**Card Identification Mode**

**Card Reset**:

Command: MMC\_GO\_IDLE\_STATE **[CMD0]**

Function: mmc\_go\_idle( ); [core/mmc\_ops.c]

**Verify SD Memory Card Interface Operating Condition** :

Command: SD\_SEND\_IF\_COND **[CMD8]**

Function: mmc\_send\_if\_cond( ); [core/sd\_ops.c]

**Designed to provide SD Memory Card hosts with a mechanism to**

**identify and reject cards which do not match the VDD range desired by the host.**

APP\_CMD (CMD55) before ACM41

Command: [SD\_APP\_OP\_COND](http://lxr.free-electrons.com/ident?i=SD_APP_OP_COND) **[ACM41]**

Function: [mmc\_send\_op\_cond](http://lxr.free-electrons.com/ident?i=mmc_send_op_cond)( ); **[**core/mmc\_ops.c]

**Get CID Number**

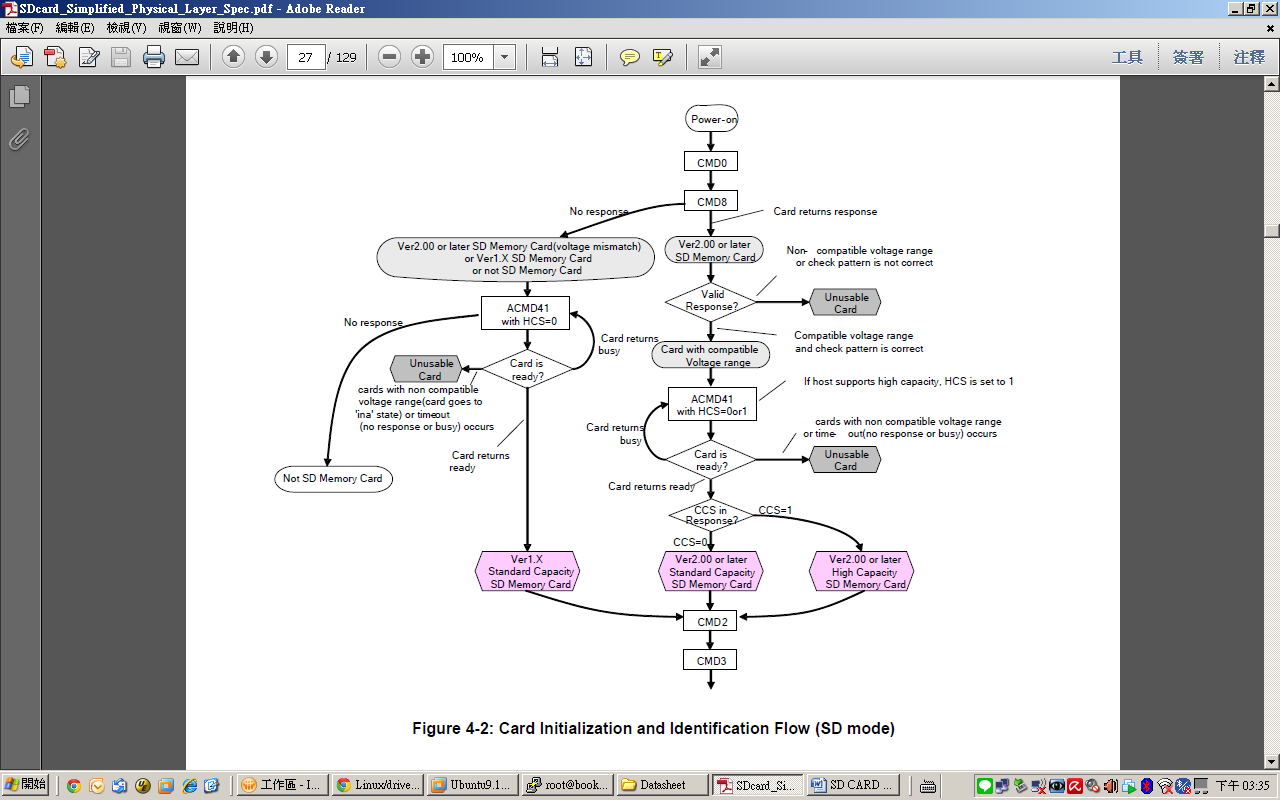
Command: MMC\_ALL\_SEND\_CID **(CMD2)**

Functions: mmc\_all\_send\_cid( ); [core/mmc\_ops.c]

**Get Relative Address**

Command: MMC\_SET\_RELATIVE\_ADDR **(CMD3)**

Functions: mmc\_set\_relative\_addr( ); [core/mmc\_ops.c]



**Data Transfer Mode**

**The host issues SEND\_CSD (CMD9) to obtain the Card Specific Data (CSD register) such as block length, card storage capacity**

Command: MMC\_SEND\_CSD [CMD9]

Function: mmc\_send\_csd( ); [core/mmc\_ops.c]

**Select one card and put it into the *Transfer State***

Command: MMC\_SELECT\_CARD [CMD7]

Function: mmc\_select\_card( ); [core/mmc\_ops.c]

Mmc\_deselect\_card( ); [core/mmc\_ops.c]

=============================================================

**Data\_Transfer\_State\_Setting:**

CMD16 : **MMC\_SET\_BLOCKLEN**

Mean : Define the block length

Function : mmc\_set\_blocklen( ); [core/core.c]

CMD32 : **SD\_ERASE\_WR\_BLK\_START**

Mean : Sets the address of the first write block to be erased.

Function : mmc\_do\_erase( ); [core/core.c]

CMD33 : **SD\_ERASE\_WR\_BLK\_END**

Mean : Sets the address of the last write block of the continuous range to be

erased.

Function : mmc\_do\_erase( ); [core/core.c]

=====================Application Command =======================

ACMD6 : **SD\_APP\_SET\_BUS\_WIDTH**

Mean : Define the data bus width

Function : mmc\_app\_set\_bus\_width( ); [core/sd\_ops.c]

ACMD23:

ACMD42:

**Read:**

CMD6 : **SD\_SWITCH**

Mean : Check switchable function and switch card function.

Function : mmc\_sd\_switch( ); [core/sd\_ops.c]

CMD17 : **MMC\_READ\_SINGLE\_BLOCK**

Mean : Single block read [read the block size of SET\_BLOCKLEN]

Function : mmc\_rw\_rq\_prep( ); [card/block.c]

CMD18 : **MMC\_READ\_MULTIPLE\_BLOCK**

Mean : Multiple Block Read [read the block size of SET\_BLOCKLEN]

Function : mmc\_rw\_rq\_prep( ); [card/block.c]

CMD30 : **MMC\_SEND\_WRITE\_PROT**

Mean : If the card provides write protection features, this command asks the

card to send the status of the write protection bits.

Function : not found yet. [mmc/mmc.h]

CMD56 : **MMC\_GEN\_CMD**

Mean : 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.

Function : not found yet. [mmc/mmc.h]

=====================Application Command =======================

ACMD13: **SD\_APP\_SD\_STATUS**

Mean : Send the SD status.

Function : mmc\_app\_sd\_status( ); [core/sd\_ops.c]

ACMD22: **SD\_APP\_SEND\_NUM\_WR\_BLKS**

Mean : Send the number of the written write blocks.

Function : mmc\_sd\_num\_wr\_blocks( ); [card/block.c]

ACMD51: **SD\_APP\_SEND\_SCR**

Mean : Read the SD Configuration Register (SCR)

Function : mmc\_app\_send\_scr( ); [core/sd\_ops.c]

**Stop:**

CMD12 : MMC\_STOP\_TRANSMISSION

Mean : Forces the card to stop transmission.

Function : send\_stop( ); [card/block.c]

=============================================================

**Write:**

CMD24 : **MMC\_WRITE\_BLOCK**

Mean : Write Single Block

Function : mmc\_blk\_rw\_rq\_prep( ); [card/block.c]

CMD25 : **MMC\_WRITE\_MULTIPLE\_BLOCK**

Mean : Write Multiple Blocks

Function : mmc\_blk\_rw\_rq\_prep( ); [card/block.c]

Mmc\_blk\_packed\_hdr\_wrq\_prep( ); [card/block.c]

CMD26 : Command Reserved

CMD27 : **MMC\_PROGRAM\_CSD**

Mean : Programming of the programmable bits of the CSD.

Function : Not found yet. [mmc/mmc.h]

CMD42 : **MMC\_LOCK\_UNLOCK**

Mean : Used to set/reset the password or lock/unlock the card.

Function : Not found yet. [mmc/mmc.h]

CMD56 : **MMC\_GEN\_CMD**

Mean : 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.

Function : not found yet. [mmc/mmc.h]

=============================================================

**Stand-by State Setting:**

CMD3 : **MMC\_SET\_RELATIVE\_ADDR**

Mean : Ask the card to publish a new relative address (RCA)

Function : mmc\_set\_relative\_addr( ); [core/mmc\_ops.c]

CMD4 : **MMC\_SET\_DSR**

Mean : Programs the DSR of all cards

Function : Not found yet. [mmc/mmc.h]

CMD9 : **MMC\_SEND\_CSD**

Mean : Addressed card sends its card-specific data (CSD) on the CMD line.

Function : mmc\_send\_csd( ); [core/mmc\_ops.c]

CMD10 : **MMC\_SEND\_CID**

Mean : Addressed card sends its card identification (CID) on CMD the line.

Function : mmc\_send\_cid( ); [core/mmc\_ops.c]

=============================================================

**Transfer State to Programming State:**

CMD28 : **MMC\_SET\_WRITE\_PROT**

Mean : If the card has write protection features, this command sets the write

protection bit of the addressed group.

Function : Not found yet. [mmc/mmc.h]

CMD29 : **MMC\_CLR\_WRITE\_PROT**

Mean : If the card provides write protection features, this command

clears the write protection bit of the addressed group.

Function : Not found yet. [mmc/mmc.h]

CMD30 : **MMC\_SEND\_WRITE\_PROT**

Mean : If the card provides write protection features, this command asks

the card to send the status of the write protection bits.

Function : Not found yet. [mmc/mmc.h]

CMD38 : **MMC\_ERASE**

Mean : Erases all previously selected write blocks.

Function : mmc\_do\_erase( ); [core/core.c]

**Note:**

Before ACMD Command must give CMD55 first.

CMD55 : **MMC\_APP\_CMD**

Mean : Indicates to the card that the next command is an application specific

command rather than a standard command

Function : mmc\_app\_cmd( ); [core/sd\_ops.c]

**Read Flow and Write Flow**

Macintosh HD:Users:RichardChen:Desktop:SD CARD       (Block Device):SD_Read_Flow:SD_Read_Flow.pdf

Data Read Flow

Macintosh HD:Users:RichardChen:Desktop:SD CARD       (Block Device):SD_Write_Flow:SD_Write_Flow.pdf

Data Write Flow