSP2(SDI 响应寄存器 2)

寄存器名	地址	R/W	描述	重置值
------	----	-----	----	-----

SDIRSP2	0x5A00001C	R	SDI 响应寄存器 2	0x0
---------	------------	---	-------------	-----

SDIRSP1	位	描述	初始值
Response2	[31:0]	未用(带结束位, 短格式); 卡状态[63:32](长格式)	0x00000000

### 3.9 SDIRSP3(SDI 响应寄存器 3)

寄存器名	地址	R/W	描述	重置值
SDIRSP3	0x5A000020	R	SDI 响应寄存器 3	0x0

SDIRSP1	位	描述	初始值
Response3	[31:0]	未用(带结束位, 短格式); 卡状态[31:0](长格式)	0x00000000

### 3.10 SDIDTIMER(SDI 数据/忙时钟)

寄存器名	地址	R/W	描述	重置值
SDIDTIMER	0x5A000024	R/W	SDI 数据/忙时钟寄存器	0x0

SDIDTIMER	位	描述	初始值
DataTimer	[15:0]	数据/忙超时周期 (0~65535 周期)	0x2000

### 3.11 SDIBSIZE(SDI 块大小寄存器)

寄存器名	地址	R/W	描述	重置值
SDIBSIZE	0x5A000028	R/W	SDI 块大小寄存器	0x0

SDIBSIZE	位	描述	初始值
BlkSize	[11:0]	块大小值(0~4095byte), 在流模式中没用。在多块传输中, BlkSize 必须是字对齐的, 即 BlkSize[1:0]=00	0x000

### 3.12 SDIDCON(SDI 数据控制寄存器)

寄存器名	地址	R/W	描述	重置值
SDIDCON	0x5A00002C	R/W	SDI 数据控制寄存器	0x0

SDIDCON	位	描述	初始值
SDIO 中断周期(Period)类型 PryType	[21]	确定最后一个块传输时 SDIO 中断周期是 2 个时钟周期(cycle)还是更多周期 0: 2 个周期          1:更多周期(如单块传输)	0
响应后传输 TARSP	[20]	确定是否在接收到响应后开始数据传输 0: DatMode 设置后开始 1:接收到响应后开始(假设 DatMode 设置为 11)	0
命令后数据接收方式 RACMD	[19]	确定何时数据开始接收 0: DatMode 设置后开始 1:命令发送后开始(假设 DatMode 设置为 10)	0
命令后忙信号接收方式 BACMD	[18]	确定何时开始接收忙信号 0: DatMode 设置后开始 1:命令发送后开始(假设 DatMode 设置为 01)	0
块模式 BlkMode	[17]	数据传输模式 0: 流模式          1:块模式	0
宽总线使能 WideBus	[16]	使能宽总线 0:标准总线(只使用 SDIDAT[0]) 1:宽总线模式(使用 SDIDAT[3:0])	0
DMA 使能 EnDMA	[15]	使能 DMA 0: 禁止(polling)          1:允许 *DMA 操作结束后, 需要清除该位	0
强制停止 (STOP)	[14]	确定是否允许强制停止 0:标准 (normal)          1:强制停止	0
数据传输模式 DatMode	[13:12]	数据传输方向 00=ready                  01=only busy check start 10=data receive start      11=data transmit start	00
BlkNum	[11:0]	块数目(0~4095),在流模式中无用	0x000

### 3.13 SDIDCNT(SDI 数据剩余计数寄存器)

寄存器名	地址	R/W	描述	重置值
SDIDCNT	0x5A000030	R	SDI 数据剩余计数寄存器	0x0

SDIDCNT	位	描述	初始值
BlkNumCnt	[23:12]	剩余块数目	0x000
BlkCnt	[11:0]	1 块中的剩余字节	0x000

### 3.14 SDIDSTA(SDI 数据状态寄存器)

寄存器名	地址	R/W	描述	重置值
SDIDSTA	0x5A000034	R/(W)	SDI 数据状态寄存器	0x0

SDIDSTA	位	描述	初始 值
读等待请求 (RWaitReq)	[10] R/W	读等待请求信号发送给 SD 卡。通过写 1 清 0 清除该位，并停止请求。写 1 清 0。 0=未发生 (occur)      1=读等待请求发生	0
SDIO 中断检测 (IOIntDet)	[9] R/W	中断检测，该位通过写 1 清 0 0=没有检测到      1= 检测到中断	0
FIFO 失败错误 (Fffail)	[8] R/W	是否 FIFO 出现满栈写错误、空栈读错误、未对准数据保存等错误。写 1 清 0。 0: 无      1:FIFO fail	0
CRC 状态错误 (CrcSta)	[7] R/W	数据块发送时出现 CRC 状态错误（从卡返回的 CRC 校验失败）。写 1 清 0。 0: 无      1=CRC status fail	0
数据接收 CRC 错误 (DatCRC)	[6] R/W	数据块接收错误（主机 CRC 校验失败）。写 1 清 0。 0: 无      1=CRC status fail	0
数据超时 (DatTout)	[5] R/W	数据/忙接收超时。写 1 清 0。 0: 无      1=超时	0
数据传输结束 (DatFin)	[4] R/W	数据传输结束，计数为 0。写 1 清 0。 0: 无      1=数据结束	0
忙结束 (BusyFin)	[3] R/W	忙检测结束。写 1 清 0。 0: 无      1=忙结束	0
开始位错误 (SbitErr)	[2] R/W	在宽总线模式中在所有的信号中没有检测到起始位。写 1 清 0。 0: 无      1=命令结束，即无开始位	0
Tx 数据操作进行中 (TxDatOn)	[1] R	数据传输进行中 0: 没有      1=进行中	0
Rx 数据操作进行中 (RxDatOn)	[0] R	数据接收进行中 0: 没有      1=进行中	0

### 3.15 SDIFSTA (SDI 数据 FIFO 状态寄存器)

寄存器名	地址	R/W	描述	重置值
SDIFSTA	0x5A000038	R	SDI 数据 FIFO 状态寄存器	0x0

SDIFSTA	位	描述	初始值
传输时检测 FIFO 是否可用 (TFDET)	[13]	当 DatMode(SDIDCON[12])设置为数据传输模式,指示 FIFO 数据是否可用。如果使能了 DMA 模式, SD 主机请求 DMA 操作。 0=没有发现 (FIFO 满) 1=发现 ( $0 \leq \text{FIFO} \leq 63$ )	0
接收时检测 FIFO 是否可用 (RFDET)	[12]	当 DatMode(SDIDCON[12])设置为数据接收模式,指示 FIFO 是否可用来接收。如果使能了 DMA 模式, SD 主机请求 DMA 操作。 0=没有发现 (FIFO 空) 1=发现 ( $0 \leq \text{FIFO} \leq 64$ )	0
Tx FIFO 半满 (TFHalf)	[11]	Tx FIFO 数据少于 33 个字节时置为 1 $0=33 \leq \text{Tx FIFO} \leq 64$ $1=0 \leq \text{Tx FIFO} \leq 32$	0
Tx FIFO 空 (TFEmpty)	[10]	Tx FIFO 为空时置为 1 $0=1 \leq \text{Tx FIFO} \leq 64$ $1=\text{Empty}$	0
Rx FIFO 最后数据就绪 (RFLast)	[9]	当 Rx FIFO 有所有数据块的最后数据时 0=还没有接收到 1=最后数据就绪	0
Rx FIFO FULL (RFFull)	[8]	当 Rx FIFO 满时置为 1 $0=0 \leq \text{Rx FIFO} \leq 63$ $1=\text{Full}(64\text{byte})$	0
Rx FIFO Half FULL (RFHalf)	[7]	当 Rx FIFO 超过 31byte 时置为 1 $0=0 \leq \text{Rx FIFO} \leq 31$ $1=32 \leq \text{Rx FIFO} \leq 64$	0
FIFO 计数 (FIFOCNT)	[6:0]	FIFO 中数据的字节数	0000000

### 3.16 SDIDAT(SDI 数据寄存器)

寄存器名	地址	R/W	描述	重置值
SDIDAT	0x5A00003C(Li/W, Li/B, Bi/W) 0x5A00003F(Bi/B)	R/W	SDI 数据寄存器	0x0

1. (Li/W, Li/B): 小端模式以 Word/Byte 为单位

2. (Bi/W): 大端模式以 Word 为单位

3. (Bi/B): 大端模式以 Byte 为单位

SDIDAT	位	描述	初始值
数据寄存器	[31:0]	包含通过 SDI 通道传输和接收的数据	0x00000000

### 3.17 SDIIMSK(SDI 中断屏蔽寄存器)

寄存器名	地址	R/W	描述	重置值
SDIIMSK	0x5A000040	R/W	SDI 中断屏蔽寄存器	0x0

SDIIMSK	位	描述	初始值
RspCrc 中断使能	[17]	响应 CRC 错误中断 0 = 禁止, 1 = 使能	0
CmdSent 中断使能	[16]	命令发送(无响应) 中断. 0 = 禁止, 1 = 使能	0
CmdTout 中断使能	[15]	命令响应超时中断. 0 = 禁止, 1 = 使能	0
RspEnd 中断使能	[14]	命令响应接收中断, 收到时触发 0 = 禁止, 1 = 使能	0
RWaitReq 中断使能	[13]	读等待请求中断. 0 = 禁止, 1 = 使能	0
IOIntDet 中断使能	[12]	SD 主机从卡接收到 SDIO 中断 0 = 禁止, 1 = 使能	0
FFail 中断使能	[11]	FIFO 错误中断. 0 = 禁止, 1 = 使能	0
CrcSta 中断使能	[10]	CRC 状态错误中断. 0 = 禁止, 1 = 使能	0
DatCrc 中断使能	[9]	数据 CRC 错误中断. 0 = 禁止, 1 = 使能	0
DatTout 中断使能	[8]	数据超时 中断. 0 = 禁止, 1 = 使能	0
DatFin 中断使能	[7]	数据计数为 0 中断. 0 = 禁止, 1 = 使能	0
BusyFin 中断使能	[6]	忙检测结束中断. 0 = 禁止, 1 = 使能	0
SBitErr 中断使能	[5]	起始位错误中断. 0 = 禁止, 1 = 使能	0
TFHalf 中断使能	[4]	Tx FIFO 半满中断. 0 = 禁止, 1 = 使能	0
TFEmpty 中断使能	[3]	Tx FIFO 空中断. 0 = 禁止, 1 = 使能	0

RFLast 中断 使能	[2]	Rx FIFO 有最后数据中断. 0 = 禁止, 1 = 使能	0
RFFull 中断 使能	[1]	Rx FIFO 满中断. 0 = 禁止, 1 = 使能	0
RFHalf 中断 使能	[0]	Rx FIFO 半满中断. 0 = 禁止, 1 = 使能	0

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Single Board Computers](#) category:*

*Click to view products by [Waveshare](#) manufacturer:*

Other Similar products are found below :

[MANO882VPGGA-H81](#) [SSD3200W-S-SLC-INN](#) [AmITX-SL-G-Q170](#) [IB100](#) [MVME61006E-2173R](#) [20-101-0738](#) [PCE-4128G2-00A1E](#)  
[RSB-4220CS-MCA1E](#) [SHB230DGGA-RC](#) [IB909AF-5650](#) [AmITX-BT-I-E3815](#) [PICO841VGA-E3827](#) [IMB210VGGA](#) [MI981AF](#) [RSB-4221CS-MCA1E](#) [PCE-9228G2I-00A1E](#) [IB915F-3955](#) [IB909F-5010](#) [MI958F-16C](#) [UPS-P-8G-64GB-PACK](#) [S2600WFT](#) [IB915AF-6300](#)  
[S2600STB](#) [BBS2600BPS](#) [IB915F-6100](#) [Nit6QP\\_MAX](#) [MI990VF-X28-E](#) [MI990VF-6820](#) [MI991AF-C236](#) [94AC6636](#) [BANANA PI BPI-M4](#)  
[BLKNUC7I3DNHNC1978015](#) [BLKNUC7I5DNK1E 960791](#) [IOT-LS1012A-OXALIS](#) [NITX-300-ET-DVI](#) [94AC6633](#) [A33-OLINUXINO-N8G](#) [A64-OLINUXINO-1GE16GW](#) [A20-SOM-E16GS16M](#) [A20-SOM204-1G-M](#) [EMB-APL1-A10-3350-F1-LV](#) [PICO-APL1-A10-F001](#)  
[PICO-APL4-A10-F003](#) [ODYSSEY - STM32MP157C BOARD WITH SOM](#) [BEAGLEBONE GREEN GATEWAY DEV BOARD](#) [ODYSSEY - X86J4105864 8GB RAM 64GB EMMC](#) [ODYSSEY -X86J4105864 8GB/64GB ENTERPRISE](#) [VISIONDK-STM32MP1 V.1.0](#) [VISIONDK-6ULL V.2.0](#) [VISIONDK-8MMINI V.1.0](#)