SD 卡驱动编写流程

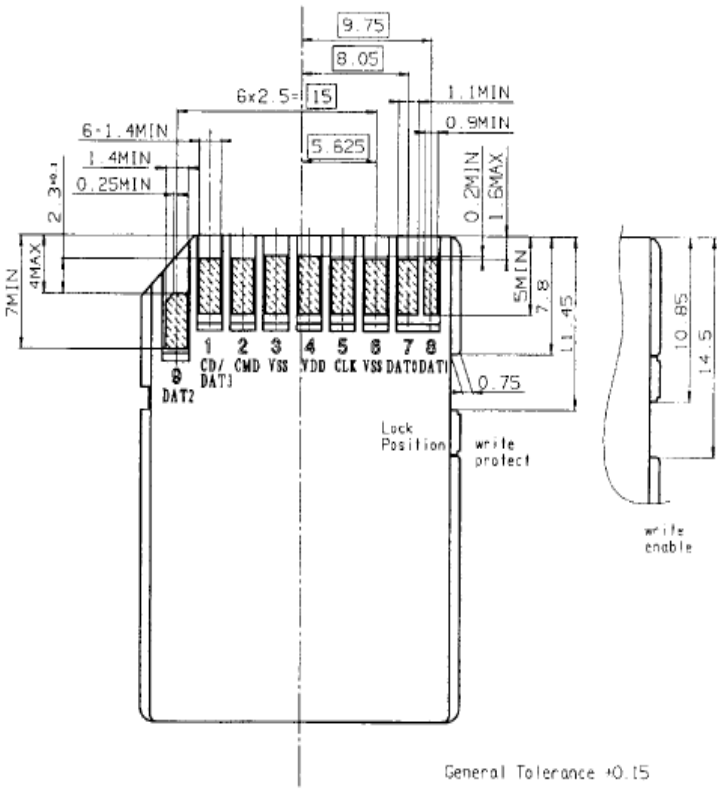
现在对SD卡驱动编写做个总结，以及本人在调试中遇到的问题。先介绍一下SD卡规格，其全称Secure Digital Memory Card 简写SD，是以Flash Memory 为基础的存储卡，自带控制器。所以编写的驱动是针对控制器接口的，不是针对Flash Memory的。

SD卡支持三种驱动模式，见下图！

Table 2. MultiMediaCard and SD Card Clock Speed and Burst Rate

Product	Maximum Clock Speed and Burst Rate	
MultiMediaCard	**Clock Speed**	**Burst Rate**
SPI Bus mode	20 MHz	2.5 MB/s
MMC 1-bit mode	20 MHz	2.5 MB/s
SD Card		
SPI Bus mode	25 MHz	3.125 MB/s
SD 1-bit mode	25 MHz	3.125 MB/s
SD 4-bit mode	25 MHz	12.5 MB/s

外观图和PIN定义看下图：



分别为SPI模式和SD BUS模式，不同的模式，注意相同编号的 PIN意义不一样！

Pin #	SD Mode			SPI Mode		
	Name	Type[1]	Description	Name	Type	Description
1	CD/DAT3[2]	I/O/PP[3]	Card Detect / Data Line [Bit 3]	CS	I	Chip Select (neg true)
2	CMD	PP	Command/Response	DI	I	Data In
3	V_{SS1}	S	Supply voltage ground	VSS	S	Supply voltage ground
4	V_{DD}	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS2}	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1	I/O/PP	Data Line [Bit 1]	RSV		
9	DAT2	I/O/PP	Data Line [Bit 2]	RSV		

下面以SD 1bit mode 为例子介绍怎么驱动SD 卡。

SD卡是以命令驱动的，所以应先了解命令的基本格式，见下图！

4.8.2. Command Format

(Command length 48 bits, 1.92 µs @ 25 MHz)

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC7[1]	end bit

Commands and arguments are listed in Table 5-3 through Table 5-10.

7-bit CRC Calculation: $G(x) = x^7 + x^3 + 1$

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

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

其中需要用到CRC7

下面的图是用示波器实际采集到的　CMD6发送时的波形

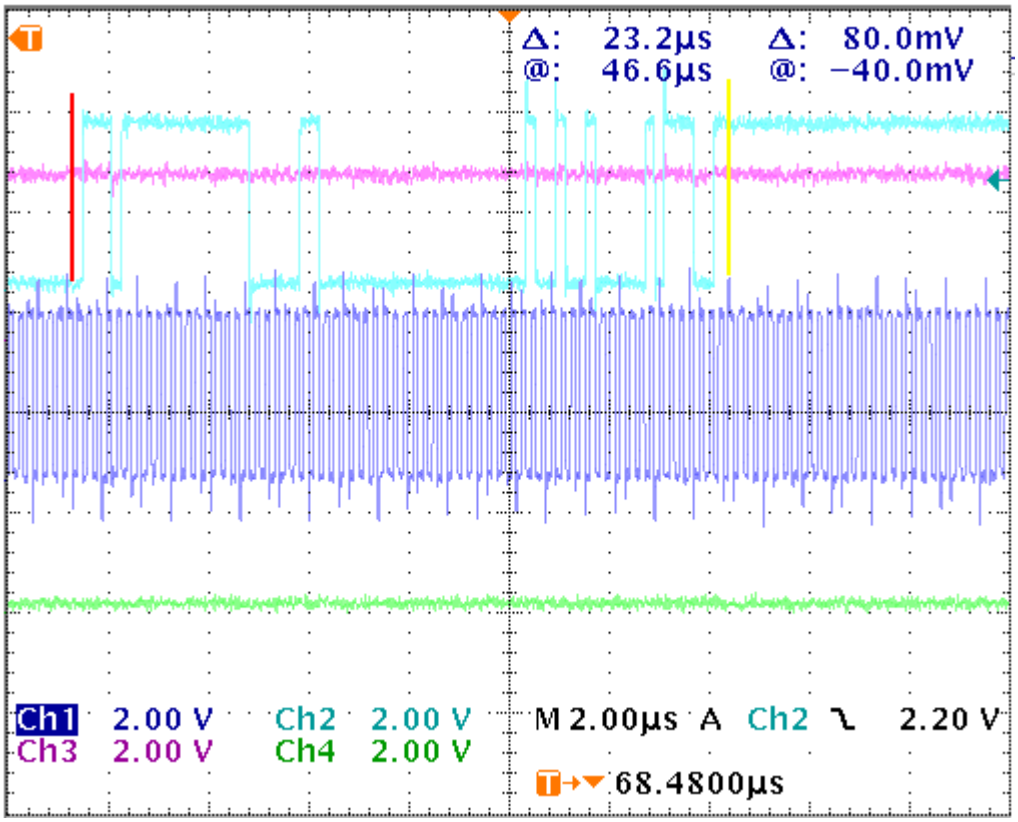
图例说明：

深蓝为CLK信号

浅蓝为CMD信号

红色CMD开始

黄色CMD结束



结合上面的图很容易理解命令发送格式

CRC7码可以用下面函数计算

```
//=====================================================================
unsigned char SD_CRC7(unsigned char *p, unsigned char len)
{
    unsigned char i, CRC=0;
    while(len--)
    {
        for(i=0x80;i!=0;i>>=1)
        {
            if( (CRC&0x40)!=0 )
            {
                CRC<<=1; CRC^=0x09;
            }
            else CRC<<=1;
        if( (*p&i)!=0 ) CRC^=0x09;
        }
    p++;
    }
    CRC=CRC&0X7F;
    return CRC;
}
```

另外还会用到CRC16码，在发送数据和接收数据时需要用到，可以用查表方法，那样速度最快

SD卡命令实际使用到的命令有 32条，其中基本命令 25条，用户应用命令7条。

下面表中详细介绍了每条命令的功能

Table 4-3. Basic Commands (Class 0 And Class 1)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] don't care*	-	GO_IDLE_STATE	Resets all cards to Idle State.
CMD1	Reserved				
CMD2	bcr	[31:0] don't care*	R2	ALL_SEND_CID	Asks any card to send their CID numbers on the CMD line. (Any card that is connected to the host will respond.)
CMD3	bcr	[31:0] don't care*	R6	SEND_RELATIVE_ ADDR	Asks the card to publish a new relative address (RCA).
CMD4[1]	Not Supported				
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16] RCA [15:0] don't care*	R1 (only from the selected card)	SELECT/DESELECT_ CARD	Command toggles a card between the 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 de-selection or —re-send CMD3 to change its RCA number to other then 0 and then use CMD7 with RCA=0 for card de-selection.
CMD8	Reserved				
CMD9	ac	[31:16] RCA [15:0] don't care*	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] don't care*	R2	SEND_CID	Addressed card sends its card identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address[2]	R1	READ_DAT_UNTIL_ STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] don't care*	R1b[3]	STOP_ TRANSMISSION	Terminates a multiple block read/write operation.
CMD13	ac	[31:16] RCA [15:0] don't care*	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] don't care*	-	GO_INACTIVE_ STATE	Sets the card to inactive state.

* The bit places must be filled but the value is irrelevant.

Table 4-4. Block Read Commands (Class 2)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read and write).[1]
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[2]
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously send blocks of data until interrupted by a stop transmission command.
CMD19 – CMD23	Reserved				

Table 4-5. Block Write Commands (Class 4)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[3]
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_ BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	Not Applicable				
CMD27	adtc	[31:0] don't care*	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

* The bit places must be filled but the value is irrelevant.

Table 4-6. Write Protection (Class 6)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD28*	ac	[31:0] data address	R1b	SET_WRITE_PROT	This command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29*	ac	[31:0] data address	R1b	CLR_WRITE_PROT	This command clears the write protection bit of the addressed group.
CMD30*	adtc	[31:0] write protect data address	R1	SEND_WRITE_ PROT	This command asks the card to send the status of the write protection bits.
CMD31	Reserved				

Table 4-7. Erase Commands (Class 5)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK_START	Sets the address of the first write block to be erased.
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK_END	Sets the address of the last write block of the continuous range to be erased.
CMD34 … CMD37	Reserved				
CMD38	ac	[31:0] don't care*	R1b	ERASE	Erases all previously selected write blocks.
CMD39 … CMD41	Reserved				

* The bit places must be filled but the value is irrelevant.

Table 4-8. Lock Card Commands (Class 7)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD42 …. CMD54	SDA Optional Commands, currently supported by SanDisk SD Card.				

Table 4-9. Application Specific Commands (Class 8)

CMD INDEX	Type	Argument	Resp.	Abbreviation	Command Description
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command
CMD56	adtc	[31:1] stuff bits. [0]: RD/ WR[1]	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block shall be set by the SET_BLOCK_LEN command.
CMD57 ... CMD59	Reserved				
CMD60 -63	Reserved for Manufacturer				

下面的图表为用户命令：

Table 4-10. Application Specific Commands Used/Reserved by SD Card

ACMD INDEX	Type	Argument	Resp.	Abbreviation	Command Description
ACMD6	ac	[31:2] stuff bits [1:0]bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4 bits bus) to be used for data transfer.
ACMD13	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Card status. The status fields are given in Table 4-28.
ACMD17	Reserved				
ACMD18	--	--	--	--	Reserved for SD security applications.[1]
ACMD19 to ACMD21	Reserved				
ACMD22	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_ BLOCKS	Send the number of the written (without errors) write blocks. Responds with 32bit+CRC data block.
ACMD23	ac	[31:23] stuff bits [22:0]Number of blocks	R1	SET_WR_BLK_ ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"=default (one wr block)[2].
ACMD24	Reserved				
ACMD25	--	--	--	--	Reserved for SD security applications.[1]
ACMD26	--	--	--	--	Reserved for SD security applications.[1]
ACMD38	--	--	--	--	Reserved for SD security applications.[1]
ACMD39 to ACMD40	Reserved				
ACMD41	bcr	[31:0]OCR without busy	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) con tent in the response on the CMD line.
ACMD42	ac	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_ DETECT	Connect[1]/Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin 1) of the card. The pull-up may be used for card detection.
ACMD43 ACMD49	--	--	--	--	Reserved for SD security applications.[1]
ACMD51	adtc	[31:0] staff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

NOTES: 1) Refer to *SD Card Security Specification* for detailed explanation about the SD Security Features
2) Command STOP_TRAN (CMD12) shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

要运行用户命令需要先发送　基本命令CMD55。每次运行都需要先发送 CMD55

其中 R1, R2, R3, R6 为返回数据格式

Table 4-12. Response R1

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	x	x	x	'1'
Description	start bit	transmission bit	command index	card status	CRC7	end bit

Table 4-13. Response R2

Bit Position	135	134	[133:128]	[127:1]	0
Width (bits)	1	1	6	127	1
Value	'0'	'0'	'111111'	x	'1'
Description	start bit	transmission bit	reserved	CID or CSD register incl. internal CRC7	end bit

Table 4-14. Response R3

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	'111111'	x	'1111111'	'1'
Description	start bit	transmission bit	reserved	OCR register	reserved	end bit

Table 4-15. R6 Response

Bit Position	47	46	[45:40]	[39:8] Argument Field		[7:1]	0
Width (bits)	1	1	6	16	16	7	1
Value	'0'	'0'	x	x	x	x	'1'
Description	start bit	transmission bit	Command index ('000011')	New published RCA [31:16] of the card	[15:0] *card status* bits: 23,22,19,12:0 (see Table 4-28)	CRC7	end bit

SD卡内部寄存器介绍：

Table 3-3. SD Card Registers

Name	Width	Description
CID	128	Card identification number: individual card number for identification.
RCA[1]	16	Relative card address: local system address of a card, dynamically suggested by the card and approved by the host during initialization.
CSD	128	Card specific data: information about the card operation conditions.
SCR	64	SD Configuration Register: information about the SD Card's special features capabilities.
OCR	32	Operation Condition Register

NOTE: 1) The RCA register is not available in SPI Mode.

R2,R3,R6,能返回寄存器状态或使用 CMD9.CMD10读寄存器内容

下面的截图为发送 CMD24后 ， SD卡回复的数据信息

图例说明：

深蓝为CLK信号

浅蓝为CMD信号

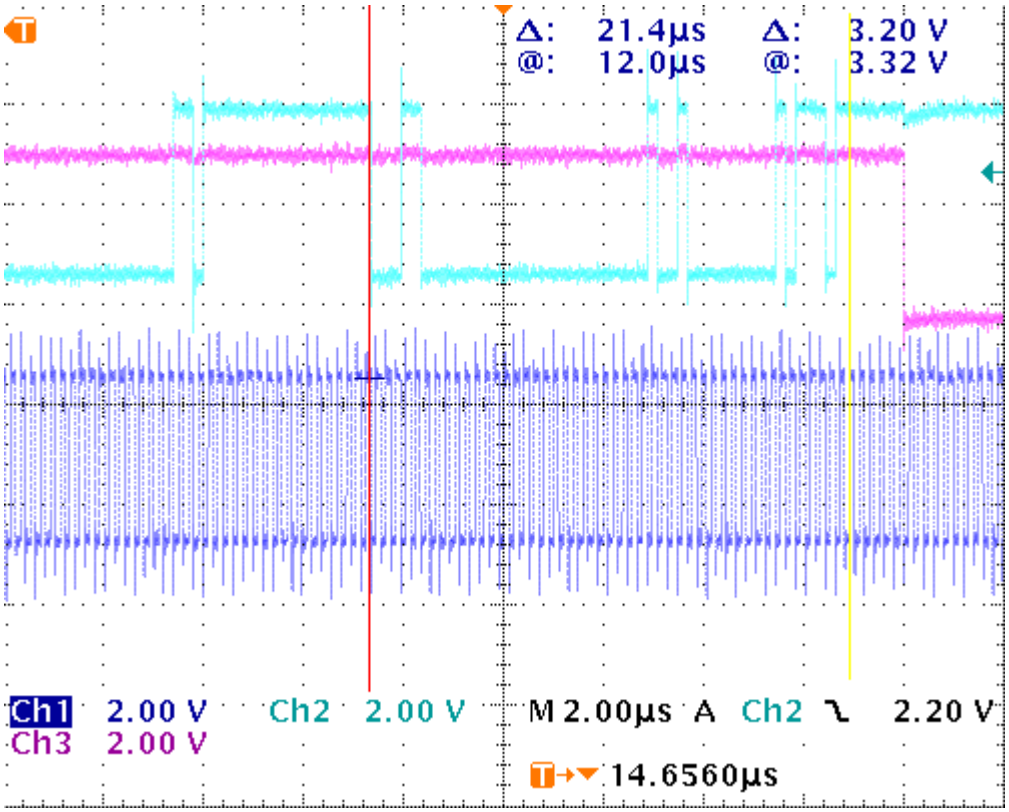
红色CMD开始

黄色CMD结束

紫色为DATA0

了解命令发送格式后，就可以继续下一步了，向 SD卡发送初始化命令和读写数据了，调试过程中这一步很容易出问题，主要是时序匹配不上，发送命令后 SD卡没有反映，不过 CMD0是个例外

用I/O模拟CLK,CMD时 需要注意 数据是上升沿有效，下降沿触发。



SD卡操作总流程！

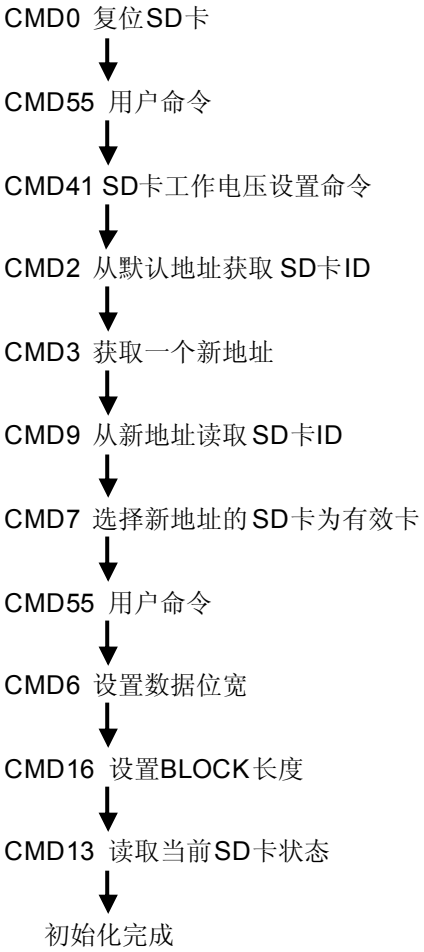
SD卡支持级联，可以在总线上挂多个设备，通过初始化命令选择卡，状态转换命令退出。不支持单个SD卡模式选择。

驱动模式选择：

SD卡上电时 如果CS接地，SD卡将进入SPI模式，CS接高电平，将进入 SD BUS模式
在SD BUS模式时 发送CMD0 SD卡不会回应，在SPI模式时 发送CMD0, SD卡能回应R1信息。

初始化SD卡：

进入SD BUS 模式后，发送CMD0复位SD卡。由于处于SD BUS模式，SD卡此时不会回复信息。发送命令的时候需要注意 SD卡命令状态，不同的状态只能响应特定的命令 ，下面为初始化命令发送流程：



初始化完成后，就可以对 SD卡进行读写操作了 。
调试的时候需要注意 ACMD41 用户命令，只有这个命令操作成功后，下面的 CMD2才能响应，ACMD41命令失败时，需要重试，从 CMD0执行到ACMD41命令执行成功需要大楷 300ms。
这个过程中的SD卡状态转换如下图：

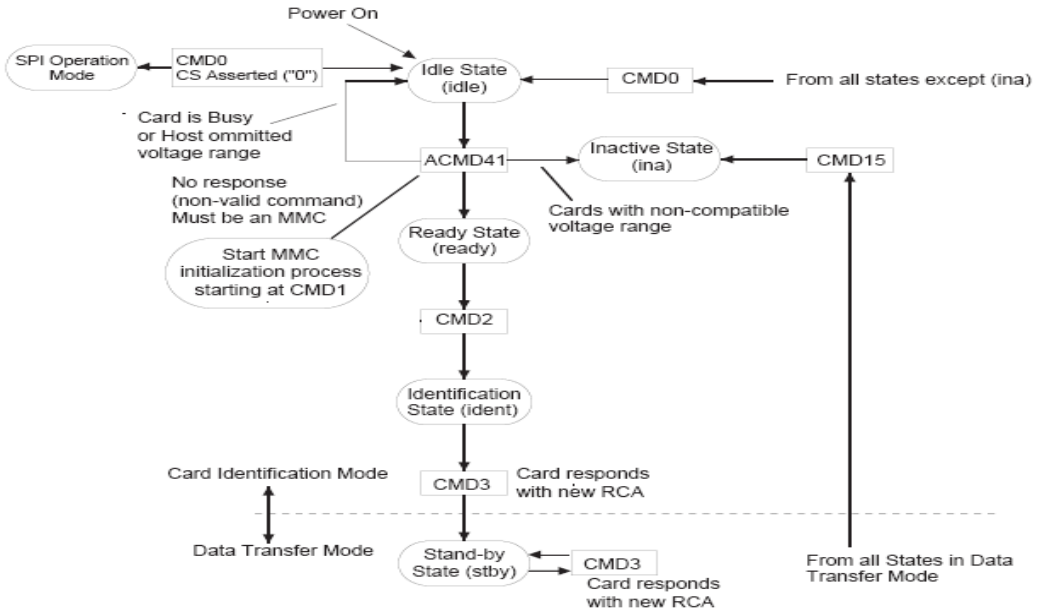


Figure 4-7. SD Card State Diagram (Card Identification Mode)

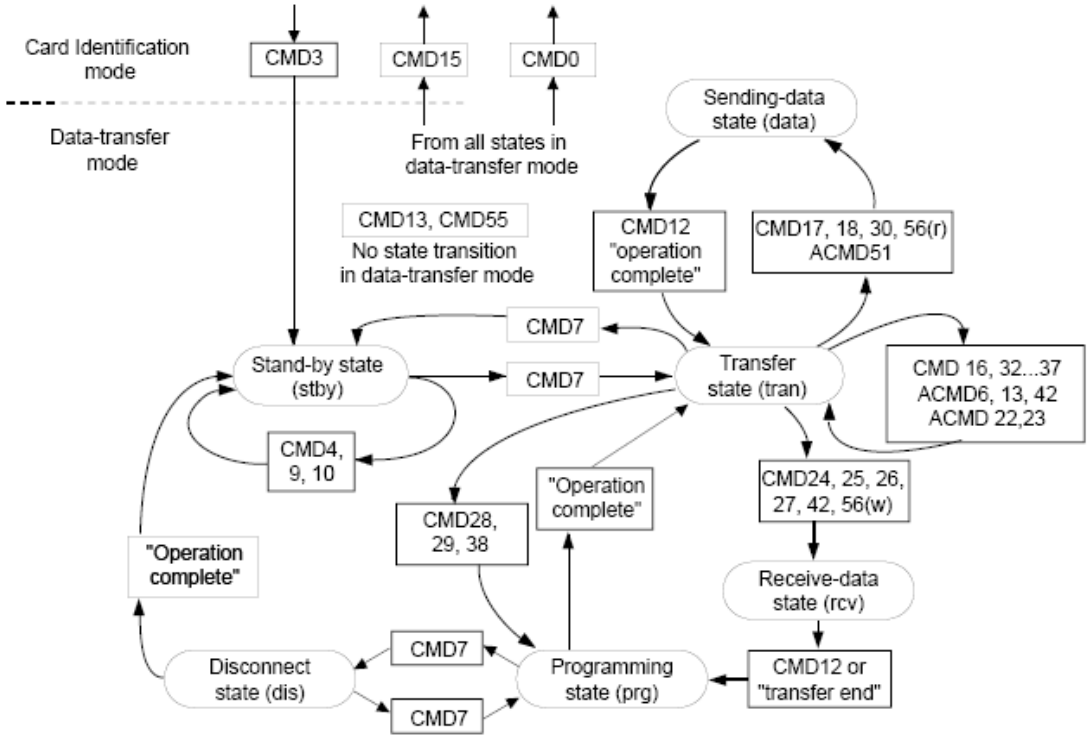


Figure 4-8. SD Card State Diagram (Data Transfer Mode)

发送SD卡支持的操作命令：

不同的SD卡所支持的命令稍有不用，但基本命令都是支持的。详细支持那类命令要从 SD卡的ID中获得。这一步主要是要调通 读，写，擦，这三个功能，这个三个命令调通后其它命令大同小异。读CMD17，写CMD24，擦CMD38.

需要注意的是读写命令都是单个 BLOCK操作，擦可以多个BLOCK操作，需要先设置开始地址 命令CMD32，结束地址命令CMD33。最后发送CMD38指令擦除。下面为详细操作流程：

先说一下SD卡数据发送格式，见下图， sbit1 和sbit4 MODE的区别！

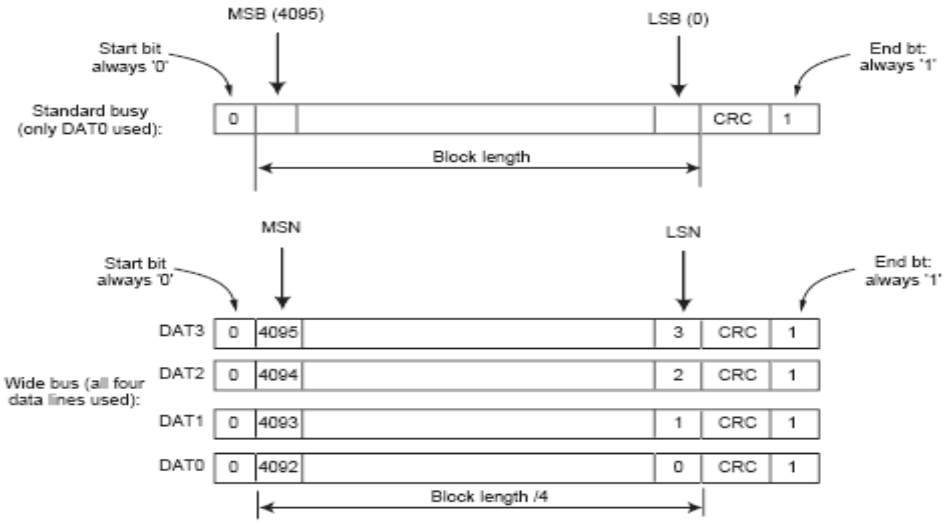
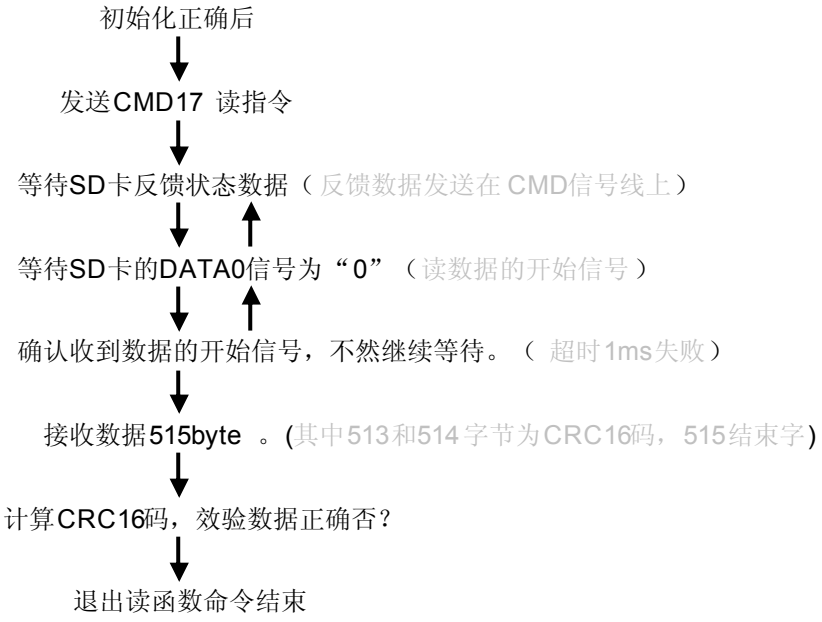


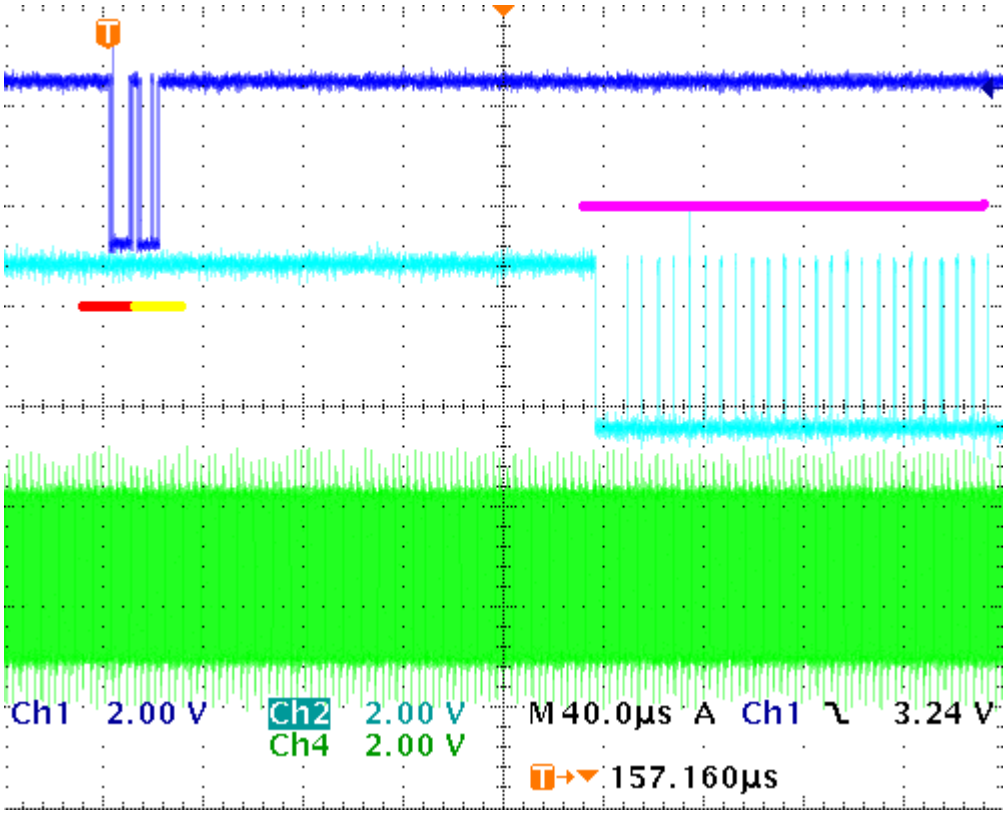
Figure 4-6. Data Packet Format

数据包以512byte为基本单位发送

读的操作流程：CMD17



下图为执行CMD17的截图：



图例说明：

深蓝为CMD信号

浅蓝为DATA0信号

红色段为CMD17命令

黄色段为SD卡回复的状态数据

紫色段为开始读取 SD卡数据

绿色为CLK信号

读命令需要注意的是等待回复问题， “0”为start bit;

结合上面的流程说明和 CMD17的截图，编写代码就容易了。

写的操作流程：CMD24

初始化正确后

↓

准备好要发送的数据和 CRC16效验码

↓

发送CMD24 写指令

↓

等待SD卡反馈状态数据（反馈数据发送在 CMD信号线上）

↓ ↑

写SD卡的DATA0信号为“0”（写数据的开始信号）

↓

写数据515byte 。(其中513和514字节为CRC16码，515结束字)

↓

准备接收SD卡反馈的写状态字 （反馈数据发送在 DATA0信号线上）

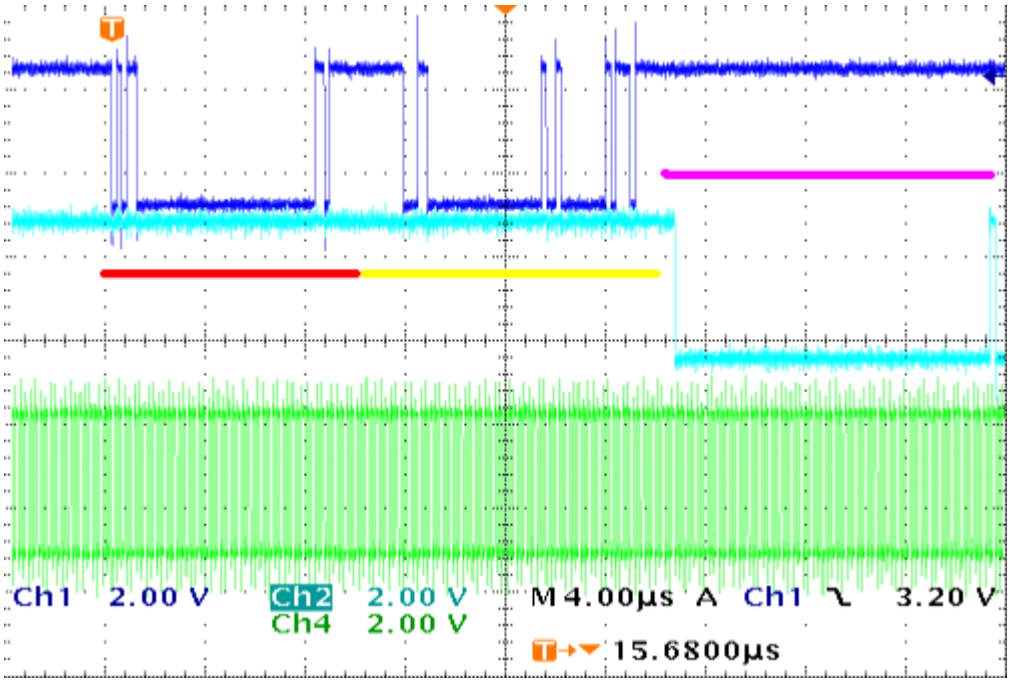
↓

判断接收到的状态字是否为 “0，1，0”如果是则写正确，不然则出错

↓

退出写函数命令结束

下图为CMD24的截图：



图例说明：

深蓝为CMD信号

浅蓝为DATA0信号

红色段为CMD24命令

黄色段为SD卡回复的状态数据

紫色段为开始写数据到 SD卡

绿色为CLK信号

写命令和读命令差不多，但这个命令本人则调试了很久才调通，主要是在 SD卡回复后CPU对卡的写操作要延迟7ms之多，黄色段到紫色段的严重超时，正常情况看上图，基本没有延迟，具 体原因跟（判断状态函数）有关系。

本人用的调试系统为 (ARM 44B0 60M＋8M SDRAM＋2M ROM)

擦除块的操作流程:CMD32 CMD33 CMD38

初始化正确后

↓

发送CMD32命令指定擦除块的开始地址

↓

发送CMD33命令指定擦除块的结束地址

↓

发送CMD38命令 擦除开始

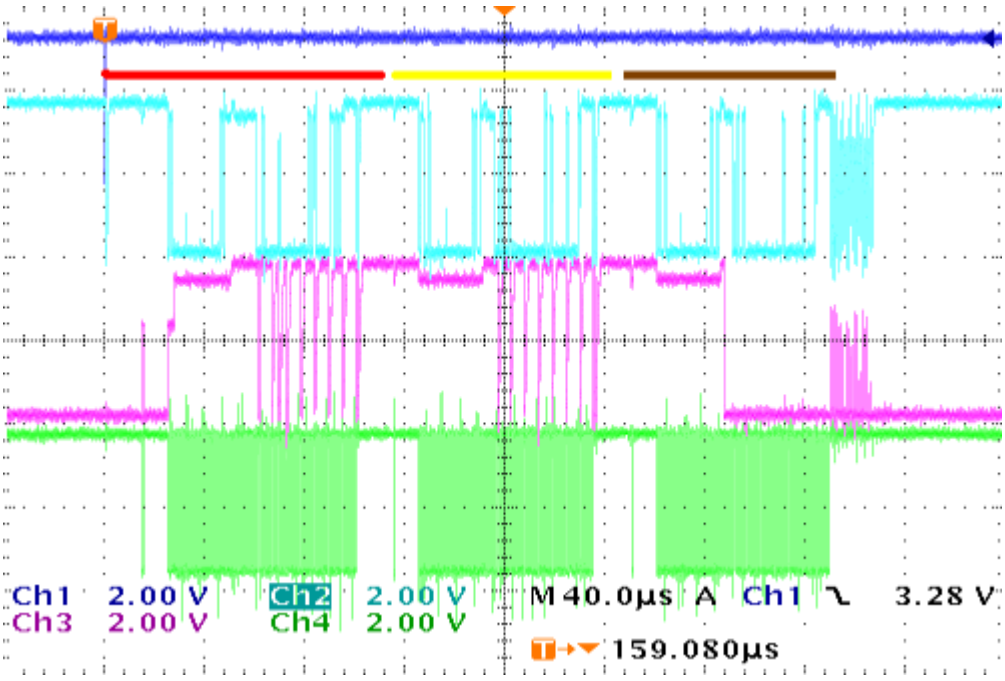
↓

接收SD卡返回的状态字

↓

命令结束

下图为整个擦除过程中的截图



图例说明：

浅蓝为CMD信号

红色段为CMD32命令和回复信号

黄色段为CMD33命令和回复信号

棕色为CMD38命令和回复信号

不用关注紫色

绿色为CLK信号

擦除命令比较容易处理，一般命令正确发送 ,就能执行。其他命令如连续读，连续写，都大同小异，需要注意的是连续写或读 需要用CMD12停止数据发送或写的状态，转换到空闲状态。单个块读或写不需要停止命令。

退出CD卡操作：

如果总线上挂有多个 SD卡，就需要用 CMD7命令退出当前 SD卡，从而选择其他的 CD卡操作。CMD7命令为 选择卡或取消卡，在初始化的时候发送 CMD7命令， SD卡将进入数据发送模式，在数

据模式里发送CMD7则退出当前卡的操作。需要注意的是，在数据模式里不同的状态只支持特定 命令，如在数据模式里发送读 ID指令则无效。

总结：

该驱动流程主要是针对 I/O模拟SD BUS时序写的，如果使用专用 SD桥接IC的话，则编程方法有所不同，但命令操作顺序是一样的。如果 CPU支持SPI模式则推荐用，如果采用 I/O口模拟时钟，延迟不容易控制，特别是在写命令的时候，特容易出错。 SPI模式和SD BUS 1bit MODE的操作效果一样，只是SPI模式支持的命令多一点。 其它并无区别。 I/O口模拟时需要注意上拉电阻，不加的话，读的数据容易出错。另外在调试 SD卡驱动的时候发现，在（状态检测函数）里加 printf函数会增加不稳定行。程序经常因此跑飞， WHY? 这个问题我也没有搞清楚！去掉后就一切正常 。我们经常用printf函数来打印程序运行状态，虽然方便直观，但是用的太多，问题似乎也多多。在这个SD卡驱动编写调试过程中，表现很明显，很多次程序跑飞就是因为它。

Jack Peng
2005 10.19