

File System Brief Introduction Based on SD

Dept. 2
Personal Entertainment Div.
12/1/2006



目录

- ❖ SD卡简介
- ❖ 文件系统概述
- ❖ FAT介绍
- ❖ 基于SPCA536文件系统介绍
- ❖ Q&A





SD卡简介



SD卡简介

- SD card(Secure Digital Memory Card) 是一种基于flash的存储卡。
- SD 卡是由松下电器Matsushita Electric Industrial Co.,Ltd, 东芝Toshiba Corporation 和SanDisk Corporation联合推出,1999年8月首次发布。
- SD卡主要是为了解决新兴的音频、视频电子设备的安全、存储容量、性能、操作环境等方面需求。



SD卡系统特征（1）

- 主要是向portable及stationary的应用。

- 电压范围

SD 卡：

基本通讯(CMD0,CMD15,CMD55,ACMD41)：
2.0v-3.6v.

其它命令：2.7v-3.6v

SDLV(low voltage) 卡：1.6v-3.6v

- Read-only和read/write两种卡。
- Clock rate 0-25mhz
- 兼容MMC卡
- 支持版权保护，遵守SDMI标准。



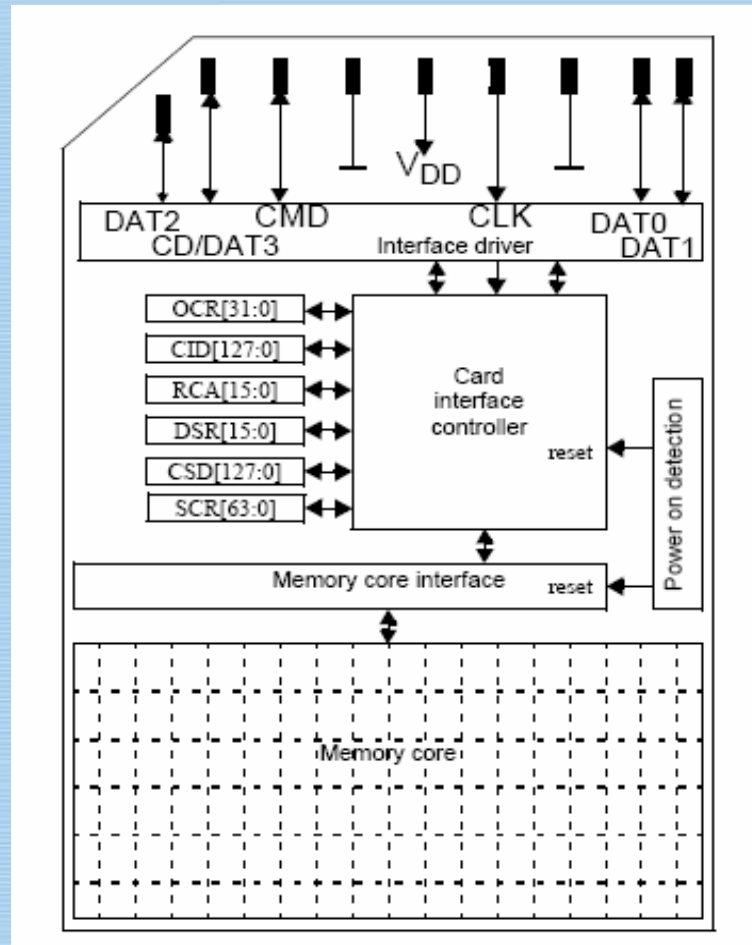
SD卡系统特征（2）

- 支持密码保护机制。
- 支持机械写保护机制。
- 支持内嵌临时或永久写保护机制。



SD卡内部结构

SD卡内部结构可分为：
卡控制器(CardController)、
内存控制器
(MemoryController)、
及内存
(Memory Chip)
三大部分



SD卡外部接口（1）

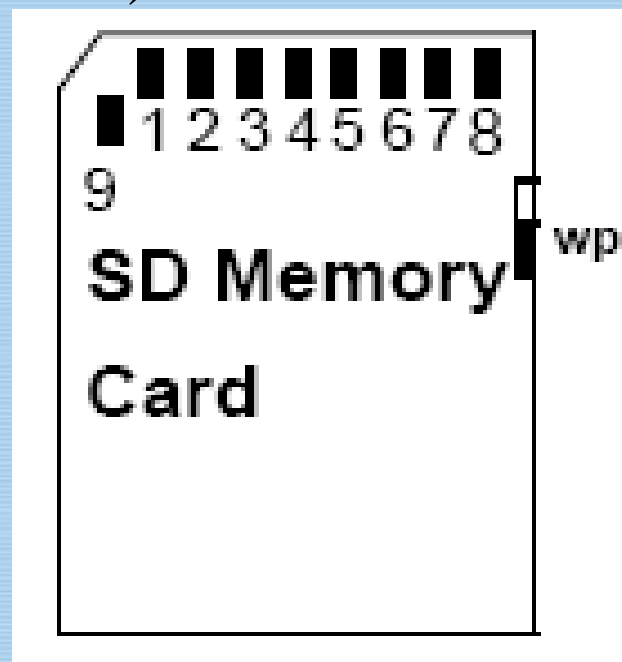
SD卡采用九pin外部通讯接口。

3支power pin(include 2 ground)

1支CMD pin

1支clock pin

4支data pin



SD卡外部接口（2）

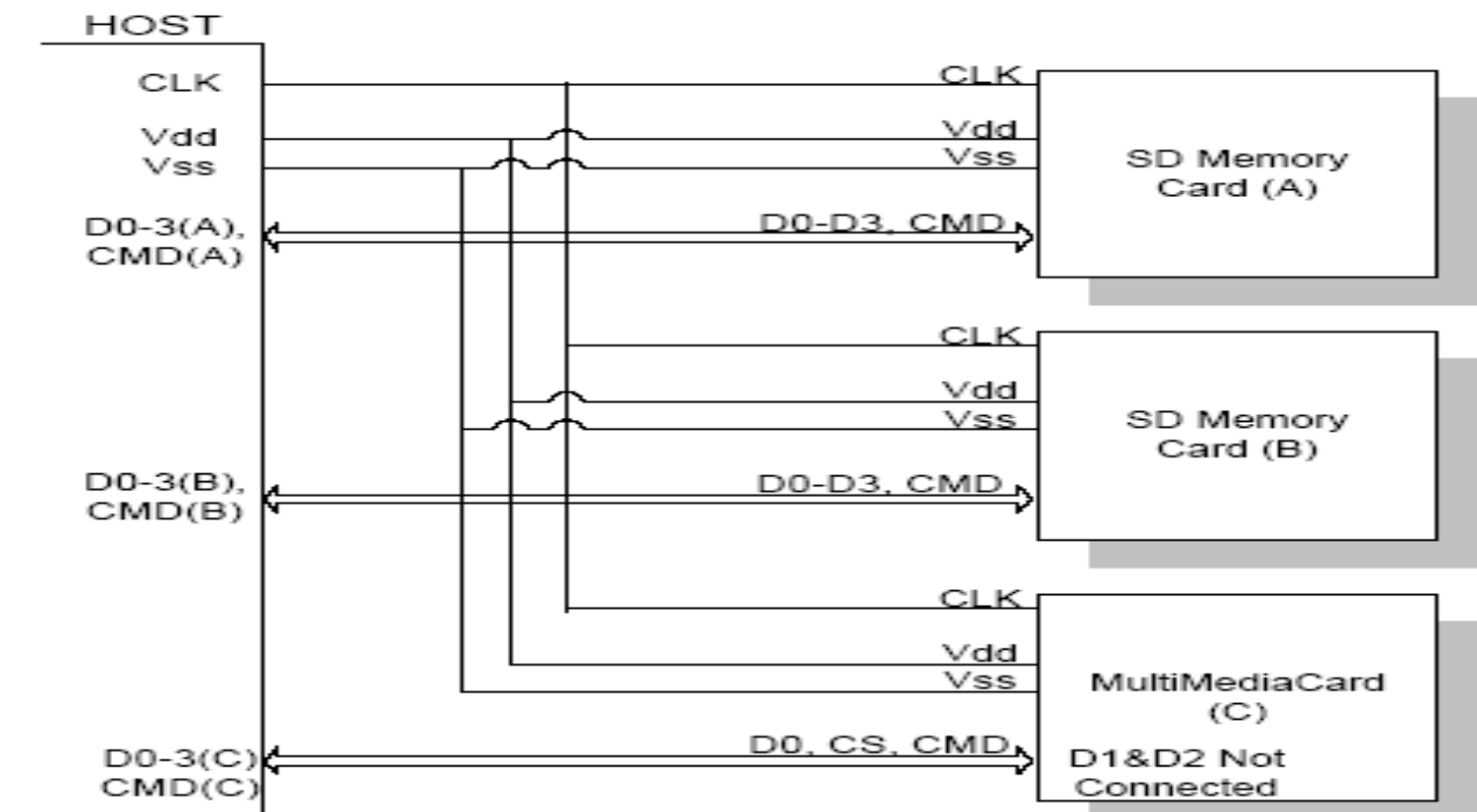
Pin #	SD Mode			SPI Mode		
	Name	Type ¹	Description	Name	Type	Description
1	CD/DAT3 ²	I/O/PP ³	Card Detect / Data Line [Bit 3]	CS	I	Chip Select (neg true)
2	CMD	PP	Command/Response	DI	I	Data In
3	V _{SS1}	S	Supply voltage ground	VSS	S	Supply voltage ground
4	V _{DD}	S	Supply voltage	VDD	S	Supply voltage
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		

SD卡的bus架构

SD卡系统定义了两种通讯协议：SD与SPI。



BUS Type1(1)



BUS Type1(2)

SD bus 包括以下信号：

CLK	host to card clock signal
CMD	bidirectional signal
DAT0-DAT3	bidirectional signal
VDD,VSS1,VSS2	power & ground

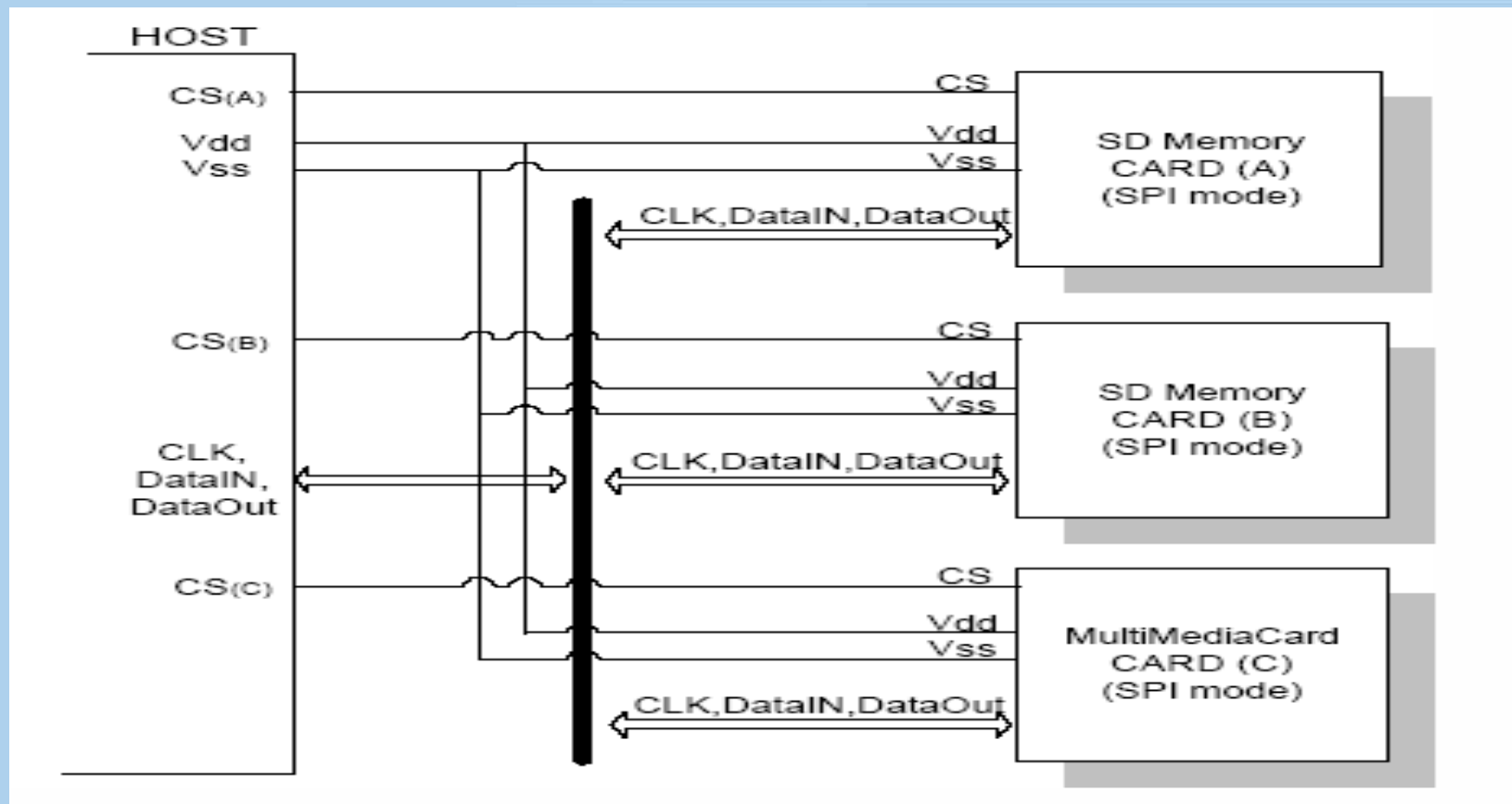
SD bus 特点：

缺点：设计复杂。

优点：效率高。



BUS Type2(1)



BUS Type2(2)

SPI bus 包括以下信号:

CS: host to card chip select signal

CLK: host to card clock signal

DataIn: host to card data signal

DataOut: card to host data signal

SD bus 特点:

优点: 设计相对简单。

缺点: 效率低。



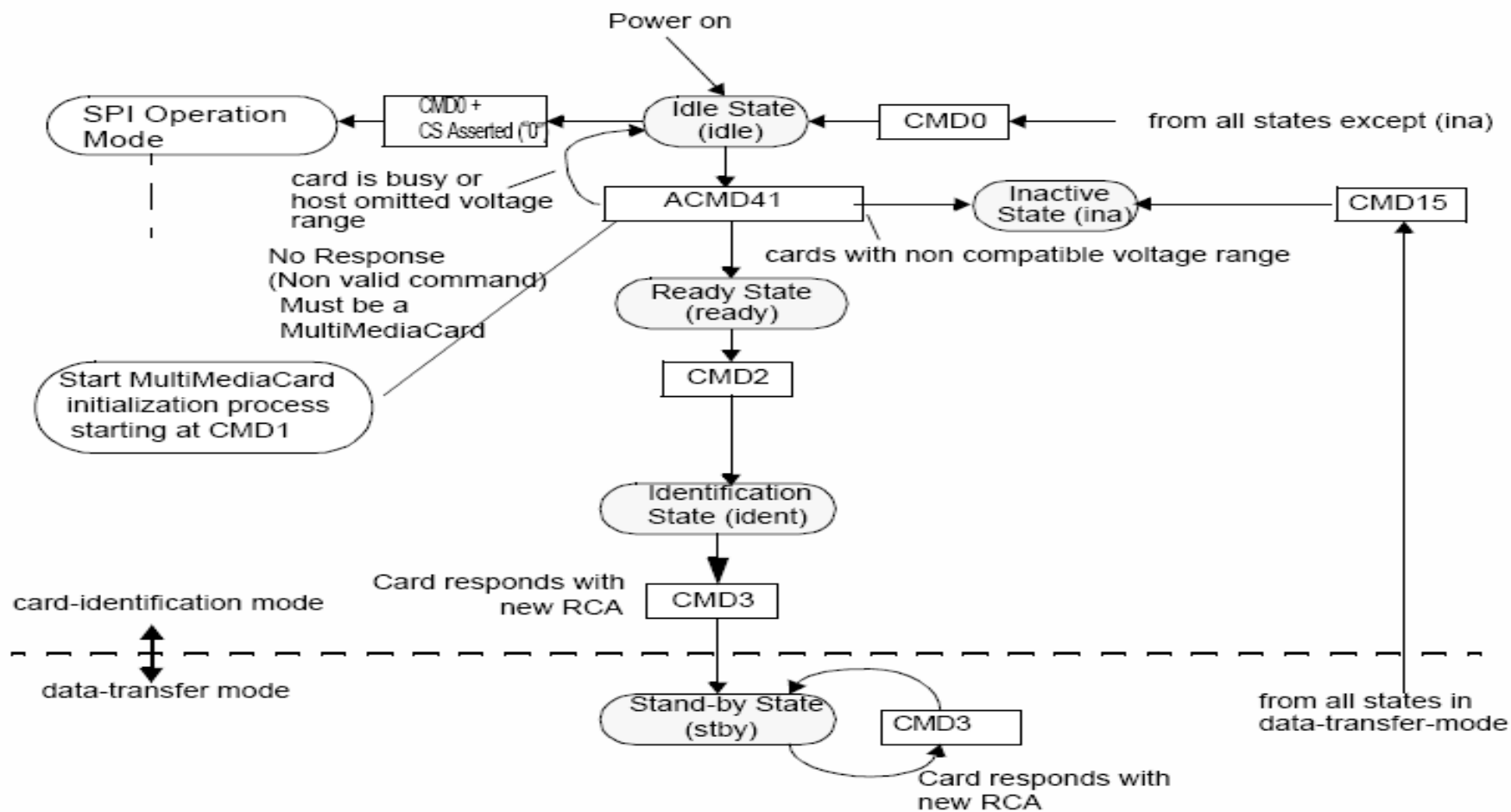
SD卡register

Name	Width	Description
CID	128	Card identification number; card individual number for identification. Mandatory .
RCA ¹	16	Relative card address; local system address of a card, dynamically suggested by the card and approved by the host during initialization. Mandatory .
DSR	16	Driver Stage Register; to configure the card's output drivers. Optional .
CSD	128	Card Specific Data; information about the card operation conditions. Mandatory
SCR	64	SD Configuration Register; information about the SD Memory Card's Special Features capabilities. Mandatory
OCR	32	Operation condition register. Mandatory .

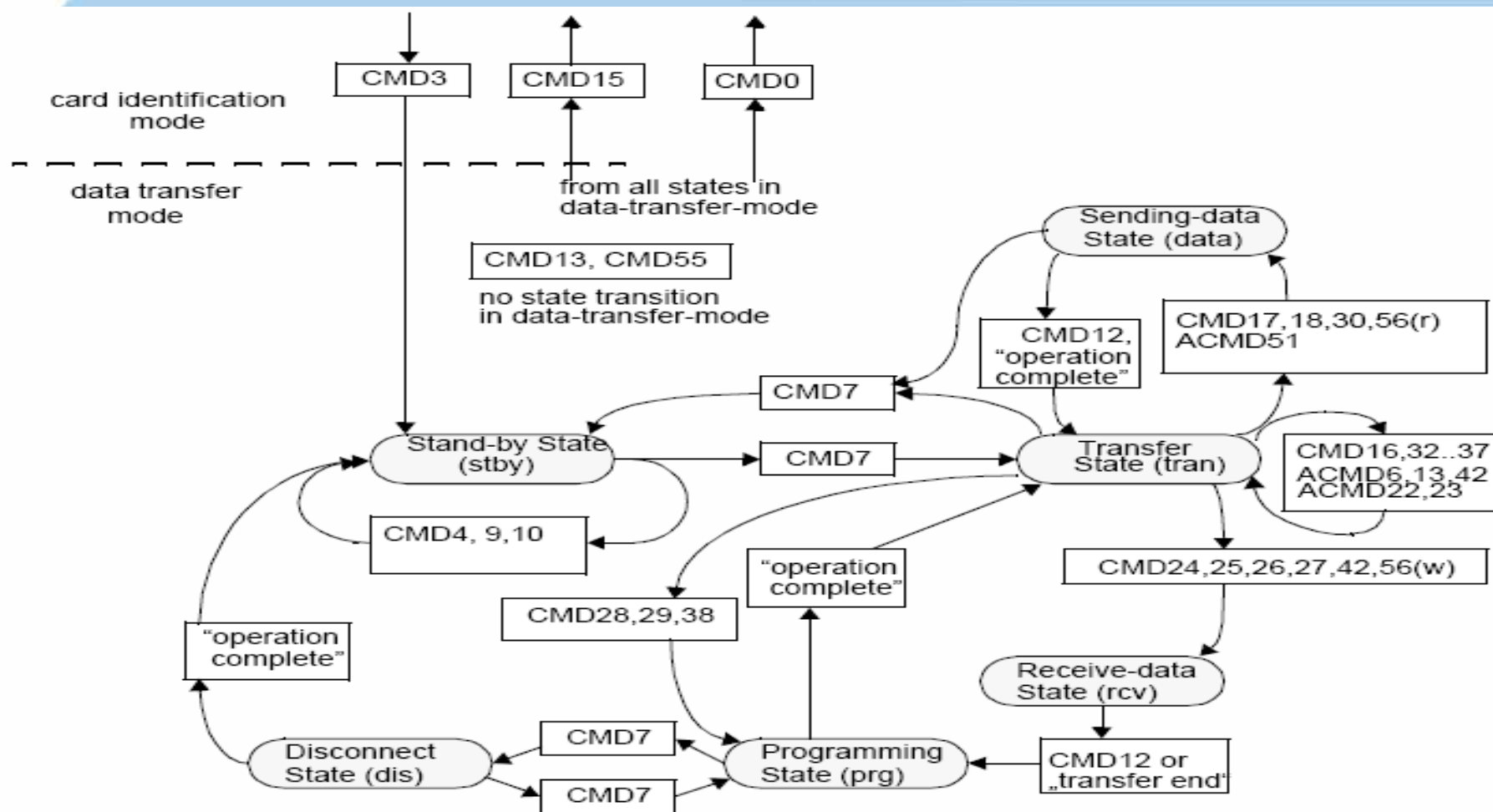
SD卡工作状态

Card state	Operation mode
Inactive State	inactive
Idle State	card identification mode
Ready State	
Identification State	
Stand-by State	data transfer mode
Transfer State	
Sending-data State	
Receive-data State	
Programming State	
Disconnect State	

SD卡识别mode



SD卡数据传输mode



SD卡三种写保护方法

- 机械写保护开关。
- 卡内部写保护(在CSD寄存器设置永久或临时写保护)
- 密码加密(密码和密码长度存在PWD和PWN-LEN寄存器)



SD卡的分区（1）

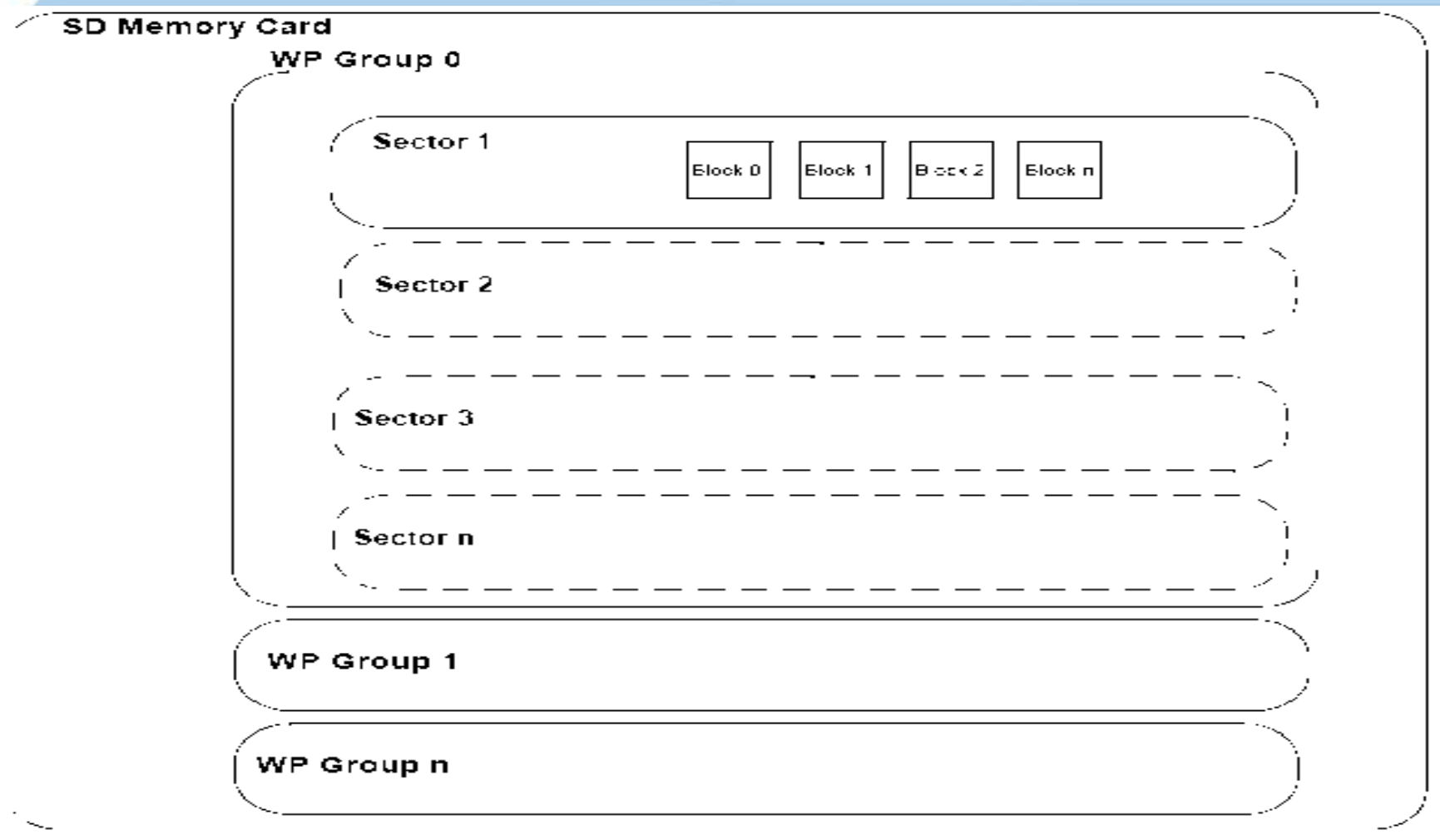
BLOCK:与读写对应

SECTOR:与擦除对应

GROUP:与写保护对应



SD卡的分区 (2)



SDHC 卡

- SDHC (High capacity SD card)的容量从2GB到32GB。
- SDHC的文件系统从FAT12/FAT16升级到FAT32。
- SDHC在速度标识上采用“等级”的概念。

Class2	2MB/s
Class4	4MB/s
Class6	6MB/s





文件系统概述



磁盘的文件系统

文件系统描述：

文件系统是操作系统对数据进行组织管理的系统。操作系统的文件系统包括两个方面：一方面是负责管理文件的系统软件，另一方面是包括被管理的文件。



常见的文件系统

➤ FAT12

一种非常“古老”的磁盘分区方式，它采用12位的文件分区表，能够管理的磁盘容量极为有限。

➤ FAT16

采用16位的磁盘分区表，最大能支持2GB的磁盘分区。FAT16有一个独特的优点，那就是它的兼容性非常好，几乎所有的操作系统（如DOS、Windows 95、Windows 98、Windows NT、Linux等）都支持该分区模式，嵌入式操作系统一般利用它，以在不同操作系统中实现数据交流和交换的。FAT16的缺点，那就是磁盘利用效率较低（以簇为分配单位）。

➤ FAT32:

采用了32位的文件分配表，管理硬盘的能力得以极大的提高，达到了2000GB。

➤ NTFS

Windows NT所采用的一种磁盘分区方式，存在着兼容性的问题，但它的安全性及稳定性却独树一帜——NTFS分区对用户权限作出了非常严格的限制，每个用户都只能按照系统赋予的权限进行操作，同时它还提供了容错结构日志，可以将用户的操作全部记录下来，从而保护了系统的安全。另外，NTFS还具有文件级修复及热修复功能、分区格式稳定、不易产生文件碎片等优点。这些优点进一步增强了系统的安全性。



➤ Linux

Linux一般采用 EXT2/EXT3文件系统。与NTFS一样，Linux分区的安全性及稳定性都比较好。

➤ 其它格式

Nfs（网络文件系统）、Hfs（apple macintosh的文件系统）、ffs（快速文件系统）等等。



基本名词（1）

- 磁头(head)：

硬盘最基本的组成部分是盘片，每个盘片有两面，每个面都有一个磁头。

- 磁道(track)/柱面(cylinder)：

盘片表面上以盘片中心为圆心，不同半径的同心圆称为磁道

- 扇区(sector)：

盘片被分成许多扇形的区域，每个区域叫一个扇区

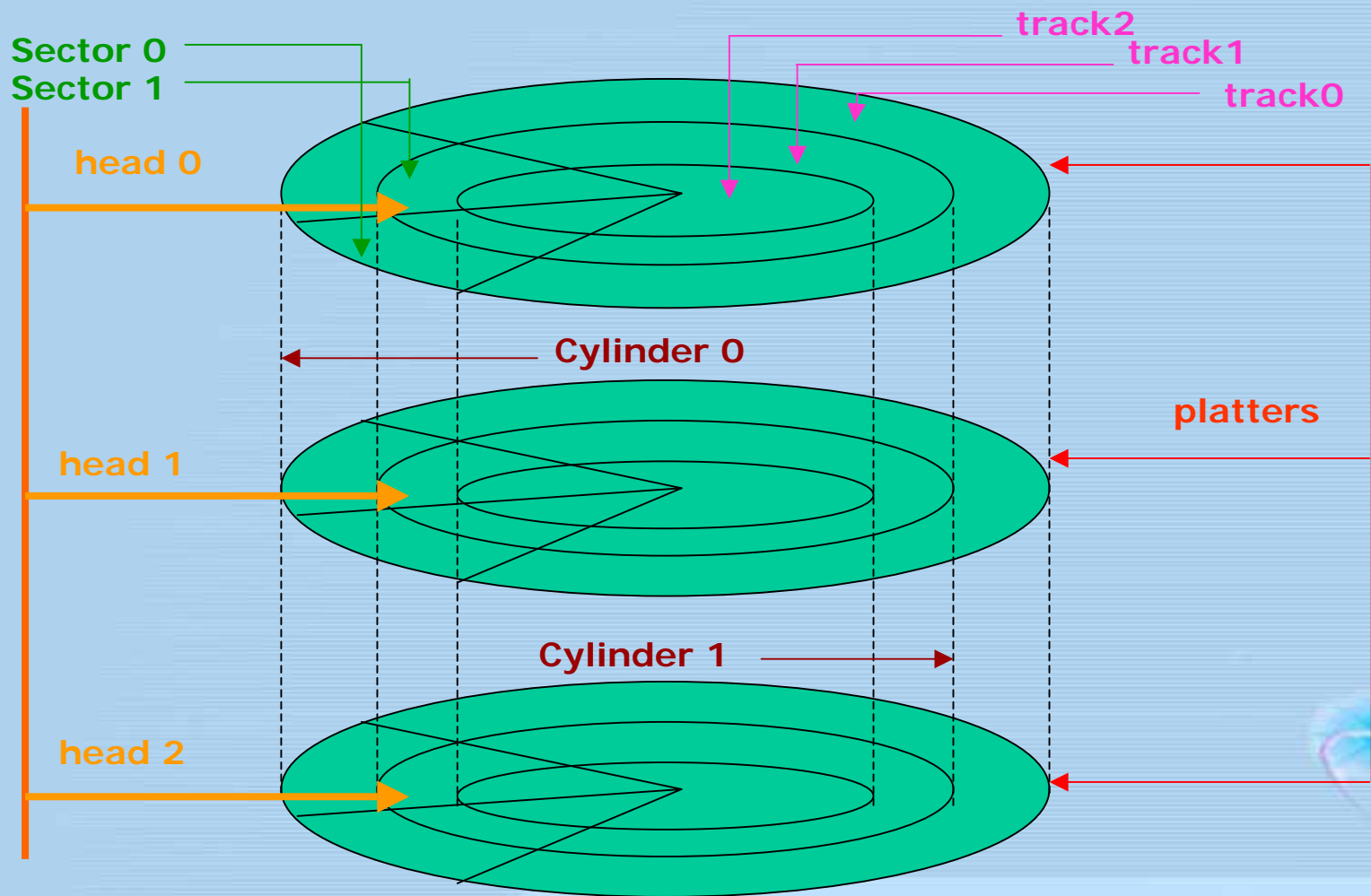
基本名词（2）

- 簇(cluster)

磁盘空间分配的基本单位，它由一个或多个连续的sector组成。



硬盘上数据区域





DOS FAT介绍

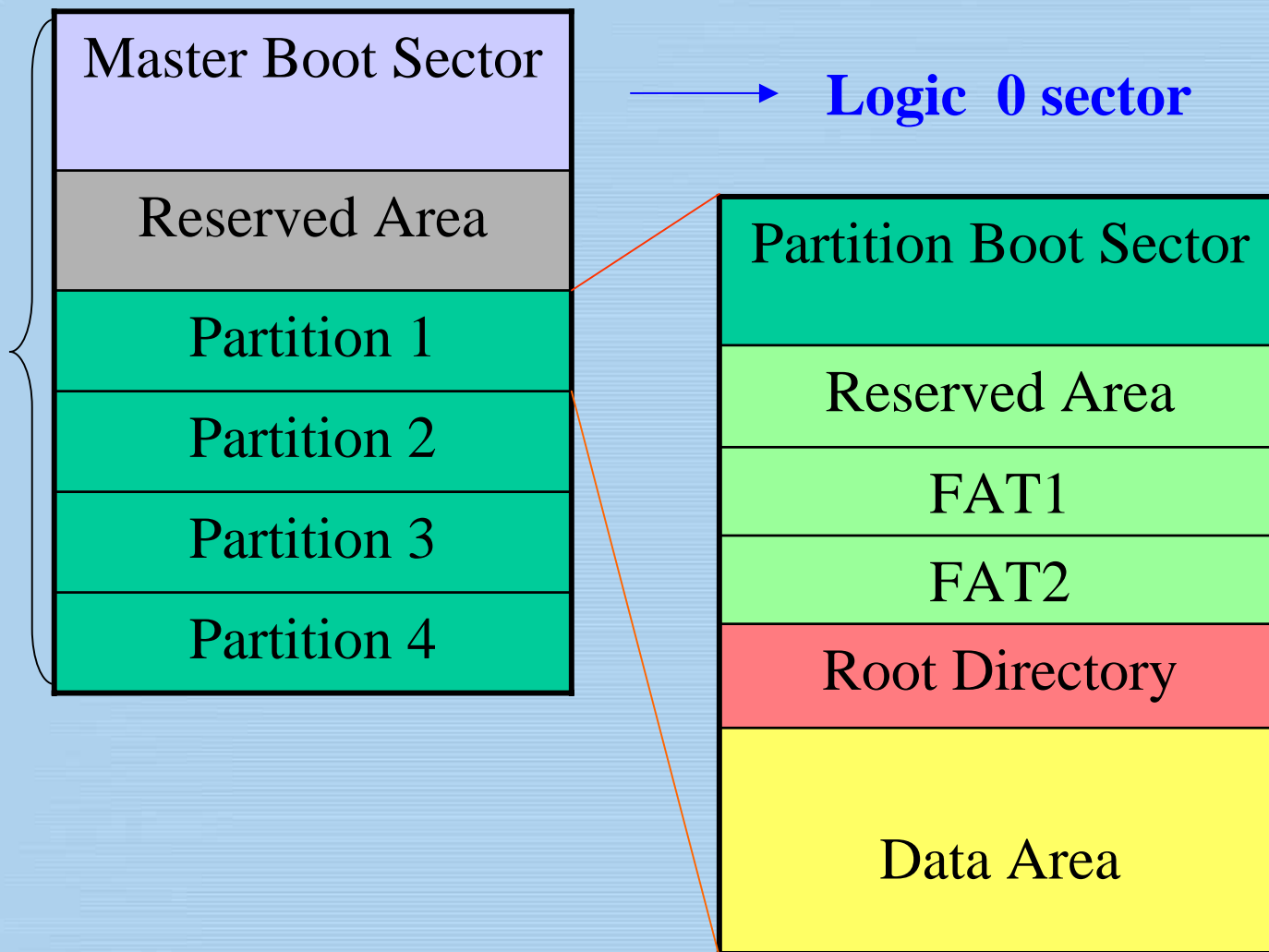


FAT文件系统组成

一个FAT文件系统由以下五个基本的区域组成：

- 0 - 主引导区
- 1 - 分区引导区
- 2 - FAT 区域（两组，一组备份）
- 3 - 根目录区域(FAT32没有)
- 4 - 数据区域





主引导区的内容

Offset	Parameter		Data
000h-1BDh	Boot command(Not in use)		
1BEh	Partition 1	Boot ID	
1BFh		Start head No.	
1C0h		Start sector No.	
1C1h		Start cylinder No.	
1C2h		System ID	
1C3h		End head NO.	
1C4h		End sector No.	
1C5h		End cylinder No.	
1C6h-1C9h		Start logical sector NO. PBS L. A.	
1CAh-1CDh		Partition size Y_SMC_FatEndAddr	
1CEh-1DDh		partition 2 (Not in use)	
1DEh-1EDh		partition 3 (Not in use)	
1EEh-1FDh		partition 4 (Not in use)	
1FEh-1FFh		Fixcd data (Signature)	

分区引导区内容

Offset	Parameter	Data
000h-002h	Jump command	
003h-00Ah	Manufacturer's name and version(ASCII,8 Bytes)	
00Bh-00Ch	Sector Size	
00Dh	Sectors per Cluster	
00Eh-00Fh	Reserve sector count	
010h	Number of FATs (File Allocation Tables)	
011h-012h	Number of root directory entries	
013h-014h *	Total number of partition sectors(Total sector)	
015h	Medium Identifier	0xf8:fixxed
016h-017h	Sectors per FAT	
018h-019h	Sector per Track	
01Ah-01Bh	Number of heads	
01Ch-01Fh	Number of hidden sectors	
020h-023h *	Total number of partition sectors (32 Bits)	
024h	Physical drive NO.	0x80
025h	Reserved	
026h	Extended boot record signatures	0x29
027h-02Ah	Volume ID (4 Bytes)	
02Bh-035h	Volume label (ASCII,11 Bytes)	
036h-03Dh	File system type (ASCII,8 Bytes)	
03Eh-1FDh	Reserved (IPL code area)	boottrap启动程式
1FEh-1FFh	Fixed data (Signature)	0x55AA

文件分配表(1)

FAT Entry Value		Contents
FAT12	FAT16	
000	0000	Indicates that the corresponding cluster is not in use and may be allocated to a file or a directory.
002 to MAX	0002 to MAX	Indicates that the corresponding cluster is already allocated. The value of the entry is the cluster number of the next cluster following this corresponding cluster. Max shall be the Maximum Cluster Number.
MAX+1 to FF6	MAX+1 to FFF6	Shall be reserved for future standardization and shall not be used.
FF7	FFF7	Indicates that the corresponding cluster has a defective cluster.
FF8 to FFF	FFF8 to FFFF	The corresponding cluster is already allocated, and it is the final cluster of the file.

文件分配表(2)

1.文件分配表入口：

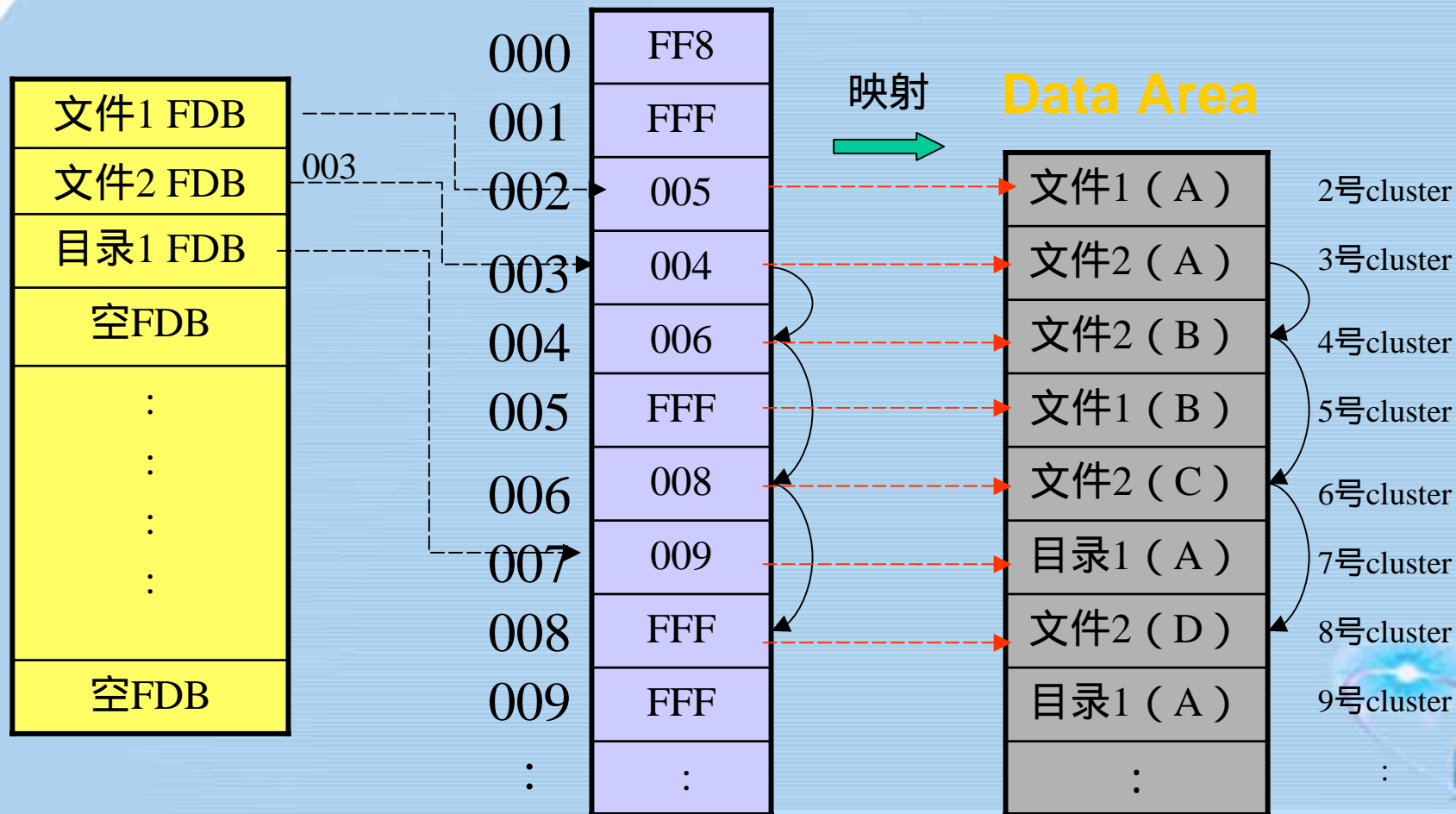
FAT12：12bits; FAT16:16bits; FAT32:32bits

2.第一块数据的cluster编号是从#2开始。

3.FAT12的文件分配表不能超过6k个扇区， FAT16的文件分配表不能起过128K个扇区， FAT32没有限制。

4.文件分配表的FAT[0],FAT[1]保留，FAT[0]低8bit记录Medium Identifier，FAT16/FAT32的FAT[1]的高两位用来dirty flag与read/write error。

文件分配表(3)



文件目录入口

BP	Length	Field Name	Contents
0	8	Name	Depends on entry type
8	3	Name Extension	d-characters
11	1	Attributes	8 bits
12	10	Reserved Field	bytes
22	2	Time Recorded	Numeric Value
24	2	Date Recorded	Numeric Value
26	2	Starting Cluster Number	Numeric Value
28	4	File Length	Numeric Value

文件名

- 0x00:没有使用
- 0xE5档案被删除
- 0x05文件名第一个字为0xE5
- 字母全部是大写
- \$ % ` - _ @ ~ ‘ ! () { } ^ # &可以出现在文件名中。
- 第一个符号不能为空格



文件属性

- Bit0 : 只读
- Bit1 : 隐藏
- Bit2 : 系统
- Bit3 : Volume label
- Bit4 : 目录
- Bit5 : 文件有被改动
- Bit6/Bit7 : 保留

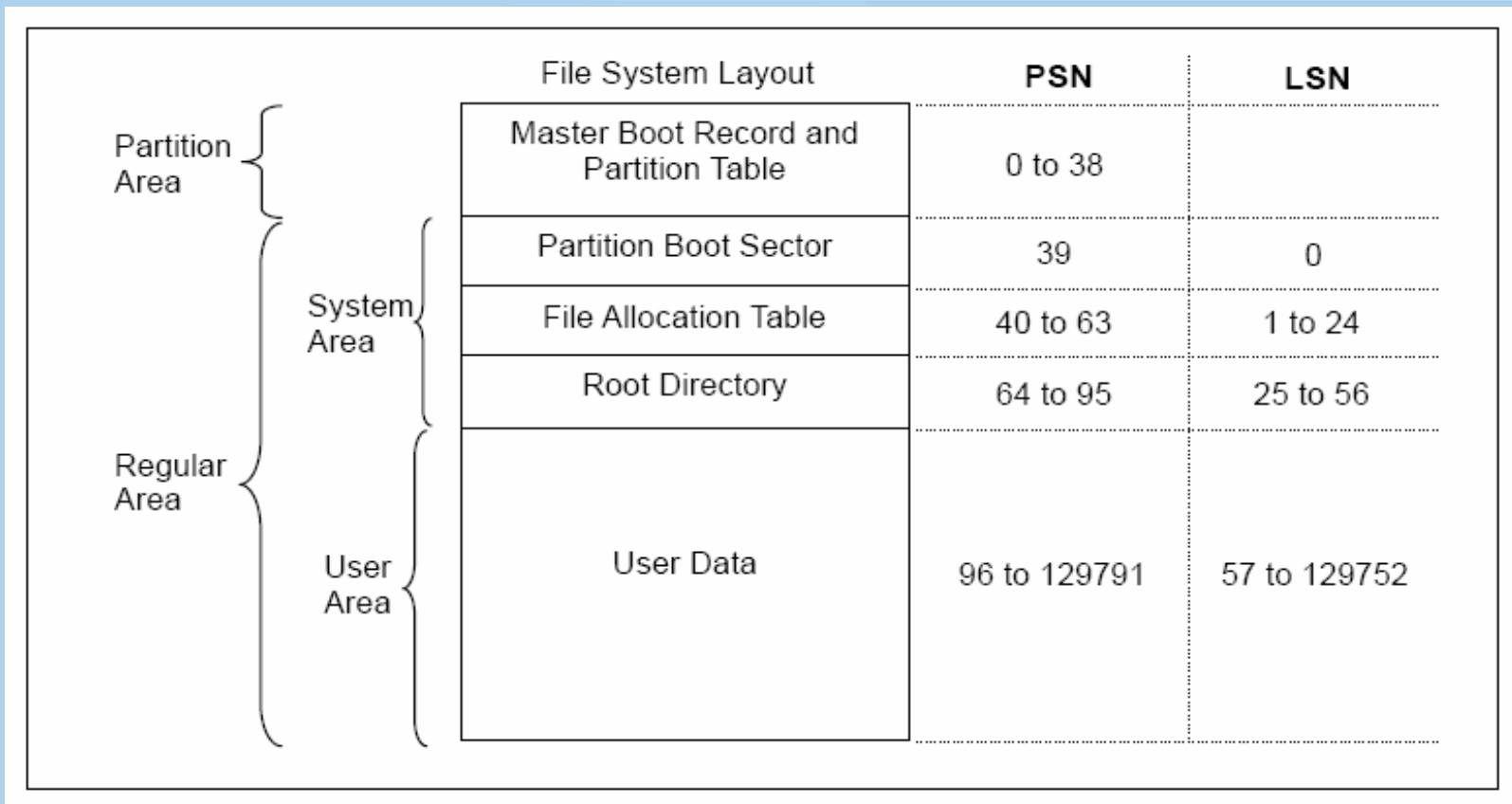


根目录

根目录是一个比较特殊的目录，根目录没
“.”与“..”入口，根目录没有data/time
stamps，只有根目录下才能有属性为
ATTR_VOLUME_ID的入口。



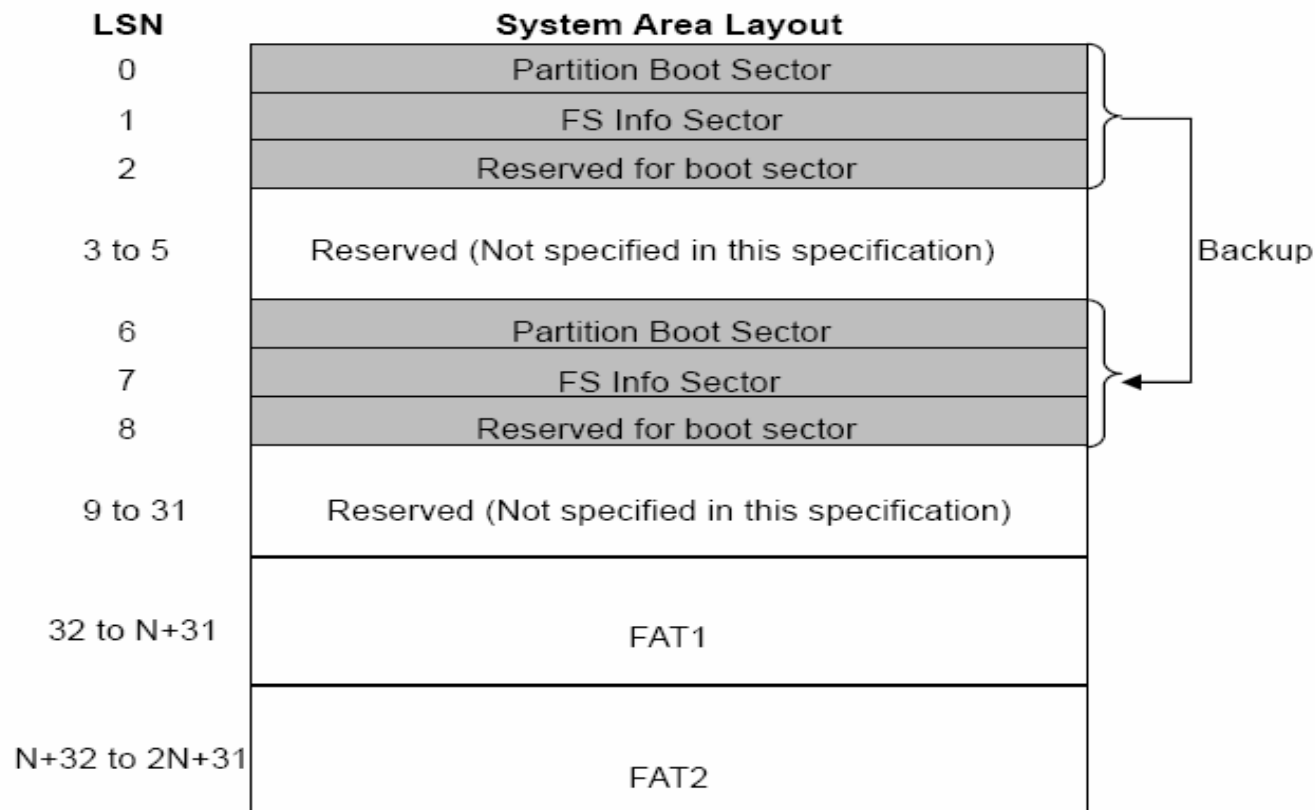
SD FAT12/FAT16 System Area Layout



SD FAT32 System Area Layout

		File System Layout	PSN	LSN
Partition Area	System Area	Master Boot Record and Partition Table	0 to 14321	
		Partition Boot Sector and FS Info	14322 to 14353	0 to 31
Regular Area		File Allocation Table	14354 to 16383	32 to 2061
	User Area	User Data	16384 to 8323071	2062 to 8308749

FAT32 System Area Layout

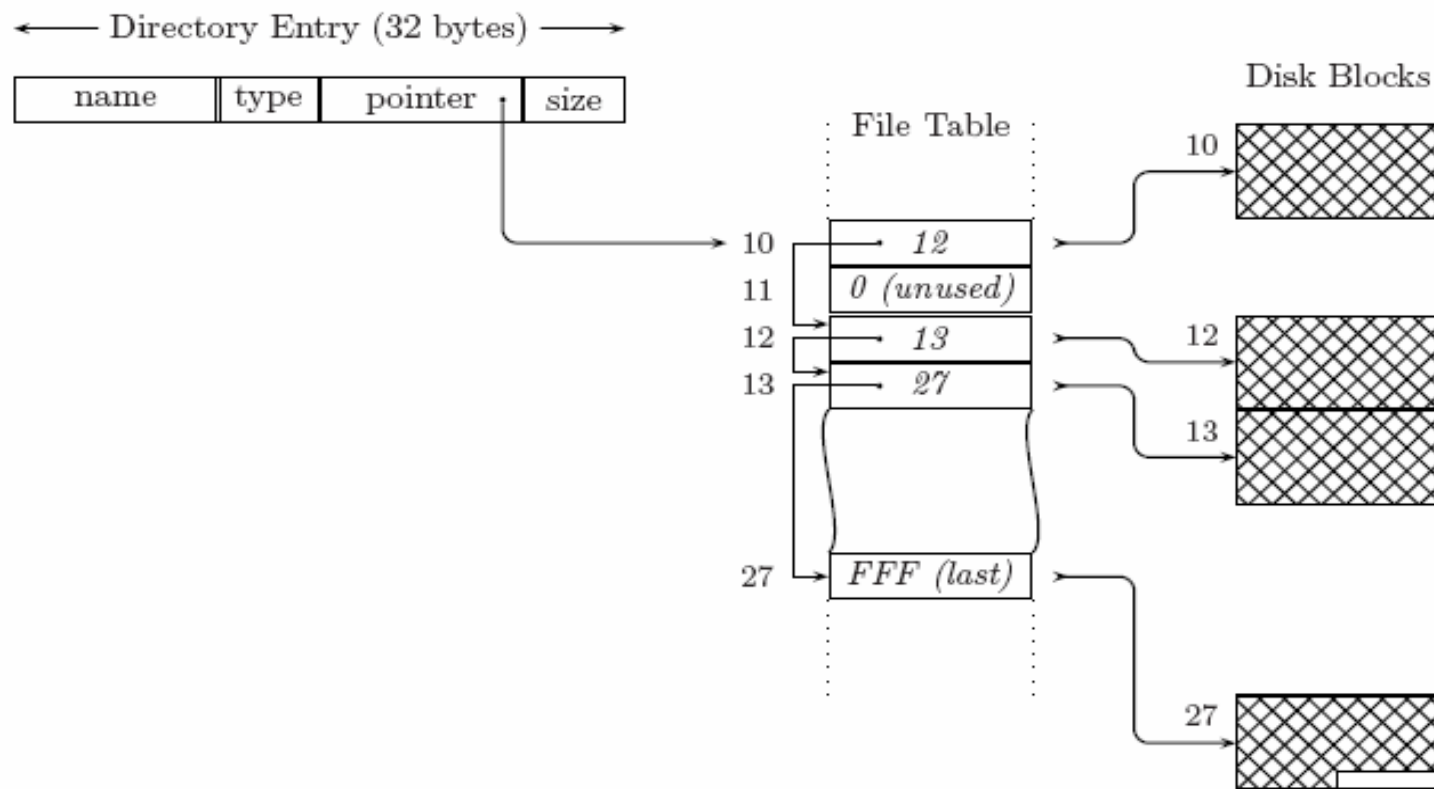


FAT32的根目录

FAT32没有特别为根目录设置一个特殊的位置，放置根目录的第一个cluster被记录在partition boot sector入口的ROOT CLUSTER中。



FAT32文件分配



文件系统的类型的决定

FAT的类型仅由cluster的数目决定:

```
If (CountofClusters < 4085) {  
    /* Volume is FAT12 */  
} else if (CountofClusters < 65525) {  
    /* Volume is FAT16 */  
} else {  
    /* Volume is FAT32 */  
}
```



长文件名（1）

- 长文件名入口紧挨的短文件名入口，并在短文件名之前。
- 长文件名入口以相反的顺序存放。
- 长文件名的最小属性是0x0f。
- 长文件名入口的LDIR_Ord = 0x40|N时表示长文名结束。
- 长文件名是以UNICODE存放，长文件名允许小写字母。
- 长文件名最长为255个字符



长文件名（2）

Descriptive name	Offset(byte)	Size(bytes)	description
LDIR_Ord	0	1	
LDIR_Name1	1	10	
LDIR_Attr	11	1	
LDIR_Type	12	1	
LDIR_Checksum	13	1	
LDIR_Name2	14	12	
LDIR_FstClusLO	26	2	
LDIR_Name3	28	4	

Entry	Ordinal
N th long name directory entry	LAST_LONG_ENTRY (0x40) N
... Additional long name directory entries	...
1 st long name directory entry	1
Short name directory entry associated with preceding long name directory entry set	N/A

长文件名 (3)

Example: “the quick brown.fox”

2nd long entry
(and last) →

42h	w	n	.	f	o	0Fh	00h	chk-sum	x
0000h	FFFFh	FFFFh	FFFFh	FFFFh	0000h	FFFFh	FFFFh		
01h	T	h	e		q	0Fh	00h	chk-sum	u
	i	c	k		b	0000h		r	o
T	H	E	Q	U	I	~	1	F	O
Created Date	Last Access Date	0000h	Last Modified Time	Last Modified Date	First Cluster	20h	NT	Rsvd	Created Time

1st long entry →

Short entry →

目录的生成

1. 目录入口 DIR_Attr | ATTR_DIRECTORY。
2. DIR_FILESize = 0。
3. 至少一个 cluster 分配给该入口
(DIR_FstClusLO, DIR_FstClusHI)。
4. 如果非根目录，两个附带的目录入“.”与“..”生成。



Boot directory

Boot dir address	name	extension	attribute	reserved	time	date	First cluster	File size
	DICM		0x10				002	0

DICM

002 cluter	name	extension	attribute	reserved	time	date	First cluster	File size
	.		0x10				002	0

002 cluter	name	extension	attribute	reserved	time	date	First cluster	File size
	..		0x10				0	0

002 cluter	name	extension	attribute	reserved	time	date	First cluster	File size
	100Media		0x10				003	0

100Media

003 cluter	name	extension	attribute	reserved	time	date	First cluster	File size
	.		0x10				003	0

003 cluter	name	extension	attribute	reserved	time	date	First cluster	File size
	..		0x10				002	0

文件生成 (1)

```
WORD File_Open(BYTE  
*name, BYTE rwMode,  
WORD numCluster)
```

```
BYTE  
DOS_MakeFatChain(WOR  
D* fatReq, BYTE  
continueSearch)
```

```
BYTE File_LWrite(WORD  
fHandle, BYTE srcPath,  
ULONG writeSize, ULONG  
ramAddr, BYTE  
findHeaderPage)
```

FAT chain

G_DOS_FreeFatTable[0]
G_DOS_FreeFatTable[1]
G_DOS_FreeFatTable[2]
...
G_DOS_FreeFatTable[G_DOS_FreeTableTotal]

Free cluser

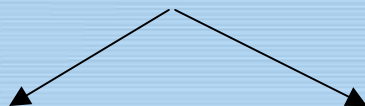
004
005
007
009
00a

file

data	004
data	005
data	007
data	009
data	00a

文件生成 (2)

BYTE File_Close(WORD fHandle, BYTE* name, BYTE rwMode, ULONG fileSize)



FAT

BYTE
DOS_SetFatChain(
WORD fatReq)



005
007
009
00a
FFF

BYTE
DOS_AddBaseFdb(
BYTE *name,
WORD startCluster,
ULONG fileSize)

name
extension
...
004
File size



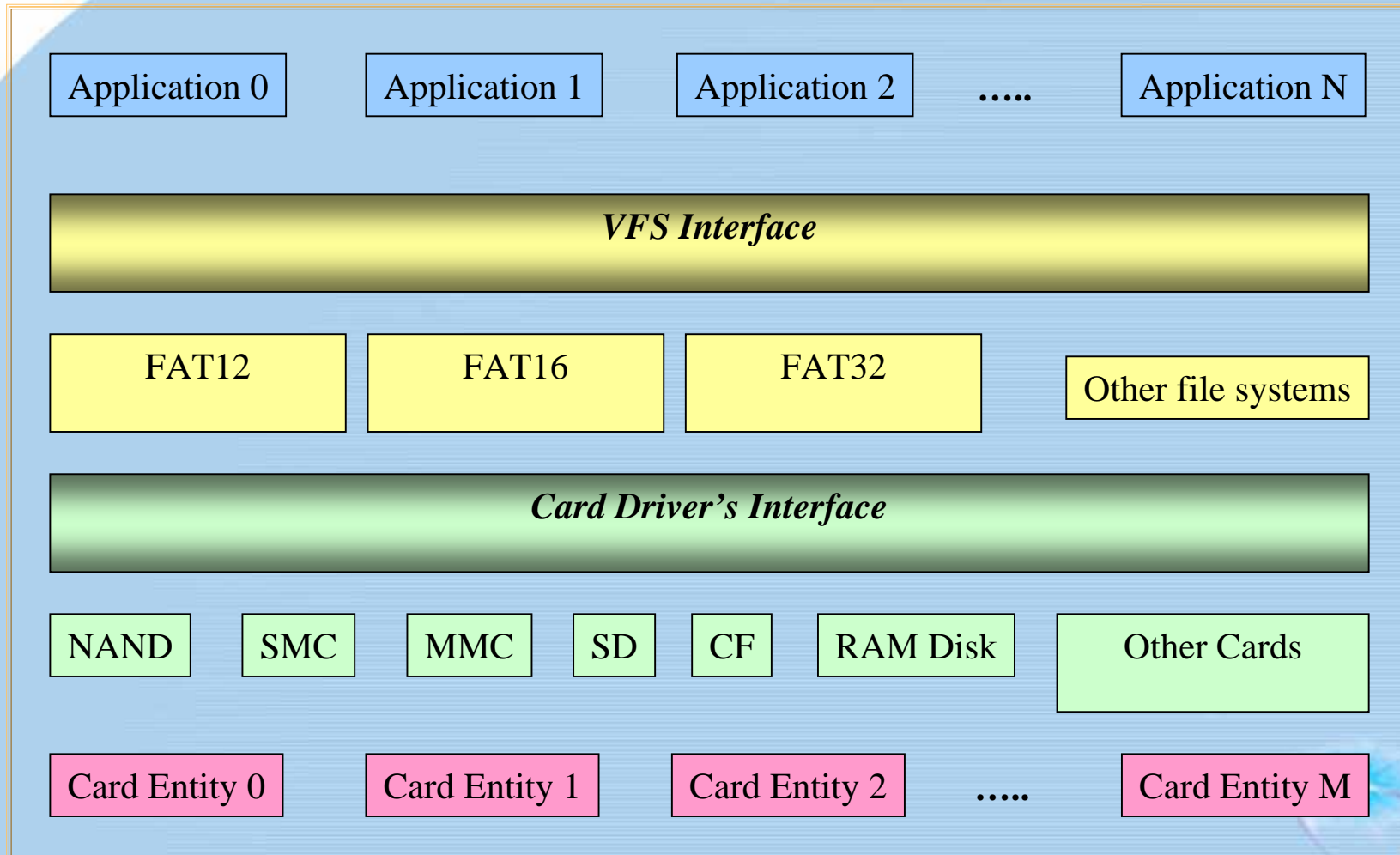
基于SPCA536文件系统介绍



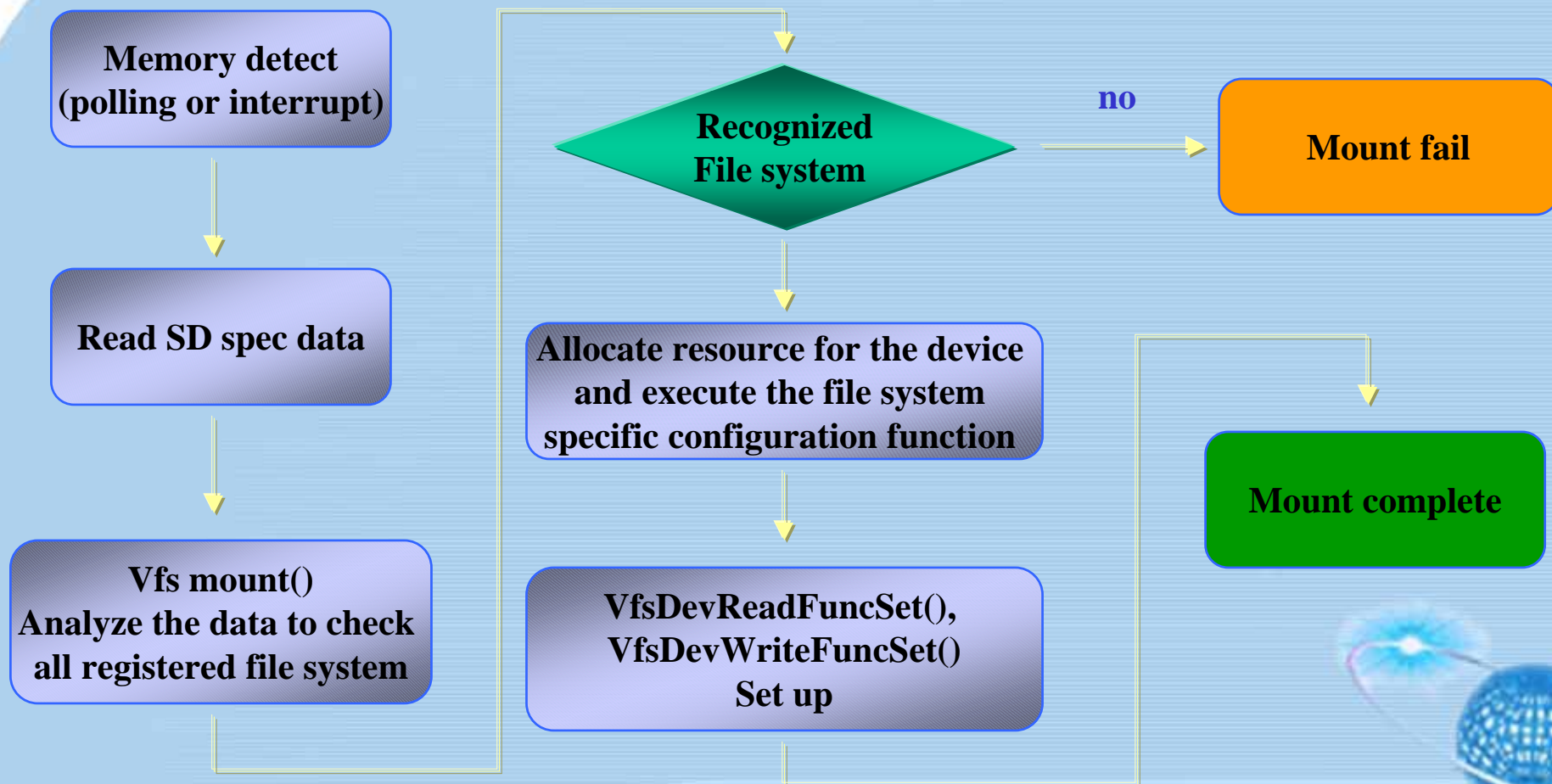
Virtual File system

SPCA536采用virtual file system机制，
virtual file system将具体的文件系统及
storage driver与应用接口分离开来。

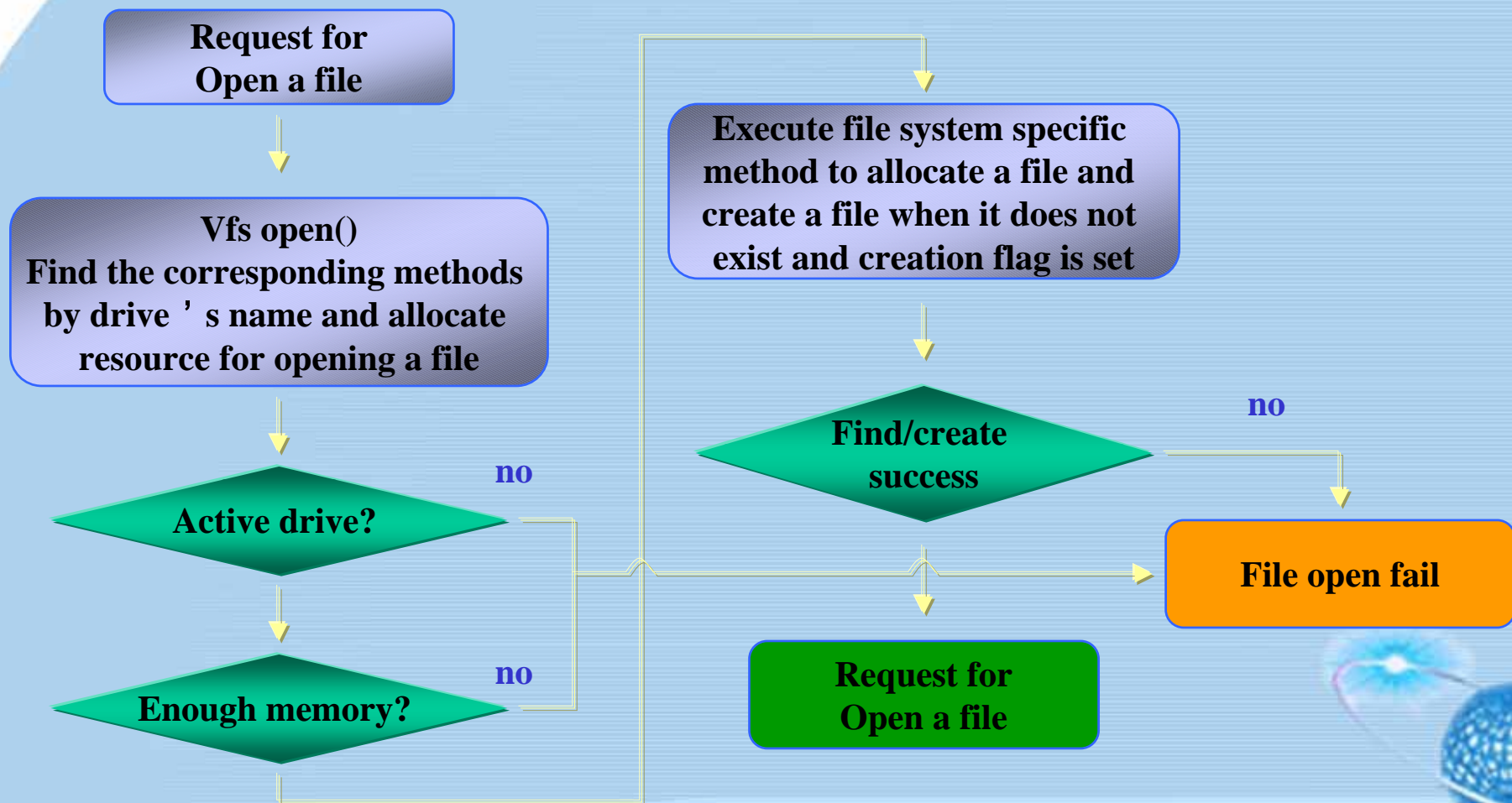




SD卡mount



File open flow





Q&A



Thank you for your time
Thank you for your time

