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 **[ACM41]**

Function: 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]

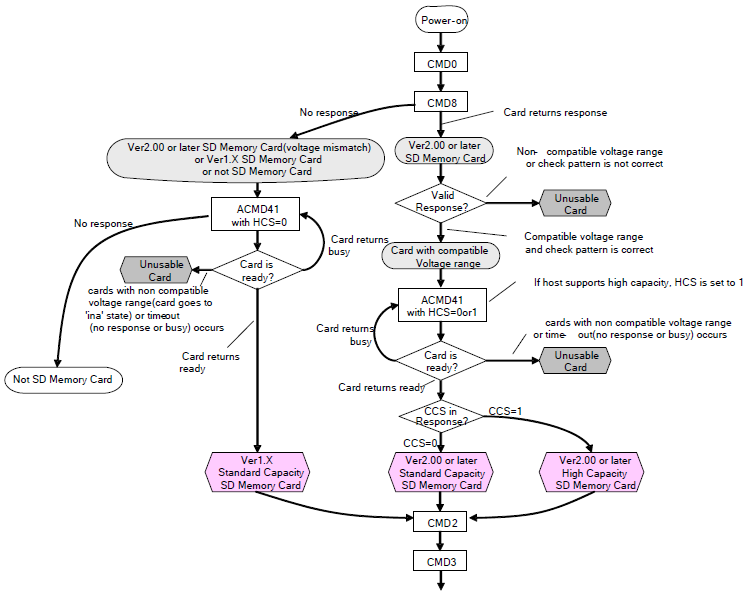


Figure 4-2: Card Initialization and Identification Flow (SD mode)

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:

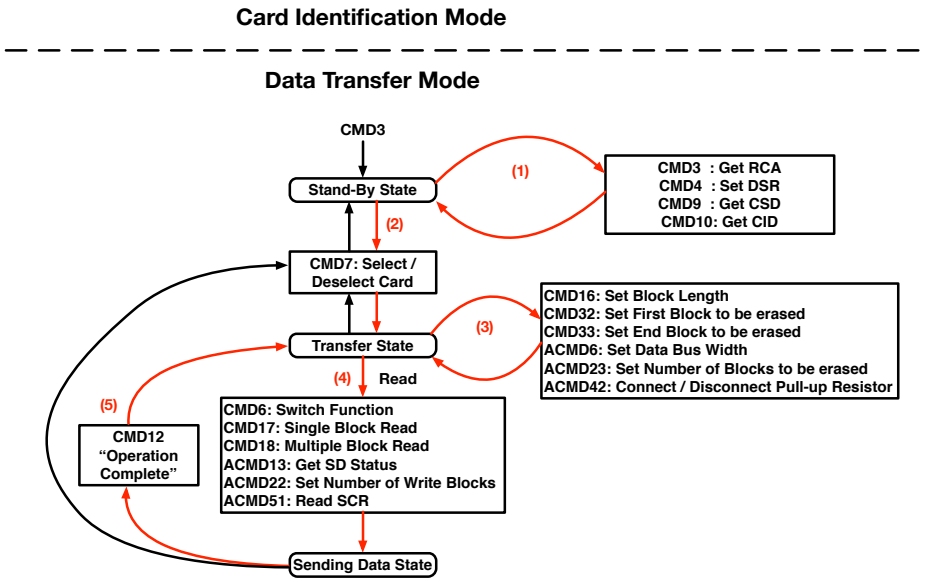
Before ACMD Command must give CMD55 first.

CMD55 : **MMC_APP_CMD**

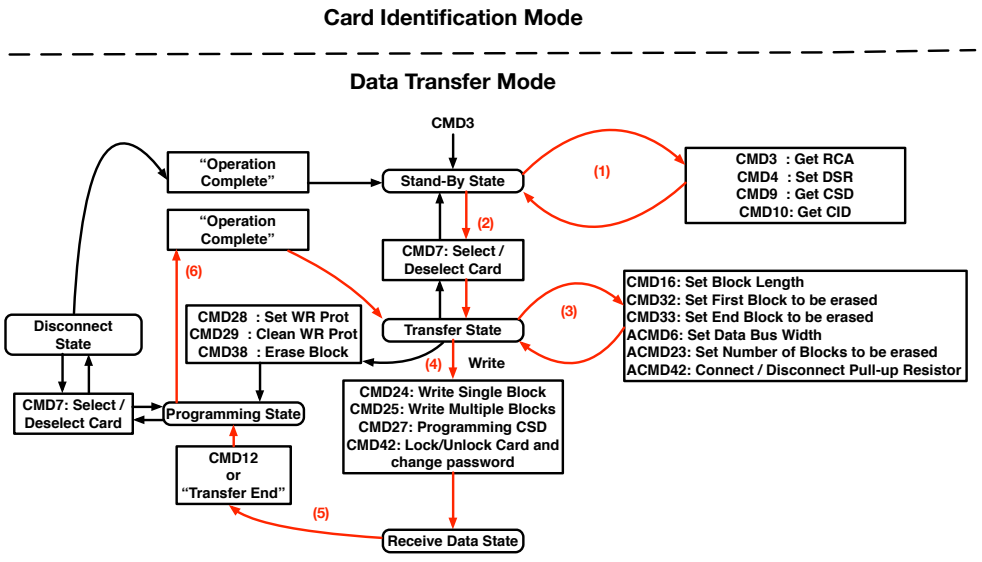
Mean : Indicates to the card that the next command is an application specific command rather than a standard command

Function : mmc_app_cmd( ); [core/sd_ops.c]

Read Flow and Write Flow



Data Read Flow



Data Write Flow