1.1 SD 卡的介绍

SD 卡（Secure Digital Memory Card）是一种基于半导体快闪记忆器的新一代记忆设备，它被广泛地于便携式装置上使用，例如数码相机、个人数码助理(PDA)和多媒体播放器等。SD 卡由日本松下、东芝及美国 SanDisk 公司于 1999 年 8 月共同开发研制。

按容量分类，可以将 SD 卡分为 3 类： SD 卡、 SDHC 卡、 SDXC 卡。

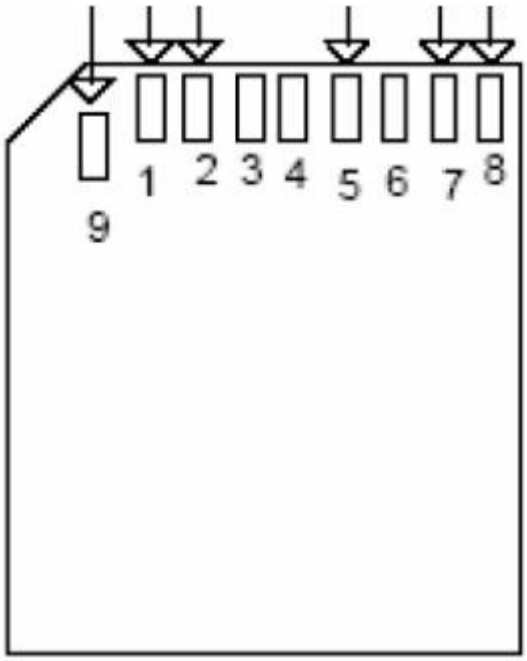
容量	命名	简称
0~2G	Standard Capacity SD Memory Card	SDSC 或 SD
2G~32G	High Capacity SD Memory Card	SDHC
32G~2T	Extended Capacity SD Memory Card	SDXC

SD 卡一般支持 2 种操作模式：

1， SD 卡模式（通过 SDIO 通信）--速度快；

2， SPI 模式——速度比较慢；

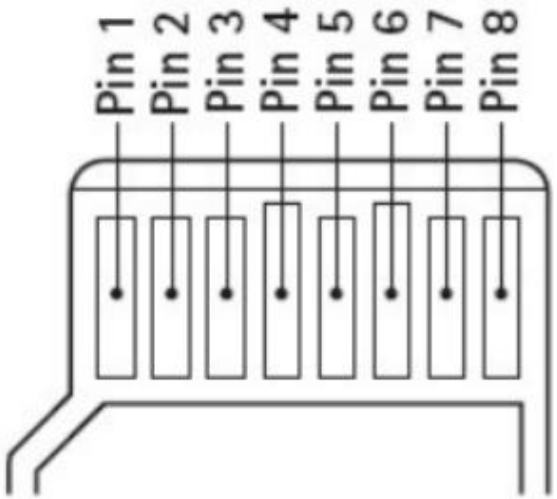
主机可以选择以上任意一种模式同 SD 卡通信， SD 卡模式允许 4 线的高速数据传输。 SPI 模式允许简单的通过 SPI 接口来和 SD 卡通信，这种模式同 SD 卡模式相比就是丧失了速度。



SD 卡引脚功能描述如下：

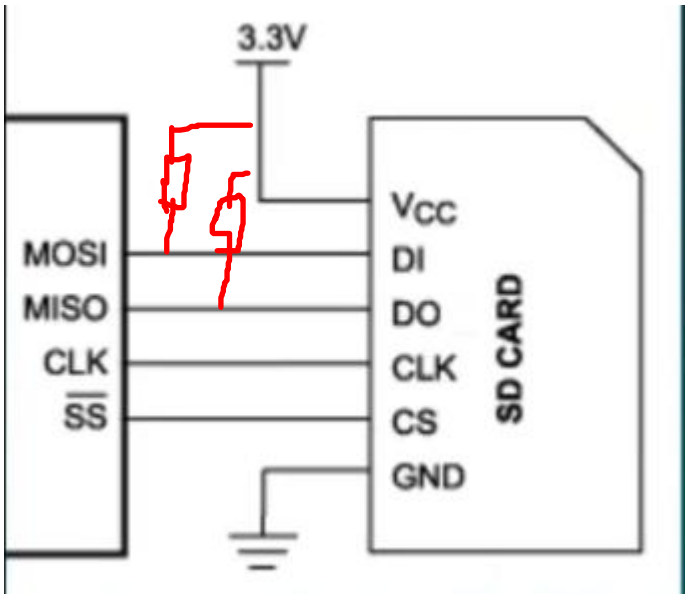
引脚	SD 模式		SPI 模式	
	名称	描述	名称	描述
1	CD/DAT3	卡检测 /数据线（位 3）	CS	芯片选择
2	CMD	命令响应	DI MOSI	数据输入
3	VSS1	电源地 1	VSS1 GND	电源地 1
4	VDD	提供电压	VDD VCC	提供电压
5	CLK	时钟	SCLK	时钟
6	VSS2	电源地 2	VSS2	电源地 2
7	DAT0	数据线（位 0）	DO MISO	数据输出
8	DAT1	数据线（位 1）	RSV	保留
9	DAT2	数据线（位 2）	RSV	保留

Micro SD　引脚示意图及模式区别：



引脚	SD 模式		SPI 模式	
	名称	描述	名称	描述
1	DAT2	数据线（位 2）	RSV	保留
2	CD/DAT3	卡检测 /数据线 （位 3）	CS	芯片选择
3	CMD	命令响应	DI	数据输入
4	VDD	提供电压	VDD	提供电压
5	CLK	时钟	SCLK	时钟
6	VSS	电源地	VSS	电源地
7	DAT0	数据线（位 0）	DO	数据输出
8	DAT1	数据线（位 1）	RSV	保留

SPI 接线方式,在此模式下，CS/MOSI/MISO/CLK 都需要加 10~100K 左右的上拉电阻。



1.1.1 1. SD 卡寄存器介绍 -SDIO

（1） **1.1.SD 卡有 6 个寄存器**

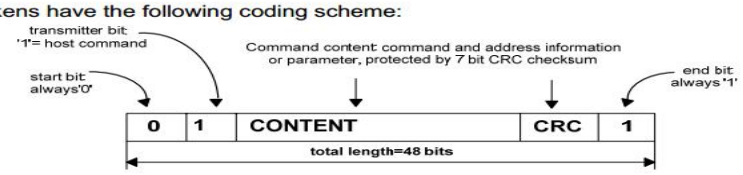
SD 卡有 6 个寄存器这些寄存器的详细介绍，请参考《SD 卡 2.0 协议.pdf》第五章 P84.，中文参考 STM32_SDIO 应用_SD 卡.pdf

名称	宽度	描述
CID	128	卡标识寄存器
RCA	16	相对卡地址（Relative card address）寄存器: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

（2） 1.2.SD 卡命令格式 重点

手册 P19

Command tokens have the following coding scheme:



、SD 命令格式

位	47	46	[45:40]	[39:8]	[7:1]	0
宽度	1	1	6	32	7	1
数值	0	1	-	-	-	1
说明	开始位	传输位	命令索引	参数	CRC7	结束位

SD 卡的指令由 6 个字节组成，字节 1 的最高 2 位固定为 01，低 6 位为命令号.字节 2~5 为命令参数，有些命令是没有参数的。

字节 6 的高七位为 CRC 值，最低位恒定为 1。 SPI通信使用，SDIO不看这部分

参考代码:SD_SendCmd(u8 cmd, u32 arg, u8 crc)

u8 cmd --> 8 位; u32 arg--> 32 位; u8 crc -->8 位; 刚好 48 位

现在我要发送 CMD25,那么是怎么发送的???

首先是把 CMD25 中的 十进制 25 转化为 十六进制为 0x19-->即 0001 1001

所以第一个字节应该为 0001 1001+0100 0000,因此可以得到字节 1 应该是按下面这样来发送的 SD_SPI_Read_Write_Byte(cmd | 0x40);

如 :cmd 为 CMD25 ; 0X40-> 0100 0000 即第一个字节的第 6 位为 1.

接着发送 32 位的命令参数,最后发送的是 CRC 校验(参考程序).

SD_SendCmd(CMD0,0,0x95);

0x95 怎么得到?? 手册 P54 \ P107

- **CRC7 Examples**

The CRC section of the command/response is bolded. 0x95

CMD0 (Argument=0) --> 01 000000 00000000000000000000000000000000 **"1001010"** 1
CMD17 (Argument=0) --> 01 010001 00000000000000000000000000000000 **"0101010"** 1
Response of CMD17 --> 00 010001 00000000000000000000100100000000 **"0110011"** 1

Since CMD0 has no arguments, the content of all the fields, including the CRC field, are constants and need not be calculated in run time. A valid reset command is:

0x40, 0x0, 0x0, 0x0, 0x0, 0x95

After the card is put into SPI mode, CRC check for all commands including CMD0 will be done

（3） **1.3.SD 卡的命令**

SD 卡的命令总共有 12 类，分为 Class0~Class11（SPI MODE），SD 卡 2.0 协议 P112 以下只是其中的几处命令 .其他参考 SD 卡 2.0 协议 P112

命令	参数	回应	描述
CMD0(0X00)	NONE	R1	复位 SD 卡
CMD8(0X08)	VHS+Check pattern	R7	发送接口状态命令
CMD9(0X09)	NONE	R1	读取卡特定数据寄存器
CMD10(0X0A)	NONE	R1	读取卡标志数据寄存器
CMD16(0X10)	块大小	R1	设置块大小（字节数）
CMD17(0X11)	地址	R1	读取一个块的数据
CMD24(0X18)	地址	R1	写入一个块的数据
CMD41(0X29)	NONE	R3	发送给主机容量支持信息和激活卡初始化过程
CMD55(0X37)	NONE	R1	告诉 SD 卡，下一个是特定应用命令
CMD58(0X3A)	NONE	R3	读取 OCR 寄存器

3.1 CMD12 指令说明　P115

命令编号	参数	响应	缩写	说明
CMD12	[31:0]无效	R1b	STOP_TRANSMISSION	强制结束当前SD的数据传输。用于多块数据读取时，结束传输用。 注意：该指令仅用于结束多块数据读取。

3.2 CMD16 指令说明　P115

命令编号	参数	响应	缩写	说明
CMD16	[31:0]块长度(单位：字节)	R1	SET_BLOCKLEN	该指令设置SD卡的块大小(字节)，该指令设置后，后续所有的块操作指令(读/写)，都是以该指令设置的块大小为基准的。 注意：对高容量SD卡，块大小固定为512字节，不受此指令影响。

3.3 CMD17 指令说明　P115

命令编号	参数	响应	缩写	说明
CMD17	[31:0]数据地址（单位：字节（SDSC）、512字节（SDHC））	R1	READ_SINGLE_BLOCK	该指令用于读取SD卡一个块的数据。该指令带一个参数，表示要读取的数据块首地址。 块大小由CMD16设置。 注意：对高容量SD卡，块大小固定为512字节。

3.4 CMD18 指令说明　P115

命令编号	参数	响应	缩写	说明
CMD18	[31:0]数据地址（单位：字节（SDSC）、512字节（SDHC））	R1	READ_MULTIPLE_BLOCK	连续读取SD卡多数据块数据，直到主机发送CMD12指令。该指令带一个参数，表示要读取的数据块首地址。 块大小由CMD16设置。 注意：对高容量SD卡，块大小固定为512字节。

3.5 ACMD23 指令说明　P65

命令编号	参数	响应	缩写	说明
ACMD23	[31:23]无效 [22:0]数据块个数	R1	SET_WR_BLK_ERASE_COUNT	设置需要预擦除的数据块个数，以提高SD卡多数据块写入性能(速度)。

注意 :发送 ACMD 之前，必须先发送 CMD55，通知 SD 卡，接下来要发送的是应用命令（APP CMD），而非标准命令!

3.6 CMD24 指令说明　P62

命令编号	参数	响应	缩写	说明
CMD24	[31:0]数据地址（单位：字节（SDSC）、512字节（SDHC））	R1	WRITE_BLOCK	该指令用于写入SD卡一个块的数据。该指令带一个参数，表示要写入的数据块首地址。 块大小由CMD16设置。 注意：对高容量SD卡，块大小固定为512字节。

3.7 CMD25 指令说明　P62

命令编号	参数	响应	缩写	说明
CMD25	[31:0]数据地址（单位：字节（SDSC）、512字节（SDHC））	R1	WRITE_MULTIPLE_BLOCK	连续写入SD卡多数据块数据，直到主机发送0XFD令牌。该指令带一个参数，表示要写入的数据块首地址。 块大小由CMD16设置。 注意：对高容量SD卡，块大小固定为512字节。

3.7 CMD55 指令说明　P64

命令编号	参数	响应	缩写	说明
CMD55	[31:0]无效	R1	APP_CMD	该指令通知SD卡，接下来要发送的是应用命令(APP CMD)。 注意：所有APP CMD(ACMD)发送之前，都应该先发送CMD55！！

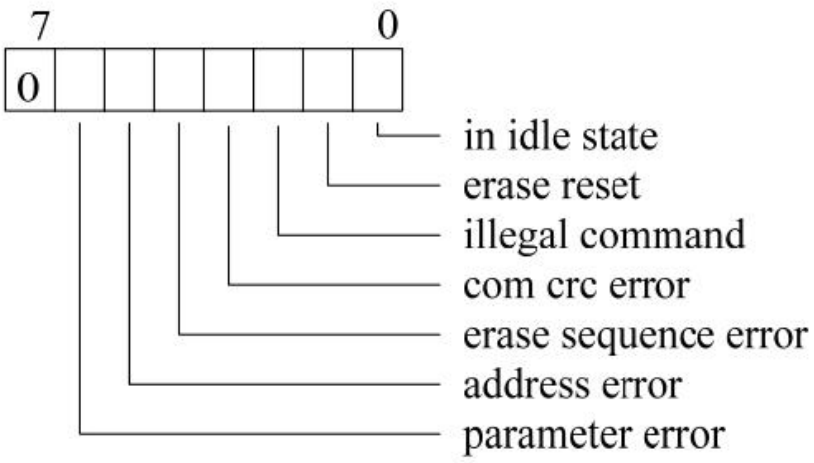
（4） 1.4.SD 卡的响应

每发送一个命令， SD 卡都会给出一个应答，以告知主机该命令的执行情况，或者返回主机需要获取的数据。

SD 卡响应参考 SD 卡 2.0 协议　7.3.2　P120

SPI 模式下， SD 卡针对不同的命令，应答可以使 R1~R7， R1 的应答，各位描述如下表所述：

R1 响应格式								
位	7	6	5	4	3	2	1	0
含义	开始位始终为0	参数错误	地址错误	擦除序列错误	CRC 错误	非法命令	擦除复位	闲置状态



如果第 0 位为 1，说明 SD 卡为空闲状态 .

1.1.2 2 SD 卡初始化流程图--SDIO 模式

（1） 2.1

在发送 CMD0 之前，要发送大于 74 个时钟周期(为什么?手册 P102-P103. 如何发送???)，这是因为 SD 卡内部有个供电电压上升时间，大概为 64 个 CLK，剩下的 10 个 CLK 用于 SD 卡同步，之后才能开始 CMD0 的操作。

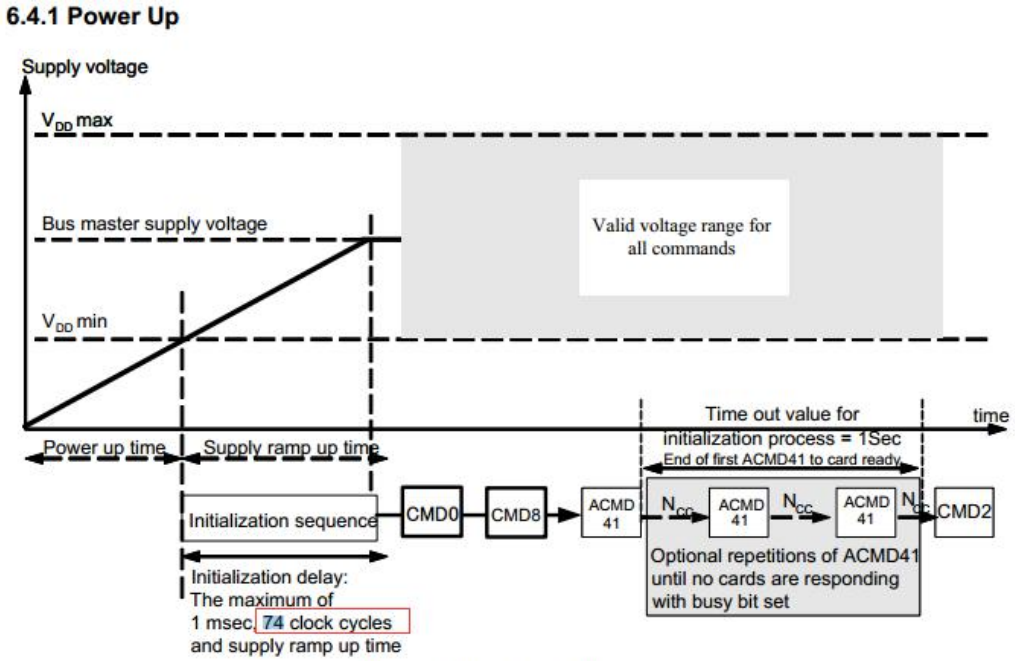
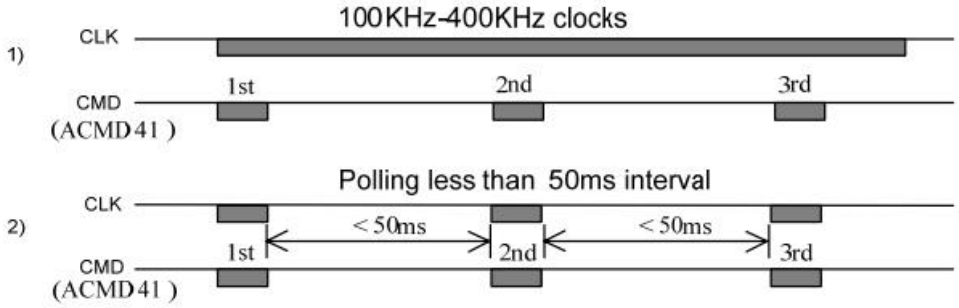


Figure 6-1: Power-up Diagram

- The host shall supply power to the card so that the voltage is reached to Vdd_min within 250ms and start to supply at least 74 SD clocks to the SD card with keeping CMD line to high. In case of SPI mode, CS shall be held to high during 74 clock cycles.
- After power up (including hot insertion, i.e. inserting a card when the bus is operating) the SD Card enters the *idle state*. In case of SD host, CMD0 is not necessary. In case of SPI host, CMD0 shall be the first command to send the card to SPI mode.

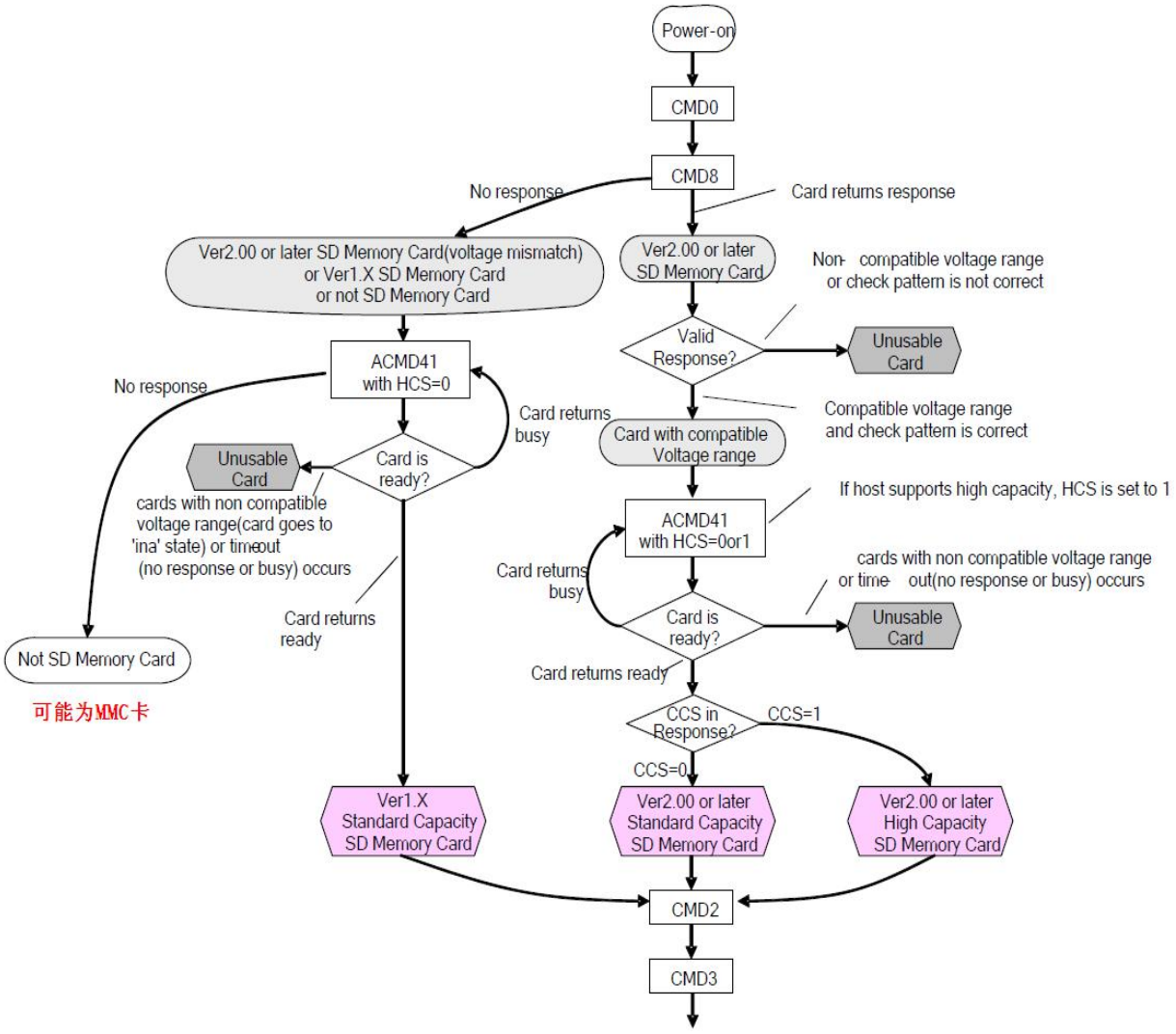
注意：在卡初始化的时候，CLK 时钟（**SPI 的通信速度**）最大不能超过 400Khz！ 手册 P53



如何发送 74 个时钟???

发送一个位需要一个时钟周期,那么发送一个字节就需要 8 个时钟周期,所以我们只需要连续发送 10 个字节的数据不就是相当于送了 80 个时钟周期了么!

（2） **2.2. SDIO 初始化流程图**



STM32_SDIO 应用_SD 卡.pdf

SD 卡的初始化过程如下：

1、初始化与 SD 卡连接的硬件条件（ MCU 的 SPI 配置， IO 口配置）；

2、上电延时（发送大于 74 个 CLK 时钟周期）；

3、复位卡（发送 CMD0），SD 卡进入空闲状态；

4、发送 CMD8，检查是否支持 2.0 协议；

5、根据不同协议检查 SD 卡（命令包括： CMD55、 CMD41、 CMD58 和 CMD1 等）；

6、取消片选，发多 8 个 CLK，结束初始化

这样就完成了对 SD 卡的初始化，注意末尾发送的 8 个 CLK 是提供 SD 卡额外的时钟，完成某些操作。通过 SD 卡初始化，我们可以知道 SD 卡的类型（ V1、 V2、 V2HC 或者 MMC），在完成了初始化之后，就可以开始读写数据了。

说明:

第 3 步:参考 卡命令格式章节

第 4 步:SD_SendCmd(CMD8,0x1AA,0x87).为什么是 0x1AA???

在发送 CMD8 的时候，通过其带的参数可以设置 VHS 位，用来告诉 SD 卡，主机的供电情况，让 SD 卡知道主机的供电范围。

VHS 位定义如下表所示：

Voltage Supplied	Value Definition
0000b	Not Defined
0001b	2.7-3.6V
0010b	Reserved for Low Voltage Range
0100b	Reserved
1000b	Reserved
Others	Not Defined

0X1AA(0X1AA 中的 1 为上表中 0001b 的 1、b 并不是 16 进制的 b,这个 b 说明这里 0001 为二进制数，AA 为 CMD8 格式命令表中的 8－15 位,AA 是其默认值，手册 P51)，即告诉 SD 卡，主机供电为 2.7~3.6V 之间，如果 SD 卡支持 CMD8，且支持该电压范围，则会通过 CMD8 的响应（R7，关于 SD 卡响应，请参考《SD 卡 2.0 协议.pdf》第 4.9 节）将参数部分原本返回给主机，如果不支持 CMD8，或者不支持这个电压范围，则不响应。

Application Note:
It is recommended to use '10101010b' for the 'check pattern'. 0xaa

注意:在发送 CMD8 后，发送 ACMD41（注意：发送 ACMD41 之前，要先发送 CMD55.因为在发送 ACMD 指令之前需要发送 CMD55 指令,手册 P65 等,)，来进一步确认卡的操作电压范围，并通过 HCS 位来告诉 SD 卡，主机是不是支持高容量卡.

The following table describes all the application-specific commands supported/reserved by the SD Memory Card. All the following ACMDs shall be preceded with APP_CMD command (CMD55).

1.1.3 3. SD 卡读数据

（1） 3.1 SD 卡 读单块数据块流程

1. 发送 CMD16 指令，设置数据块大小
2. 等待 CMD16 响应(R1)
3. 发送 CMD17 指令，开始读数据块
4. 等待 CMD17 响应(R1)
5. 接收数据
6. 读一个数据块的数据，完成单块读取

注意：CMD16 设置的数据块大小，一般为 512 字节，此设置直接决定 SD 卡的块大小，SD 卡默认的块大小自动失效。

（2） 3.2 SD 卡 读多块数据块流程

1. 发送 CMD16 指令,设置数据块大小
2. 等待 CMD16 响应(R1)
3. 发送 CMD18 指令,开始读数据块
4. 等待 CMD18 响应(R1)
5. 读第一个数据块的数据
6. 读第二个数据块的数据
7. ……读取第 N 个数据块的数据
8. 读第二个数据块的数据
9. 发送 CMD12 指令,结束数据块读取
10. 读取第 N 个数据块的数据
11. 等待 CMD12 响应(R1)

12.结束多块数据块读取

（3） 3.4 SD卡 写单块数据块流程

1.发送CMD16指令，设置数据块大小
2.等待CMD16 指令的响应。
3.发送CMD13指令，查询卡状态
4.等待READY_FOR_DATA位=1
5.发送CMD24指令，开始写入数据
6.写一个数据块的数据，完成单块写入
注意：此处省略了等待指令响应的步骤

（4） 3.5 SD卡 写多块数据块流程

1.发送CMD16指令,设置数据块大小
2.发送ACMD23指令,预擦除数据块（发ACMD指令之前需要发CMD55）
3.发送CMD25指令,开始写数据块
4.写入第一个数据块的数据
5.写入第二个数据块的数据
6.……写入第N个数据块的数据
7.发送CMD12指令,结束数据块写入
8.发送CMD13指令,查询卡状态
9.等待SD卡写入过程结束
10.完成多数据块写入

总结：响应（R1 ~ R7 ）,响应格式。
初始化流程 。初始化流程图，协议 P27
上电后，这时候的时钟不能大于400K HZ.
先发送74个时钟周期。
CMD8 命令 （0x1AA）。->两个功能1.电压。2.判断SD版本
ACMD41 命令：发ACMD41(功能：卡容量) 需要发送CMD55

SD 卡的读写：

CMD16 设置块的大小
CMD17 写一个块
CMD18 写多个块
CMD12 停止读写数据

CMD13 查状态 （读数据）
CMD24 读一个块
CMD25 读多个块

1.2 SDIO 介绍

（1） SDIO 概述

Secure Digital Input and Output Card，简称 SDIO，即安全数字输入/输出接口。SDIO 在 SD 标准上定义了一种外设接口。目前，SDIO 主要有两类应用——可移动和不可移动。可移动设备作为 Palm 和 Windows Mobile 的扩展设备，用来增加蓝牙、照相机、GPS 等功能。不可移动设备遵循相同的电气标准，但不要求符合物理标准。

（2） SDIO 基本特性

SDIO 具有以下特性：

- 完全兼容多媒体卡系统规范版本 4.2。卡支持三种不同数据总线模式： 1 位（默认）、4 位和 8 位
- 完全兼容先前版本的多媒体卡（向前兼容性）
- 完全兼容 SD 存储卡规范版本 2.0
- 完全兼容 SD I/O 卡规范版本 2.0： 卡支持两种不同数据总线模式： 1 位（默认）和 4 位
- 完全支持 CE-ATA 功能（完全符合 CE-ATA 数字协议版本 1.1）
- 对于 8 位模式，数据传输高达 48 MHz
- 数据和命令输出使能信号，控制外部双向驱动程序。

（3） SDIO 框图分析

STM32F4 SDIO 主要由两部分组成：APB2 总线接口和 SDIO 适配器。APB2 接口访问 SDIO 适配器寄存器，并且生成中断和 DMA 请求信号。SDIO 适配器块提供特定于 MMC/SD/SD I/O 卡的所有功能，如时钟生成单元、命令和数据传输。如图 9.3.1 所示：

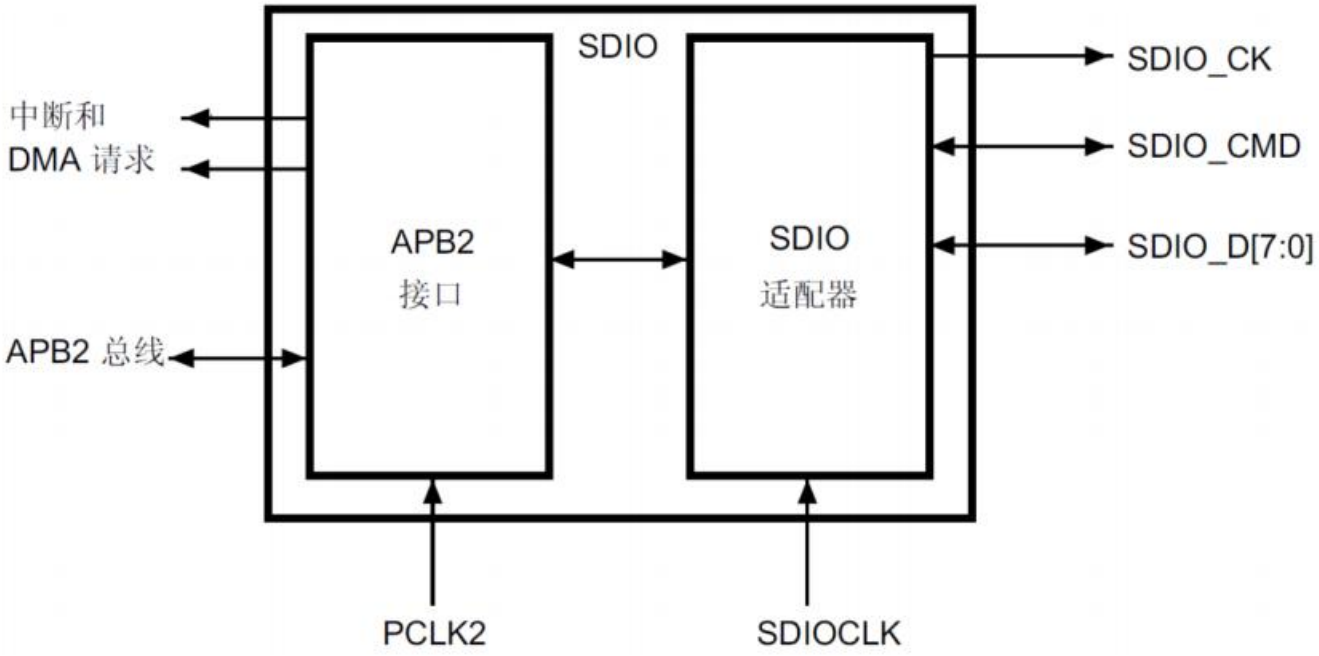


图 9.3.1 STM32F4 SDIO 框图

SDIO 主要信号线：

SDIO_CK：CLK 信号线，由主机给从机的时钟信号。

SDIO_CMD：双向的信号，用于传输命令和响应。

SDIO_D[7:0]：8 条用于传输的信号线。

默认情况下， SDIO_D0 用于数据传输。初始化后，主机可以更改数据总线宽度。

如果多媒体卡连接到总线，则 SDIO_D0、 SDIO_D[3:0] 或 SDIO_D[7:0] 可以用于数据传输。MMC V3.31 或更低版本仅支持 1 位数据，因此只能使用 SDIO_D0。

如果 SD 或 SD I/O 卡连接到总线，则主机可以将数据传输配置为使用 SDIO_D0 或 SDIO_D[3:0]。所有数据线均以推挽模式运行。

SDIO_CMD 有两种操作模式：

- 开漏引脚，用于初始化（仅限于 MMC V3.31 或更低版本）
- 推挽，用于命令传输（ SD/SD I/O 卡 MMC4.2 还将推挽驱动程序用于初始化）

（4） SDIO 时钟

SDIO_CK 是与卡相连的时钟：每个时钟周期在命令和数据线上传输一个位的命令或数据。SDIO_CK 根据卡的类型不同，它的时钟频率也是不同的。对于 MMC 卡 V3.31 协议来说，时钟频率可以在 0 MHz 到 20 MHz 之间变化，对于 MMC 卡 V4.0/4.2 协议来说，可以在 0 到 48 MHz 之间变化，对于 SD/SD I/O 卡，可以在 0 到 25 MHz 之间变化，此实验中使用的是 SD 卡。

SDIO 使用如下两个时钟信号：

1．SDIO 适配器时钟（SDIOCLK）

此时钟主要用于驱动 SDIO 适配置器，一般为 48M，用于产生 SDIO_CK 时钟。SDIO 适配器的时钟（SDIOCLK）与卡相连时钟（SDIO_CK）存在以下关系：

SDIO_CK=SDIOCLK/(2+CLKDIV)

SDIO_CK：卡相连时钟。

SDIOCLK：适配器时钟。

CLKDIV：时钟分频系数。通过 SDIO_CLKCR 寄存器进行设置，不能超过卡的最大操作频率。需要注意的是，在 SD 卡初始化的过程中，其频率是不能超过 400K Hz 的。在初始化完成之后就可以把时钟频率加大了，但不不能超过 SD 卡的最大操作频率。

2．APB2 总线时钟 (PCLK2)

此时钟来自于 PCLK2，用于驱动 SDIOA 的 APB2 总线接口，一般为 84M。

（5） SDIO 命令

命令：命令是用于启动操作的令牌。命令是从主机发出的，命令在 CMD 线上以串行的方式传输。所有命令的开度都为 48 位。如表 9.3.1 所示：

SDIO 的命令分为两类：应用程序特定的命令（ACMD)和常规命令（GEN_CMD)。当卡接收到 APP_CMD (CMD55) 命令时，卡所需的下一个命令是应用程序特定的命令，即在发送 ACMD 命令之前需要先发送 CMD55 这个命令。

表 9.3.1 命令格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	1	传输位
[45：40]	6	—	命令索引
[39：8]	32	—	参数
[7：1]	7	—	CRC7
0	1	1	结束位

由表可知：SDIO 的命令是由起始位、传输位、命令索引、参数、CRC7 和结束位组成的。并且都是由主机发出的，其中起始位、传输位、CRC7 和结束位不需要软件控制，由硬件自主产生，我们需要设置的是命令索引和参数这两部分。命令索引是在 SDIO_CMD 寄存器中设置，参数是在 SDIO_ARG 寄存器中设置。

（6） SDIO 响应

响应：响应是一个令牌，它作为对先前接收命令的应答，是从卡发送到主机的。响应在 CMD 线上以串行的方式传输。SDIO 一共有两种响应：48 位短响应和 136 位长响应。两种响应都使用 CRC 错误校验，如果响应不包含 CRC（ CMD1 响应），必须忽略 CRC 失败状态。短响应的格式如表 9.3.2 所示，长响应格式如表 9.3.3 所示：

表 9.3.2 短响应格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45：40]	6	—	命令索引
[39：8]	32	—	参数

[7：1]	7	—	CRC7(或 1111111）
0	1	1	结束位

表 9.3.3 长响应格式

位的位置	宽度	值	说明
135	1	0	起始位
134	1	0	传输位
[139：128]	6	111111	保留
[127：1]	127	—	CID 或 CSD（包括内部 CRC7）
0	1	1	结束位

从表 9.3.2 和表 9.3.3 中我们可以知道，短响应的格式和命令格式一样的，而长响应的格式也和命令格式差不多。因此，起始位、传输位、CRC7 和结束位不需要软件控制的，由硬件自动控制。短响应的命令索引存放在 SDIO_RESPCMD 寄存器中，参数存放在 SDIO_RESP1 寄存器中。长响应的，只有 CID 或 CSD，存放在 SDIO_RESP1 到 SDIO_RESP4 这四个寄存器中。

根据不同的命令，SDIO 有不同的响应，不同的响应其长度也是不一样的。SD 卡的响应一共有 6 种，分别为 R1、R1b、R2、R3、R6 和 R7。下面简单介绍一下。

1．R1 响应（正常命令响应）。长度为 48 位。当收到 R1 响应后，命令索引的内容是存放在 SDIO_RESPCMD 寄存器中的，卡状态的内容是存放在 SDIO_RESP1 寄存器中的。R1 响应格式如表 9.3.4 所示：

表 9.3.4 R1 响应格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45：40]	6	—	命令索引
[39：8]	32	—	卡状态
[7：1]	7	—	CRC7
0	1	1	结束位

2．R1b 响应。与 R1 响应类似，只是把 BUSY 位加入响应中。

3．R2 响应（ CID 和 CSD 寄存器）。长度为 136 位。CID 是命令 CMD2 和命令 CMD10 的响应，CSD 是命令 CMD9 的响应。当收到 R2 响应后，CID 和 CSD 的内容存放在 SDIO_RESP1 到 SDIO_RESP4 这四个寄存器中。R2 响应的格式如表 9.3.5 所示：

表 9.3.5 R2 响应格式

位的位置	宽度	值	说明
135	1	0	起始位
134	1	0	传输位
[133：128]	6	111111	命令索引
[127：1]	127	—	卡状态
0	1	1	结束位

4．R3 响应（OCR 寄存器）。长度为 48 位。作为命令 ACMD41 的响应。发送 OCR 寄存器的内容以响应命令 CMD1。R3 响应的格式如表 9.3.6 所示：

表 9.3.6 R3 响应格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45：40]	6	111111	保留
[39：8]	32	—	OCR 寄存器
[7：1]	7	111111	保留

0	1	1	结束位

5．R6 响应（RCA 寄存器）。长度为 48 位。参数字段包含寻址卡的 RCA、要读出或写入的寄存器地址及其内容。R6 响应格式如表 9.3.7 所示：

表 9.3.7 R6 响应格式

位的位置		宽度	值	说明
47		1	0	起始位
46		1	0	传输位
[45：40]		6	101000	CMD40
[39：8]		16	—	胜出的卡或主机的 RCA [31:16]
参数字段		16	—	未定义。可用于 IRQ 数据
[7：1]		7	—	CRC7
0		1	1	结束位

6．R7 响应（卡接口条件）。长度为 48 位。此响应是 SD 卡专有的响应。卡支持的电压信息是通过 CMD8 的响应发送。卡会在响应的参数中返回电压范围。R7 响应格式如表 9.3.8 所示：

表 9.3.8 R7 响应格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45：40]	6	001000	CMD8
[39：20]	20	00000	保留
[19：16]	4	—	可接受的电压范围
[15：8]	8	—	检查模式
[7：1]	7	—	CRC7
0	1	1	结束位

（7） SDIO（多个）块读取操作

SDIO 的读数据操作过程如图 9.3.2 所示：

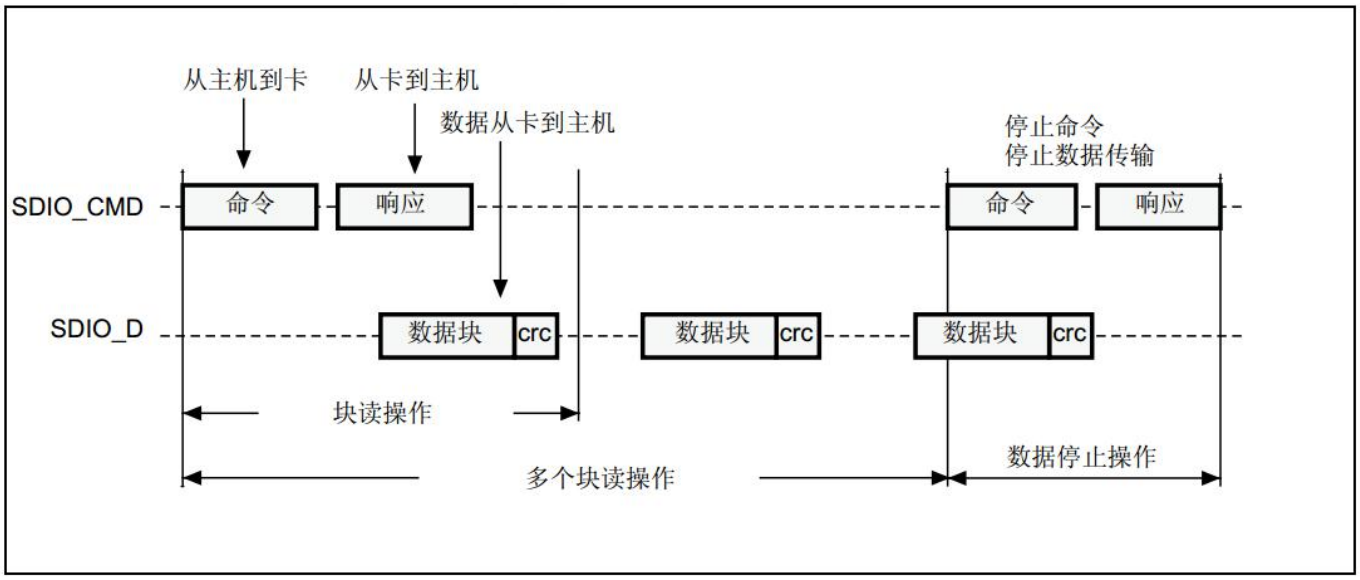


图 9.3.2 SDIO（多个）块读取

从图 9.3.2 中，我们可以知道，首先主机通过 SDIO_CMD 命令线发送命令给卡，之后卡给主机一个响应，然后开始给主机发送数据，所有的数据块都包含有 CRC 校验。CRC 的校验是由 SDIO 硬件自动处理的。如果是读取

一个数据块，那么在收到一个数据块的内容之后就会停止了，不需要发送停止命令。但是，读取多个数据块时，卡将会一直给主机发送数据，直到主机给卡发送停止命令时，卡才会停止给主机发送数据。

（8） SDIO（多个）块写入操作

SDIO 的写数据操作过程如图 9.3.3 所示：

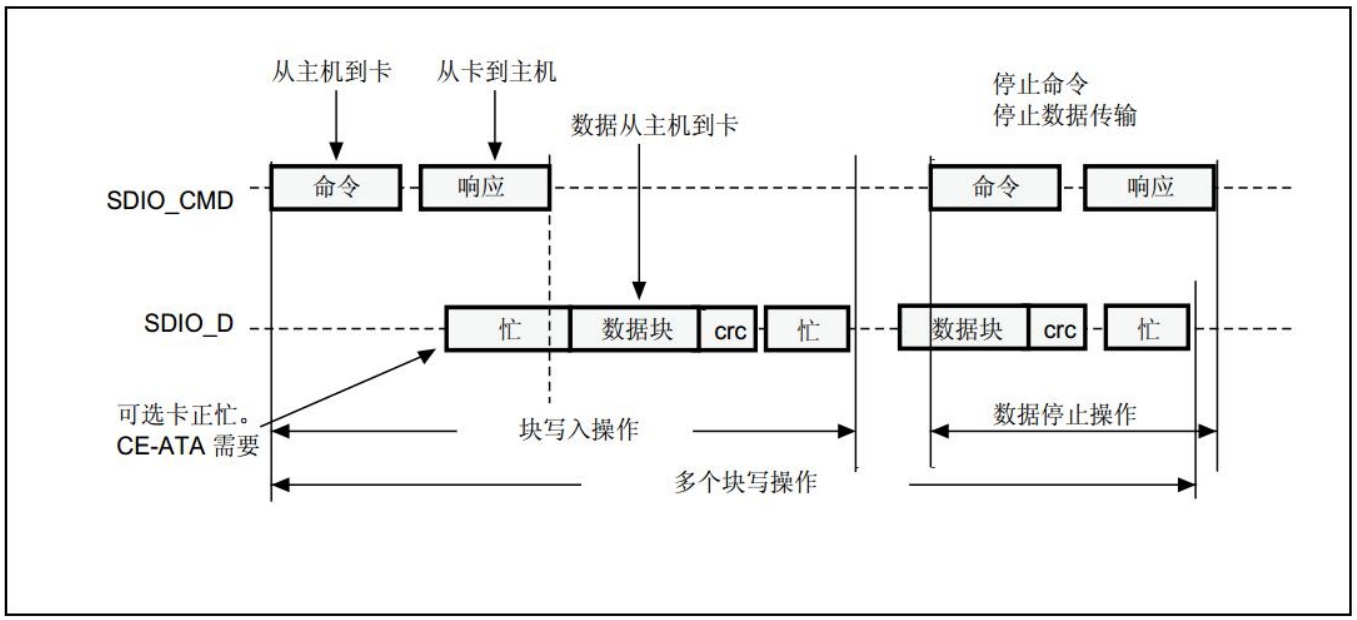


图 9.3.3 SDIO（多个）块写入

从图 9.3.3 中，我们可以知道，数据块的写入过程和数据块的读取是相类似的，只不过数据块写入的时候多了一个忙检测，在数据块写入之前必须先执行判断，只有在 SD 卡处于空闲状态时才能够写入数据。当 SD 卡处于忙状态时，卡会把 SDIO_D0 线拉低。

1.2.2 STM32F4 SDIO 相关寄存器介绍

（1） SDIO 电源控制寄存器 (SDIO_POWER)

偏移地址： 0x00
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														PWRCTRL	
																														rw	rw

位 31:2	保留，始终为0。
位 1:0	PWRCTRL： 电源控制位 (Power supply control bits)。 这些位用于定义卡时钟的当前功能状态： 00：掉电：停止为卡提供时钟。 01：保留 10：保留，上电 11：通电：为卡提供时钟。

芯片复位后，SDIO 是处于不给卡上电，也不给卡提供时钟的状态，所以，我们想要使用 SDIO 的功能，就必须给卡上电和提供时钟。

（2） SDI 时钟控制寄存器 (SDIO_CLKCR)

偏移地址： 0x04
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																	HWFC_EN	NEGEDGE	WID BUS		BYPASS	PWRSAV	CLKEN	CLKDIV							
																	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15	保留，始终为0。
位 14	HWFC_EN： 硬件流控制使能 (HW Flow Control enable) 0：禁止硬件流控制 1：使能硬件流控制
位 13	NEGEDGE： SDIO_CK 移相选择位 (SDIO_CK dephasing selection bit) 0：在主时钟 SDIOCLK 的上升沿产生 SDIO_CK 1：在主时钟 SDIOCLK 的下降沿产生 SDIO_CK
位 12:11	WIDBUS： 宽总线模式使能位 (Wide bus mode enable bit) 00：默认总线模式：使用 SDIO_D0 01： 4 位宽总线模式：使用 SDIO_D[3:0] 10： 8 位宽总线模式：使用 SDIO_D[7:0]
位 10	BYPASS： 时钟分频器旁路使能位 (Clock divider bypass enable bit) 0：禁止旁路：在驱动 SDIO_CK 输出信号前，根据 CLKDIV 值对 SDIOCLK 进行分频。 1：使能旁路： SDIOCLK 直接驱动 SDIO_CK 输出信号。
位 9	PWRSAV： 节能模式配置位 (Power saving configuration bit) 要实现节能模式，可在总线空闲时通过将 PWRSAV 置 1 来禁止 SDIO_CK 时钟输出： 0：始终使能 SDIO_CK 时钟 1：仅在总线激活时使能 SDIO_CK
位 8	CLKEN： 时钟使能位 (Clock enable bit) 0：禁止 SDIO_CK 1：使能 SDIO_CK
位 7:0	CLKDIV： 时钟分频系数 (Clock divide factor) 该字段定义输入时钟 (SDIOCLK) 与输出时钟 (SDIO_CK) 之间的分频系数： SDIO_CK 频率 = SDIOCLK / [CLKDIV + 2]。

在 SD 卡的使用中，我们需要把总线宽度设置为 4 位，并且禁止使用旁路。需要把时钟的使能位置 1 以开启时钟。时钟分频系数用于设置 SDIO 的时钟分频。

（3） SDIO 参数寄存器 (SDIO_ARG)

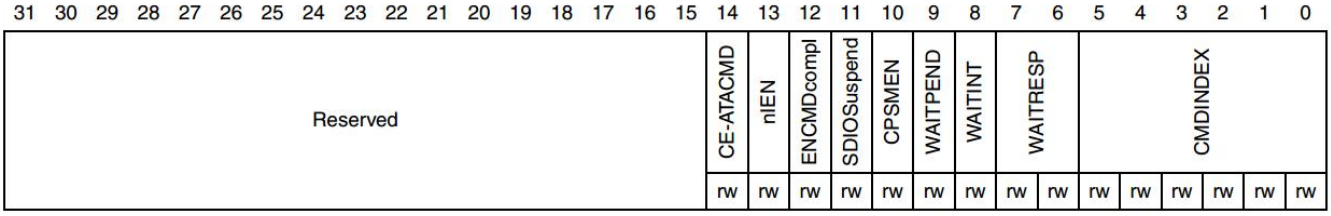
偏移地址： 0x08
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG																															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0	CMDARG： 命令参数 (Command argument) 作为命令消息的一部分发送给卡的命令参数。如果命令包含参数，则在将命令写入到命令寄存器之前，必须将参数加载到此寄存器中。

（4） SDIO 命令寄存器 (SDIO_CMD)

偏移地址： 0x0C
复位值： 0x0000 0000

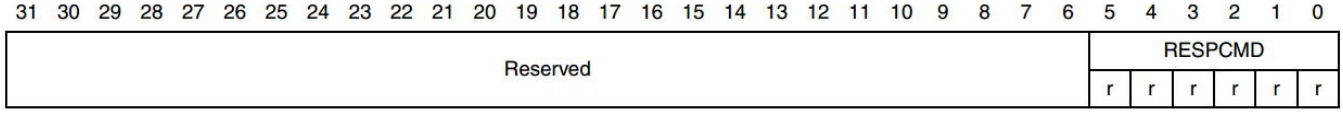


位 31:15	保留，始终为0
位 14	ATACMD： CE-ATA 命令 (CE-ATA command) 如果 ATACMD 置 1，则 CPSM 将传输 CMD61。
位 13	nIEN： 非中断使能 (not Interrupt Enable) 如果该位为 0，则使能 CE-ATA 设备中的中断。
位 12	ENCMDcompl： 使能 CMD 完成 (Enable CMD completion) 如果此位置 1，则使能命令完成信号。
位 11	SDIOSuspend： SD I/O 挂起命令 (SD I/O suspend command) 如果此位置 1，则要发送的命令为挂起命令（仅用于 SDIO 卡）。
位 10	CPSMEN： 命令路径状态机 (CPSM) 使能位 (Command path state machine (CPSM) Enable bit) 如果此位置 1，则使能 CPSM。
位 9	WAITPEND： CPSM 等待数据传输结束（ CmdPend 内部信号） (CPSM Waits for ends of data transfer (CmdPend internal signal))。 如果此位置 1，则 CPSM 将等到数据传输结束后才开始发送命令。
位 8	WAITINT： CPSM 等待中断请求 (CPSM waits for interrupt request) 如果此位置 1，则 CPSM 禁止命令超时并等待中断请求。
位 7:6	WAITRESP：等待响应位 (Wait for response bits) 这些位用于配置 CPSM 是否等待响应，如果等待，将等待哪种类型的响应。 00：无响应，但 CMDSENT 标志除外 01：短响应，但 CMDREND 或 CCRCFAIL 标志除外 10：无响应，但 CMDSENT 标志除外 11：长响应，但 CMDREND 或 CCRCFAIL 标志除外
位 5:0	CMDINDEX： 命令索引 (Command index) 命令索引作为命令消息的一部分发送给卡。

此寄存器第 10 位是设置命令通道状态机，需要开启。第[7：6]位是设置 CPSM 是否需要等待和响应的模式是长响应还是短响应。第[5：0]位为命令索引，就是前面介绍的命令格式中的命令索引，也就是我们需要发送的命令，如发送 CMD12，则把 12 写入这几个位中即可。

（5） SDIO 命令响应寄存器 (SDIO_RESPCMD)

偏移地址： 0x10
复位值： 0x0000 0000

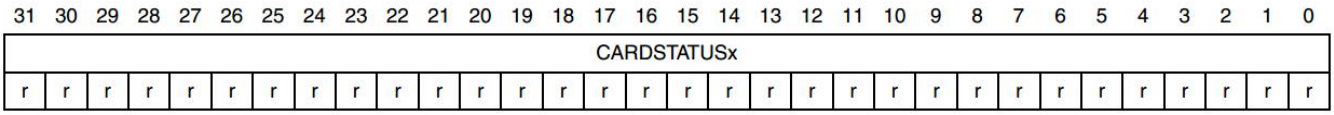


位 31:6	保留，始终为0
位 5:0	RESPCMD： 响应命令索引 (Response command index) 只读位域。包含接收到的最后一个命令响应的命令索引。

SDIO_RESPCMD 寄存器包含接收到的最后一个命令响应的命令索引字段。如果命令响应传输不包含命令索引字段（长响应或 OCR 响应），则 RESPCMD 字段为未知，但该字段必须包含 111111b（响应中保留字段的值）。

（6） SDIO 响应 1..4 寄存器 (SDIO_RESPx)

偏移地址： (0x10 + (4 × x)); x = 1..4
复位值： 0x0000 0000



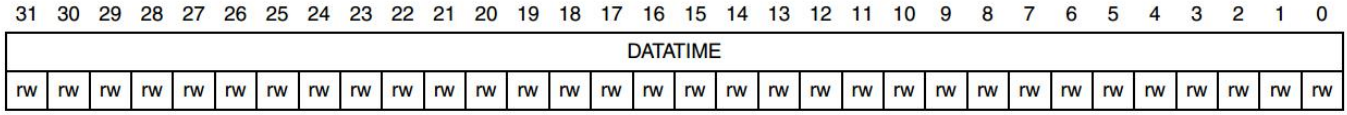
此寄存器组一共有四个寄存器，是用来存放卡响应的信息。如果是短响应，则数据存放在 SDIO_RESP1 寄存器里面，其他三个寄存器没有用到。如果是长响应，则依次存放在 SDIO_RESP1 至 SDIO_RESP4 寄存器中。此寄存器组说明如表 9.3.9 所示：

表 9.3.9 响应类型和 SDIO_RESPx 寄存器

寄存器	短响应	长响应
SDIO_RESP1	卡状态[31:0]	卡状态 [127:96]
SDIO_RESP2	未使用	卡状态 [95:64]
SDIO_RESP3	未使用	卡状态 [63:32]
SDIO_RESP4	未使用	卡状态 [31:1]0b

（7） SDIO 数据定时器寄存器 (SDIO_DTIMER)

偏移地址： 0x24
复位值： 0x0000 0000

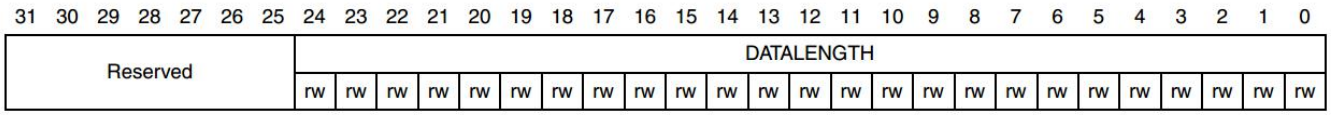


位 31:0	DATATIME： 数据超时周期 (Data timeout period) 以卡总线时钟周期表示的数据超时周期。

此寄存器用于存储以卡总线时钟（ SDIO_CK）为周期的数据超时周期， 计数器 加载 SDIO_DTIMER 寄存器的值，并在数据通道状态机（DPSM）进入 Wait_R 或繁忙状态后开始递减计数，如果计数器减到 0，则会将超时标志置 1。数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。

（8） SDIO 数据长度寄存器 (SDIO_DLEN)

偏移地址： 0x28
复位值： 0x0000 0000



位 31:25	保留，始终为0
位 24:0	DATALENGTH： 数据长度值 (Data length value) 要传输的数据字节数量。

此寄存器用于设置需要传输的数据字节数，当数据开始传输时，这个值将会加载到数据计数器中。对于块数据传输，数据长度寄存器中的值必须是块大小的倍数（在 SDIO_DCTRL 中设置）。数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。

（9） SDIO 数据控制寄存器 (SDIO_DCTRL)

偏移地址： 0x2C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE				DMAEN	DTMODE	DTDIR	DTEN
																				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

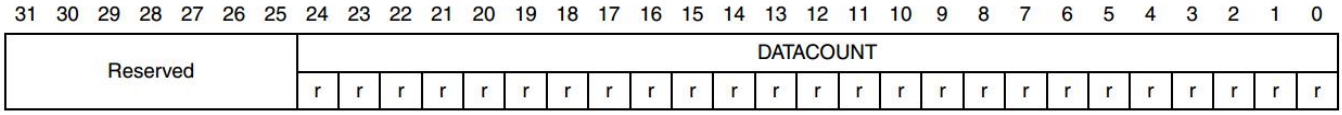
位 31:12	保留，始终为0
位 11	SDIOEN： SD I/O 使能功能 (SD I/O enable functions) 如果将该位置 1，则 DPSM 执行特定于 SD I/O 卡的操作。
位 10	RWMOD： 读取等待模式 (Read wait mode) 0：通过停止 SDIO_D2 进行读取等待控制 1：使用 SDIO_CK 进行读取等待控制
位 9	RWSTOP： 读取等待停止 (Read wait stop) 0：如果将 RWSTART 位置 1，则读取等待正在进行中 1：如果将 RWSTART 位置 1，则使能读取等待停止
位 8	RWSTART： 读取等待开始 (Read wait start) 如果将该位置 1，则读取等待操作开始。
位 7:4	DBLOCKSIZE： 数据块大小 (Data block size) 定义在选择了块数据传输模式时数据块的长度： 0000：（十进制数 0）块长度 = 20 = 1 字节 0001：（十进制数 1）块长度 = 21 = 2 字节 0010：（十进制数 2）块长度 = 22 = 4 字节 0011：（十进制数 3）块长度 = 23 = 8 字节 0100：（十进制数 4）块长度 = 24 = 16 字节 0101：（十进制数 5）块长度 = 25 = 32 字节 0110：（十进制数 6）块长度 = 26 = 64 字节 0111：（十进制数 7）块长度 = 27 = 128 字节 1000：（十进制数 8）块长度 = 28 = 256 字节 1001：（十进制数 9）块长度 = 29 = 512 字节 1010：（十进制数 10）块长度 = 210 = 1024 字节 1011：（十进制数 11）块长度 = 211 = 2048 字节 1100：（十进制数 12）块长度 = 212 = 4096 字节 1101：（十进制数 13）块长度 = 213 = 8192 字节 1110：（十进制数 14）块长度 = 214 = 16384 字节 1111：（十进制数 15）保留
位 3	DMAEN： DMA 使能位 (DMA enable bit) 0：禁止 DMA。 1：使能 DMA。
位 2	DTMODE： 数据传输模式选择 (Data transfer mode selection) 0：块数据传输 1：流或 SDIO 多字节数据传输
位 1	DTDIR： 数据传输方向选择 (Data transfer direction selection) 0：从控制器到卡。 1：从卡到控制器。
位 0	DTEN： 数据传输使能位 (Data transfer enabled bit) 如果 1 写入到 DTEN 位，则数据传输开始。根据方向位 DTDIR，如果在传输开始时立即将 RW 置 1开始，则 DPSM 变为 Wait_S 状态、 Wait_R 状态或读取等待状态。在数据传输结束

	后不需要将使能位清零，但必须更新 SDIO_DCTRL 以使能新的数据传输

此寄存器控制数据路径状态机 (DPSM)。需要我们配置的有：数据传输使能、传输方向、传输模式、DMA 使能、数据块长度等信息。

（10） SDIO 数据计数器寄存器 (SDIO_DCOUNT)

偏移地址： 0x30
复位值： 0x0000 0000

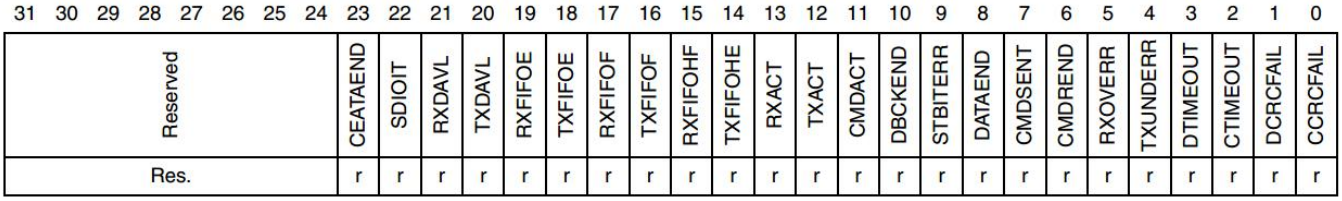


位 31:25	保留，始终为0
位 24:0	DATACOUNT： 数据计数值 (Data count value) 读取该位时，将返回要传输的剩余数据字节数量。写入没有任何效果。

当 DPSM 从空闲状态变为 Wait_R 或 Wait_S 状态时， SDIO_DCOUNT 寄存器将从数据长度寄存器（SDIO_DLEN）中加载值。在传输数据时，计数器将值递减直至计数器达到 0。然后 DPSM 将变为空闲状态，并且将数据状态结束标志 DATAEND 置 1。仅当数据传输完成后才应读取该寄存器。

（11） SDIO 状态寄存器 (SDIO_STA)

偏移地址： 0x34
复位值： 0x0000 0000



位 31:24	保留，始终为0
位 23	CEATAEND： 针对 CMD61 收到了 CE-ATA 命令完成信号 (CE-ATA command completion signal received for CMD61)
位 22	SDIOIT： 收到了 SDIO 中断 (SDIO interrupt received)
位 21	RXDAVL： 接收 FIFO 中有数据可用 (Data available in receive FIFO)
位 20	TXDAVL： 传输 FIFO 中有数据可用 (Data available in transmit FIFO)
位 19	RXFIFOE： 接收 FIFO 为空 (Receive FIFO empty)
位 18	TXFIFOE： 发送 FIFO 为空 (Transmit FIFO empty) 如果使能了硬件流控制，则 TXFIFOE 信号在 FIFO 包含 2 个字时激活。
位 17	RXFIFOF： 接收 FIFO 已满 (Receive FIFO full) 如果使能了硬件流控制，则 RXFIFOF 信号在 FIFO 差 2 个字便变满之前激活。
位 16	TXFIFOF： 传输 FIFO 已满 (Transmit FIFO full)
位 15	RXFIFOHF： 接收 FIFO 半满： FIFO 中至少有 8 个字 (Receive FIFO half full: there are at least 8 words in the FIFO)
位 14	TXFIFOHE： 传输 FIFO 半空：至少可以写入 8 个字到 FIFO (Transmit FIFO half empty: at least 8 words can be written into the FIFO)
位 13	RXACT： 数据接收正在进行中 (Data receive in progress)
位 12	TXACT： 数据传输正在进行中 (Data transmit in progress)
位 11	CMDACT： 命令传输正在进行中 (Command transfer in progress)
位 10	DBCKEND： 已发送/ 接收数据块（ CRC 校验通过 ） (Data block sent/received (CRC check

	passed))
位 9	STBITERR： 在宽总线模式下，并非在所有数据信号上都检测到了起始位 (Start bit not detected on all data signals in wide bus mode)
位 8	DATAEND： 数据结束（数据计数器 SDIDCOUNT 为零） (Data end (data counter, SDIDCOUNT, is zero))
位 7	CMDSENT： 命令已发送（不需要响应） (Command sent (no response required))
位 6	CMDREND： 已接收命令响应（ CRC 校验通过） (Command response received (CRC check passed))
位 5	RXOVERR： 收到了 FIFO 上溢错误 (Received FIFO overrun error)
位 4	TXUNDERR： 传输 FIFO 下溢错误 (Transmit FIFO underrun error)
位 3	DTIMEOUT： 数据超时 (Data timeout)
位 2	CTIMEOUT： 命令响应超时 (Command response timeout) 命令超时周期为固定值 64 个 SDIO_CK 时钟周期。
位 1	DCRCFAIL： 已发送/ 接收数据块（ CRC 校验失败） (Data block sent/received (CRC check failed))
位 0	CCRCFAIL： 已接收命令响应（ CRC 校验失败） (Command response received (CRC check failed))

此寄存器是一个只读寄存器。它包含两种类型的标志：

- 静态标志（位 [23:22,10:0]）：在通过写入到 SDIO 中断清零寄存器来清零这些位之前，会一直保持这些位（详见 SDIO_ICR）。
- 动态标志（位 [21:11]）：这些位根据底层逻辑的状态来更改状态（例如，随着数据写入到 FIFO，发出和停止发出 FIFO 满和空标志）

（12） SDIO 中断清零寄存器 (SDIO_ICR)

偏移地址： 0x38
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CEATAENDC	SDIOITC	Reserved											DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
								rw	rw												rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

此寄存器是一个只写寄存器。以 1 写入某个位会将 SDIO_STA 状态寄存器中的对应位清零。

（13） SDIO 屏蔽寄存器 (SDIO_MASK)

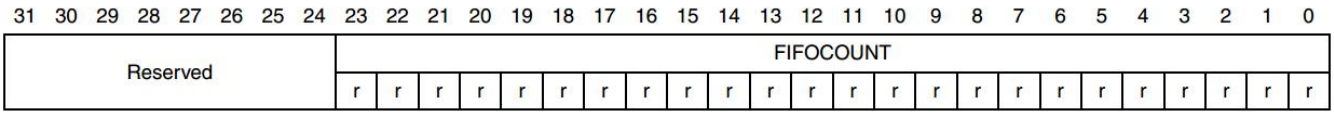
偏移地址： 0x3C
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CEATAENDIE	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIFOEIE	TXFIFOEIE	RXFIFOFIE	TXFIFOFIE	RXFIFOHFIE	TXFIFOHEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	STBITERRIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
								rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

中断屏蔽寄存器通过将对应的位置 1 来确定哪一个状态标志位可以产生中断。对应的位为 SDIO_STA 状态寄存器中的对应位。

（14） SDIO FIFO 计数器寄存器 (SDIO_FIFOCNT)

偏移地址： 0x48
复位值： 0x0000 0000

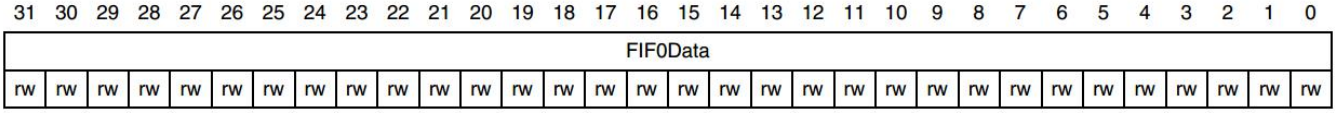


位 31:24	保留，始终为0
位 23:0	FIFOCOUNT： 要在 FIFO 中写入或读取的剩余字数。

SDIO_FIFOCNT 寄存器包含要在 FIFO 中写入或读取的剩余字数。如果数据控制寄存器（ SDIO_DCTRL 寄存器）中将数据传输使能位 DTEN 置 1，并且 DPSM 处于空闲状态，则 FIFO 计数器从数据长度寄存器中加载值（详见 SDIO_DLEN）。如果数据长度未进行字对齐（ 4 的倍数），则剩余的 1 到 3 个字节被视为一个字。

（15） SDIO 数据 FIFO 寄存器 (SDIO_FIFO)

偏移地址： 0x80
复位值： 0x0000 0000



位 31:0	FIFOData： 接收和传输 FIFO 数据 (Receive and transmit FIFO data) FIFO 数据占用 32 位字的 32 个入口，从地址： SDIO 基址 + 0x080 到 SDIO 基址 + 0xFC。

此寄存器包括接收 FIFO 和发送 FIFO，他们由一组连续的 32 个地址上的 32 个寄存器组成，CPU 可以使用 FIFO 读写多个操作数。例如，要从 SD 卡读数据，就必须读 SDIO_FIFO 寄存器，要写数据到 SD 卡，就必须写 SDIO_FIFO 寄存器。
SDIO 将这 32 个地址分为 16 个一组，发送接收各占一半。而每次读写时，最多就是读取接收 FIFO 或写入发送 FIFO 的一半大小的数据，也就是 8 个字（32 个字节）。注意：操作 SDIO_FIFO 必须是以 4 字节对齐的内存操作，否则可能出错！