English Version
Die deutsche Version
Карта сайта

Поделиться

...
Расширенный поиск

Нашли опечатку?
Пожалуйста, сообщите об этом - просто выделите ошибочное слово или фразу и нажмите **Shift Enter**.

Блог одного Сумасшествия

Упрощенное описание стандарта физического уровня карт SD

Добавил(а) microsin

Здесь приведен перевод документа [1], описывающий принцип работы карт памяти **Security Digital** (SD). Все непонятные термины и сокращения см. в разделе "Словарик", в конце статьи. Описание режима подключения карты через **SPI** (Глава 7) перемещено в отдельный перевод [2].

[1. Общее описание]

SD Memory Card это карта памяти, которая специально разработана для удовлетворения требований по безопасности, емкости и быстродействию в применении в современных электронных аудио- и видеоустройствах. SD Memory Card может включать в себе механизм защиты контента, совместимый со стандартом безопасности SDMI. Система безопасности SD Memory Card использует взаимную аутентификацию и "новый алгоритм шифрования", чтобы защитить карту от нелегального использования её контента. Также доступен незащищенный доступ пользователя к его собственному контенту.

Карты памяти SD могут также поддерживать вторую систему безопасности, основанную на общепринятых стандартах, таких как ISO-7816, которая может использоваться для интерфейса карты SD с публичными сетями и другими системами мобильной электронной коммерции и приложениями цифровой подписи.

Карта SDIO

Обмен данными с SD Memory Card осуществляется через продвинутый 9-выводный интерфейс (такты Clock, сигнал команды Command, 4 линии данных Data и 3 линии питания Power), разработанные в расчете на максимальную рабочую частоту 50 МГц и низкое рабочее напряжение. Протокол обмена описан как часть этого стандарта.

Спецификации стандарта SD поделены на несколько отдельных документов. Структура этой документации показана на рис. 1-1.

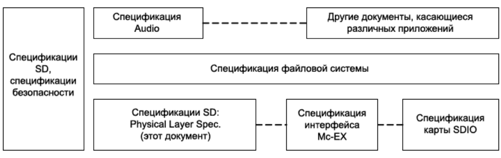

Спецификации SD, спецификации безопасности
Спецификация Audio
Другие документы, касающиеся различных приложений
Спецификация файловой системы
Спецификации SD: Physical Layer Spec. (этот документ)
Спецификация интерфейса Mc-EX
Спецификация карты SDIO

Рис. 1-1. Документация, входящая в состав SD Specification.

Audio Specification: документация, которая вместе с другими спецификациями приложений описывает применение карт SD для аудио и требования, связанные с этим приложением.
File System Specification: эта спецификация описывает структуру файлового формата данных, сохраняемых на SD Memory Card (в защищенных и не защищенных областях).

Security Specification: эта спецификация описывает механизм защиты контента, и специальных команд, специфичных для приложения, поддерживающих его.

Physical Layer Specification (спецификация физического уровня, этот документ): описание физического интерфейса и командного протокола SD Memory Card.

Назначение спецификации физического уровня - определить саму SD Memory Card, её рабочее окружение, работу с картой.

Этот документ поделен на несколько частей (глав). Глава 3 дает общий обзор системных концепций. Общие характеристики SD Memory Card описаны в главе 4. Регистры карты описаны в главе 5.

Глава 6 описывает электрические параметры аппаратного интерфейса SD Memory Card. Глава 8 описывает физические и механические свойства SD Memory Card, и минимальные рекомендации для слотов карт или картриджей.

В документе используются такие слова, как "должен" или "будет", что означает обязательные требования стандарта. "Должно быть" означает рекомендацию, что не является обязательным требованием. "Может быть" обозначает функцию, которая может как не присутствовать, так и присутствовать - т. е. это не обязательная опция, оставленная на усмотрение производителя карты, и она не касается совместимости со стандартом.

- Mc-EX Interface Specification (This section was added in version 1.10). Часть A1 спецификации на карты памяти SD (см. рис. 1-1) обслуживается как расширение спецификации физического уровня карт SD, и предоставляет все определения, требуемые для передачи пакетов команд мобильного коммерческого расширения (Mobile Commerce Extension, сокращенно Mc-EX) из хоста Mc-EX в карту SD с поддержкой Mc-EX, и наоборот.

[2. Системные функции]

- Предназначено для портативных устройств и стационарных приложений.

- Емкость памяти: карты стандартной емкости (Standard Capacity SD Memory Card) до 2 гигабайт включительно и повышенной емкости (High Capacity SD Memory Card) больше 2 гигабайт (эта версия спецификации ограничивается лимитами емкости до 32 гигабайт включительно).

- Диапазон напряжений: High Voltage SD Memory Card – рабочий диапазон 2.7-3.6 V. Dual Voltage SD Memory Card – два диапазона напряжений - низкое Low Voltage Range (TBD) и 2.7-3.6 V.

- Стандарт разработан для карт только для чтения (read-only) и карт чтения и записи (read/write).

- Режим по умолчанию: переменная частота тактов 0..25 МГц, скорость интерфейса до 12.5 мегабайт/сек (при использовании 4 параллельных линии данных.

- Высокоскоростной режим (High-Speed mode): изменяемая тактовая частота 0..50 МГц, скорость интерфейса до 25 мегабайт/сек (при использовании 4 параллельных линии данных.

- Переключение функций командой, поддерживающее High-Speed, eCommerce и другие будущие функции.

- Коррекция ошибок в области памяти.

- Извлечение карты во время операции чтения никогда не повредит данные на карте.

- Механизм защиты контента на карте, совместимый со стандартом SDMI самой высокой степени защиты.

- Защита карты паролем (CMD42 - LOCK_UNLOCK).

- Функция защиты от записи с помощью механического выключателя и соответствующей программной поддержки.

- Встроенные функции защиты от записи (постоянной и временной).

- Детектирование карты (установка/извлечение).

- Команды, специфичные для приложения.

- Удобный механизм стирания данных.

- Атрибуты протокола канала обмена данными:

Коммуникационный канал карты памяти SD
6-проводный канал обмена данными (такты, команда, 4 сигнала данных)
Защищенная от ошибок передача данных карта-хост
Передача, основанная на одном блоке данных или нескольких блоков

- Форм-фактор SD Memory Card

SD Memory Card стандартного размера описан в этом документе (см. главы 6 и 8). Форм-фактор miniSD Memory Card приведен в документе "miniSD Memory Card Specification". Форм-фактор microSD Memory Card приведен в документе "microSD Memory Card Specification".

- Толщина SD Memory Card стандартного размера составляет 2.1 мм (обычная толщина карты) и 1.4 мм (Thin SD Memory Card, тонкая карта памяти).

Все функции в этой спецификации относятся к картам SD Memory Card стандартного размера, однако карты других размеров имеют аналогичные электрические характеристики и полностью совместимы по протоколу шины (фактически отличаются только размерами и формой контактов слота).

[3. Концепция системы SD Memory]

Описание концепции не входит в упрощенную спецификацию, приведенную в этом документе.

3.1 Свойство чтения-записи (Read-Write). В контексте возможности чтения и записи определены 2 типа карт SD:

- Карты, работающие и на чтение, и на запись (Read/Write, RW). К ним относятся обычные карты (Flash), карты OTP (встречаются редко), карты MTP (также не встречал). Эти карты обычно продаются с чистым (пустым) носителем, иногда отформатированными в файловой системе FAT, и они предназначены для записи и воспроизведения видео, аудио или цифровых изображений пользователей.

- Карты только для чтения (Read Only Memory, ROM). Эти карты производятся с фиксированным содержанием, некими данными, которые уже записаны на карты. Они обычно используются для как носитель информации для программного обеспечения, аудио, видео и т. д.

3.2 Supply Voltage, напряжение питания. По рабочему напряжению питания различают 2 типа SD Memory Card:

- High Voltage SD Memory Card, которые работают в диапазоне рабочих напряжений 2.7-3.6V.

- Dual Voltage SD Memory Card, поддерживающие низкое напряжение питания (Low Voltage Range, TBD) и диапазон 2.7-3.6 V.

Обратите внимание, что параметры для последней категории, Dual Voltage SD Memory Card, пока не определены, это должно быть сделано в будущем.

3.3 Емкость карты. По объему данных SD Memory Card делятся на 2 типа:

- Карты стандартной емкости, Standard Capacity SD Memory Cards, поддерживающие размер до 2 гигабайт (231 байт). Все версии физических спецификаций определяют Standard Capacity SD Memory Card.

- Карты высокой емкости, High Capacity SD Memory Cards, поддерживающие размер свыше 2 гигабайт, и эта версия спецификации ограничена рассмотрением карт размером до 32 гигабайт включительно. High Capacity SD Memory Card были определены в новой спецификации Physical Layer Version 2.00.

Только те хосты, которые совместимы с Physical Layer Specification version 2.00 или более новую спецификацию, могут работать с картами High Capacity SD Memory. Другие хосты не смогут инициализировать High Capacity SD Memory Card (см. рис. 3-1).

Примечания:

1. Спецификации Part 1 Physical Layer Version 2.00 и Part 2 File System Version 2.00 позволяют картам стандартной емкости быть размером до 2 гигабайт включительно и картам High Capacity SD Memory Cards быть размером до 32 гигабайт включительно. Карты SD Memory емкостью свыше 32 гигабайт будут доступны в обновленных версиях спецификаций Part 1 и Part 2.

2. Хосты, которые могут получить доступ (на чтение и/или запись) к картам памяти SD с емкостью свыше 2 гигабайт и до 32 гигабайт включительно, также могут работать и с картами SD емкостью 2 гигабайта и менее.

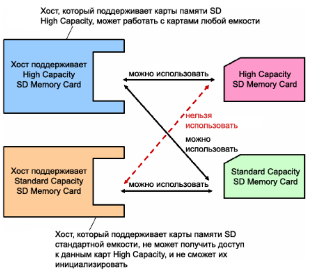

Хост, который поддерживает карты памяти SD High Capacity, может работать с картами любой емкости
Хост поддерживает High Capacity SD Memory Card
можно использовать
High Capacity SD Memory Card
нельзя использовать
можно использовать
Хост поддерживает Standard Capacity SD Memory Card
можно использовать
Standard Capacity SD Memory Card
Хост, который поддерживает карты памяти SD стандартной емкости, не может получить доступ к данным карт High Capacity, и не сможет их инициализировать

Рис. 3-1. Совместимость хостов и карт разных спецификаций.

- Определено 2 типа карт памяти высокой емкости (High Capacity SD Memory Card). Type A (Single State Card, карта с одним состоянием) имеет одну область памяти высокой емкости. Подробное описание Type A приведено в спецификации Physical Layer version 2.00. Type B (карты с двумя состояниями, Dual State Card) имеют как область памяти высокой емкости (High Capacity memory area), так и область памяти стандартной емкости (Standard Capacity memory area). На картах Type B в любой момент времени может использоваться только одна из этих областей памяти. Для выбора области используется механический переключатель. Подробно Type B будет описан в будущих спецификациях. Для хоста нет необходимости как-то дифференцировать эти два типа карт - A и B.

3.4 Класс скорости. Определены 4 класса скорости, определяющие минимальное быстродействие карт.

- Class 0 - для этого класса карт не указано быстродействие. Сюда относятся все старые карты, которые появились до разработки этой спецификации, независимо от их быстродействия.
- Class 2 - карты, скорость которых не меньше 2 мегабайт/сек.
- Class 4 - карты, скорость которых не меньше 4 мегабайт/сек.
- Class 6 - карты, скорость которых не меньше 6 мегабайта/сек.

Карты High Capacity SD Memory должны поддерживать спецификацию класса скорости не хуже, чем Class 2.

Обратите внимание, что единица производительности [мегабайт/сек] означает 1000x1000 [байт/сек], в то время как единица размера блока данных [мегабайт] означает 1024x1024 [байт]. Это потому, что максимальная скорость шины SD указывается по максимальной частоте тактов SD (25 [мегабайт/сек] = 25000000 [байт/сек] 50 МГц), и размер данных базируется на границе памяти (степени двойки).

3.5 Топология шины. *Содержимое этой секции не входит в упрощенную спецификацию.*

3.6 Протокол шины

3.6.1 Шина SD. Обмен по шине SD базируется на команде на битовых потоков команды (command) и данных (data), которые инициируются стартовым битом и завершается стоповым битом.

- Command: команда это токен, который запускает операцию. Команда отправляется от хоста либо к одной карте (адресная команда) или ко всем подключенным картам (broadcast command, широковещательная команда). Команда передается последовательно через сигнал CMD.

- Response: ответ это токен, который отправляется хосту от адресованной карты или (синхронно) от всех подключенных карт, как ответ на ранее полученную от хоста команду. Ответ передается последовательно также через сигнал CMD.

- Data: данные могут быть переданы от карты к хосту или обратно. Данные передаются через сигналы данных.

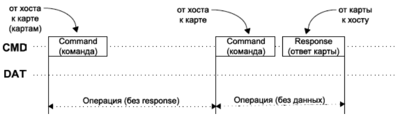


Рис. 3-2. Операции "no response" (нет ответа) и "no data" (нет данных).

Адресация карты реализована с использованием адрес сессии, назначенной карте во время фазы инициализации. Структура команд (command), ответов на команды (response) и блоков данных (data) описана в Главе 4. Базовая транзакция на шине SD состоит из процедуры команды/ответа (command/response, см. рис. 3-2). Этот тип транзакции напрямую передает свою информацию по шине в структуре команды (command) или ответа (response). Дополнительно некоторые операции включают токен данных.

Передачи данных в карту памяти SD / из карты памяти SD осуществляется в блоках. Блоки данных (Data block) всегда завершаются битами CRC. Определены одиночные (Single block) и многоблочные (multiple block) операции. Обратите внимание, что режим работы Multiple Block эффективнее для быстрой операции записи. Передачи multiple block завершаются, когда команда stop поступила на сигнале CMD. Передача данных может быть сконфигурирована хостом для использования одной или нескольких сигналов данных (1, 4 или 8).

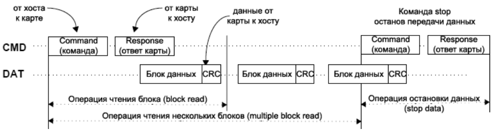

от хоста к карте
от карты к хосту
данные от карты к хосту
Команда stop останов передачи данных
CMD
DAT
Command (команда)
Response (ответ карты)
Блок данных
CRC
Command (команда)
Response (ответ карты)
Операция чтения блока (block read)
Операция чтения нескольких блоков (multiple block read)
Операция остановки данных (stop data)

Рис. 3-3. Операция чтения блока (Block Read) или нескольких блоков (Multiple Block Read).

Операция записи блока (block write) использует простую сигнализацию занятости (busy signaling) на сигнале DAT0 (см. рис. 3-4) независимо от количества используемых для передачи блока сигналов данных.

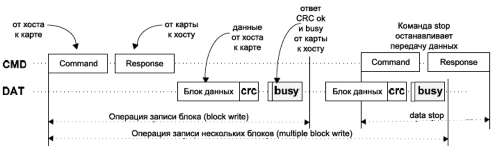

от хоста к карте
от карты к хосту
данные от хоста к карте
ответ CRC ok и busy от карты к хосту
Команда stop останавливает передачу данных
CMD
DAT
Command
Response
Блок данных
crc
busy
Операция записи блока (block write)
data stop
Операция записи нескольких блоков (multiple block write)

Рис. 3-4. Операция записи блока (Block Write) или нескольких блоков (Multiple Block Write).

Токены команд соответствуют следующей схеме:

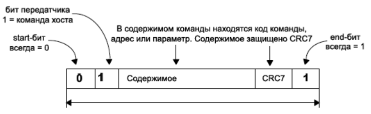

бит передатчика 1 = команда хоста
start-бит всегда = 0
В содержимом команды находятся код команды, адрес или параметр. Содержимое защищено CRC7
end-бит всегда = 1
0
1
Содержимое
CRC7
1

Рис. 3-5. Формат токена команды.

Каждому токену команды (command token) предшествует стартовый бит start bit (лог. 0), и завершается конечным битом end bit (лог. 1). Общая длина этого токена 48 бит. Каждый токен защищен битами CRC, так что можно детектировать ошибки передачи, и в случае появления ошибок операция может быть повторена.

Токены ответа (Response token) имеют одну из 4 схем кодирования, в зависимости от их содержимого. Длина токена ответа составляет либо 48, либо 136 бит. Подробное описание команд и ответов дано в главе 4.7. Алгоритм защиты CRC для блоков данных использует 16-битный полином CCITT. Разрешенные типы CRC описаны в главе 4.5.

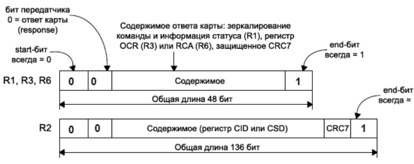

бит передатчика 0 = ответ карты (response)
start-бит всегда = 0
Содержимое ответа карты: зеркалирование команды и информация статуса (R1), регистр OCR (R3) или RCA (R6), защищенное CRC7
end-бит всегда = 1
R1, R3, R6
0
0
Содержимое
1
Общая длина 48 бит
end-бит всегда = 1
R2
0
0
Содержимое (регистр CID или CSD)
CRC7
1
Общая длина 136 бит

Рис. 3-6. Формат токена ответа (Response Token).

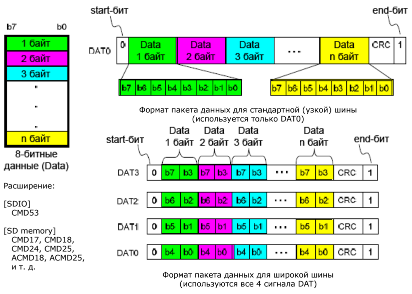
По сигналу CMD самый старший значащий бит (Most Significant Bit, MSB) передается первым, самый младший бит (Least Significant Bit, LSB) передается последним.

Когда используется опция широкой шины, данные передаются по 4 бита за такт (см. рис. 3-8). Биты start и end, как и биты CRC, передаются каждый по отдельности для каждого из сигналов DAT, т. е. биты CRC вычисляются и проверяются индивидуально для каждого сигнала DAT. CRC статуса ответа (status response) и индикации занятости (Busy indication) будут оправляться картой к хосту только через сигнал DAT0 (DAT1..DAT3 во время этого не используются).

Есть 2 типа формата пакетов данных (Data packet) для карты SD.

(1) Обычные данные (8-битной ширины): они отправляются в такой последовательности, когда младший байт передается первым, а старший байт последним. Однако имейте в виду, что каждый индивидуальный байт передается так, что старший значащий бит MSB (Most Significant Bit) передается первым, а младший бит LSB (Least Significant Bit) последним. Не перепутайте байты с битами и порядок байт и порядок бит в байте.

(2) Широкие данные (регистр карты памяти, SD Memory Register): эти данные выдвигаются, начиная со старшего значащего бита MSB.



Формат пакета данных для стандартной (узкой) шины (используется только DAT0)

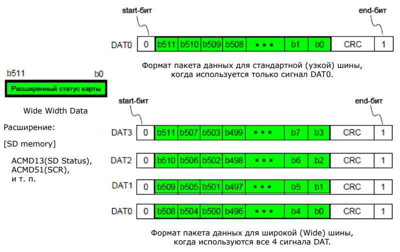
Расширение:

[SDIO]
CMD53

[SD memory]
CMD17, CMD18,
CMD24, CMD25,
ACMD18, ACMD25,
и т. д.

Формат пакета данных для широкой шины (используются все 4 сигнала DAT)

Рис. 3-7. Формат пакета для обычных данных.


start-бит
end-бит
DAT0
0
b511 b510 b509 b508
• • •
b1 b0
CRC
1
b511
b0
Расширенный статус карты
DAT3 0 b511 b507 b503 b499 • • • b7 b3 CRC 1
DAT2 0 b510 b506 b502 b498 • • • b6 b2 CRC 1
DAT1 0 b509 b505 b501 b497 • • • b5 b1 CRC 1
DAT0 0 b508 b504 b500 b496 • • • b4 b0 CRC 1

Формат пакета данных для стандартной (узкой) шины, когда используется только сигнал DAT0.

Wide Width Data

Расширение:

[SD memory]

ACMD13(SD Status),
ACMD51(SCR),
и т. п.

Формат пакета данных для широкой (Wide) шины, когда используются все 4 сигнала DAT.

Рис. 3-8. Формат пакета для широких данных (Wide Width Data).

3.6.2 Режим шины SPI. Карта может быть подключена двумя способами - традиционным (native) и в режиме SPI (до этого описывались сигналы native-подключения). Подробно протокол шины SPI описан в Главе 7 [2].

3.7 Внешние выводы и регистры карты памяти SD. *В упрощенной спецификации приведена только часть этой секции.* Также см. статью [3].

Каждая карта памяти SD имеет набор информационных регистров (см. Главу 5):

Таблица 3-1. Регистры SD Memory Card.

Имя	Бит	Описание
CID	128	Идентификационный номер карты, индивидуальный для каждой карты (см. Главу 5.2).
RCA(1)	16	Относительный адрес карты; локальный системный адрес карты, динамически предложенный картой, и подтвержденный хостом во время инициализации (см. Главу 5.4).
DSR(2)	16	Регистр стадии драйвера, применяется для конфигурирования выходных драйверов карты (см. Главу 5.5).
CSD	128	Данные, специфические для карты, информация об условиях работы карты (см. Главу 5.3).
SCR	64	Конфигурационный регистр карты, информация о специальных возможностях карты (см. Главу 5.6).
OCR	32	Регистр условий операции (см. Главу 5.1).
SSR	512	Статус SD, информация о проприетарных функциях карты (см. главу 4.10.2).
CSR	32	Статус карты (см. главу 4.10.1).

Примечания:

(1): регистр RCA не используется (не доступен) в режиме SPI (про подключение через SPI см. [2]).
(2): наличие регистра DSR опционально (не обязательно). Наличие всех остальных регистров является необязательным.

[**4. Функциональное описание SD Memory Card**]

4.1 Общая информация

Весь обмен данными между хостом и картами управляется хостом (главное устройство на шине, master). Хост посылает команды 2 типов: с широковещанием (broadcast, команда отправляется сразу всем картам) и адресно (point-to-point, т. е. команда предназначена только для одной карты).

- Команды Broadcast. Они предназначены для всех карт, подключенных к шине. Некоторые из этих команд требуют ответа (response).

- Адресованные команды (point-to-point). Они посылаются определенной карте, и приведут к ответу только одной карты. Общий обзор потока команды показан на рис. 4-1 для режима идентификации карты (card identification mode) и на рис. 4-3 для режима передачи данных (data transfer mode). Команды перечислены в таблицах команд 4-18..4-27. Зависимости между текущим состоянием, принятой командой и последующим состоянием перечислены в таблице 4-28. В последующих секциях сначала будут объяснены различные режимы работы карты. Затем будут определены ограничения для управления тактовым сигналом. Все команды карт памяти SD, вместе с их соответствующими ответами, переходами состояний, условиями ошибки и интервалами времени представлены в последующих секциях.

Для системы карт памяти SD (хост + карты) определены 2 рабочих режима:

- Card identification mode. Хост окажется в режиме идентификации карты после сброса, и пока он ищет новые карты на шине. Карты будут находиться в этом режиме после сброса, пока не будет получена команда SEND_RCA (CMD3; примечание: это не касается режима SPI [2], когда команда CMD3 не поддерживается).

- Data transfer mode. Карты войдут в режим передачи данных после публикации своего RCA (это не относится к подключению карты через SPI [2]). Хосты будут входить в режим передачи данных после идентификации всех карт на шине.

Следующая таблица показывает зависимости между рабочими режимами и состояниями карты. Каждое состояние карты памяти SD на диаграмме (см. 4-1) связано с одним из режимов работы:

Таблица 4-1. Обзор состояний карты в контексте рабочих режимов.

Состояние карты	Рабочий режим
Inactive State	Отсутствие активности карты
Idle State (состояние ожидания)	Режим идентификации карты
Ready State (состояние готовности)	
Identification State (состояние идентификации)	
Stand-by State (отключенное состояние)	Режим передачи данных (data transfer mode)
Transfer State (состояние передачи)	
Sending-data State (состояние отправки данных)	
Receive-data State (состояние приема данных)	
Programming State (состояние программирования)	
Disconnect State (отключенное состояние)	

4.2 Режим идентификации карты (Card Identification Mode). Когда активен режим идентификации карты, хост сбрасывает все карты, которые находятся в этом режиме, определяет, допустим ли рабочий диапазон напряжений, идентифицирует карты и запрашивает их опубликовать свой относительный адрес карты (Relative Card Address, RCA; последнее не относится к подключению карты через SPI [2]). Эта операция выполняется отдельно для каждой карты, по её отдельном сигналу CMD. Весь обмен данными в режиме идентификации карты осуществляется только через сигнал команд (CMD).

Во время процесса идентификации карты она должна работать с тактовой частотой identification clock rate fOD (см. главу 6.7).

4.2.1 Сброс карты (Card Reset). Команда GO_IDLE_STATE (CMD0) является командой сброса, и она переводит каждую карту в режим ожидания (Idle State) независимо от текущего состояния карты. На карты в неактивном состоянии (Inactive State) команда CMD0 никак не влияет. После включения хоста все карты находятся в состоянии ожидания (Idle State), включая случаи, когда карты ранее находились в неактивном состоянии.

После включения питания или после CMD0 все сигналы CMD карт находятся в режиме входа, ожидая появления стартового бита последующей за ним команды. Карты инициализируются с относительным адресом по умолчанию (RCA=0x0000) и с установками по умолчанию регистра стадии драйвера (низкая скорость, самая большое энергопотребление).

4.2.2 Проверка соответствия рабочих условий. В начале обмена между хостом и картой хост может не знать, поддерживает ли карта имеющееся напряжение и карта может не знать, может ли она поддерживать текущее приложенное напряжение. Хост выдает команду сброса (CMD0) с указанным напряжением, подразумевая, что оно возможно поддерживается картой. Чтобы проверить напряжение, в спецификации Physical Layer Version 2.00 определена следующая новая команда (CMD8).

SEND_IF_COND (CMD8) используется для проверки рабочих условий интерфейса SD Memory Card. Карта проверяет допустимость рабочих условий путем анализа аргумента команды CMD8, и хост проверяет допустимость рабочих условий анализом ответа (response) на команду CMD8 (см. главу 4.3.13). Предоставленное напряжение показано полем VHS в аргументе этой команды. Карта подразумевает, что напряжение, указанное полем VHS, соответствует текущему предоставленному напряжению. Только 2 бит поля VHS должен быть установлен в лог. 2 в любой момент времени. И CRC, и маска проверки (check pattern) используется хостом для проверки правильной коммуникации между хостом и картой.

Если карта может работать на предоставленном напряжении, то ответ (response) эхом отправляет назад напряжение питания и маску проверки, которые были установлены в аргументе команды.

Если карта не может работать на предоставленном напряжении, она не отправит обратно response, и будет оставаться в состоянии ожидания (idle state). Важно выдать CMD8 перед первой командой ACMD41 для инициализации High Capacity SD Memory Card (см. рис. 4-1). Прием команды CMD8 говорит карте, что хост поддерживает Physical Layer Version 2.00, и карта может разрешить в себе новые функции этого стандарта.

Также для низковольтного хоста обязательно отправить CMD8 перед ACMD41. В случае, когда Dual Voltage Card не приняла CMD8, карта будет работать только на высоком напряжении, и в этом случае низковольтный хост не отправит CMD8, карта перейдет в неактивное состояние при ACMD41.

Команда SD_SEND_OP_COND (ACMD41) разработана для предоставления хоста карт памяти SD механизма для идентификации и отбраковки карт, которые не соответствуют напряжению питания VDD, которое хочет использовать хост. Это выполнено хостом отправкой требуемого окна напряжения VDD как операнда этой команды (см. главу 5.1). Карты, которые не могут выполнить передачу данных в указанном диапазоне, должны отказаться от дальнейшей работы по шине, и оставаться в неактивном состоянии (Inactive State). Эти уровни в регистре OCR должны быть соответствующим образом определены (см. главу 5.1). Обратите внимание, что ACMD41 является командой, специфичной для приложения, так что APP_CMD (CMD55) должна всегда предшествовать команде ACMD41. RCA для использования с CMD55 в idle_state будет иметь адрес карты по умолчанию RCA = 0x0000.

После того, как хост выдал команду сброса (CMD0), чтобы сбросить карту, хост должен выдать CMD8 перед ACMD41, чтобы переинициализировать карту памяти SD.

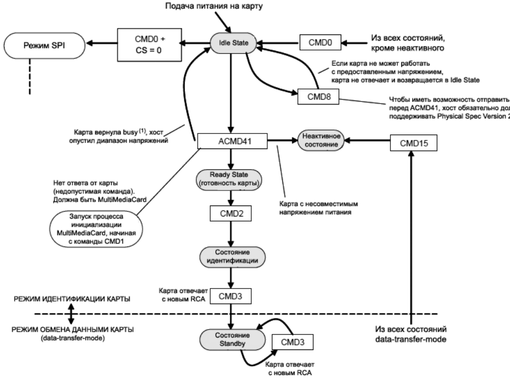

Подача питания на карту
Режим SPI
CMD0 +
CS = 0
Idle State
CMD0
Из всех состояний, кроме неактивного
Если карта не может работать с предоставленным напряжением, карта не отвечает и возвращается в Idle State
CMD8
Чтобы иметь возможность отправить CMD8 перед ACMD41, хост обязательно должен поддерживать Physical Spec Version 2.00
Карта вернула busy (1), хост опустил диапазон напряжений
ACMD41
Неактивное состояние
CMD15
Ready State (готовность карты)
Нет ответа от карты (недопустимая команда). Должна быть MultiMediaCard
Запуск процесса инициализации MultiMediaCard, начиная с команды CMD1
CMD2
Карта с несовместимым напряжением питания
Состояние идентификации
Карта отвечает с новым RCA
CMD3
РЕЖИМ ИДЕНТИФИКАЦИИ КАРТЫ
РЕЖИМ ОБМЕНА ДАННЫМИ КАРТЫ (data-transfer-mode)
Состояние Standby
CMD3
Карта отвечает с новым RCA
Из всех состояний data-transfer-mode

Рис. 4-1. Диаграмма перехода между состояниями карты памяти SD (card identification mode).

Путем установки OCR = 0 в аргументе ACMD41, хост может опросить каждую карту и определить общий диапазон напряжений перед отправкой в неактивное состояние Inactive State (query mode) карт, которые не могут работать в нужном диапазоне напряжений. Этот опрос должен использоваться, если хост может выбрать общий диапазон напряжений, или если нужно оповещение приложения о тех картах, которые нельзя использовать. Карта не запускает инициализацию, если ACMD41 была выдана как запрос. Впоследствии узел может выбрать напряжение для работы и повторно выдать ACMD41 с этими условиями, отправив несовместимые карты в неактивное состояние (Inactive State).

Во время процедуры инициализации хост не может поменять диапазон рабочего напряжения. См. последовательность включения (power up sequence), как она описана в главе 6.4.

4.2.3 Процесс инициализации и идентификации карты. После того, как шина активирована хостом, хост запускает процесс инициализации и идентификации (см. рис. 4-2). Этот процесс начинается с SD_SEND_OP_COND (ACMD41) путем установки рабочих условий и бита HCS в OCR. Этот бит HCS (аббревиатура Host Capacity Support, поддержка хостом емкости) устанавливается в 1, что означает, что хост поддерживает карты высокой емкости (High Capacity SD Memory card). Бит HCS (Host Capacity Support) устанавливается в 0, что означает отсутствие поддержки хостом карт High Capacity SD Memory.

Прием CMD8 расширяет функцию ACMD41; HCS в аргументе и CCS (Card Capacity Status, статус емкости карты) передается в ответе (response). HCS игнорируется картами, которые не отвечают на CMD8. Однако хост должен установить HCS в 0, если карта не вернет ответ на команду CMD8. Карта стандартной емкости (Standard Capacity SD Memory Card) игнорирует HCS. Если HCS установлен в 0, карта памяти High Capacity SD никогда не вернет состояние готовности (будет удерживать бит занятости busy в лог. 0). Этот бит busy в OCR используется картой, чтобы информировать хост, что завершена инициализация ACMD41. Установка бита busy в 0 показывает, что карта все еще инициализируется. Установка бита busy в 1 показывает завершение инициализации. Хост постоянно повторяет выдачу ACMD41, пока бит busy не будет установлен в 1. Карта проверяет условия функционирования и бит HCS в регистре OCR только с первой ACMD41. При повторах ACMD41 хост не должен выдавать другие команды, кроме CMD0.

Если карта ответила на CMD8, то ответ на ACMD41 включает информацию поля CCS. CCS достоверно, когда карта вернет ready (готовность, бит busy установлен в 1). CCS=1 означает, что карта повышенной емкости (High Capacity SD Memory Card), CCS=0 означает, что карта стандартной емкости (Standard Capacity SD Memory Card).

Хост выполняет ту же самую последовательность инициализации для всех новых карт в системе. Несовместимый карты отправляются в неактивное состояние (Inactive State). Затем хост выдает команду ALL_SEND_CID (CMD2) каждой карте, чтобы получить их уникальный идентификационный номер (card identification, CID). Не идентифицированная карта (которая находится в состоянии готовности, Ready State) отправит свой номер CID как ответ (response через сигнал CMD). После того, как CID был отправлен картой картой, она переходит в состояние идентификации (Identification State). Впоследствии хост выдает CMD3 (SEND_RELATIVE_ADDR), запрашивая карту опубликовать новый относительный адрес (relative card address, RCA), который короче, чем CID, и который используется для адресации карты в будущем режиме передачи данных (data transfer mode). Как только RCA был принят, состояние карты меняется на "приостановленное" (Stand-by State). В этом месте, если хост желает назначить другой номер RCA, он может запросить карту опубликовать новый номер отправкой карте другой команды CMD3. Последний опубликованный RCA будет актуальным номером RCA для карты.

Хост повторяет процесс идентификации, т. е. выдает команды CMD2 и CMD3 для каждой карты в системе.

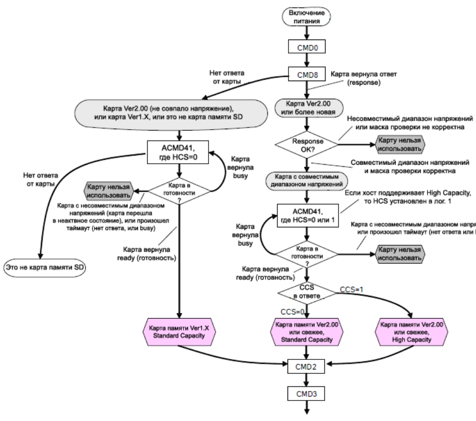

Включение питания
CMD0
CMD8
Нет ответа от карты
Карта вернула ответ (response)
Карта Ver2.00 (не совпало напряжение), или карта Ver1.X, или это не карта памяти SD
Карта Ver2.00 или более новая
Несовместимый диапазон напряжений или маска проверки не корректна
Response OK?
Карту нельзя использовать
Совместимый диапазон напряжений и маска проверки корректна
ACMD41, где HCS=0
Карта вернула busy
Нет ответа от карты
Карту нельзя использовать
Карта в готовности ?
Карта с несовместимым диапазоном напряжений (карта перешла в неактвное состояние), или произошел таймаут (нет ответа, или busy)
Карта вернула ready (готовность)
Это не карта памяти SD
Карта с совместимым диапазоном напряжений
Если хост поддерживает High Capacity, то HCS установлен в лог. 1
ACMD41, где HCS=0 или 1
Карта вернула busy
Карта с несовместимым диапазоном напряжений, или произошел таймаут (нет ответа или busy)
Карта в готовности ?
Карту нельзя использовать
Карта вернула ready (готовность)
CCS в ответе
CCS=1
CCS=0
Карта памяти Ver1.X Standard Capacity
Карта памяти Ver2.00 или свежее, Standard Capacity
Карта памяти Ver2.00 или свежее, High Capacity
CMD2
CMD3

Рис. 4-2. Поток команд и ответов инициализации и идентификации карты (SD mode).

4.3 Режим передачи данных (Data Transfer Mode). До окончания Card Identification Mode хост должен удерживать низкую частоту синхронизации fOD, потому что некоторые карты могут иметь ограничения рабочей частоты в режиме идентификации карты. В режиме передачи данных (Data Transfer Mode) хост может работать с картой в диапазоне частот fPP (см. главу 6.7). Хост выдает команду SEND_CSD (CMD9), чтобы получить специфические данные карты (Card Specific Data, регистр CSD), например длину блока, емкость карты, и т. д. Широковещательная (broadcast-команда) SET_DSR (CMD4) конфигурирует стадии драйвера для всех идентифицированных карт. Она программирует их регистры DSR соответственно разводке шины приложения (длина сигнальных линий, что влияет на емкость и индуктивность сигналов шины), количестве карт на шине (это также влияет на общую емкость сигнальных проводников) и частоте передачи данных. В этой точке также переключается тактовая частота от fOD на fPP. Команда SET_DSR является опцией (т. е. не обязательна) для карты и хоста.

CMD7 используется для выбора одной карты и перевода её в состояние передачи (Transfer State). В любой момент времени только одна карта может быть в Transfer State. Если ранее выбранная карта находится в Transfer State, её соединение с хостом освобождается, и она перейдет обратно в "приостановленное" состояние (Stand-by State). Когда выдается команда CMD7 с зарезервированным относительным (RCA) адресом 0x0000, все карты перейдут в Stand-by State (обратите внимание, что в зоне ответственности хоста резервировать RCA=0 для отмены выбора карты - см. CMD7 в таблице 4-18).

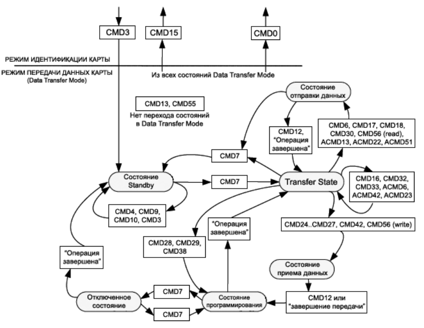

CMD3
CMD15
CMD0
РЕЖИМ ИДЕНТИФИКАЦИИ КАРТЫ
РЕЖИМ ПЕРЕДАЧИ ДАННЫХ КАРТЫ (Data Transfer Mode)
Из всех состояний Data Transfer Mode
Состояние отправки данных
CMD13, CMD55
Нет перехода состояний в Data Transfer Mode
CMD12, "Операция завершена"
CMD6, CMD17, CMD18, CMD30, CMD56 (read), ACMD13, ACMD22, ACMD51
CMD7
Состояние Standby
CMD7
Transfer State
CMD16, CMD32, CMD33, ACMD6, ACMD42, ACMD23
CMD4, CMD9, CMD10, CMD3
"Операция завершена"
CMD24..CMD27, CMD42, CMD56 (write)
CMD28, CMD29, CMD38
"Операция завершена"
Состояние приема данных
CMD7
Отключенное состояние
Состояние программирования
CMD12 или "завершение передачи"
CMD7

Рис. 4-3. Диаграмма состояний карты памяти SD в режиме передачи данных (data transfer mode).

Это может использоваться перед идентификацией новых карт без сброса уже зарегистрированных карт. Карты, которые уже имеют RCA, в этом состоянии не отвечают на команды идентификации (ACMD41, CMD2, см. главу 4.2.3).

Важное замечание: отмена выбора карты (de-selection) осуществляется, если определенная карта получила CMD7 с не совпадающим RCA. Это происходит автоматически, если сделан выбор другой карты, и сигналы CMD общие. Таким образом, в системе SD Memory Card в зоне ответственности хоста находится принцип работы - либо работать с картами через общий сигнал CMD (после того, как была осуществлена инициализация) - в этом случае отмена выбора карты будет происходить автоматически, либо если сигналы CMD отдельные для каждой карты, то хост должен знать о необходимости отменять выбор карт.

Все обмены данными в режиме Data Transfer Mode осуществляется по принципу точка-точка (point-to-point) между хостом и выбранной картой (с использованием адресованных команд). Все адресованные команды подтверждаются ответом (response) через тот же сигнал CMD, через который передаются команды.

Ниже обобщены взаимосвязи между различными режимами передачи данных.

- Все команды чтения данных могут быть оборваны в любой момент командой stop (CMD12). Передача данных будет завершена картой, и карта вернется в состояние передачи (Transfer State). Команды чтения следующие: блочное чтение (block read, CMD17), множественное блочное чтение (multiple block read, CMD18), послать защиту от записи (write protect, CMD30), послать scr (ACMD51) и общая команда режима чтения (CMD56).

- Все команды записи могут быть оборваны в любой момент командой stop (CMD12). Эта команда записи должна быть остановлена перед отменой выбора карты командой CMD7. Команды записи следующие: блочная запись (block write, CMD24 и CMD25), программирование CSD (CMD27), команда блокировки/разблокировки (lock/unlock, CMD42) и общая команда режима записи (CMD56).

- Как только передача данных была завершена, карта выйдет из состояния записи и перейдет либо в состояние программирования Programming State (при успешной передаче) или в состояние передачи Transfer State (если передача была неудачной).

- Если операция блочной записи остановлена, и длина блока и CRC последнего блока достоверны, то данные будут запрограммированы.

- Карта может предоставить буферизацию для блочной записи. Это означает, что следующий блок может быть отправлен во время, когда карта осуществляет программирование предыдущего блока. Если все буферы записи заполнены, и пока карта в состоянии программирования Programming State (см. диаграмму состояний карты памяти SD на рис. 4-3), сигнал DAT0 будет удерживаться в лог. 0 (индикация занятости BUSY).

- Отсутствует опция буферизации для записи CSD, операции защиты от записи (write protection) и стирания (erase). Это означает, что пока карта занята обработкой одной из этих команд, не могут быть приняты никакие другие команды манипуляции с данными. Сигнал DAT0 будет удерживаться в лог. 0, пока карта занята и находится в Programming State. В действительности если сигналы CMD и DAT0 сделаны отдельными, и хост может удерживать DAT0 в состоянии занятости отключенным от DAT0 других карт, то хост может обращаться к другим картам, пока одна из карт занята.

- Не разрешается использовать команды установки параметра, пока карта в состоянии программирования. Команды установки параметра следующие: установка длины блока (CMD16), команда установки начала стираемого блока (CMD32) и команда установки конца стираемого блока (CMD33).

- Не разрешены команды чтения, пока карта находится в состоянии программирования.

- Переход другой карты из Stand-by в Transfer State (с использования CMD7) не завершит операции стирания и программирования. Карта переключится в Disconnect State, и освободит линию DAT.

- Карта может быть выбрана повторно с использованием CMD7, когда она находится в Disconnect State. В этом случае карта перейдет в Programming State и заново активирует индикацию занятости.

- Сброс карты (командами CMD0 или CMD15) прервет любую ожидающую или активную операцию программирования. Это может уничтожить содержимое (полезные данные) на карте. Обязанность хоста - предотвратить это.

- CMD34-37, CMD50 и CMD57 зарезервированы для расширения системы команд SD. Переходы состояния для этих команд определены в спецификации каждой системы команд.

4.3.1 Выбор широкой шины и отмена её выбора. Режим работы широкой шины (когда под данные отведено 4 бита) может быть выбран или отменен командой ACMD6. По умолчанию, после включения питания или команды GO_IDLE (CMD0) устанавливается ширина шины в 1 бит (узкая шина).

Чтобы можно было поменять ширину шины, должны выполняться 2 условия:

a) Карта в состоянии 'tran state'.
b) Карта не заблокирована.

Заблокированная карта будет отвечать на ACMD6 как на недопустимую команду.

4.3.2 Карта на 2 гигабайта. Чтобы сделать карту 2GByte, максимальная длина блока (Maximum Block Length, READ_BL_LEN=WRITE_BL_LEN) должна быть установлена в 1024 байта. Однако длина блока Block Length, устанавливаемая командой CMD16, должна быть до 512 байт, чтобы сохранить соответствие с картами, у которых Maximum Block Length равна 512 байтам (карты, которые меньше и равны картам 2GByte).

4.3.3 Data Read (чтение данных). Когда данные не передаются, уровень шины DAT будет в лог. 1 с помощью резисторов pull-up. Переданный блок данных состоит из стартовых бит (1 или 4 бита в лог. 0), за которыми идет продолжающийся поток данных. Поток данных содержит данные полезной нагрузки (и биты коррекции ошибки, если используется ECC вне карты). Поток данных завершается конечными битами (1 или 4 бита в лог. 1). Передача данных синхронна с тактовым сигналом. Полезная нагрузка для передач блоковых данных защищается 1- или 4-битной контрольной суммой CRC (см. главу 3.6).

Операция чтения (Read) из карты SD Memory может быть прервана выключением питания. Карта SD Memory гарантирует, что данные не будут уничтожены в любых условиях, за исключением операций записи или стирания, запущенными хостом, даже с случае неожиданного выключения или извлечения карты.

Команда чтения отбрасывается, если произошла ошибка BLOCK_LEN_ERROR или ADDRESS_ERROR, и передача данных не будет выполнена.

Block Read. Блочное чтение это передача данных, основанная на блоке. Базовая единица передачи данных это блок, к которого максимальный размер всегда 512 байт. Могут передаваться блоки меньшего размера, которые начинаются и заканчиваются полностью внутри границ одного блока 512 байт.

Длина блока Block Length устанавливается командой CMD16 может быть установлена размером до 512 байт, независимо от READ_BL_LEN.

CRC добавляется в конец каждого блока, что гарантирует целостность данных. CMD17 (READ_SINGLE_BLOCK) инициирует блочное чтение и после завершения передачи карта вернется в состояние передачи (Transfer State). CMD18 (READ_MULTIPLE_BLOCK) начинается с передачи нескольких следующих друг за другом блоков. Блоки будут непрерывно передаваться, пока не будет выдана команда STOP_TRANSMISSION (CMD12). Команда stop имеет задержку выполнения в несколько последовательных передач. Передача данных остановится после конечного бита (end bit) команды stop.

Если хост использует частичные блоки, накопленная длина которых оказывается не выровненной на размер блока, и не разрешено использование не выровненных блоков, то карта должна детектировать не выровненные блоки на начале не выровненного блока, устанавливая бит ADDRESS_ERROR в регистре статуса, обрывая передачу, ожидая команды stop в состоянии Data State.

Таблица 4-2 определяет поведение карты, когда разрешены доступы с частичными блоками. Если не выровненный блок это первый блок данных команды (например был сообщен бит ADDRESS_ERROR в реальном ответе на команду), то данные не будут переданы и карта останется в состоянии TRAN.

Таблица 4-2. Длина блоков команды чтения.

Значение CSD			Current Blocklen(3)	Начальный адрес команды чтения
Макс. размер блока READ_BL_LEN	Misalign(1)	Partial(2)		
512 байт	0 (запрещено)	1 (разрешено)	1..512 байт	Допустим любой адрес(4)
1 килобайт	0 (запрещено)	1 (разрешено)	1..512 байт	Допустим любой адрес(4)
2 килобайта	0 (запрещено)	1 (разрешено)	1..512 байт	Допустим любой адрес(4)

Примечания к таблице 4-2:

(1) "Misalign" означает "нет выравнивания по границе блока".
(2) "Partial" означает "частичное чтение блока".
(3) Размер "Current Blocklen" (текущий размер блока) устанавливается или меняется командой CMD16. Если значение меньше или равно 512 байтам (нет взаимосвязи между опцией Misalign и Partial), то это будет установлено без ошибки.
(4) Когда размер данных Blocklen пересекает границу 512 байт, карта выводит данные по границы блока в 512 байт, и затем данные становятся недостоверными и также может возникнуть ошибка CRC. Карта отправит ADDRESS_ERROR в следующем ответе на команду. Для восстановления хост должен выдать CMD12.

Block Write. Во время записи блока (CMD24..CMD27, CMD42, CMD56(w)) один или большее количество блоков данных передается от хоста в карту с 1- или 4-битной CRC, добавляемой хостом в конец каждого блока. Карте, поддерживающей блочную запись, требуется установить длину блока Block Length командой CMD16 в 512 байт, независимо от того, как установлена WRITE_BL_LEN, в 1k или 2k байт.

Таблица 4-3 определяет поведение карты, когда запрещен доступ к части блока (WRITE_BL_PARTIAL = 0).

Таблица 4-3. Длина блоков команды записи.

Значение CSD			Current Blocklen(3)	Начальный адрес команды записи
Макс. размер блока WRITE_BL_LEN	Misalign(1)	Partial(2)		
512 байт	0 (запрещено)	1 (разрешено)	512 байт(4)	n*512 байт(5) (n целое число)
1 килобайт	0 (запрещено)	1 (разрешено)	512 байт(4)	n*512 байт(5) (n целое число)
2 килобайта	0 (запрещено)	1 (разрешено)	512 байт(4)	n*512 байт(5) (n целое число)

Примечания к таблице 4-3:

(1) "Misalign" означает "нет выравнивания по границе блока".
(2) "Partial" означает "частичная запись блока".
(3) Размер "Current Blocklen" (текущий размер блока) устанавливается или меняется командой CMD16. Если значение меньше или равно 512 байтам (нет взаимосвязи между опцией Misalign и Partial), то это будет установлено без ошибки. И тогда размер "Current Blocklen" тестируется при при выполнении команды записи.
(4) Если текущая Blocklen отличается от этой величины, то карта покажет BLOCK_LEN_ERROR в ответ на команду записи.
(5) Если начальный адрес отличается от этого значения, карта отправит ADDRESS_ERROR в ответ на команду записи.

Если WRITE_BL_PARTIAL разрешено (=1), то блоки меньшего размера, вплоть до 1 байта, могут использоваться при записи. Если CRC покажет ошибку, карта сообщит он ней по сигналу DAT (см. ниже); переданные данные будут отброшены и не записаны, и все будущие передаваемые блоки (в режиме multiple block write) будут проигнорированы.

Чтобы операции записи работали быстрее, вместо следующих друг за другом команд single block write должна использоваться команда multiple block write.

Если хост использует частичные блоки, которые в накопленной длине не выровнены по размеру блока, и блок не выровненный, произойдет ошибка выравнивания и прекращение программирования перед началом первого не выровненного блока. Карта должна установить бит бит ошибки ADDRESS_ERROR в регистре статуса, и будет игнорирована запись всех будущих передачи данных, ожидая Receive-data-State для команды stop.

Обратите внимание, что первый блок данных не выровнен для команды записи (например, сообщается ADDRESS_ERROR в действительном ответе на команду write), карта остается в состоянии tran, и данные не будут запрограммированы. Операция записи должна быть также оборвана, если хост пытается записать данные поверх области, защищенной от записи. Однако в этом случае карта должна установить бит WP_VIOLATION.

Программирование регистра CSD не требует установки длины предыдущего блока. Передаваемые данные также защищены CRC. Если часть регистра CSD сохранена в ROM, то недоступная часть должна соответствовать определенной части буфера приема. Если такого совпадения не произошло, то карта сообщит об ошибке и не поменяет никакие данные в регистре.

Некоторые карты могут требовать большого и не предсказуемого времени для записи блока данных. После приема блока данных и завершения проверки CRC, карта начнет запись и удерживать сигнал DAT0 в лог. 0, если её буфер записи будет заполнен, и новые данные не будут доступны от новой команды WRITE_BLOCK. Хост может опрашивать состояние карты командой SEND_STATUS (CMD13) в любой момент времени, и карта ответит информацией о своем статусе. Бит статуса READY_FOR_DATA показывает, готова ли карта принять новые данные, или процесс записи все еще продолжается. Хост может снять выбор карты выдачей команды CMD7 (чтобы выбрать другую карту) что поместит эту карту в отключенное состояние (Disconnect State), и освободит линию DAT без обрыва операции записи. Когда карта будет выбрана снова, будет повторно активирована индикация занятости переводом DAT в лог. 0, если программирование все еще продолжается, и буфер записи пока недоступен. В действительности хост может выполнить одновременно операцию записи в несколько карт, если проводить запись с чередованием между картами. Чередование может быть осуществлено путем доступа к каждой карте по отдельности - пока другие карты находятся в занятом состоянии, манипуляции происходят со свободными картами. Этот процесс может быть осуществлен правильной манипуляцией CMD и линией DAT0-DAT3 (отключение занятых карт).

Установка предварительной очистки перед операцией Multiple Block Write. Установка количества записываемых блоков, чтобы они были предварительно очищены (pre-erased, команда ACMD23) приведут к тому, что следующая операция Multiple Block Write произойдет быстрее, чем та же самая операция без предшествующей команды ACMD23. Хост будет использовать эту команду для определения, какое количество записываемых блоков будет отправлено в последующей операции записи (с использованием останова передачи, stop transmission). Перед тем, как блоки данных отправятся в карту содержимое остальных записываемых блоков не определено (оно может быть либо стерто, либо содержать старые данные). Если хост пошлет больше блоков, чем было определено в команде ACMD23, то карта будет очищать блоки по одному (по мере поступления новых данных). Это количество будет сброшено в значение по умолчанию (=1) после операции Multiple Blocks Write.

Рекомендуется использовать эту команду перед CMD25, тогда некоторые карты будут выполнять операцию Multiple Write Blocks быстрее. Обратите внимание, что хост должен отправить ACMD23 непосредственно перед командой WRITE, если хост хочет использовать функцию предварительной очистки (pre-erased feature). Если это не так, то счетчик pre-erase может быть автоматически очищен, если поступили и выполнены другие команды (например Security Application Command).

Отправка количества записываемых блоков. Системы, использующие механизм конвейера для управления буферами данных, в некоторых случаях не могут определить, какой блок был последним для успешной записи в память FLASH, если произошла ошибка посередине операции Multiple Blocks Write. Карта ответит на команду ACMD22 количеством успешно записанных блоков.

4.3.5 Erase (очистка). Желательно очистить несколько записываемых блоков одновременно, чтобы повысить скорость передачи данных. Идентификация этих стираемых блоков осуществляется командами ERASE_WR_BLK_START (CMD32), ERASE_WR_BLK_END (CMD33). Хост должен придерживаться следующей последовательности команд: ERASE_WR_BLK_START, ERASE_WR_BLK_END и наконец ERASE (CMD38).

Команда очистки (erase, CMD38) или установки адреса (CMD32, CMD33) была принята вне указанной последовательности, то карта установит бит ERASE_SEQ_ERROR в регистре статуса, и сбросит всю эту последовательность команд. Если принята команда вне последовательности (за исключением SEND_STATUS), карта установит бит статуса ERASE_RESET в регистре статуса, сбросит последовательность очистки и выполнит последнюю команду.

Если диапазон очистки включает в себя сектора с защитой от записи, то эти сектора останутся нетронутыми, и стирание коснется только незащищенных секторов. В регистре статуса установится бит WP_ERASE_SKIP.

Поле адреса в команде установки адреса это адрес записываемого блока, указанный в байтах. Карта будет игнорировать все LSB ниже размера WRITE_BL_LEN (см. CSD).

Как было описано ранее для записи блока, карта покажет, что идет процесс очистки, путем удерживания в лог. 0 линии DAT0. Реальная очистка может быть довольно долгой, и хост может выдавать команду CMD7 для отмены выборки карты или выполнить отсоединение от карты, как было описано выше в секции с описанием блочной записи (Block Write).

Данные на карте после операции очистки будут нулями или единицами, в зависимости от производителя карты. Регистр SCR, бит DATA_STAT_AFTER_ERASE (номер этого бита 55) определит, какие это будут данные после очистки - либо 0, либо 1.

4.3.6 Write Protect Management (управление защитой от записи). Картами памяти SD поддерживаются 3 метода защиты от записи:

- Механический переключатель на карте, показывающий состояние защиты - включена она или нет (поддержка защиты находится только в зоне ответственности хоста).
- Внутренняя защита от записи карты (защита данных находится только в зоне ответственности карты).
- Функционирование блокировки карты защитой паролем.

Механический переключатель на карте. Почти на всех картах имеется сбоку маленький переключатель в виде двухпозиционного ползунка (см. описание механики в Главе 8), который будет использоваться, чтобы показать хосту - хочет ли пользователь защитить эту карту от записи, или нет. Если этот ползунок установлен так, что открыт паз сбоку карты, то это означает включение защиты от записи на этой карте. Если же ползунок закрывает паз, то карта не защищена от записи.

На сокете, куда вставляется карта, имеется соответствующий пазу механический контакт, замыкание или размыкание которого покажет хосту, защищена карта от записи, или нет. Только от хоста зависит, будет работать эта защита, или нет (защита должна быть реализована хостом программно). Текущая позиция ползунка неизвестна для внутренних схем карты.

Внутренняя защита от записи в карте (необязательная опция). *В упрощенной спецификации (этот документ) описание этой секции опущено.*

4.3.7 Работа блокировки/разблокировки карты (Card Lock/Unlock)

4.3.7.1 Общее описание. Функция защиты паролем разрешает хосту заблокировать карту с установкой пароля. В дальнейшем пароль будет использоваться для разблокировки этой карты. Пароль и его размер хранятся в регистрах PWD (128 бит) и PWD_LEN (8 бит). Эти регистры энергонезависимы, так что пропадание и восстановление питания карты не очистят их.

Заблокированные карты отвечают на все команды (и выполняют их), находящиеся в "базовом" классе команд (class 0), ACMD41, CMD16 и команда "lock card". Таким образом хост может сбрасывать, инициализировать, выбирать, опрашивать статус карты и т. д., но у него нет возможности получить доступ к данным карты. Если раньше был установлен пароль (значение PWD_LEN не 0), карта будет автоматически заблокирована после включения питания.

Подобно существующей команде записи регистра CSD, команда lock/unlock доступна только в состоянии передачи (transfer state). Это означает, что команда не включает аргумент адреса и карта должна быть выбрана перед использованием.

Команда блокировки/разблокировки имеет структуру и тип транзакции по шине как и обычная команда записи одного блока (single block write). Передаваемый блок данных включает всю необходимую для команды информацию (режим установки пароля, сам PWD, заблокировать/разблокировать карту и т. д.). Таблица 4-4 описывает структуру блока данных команды. Обратите внимание, что хост, совместимый со стандартом SD Physical Specification Version 2.00, должен зарезервировать биты 7-4 в состоянии 0, когда выдается CMD42.

Таблица 4-4. Структура данных команды Lock Card.

№ байта	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0
0	Зарезервировано (должно быть сброшено в 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Данные пароля							
...								
PWDS_LEN + 1								

- ERASE: 1 определяет операцию принудительной очистки (Forced Erase). В байте 0 бит 3 будет установлен в 1 (все другие биты должны быть в 0). Все другие байты в этой команде будут игнорироваться картой.
- LOCK/UNLOCK: 1 = блокирует карту. 0 = разблокирует карту (имейте в виду, что допустимо устанавливать этот бит вместе с SET_PWD, но не разрешена их совместная установка с командой CLR_PWD).
- CLR_PWD: 1 = очищает PWD.
- SET_PWD: 1 = устанавливает в PWD новый пароль.
- PWDS_LEN: определяет длину следующего пароля (паролей) в байтах. В случает смены пароля это поле включает общую длину старого и нового пароля. Длина пароля может быть до 16 байт. В случае смены пароля общая длина старого и нового пароля может быть до 32 байт.
- Password data: в случае установки нового пароля здесь содержится новый пароль. В случае смены пароля здесь содержится старый пароль, за которым идет новый пароль.

Размер блока данных определяется хостом перед отправкой команды блокировки/разблокировки карты. Длина блока устанавливается больше или равной требуемому размеру структуры данных команды lock/unlock. В следующем объяснении смена размера блока командой CMD16 не обязательное требование для команды lock/unlock.

Следующие параграфы определяют различные последовательности команды lock/unlock:

Установка пароля

- Выбор карты (CMD7), если она не была ранее выбрана.
- Определение длины блока (CMD16), которое дается в 8-битном режиме блокировки/разблокировки, 8-битный размер пароля (в байтах), и некоторое количество байт нового пароля. В случае, когда успешно осуществлена смена пароля, тогда размер блока должен учитывать, что оба пароля - старый и новый - отправлены с командой.
- Отправка команды Card Lock/Unlock с подходящим размером блока на линии данных, включая 16-битную CRC. Блок данных показывает режим (SET_PWD), длину (PWDS_LEN) и сам пароль. В случае, когда замена пароля произведена, то значение длины (PWDS_LEN) учитывает оба пароля (старый и новый), и поле данных включает старый пароль (используемый сейчас), за которым идет новый пароль. Обратите внимание, что карта обрабатывает внутри себя вычисление длины нового пароля вычитанием длины старого пароля из значения поля PWDS_LEN.
- В случае, когда отправленный старый пароль не не подходит (не совпадает по размеру и содержанию), то будет установлен бит LOCK_UNLOCK_FAILED в регистре статуса, и старый пароль не будет изменен. В случае, когда старый пароль корректен (совпал по длине и содержимому), то указанный новый пароль и его размер будут сохранены в регистрах PWD и PWD_LEN соответственно.

Обратите внимание, что регистр длины пароля (PWD_LEN) показывает, установлен ли пароль. Когда в этом регистре 0, то пароль не установлен. Если значение PWD_LEN не равно 0, то карта блокируется после включения питания. Можно заблокировать карту немедленно в текущей сессии питания путем установки бита LOCK/UNLOCK (когда устанавливается пароль) или отправкой дополнительной команды для блокировки карты.

Сброс пароля

- Выбор карты (CMD7), если она не была ранее выбрана.
- Определение длины блока (CMD16), которое дается в 8-битном режиме блокировки/разблокировки, 8-битный размер пароля (в байтах), и некоторое количество байт пароля, который в настоящее время установлен.
- Отправка команды lock/unlock с подходящим блоком данных по линии данных, включая 16-битную CRC. Блок данных должен включать режим CLR_PWD, длину (PWDS_LEN), и сам пароль. Если содержимое PWD и PWD_LEN совпадают по содержимому и длине, то содержимое регистра PWD очищается, и PWD_LEN устанавливается в 0. Если пароль не корректный, то в регистре статуса установится бит LOCK_UNLOCK_FAILED.

Блокировка карты

- Выбор карты (CMD7), если она не была ранее выбрана.
- Определение длины блока (CMD16), которое дается в 8-битном режиме блокировки/разблокировки, 8-битный размер пароля (в байтах), и некоторое количество байт пароля, который в настоящее время установлен.
- Отправка команды блокировки/разблокировки карты с подходящего размера блоком данных на линии данных, включая 16-битную CRC. Блок данных показывает режим LOCK (блокировка), длину (PWDS_LEN) и сам пароль.

Если содержимое PWD равно отправленному паролю, то карта будет заблокирована, и в регистре статуса будет установлен бит card-locked. Если пароль не установлен, то в регистре статуса будет установлен бит ошибки LOCK_UNLOCK_FAILED.

Обратите внимание, что можно установить пароль и заблокировать карту в одной последовательности. В таком случае хост должен выполнить требуемые шаги для установки пароля (как описано ранее) включая установку бита LOCK, когда отправляется команда для установки нового пароля.

Если ранее был установлен пароль (PWD_LEN не равен 0), то карта будет заблокирована автоматически после сброса при включении питания.

Попытка заблокировать уже заблокированную карту или блокировка карты, у которой пароль не установлен, окончится неудачей и будет установлен бит ошибки LOCK_UNLOCK_FAILED в регистре статуса, исключая ситуации, когда это делается во время операций определения пароля или смены пароля.

Разблокировка карты

- Выбор карты (CMD7), если она не была ранее выбрана.
- Определение длины блока (CMD16), которое дается в 8-битном режиме блокировки/разблокировки, 8-битный размер пароля (в байтах), и некоторое количество байт пароля, который в настоящее время установлен.
- Отправка команды блокировки/разблокировки карты с подходящего размера блоком данных на линии данных, включая 16-битную CRC. Блок данных показывает режим UNLOCK (блокировка), длину (PWDS_LEN) и сам пароль.

Если содержимое PWD соответствует посланному паролю, то карат будет разблокирована, и в регистре статуса будет очищен бит card-locked. Если пароль не корректен, то в регистре статуса будет установлен бит LOCK_UNLOCK_FAILED.

Обратите внимание, что разблокировка осуществляется только для текущей сессии питания. Пока PWD не очищен, карта будет автоматически блокироваться при каждом следующем включении питания. Есть только один способ разблокировать карту, чтобы она не входила сама в блокировку - очистить пароль.

Попытка разблокировать уже разблокированную карту окажется неудачной, и в регистре статуса будет установлен бит LOCK_UNLOCK_FAILED, за исключением тех случаев, когда выполнилась операция установки или смены пароля.

4.3.7.2 Параметр и результат CMD42. Длина блока должна быть больше или равной размеру, структуры данных, требуемой для CMD42; иначе результат CMD42 не определен, и карта может быть в неожиданном, заблокированном состоянии. Таблица 4-5 показывает поведение команды CMD42. Зарезервированные биты в параметре (7..4) CMD42 не имеют значения. В случае, когда CMD42 требует пароля, подразумевается, что старый пароль и новый пароль установлены корректно; иначе карта покажет ошибку, независимо от описания в таблице 4-5. Если длина пароля 0 или больше 128 бит, то карта покажет ошибку. Если произошла ошибка при выполнении CMD42, установится в 1 бит LOCK_UNLOCK_FAILED (бит 24 статуса карты), независимо от описания в таблице 4-5. Бит CARD_IS_LOCKED (бит 25 статуса карты) в ответе (response) на команду CMD42 будет тот же самый, как и текущее состояние карты (Current Card State) в таблице 4-5. В поле статуса карты переход от 0 к 1 означает, что карта перешла в состояние блокировки, и переход от 1 к 0 означает, что карта стала разблокированной после выполнения команды CMD42. Это можно видеть в ответе на CMD13, которая отправлена после CMD42. Бит LOCK_UNLOCK_FAILED (бит 24 статуса карты) как результат CMD42 можно увидеть в ответе на либо команду CMD42, либо в ответе на следующую за ней команду CMD13.

Таблица 4-5. Как работает функция Unlock (базовая последовательность CMD42).

Параметр CMD42(1)				Текущее состояние карты	PWD_LEN и PWD	Результат функции	Состояние карты	
бит3	бит2	бит1	бит0				бит25(2)	бит24(2)
После включения питания (power on)					существует	Карта заблокирована	1	0
					очищен	Карта разблокирована	0	0
1	0	0	0	заблокирована	существует	Force Erase (см. табл. 4-6)	табл.4-6	табл.4-6
1	0	0	0	разблокирована	существует	Ошибка	0	1
1	0	0	0	разблокирована	очищен	Ошибка	0	1
0	1	0	0	заблокирована	существует	Ошибка	1	1
0	1	0	0	разблокирована	существует	Блокировка карты	0->1	0
0	1	0	0	разблокирована	очищен	Ошибка	0	1
0	1	0	1	заблокирована	существует	Смена пароля, карта заблокирована	1	0
0	1	0	1	разблокирована	существует	Смена пароля, карта разблокирована	0->1	0
0	1	0	1	разблокирована	очищен	Установка пароля и блокировка карты	0->1	0
0	0	1	0	заблокирована	существует	Сброс PWD_LEN и PWD, карта разблокирована	1->0	0
0	0	1	0	разблокирована	существует	Сброс PWD_LEN и PWD	0	0
0	0	1	0	разблокирована	очищен	Ошибка (примечание 4 к таблице 4-8)	0	1
0	0	0	1	заблокирована	существует	Замена пароля, карта разблокирована	1->0	0
0	0	0	1	разблокирована	существует	Замена пароля, карта разблокирована	0	0
0	0	0	1	разблокирована	очищен	Установка пароля, карта разблокирована	0	0
0	0	0	0	заблокирована	существует	Разблокировка карты	1->0	0
0	0	0	0	разблокирована	существует	Ошибка	0	1
0	0	0	0	разблокирована	очищен	Ошибка	0	1
Другие комбинации				Не имеет значения	Не имеет значения	Ошибка (см. примечание 1 к табл. 4-8)	0->1	1

Примечания:

(1) CMD42, параметр в данных

Bit3: ERASE
Bit2: LOCK_UNLOCK
Bit1: CLR_PWD
Bit0: SET_PWD

(2) Биты в Card Status, связанные с командой

Bit25: CARD_IS_LOCKED
Bit24: LOCK_UNLOCK_FAILED

Указание по применению: чтобы поменять пароль, хост должен учесть следующие случаи. Когда существуют PWD_LEN и данные пароля, карта предполагает, что старый и новый пароли установлены в этой структуре данных. Когда PWD_LEN и PWD очищены, то карта предполагает только установку нового пароля в этой структуре данных. В последнем случае хост не устанавливает старый пароль в структуре данных; иначе будет установлен неожиданный пароль.

4.3.7.3 Forcing Erase (принудительная очистка). В случае, когда пользователь забыл пароль (содержимое PWD), есть возможность стереть все данные карты вместе с содержимым PWD (пароль будет сброшен, и все данные карты уничтожены). Эта операция называется Forced Erase.

- Выбор карты (CMD7), если она не была ранее выбрана.
- Определение длины блока (CMD16) в 1 байт (8-битная команда блокировки/разблокировки карты). Отправка команды lock/unlock с соответствующим блоком данных в 1 байт на линии данных, включая 16-битную CRC. Блок данных должен показать режим ERASE (установленным должен быть только бит ERASE).

Если в поле данных установлен не только бит ERASE, то в регистре статуса установится бит ошибки LOCK_UNLOCK_FAILED, и запрос на очистку будет отклонен. Если команда была принята, то ВСЕ СОДЕРЖИМОЕ КАРТЫ (данные) БУДЕТ СТЕРТО, включая регистры PWD и PWD_LEN, и заблокированная карта станет разблокированной. Попытка принудительной очистки разблокированной карты потерпит неудачу, и в регистре статуса установится бит ошибки LOCK_UNLOCK_FAILED.

4.3.7.3.1 Функция Force Erase для заблокированной карты. Таблица 4-6 показывает взаимосвязь между принудительной очисткой и защитой от записи (Write Protection). Принудительная очистка не очистит защищенную область. Карта останется в заблокированном состоянии во время выполнения очистки и сменит состояние на разблокированное после того, как будет выполнена очистка всей пользовательской области. Подобным образом карта сохраняет временную защиту от записи (Temporary Write Protection) и групповую защиту от записи (Group Write Protection) во время выполнения очистки, и очистит Write Protection после того, как будет завершена очистка всей области данных пользователя. В случае, когда произошла ошибка очистки, карта может продолжить принудительную очистку, если данные в ошибочных секторах были уничтожены.

Таблица 4-6. Функция Force Erase на заблокированной карте (взаимосвязь с защитами от записи).

Параметр CMD42				PWP(1)	TWP, GWP(1)	Результат функции	Состояние карты	
бит3	бит2	бит1	бит0				бит25	бит24
1	0	0	0	ДА	Не имеет значения	Ошибка (см. прим. 2 к таблице 4-8)	1	1
1	0	0	0	нет	ДА	Выполнение принудительного стирания (force erase) и очистка Temporary Write Protect и Group Write Protect (см. прим. 3 к таблице 4-8)	1->0	0
1	0	0	0	нет	нет	Выполнение принудительного стирания (force erase)	1->0	0

Примечание (1). Виды защиты от записи:

PWP постоянная защита от записи (Permanent Write Protect, бит 13 CSD).
TWP временная защита от записи (Temporary Write Protect, бит 12 CSD).
GWP групповая защита от записи (Group Write Protect, CMD28, CMD29, CMD30).

4.3.7.4 Взаимосвязь между ACMD6 и Lock/Unlock State. ACMD6 отклоняется, когда карта заблокирована, и ширина шина может быть изменена, когда карта разблокирована. Таблица 4-7 показывает взаимосвязь между ACMD6 и состоянием блокировано/разблокировано (Lock/Unlock state).

Таблица 4-7. Взаимосвязь между ACMD6 и Lock/Unlock State.

Состояние карты	Режим шины	Результат функции
Разблокирована	1 бит	Команда ACMD6 принята.
Заблокирована	1 бит	Команда ACMD6 отклонена, и шина все еще в режиме 1 бит.
Разблокирована	4 бита	Команда ACMD6 принята.
Заблокирована	4 бита	Команда ACMD6 отклонена, и шина все еще в режиме 4 бита. CMD0 поменяет режим шины на 1 бит.

Указания по применению: после включения питания (в 1-битном режиме), если карта заблокирована, хост режима SD должен выдать команду CMD42 в 1-битном режиме. Если карта заблокирована в 4-битном режиме, хост режима SD должен выдать CMD42 в 4-битном режиме.

4.3.7.5 Команды, которые может принять заблокированная карта. Заблокированная карта принимает перечисленные ниже команды и возвращает ответ (response) с установкой CARD_IS_LOCKED.

1) Basic class (0)
2) Lock card class (7)
3) CMD16
4) ACMD41
5) ACMD42

Все другие команды, включая команды безопасности, обрабатываются как недопустимые.

Указание по применению: после включения питания хост может распознать состояние карты заблокировано/разблокировано по состоянию бита CARD_IS_LOCKED в ответе на команду CMD7 или CMD13.

4.3.7.6 Два типа блокировки/разблокировки карты. Есть 2 типа поддержки картами функции lock/unlock, Type 1 и Type 2. Type 1 относится к ранней версии SD Memory Card, и Type 2 определен в стандарте Physical Layer Specification Version 1.10 и более новом. Таблица 4-8 показывает различия между этими двумя типами карт. Карты памяти SD, которые поддерживают Lock/Unlock, и совместимы с Version 1.01, могут быть либо Type 1 или Type 2. Карты SD Memory, которые поддерживают Lock/Unlock, и совместимые с Version 1.10 и более новой, поддерживают Type 2.

Таблица 4-8. Отличия версий функций Lock/Unlock.

Примечания	Карта Type 1	Карта Type 2
Прим. 1 к таблице 4-5	Как обрабатывается параметр в CMD42: 0011b обрабатывается как 0001b 0111b обрабатывается как 0101b 0110b обрабатывается как 0010b Результатом других комбинаций будет ошибка.	Все результаты приводят к ошибке.
Прим. 2 к таблице 4-6	Выполняется принудительное стирание и устанавливается Permanent Write Protect. Если принудительное стирание завершено, то CARD_IS_LOCKED поменяется с 1 на 0. Принудительное стирание получает приоритет перед Permanent Write Protect.	Результатом будет ошибка. Установка Permanent Write Protect получает приоритет над принудительным стиранием.
Прим. 3 к таблице 4-6	Выполняется принудительное стирание, но не очищаются Temporary Write Protect и Group Write Protect. Они должны быть очищены хостом.	Выполняется принудительное стирание, и очищаются Temporary Write Protect и Group Write Protect.
Прим. 4 к таблице 4-5	Как обрабатывается параметр в CMD42: параметр 0010b и 0110b приводят к ошибке. Бит24 статуса карты будет равен 0.	Результатом будет ошибка. Бит24 статуса карты будет равен 1.

Указание по применению: хост может использовать оба типа карт без проверки, если учитывать следующие моменты.

1. Хост не должен устанавливать параметры CMD42, которые возвращают ошибку, перечисленную в таблице 4-5 (для *1).
2. Хост не должен выдавать команду принудительной очистки (force erase), если Permanent Write Protect установлен в 1, иначе карта Type 1 не может быть больше использована, даже если пользователь помнит пароль (для *2).
3. После force erase, если Temporary Write Protect не очищен, хост должен очистить его (для *3).

4.3.8 Защита контента. *В упрощенной спецификации (этот документ) описание этой секции опущено.*

4.3.9 Команды, относящиеся к приложению

4.3.9.1 APP_CMD (CMD55). Эта команда, будучи принятой картой, интерпретируется ей так, что следующая команда будет специфична для приложения (application-specific command, ACMD). ACMD имеет ту же самую структуру, что и обычные команды, и может иметь такой же номер CMD. Карта будет распознавать это как ACMD по факту, когда она появляется после APP_CMD.

Эффект от APP_CMD только в том, что если индекс команды, непосредственно следующей за APP_CMD, есть ACMD, перезагружающая её, то используется не регулярная версия команды. Если, например, у карты есть определение для ACMD13, но не для ACMD7, то если немедленно принятая после APP_CMD команда это команда 13, то она будет интерпретирована как не регулярная ACMD13, но команда 7 как регулярная CMD7. Чтобы использовать одну из заданных производителем специфических ACMD хост должен сделать следующее:

- Отправить APP_CMD. Ответ будет содержать установленный бит APP_CMD (новый бит статуса), сигнализирующий о том, что хост теперь ожидает команду ACMD.
- Отправляется требуемая ACMD. В ответе будет установлен бит APP_CMD, показывая этим, что принятая команда интерпретирована как ACMD. Если будет отправлена не-ACMD команда, то она обрабатывается картой как обычная команда SD Memory Card, и бит APP_CMD в статусе карты останется очищенным.
- Если несколько CMD55 выдается друг за другом, то бит APP_CMD а каждом ответе устанавливается в 1. Команда, следующая немедленно за последней CMD55, интерпретируется как ACMD. Когда больше одной команды (кроме CMD55) выдается непосредственно после CMD55, то первая команда интерпретируется как ACMD и следующие команды интерпретируются как обычные команды.

Если отправлена недопустимая команда (ни ACMD, ни CMD), то она будет обработана как стандартная ошибка SD Memory Card, относящаяся к недопустимой команды.

С точки зрения протокола SD Memory Card номера ACMD будут определены производителями с некоторыми ограничениями. Следующие номера ACMD зарезервированы для проприетарных приложений SD Memory Card, и не могут использоваться любыми производителями карт SD Memory: ACMD6, ACMD13, ACMD17-25, ACMD38-49, ACMD51.

4.3.9.2 General Command - GEN_CMD (CMD56). *В упрощенной спецификации (этот документ) описание этой секции опущено.*

4.3.10 Команда переключения функции (Switch Function Command)

4.3.10.1 Общая информация. Команда Switch Function (CMD6)(1) используется для переключения или расширения функций карт памяти. В настоящее время определены 2 группы функций:

- Режим доступа к карте: скорость интерфейса 12.5 MB/sec (по умолчанию) или 25 MB/sec (highspeed, высокая скорость).
- Система команд карты: стандартный набор команд (по умолчанию) или набор команд eCommerce, или набор команд вендора (Vendor Specific Command set).

Примечание (1): CMD6 определена для карт памяти. Карта SDIO будет использовать CCCR для переключения функций.

Это было представлено в стандарте SD Physical Layer Specification Version 1.10. Таким образом карты, которые совместимы с ранними версиями стандарта, не будут поддерживать эту команду. Перед использованием CMD6 хост должен проверить поле "SD_SPEC" в регистре SCR, чтобы идентифицировать, какая версия стандарта поддерживается картой. Поддержка CMD6 обязательна для SD memory card Version 1.10 и более свежих версий.

CMD6 допустимо использовать в состоянии "Transfer State". Будучи выбранными через команду switch, все функции вернутся в функционирование по умолчанию только после цикла выключения и включения питания, CMD6 (Mode 1 работы с Function 0 в каждой функциональной группе) или CMD0. Цикл выключение/включение питания или выдача CMD0 приведет к сбросу карты с переходом в состояние ожидания (idle state), и все функции переключатся обратно в состояние по умолчанию.

В ответ на CMD6 карта памяти SD пошлет R1 response по сигналу CMD, и 512 бит статуса по сигналам DAT. С точки зрения транзакции шины SD, это стандартное одиночное чтение блока (single block read), и значение таймаута этой команды составляет 100 мс, такое же время, как и для команды чтения (Read). Если произойдет ошибка CRC в данных статуса, хост должен выдать цикл переключения питания (или сброс карты).

Период переключения функции, вызванного CMD6, занимает период 8 тактов после прохождения последнего бита данных статуса. Когда команда CMD6 меняет поведение шины (например, режим доступа), хосту разрешается использовать новые функции (повысить/понизить частоту CLK сверх текущей максимальной частоты CLK), как минимум после 8 тактов от конца транзакции команды switch (см. рисунок 4-4).

В ответе на CMD0 период переключения находится в пределах 8 тактов после прохождения последнего бита CMD0. Когда CMD6 меняет поведение шины (например режим доступа), хосту можно начать процесс инициализации как минимум через 8 тактов после CMD0.

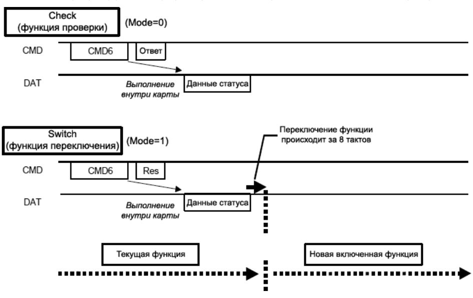


Рис. 4-4. Использование команды Switch.

CMD6 поддерживает 6 функциональных групп, и каждая функциональная группа поддерживает 16 ветвей (функций). Только одна функция может быть выбрана и активна в имеющейся функциональной группе. Функция 0 в каждой функциональной группе это функция по умолчанию (совместимо со стандартом Spec. 1.01).

CMD6 можно использовать в 2 режимах:

- Mode 0 (функция проверки) используется, чтобы опросить карту - поддерживает ли она определенную функцию или функции.
- Mode 1 (функция установки) используется для переключения функциональности карты.

4.3.10.2 Mode 0 - Check Function. CMD6 mode 0 используется для опроса функций, поддерживаемых картой, и идентификации максимального потребляемого тока картой в этих выбранных функциях.

См. таблицу 4-27: команды функции переключения (class 10) для определения аргумента CMD6.

Опрос выполняется установкой поля аргумента в команде следующим образом:

- Установка бита Mode в значение 0.
- Выбор только одной функции в каждой функциональной группе. Выбор функции по умолчанию осуществляется установкой функции в 0x0. Установка определенной функции делается путем использования подходящих значений из таблицы 4-9. Выбор 0xF сохранит выбранной текущую выбранную функцию в функциональной группе.
- Когда функция в запросе готова, карта вернет запрошенный номер функции, а когда занята, карта вернет текущий номер функции (см. таблицу 4-12).

В ответе на запрос функция переключения вернет следующие 3 статуса (см. таблицу 4-10):

- Функции, которые поддерживаются каждой функциональной группой.
- Функция, на которую переключится карта в каждой из функциональных групп. Это значение идентично предоставленному аргументу, если хост сделал допустимый выбор, или 0xF, если выбранная функция недопустима.
- Максимальное потребление тока для выбранных функций. Если одна из выбранных функций неправильная, возвращенное значение будет 0.

4.3.10.3 Mode 1 - установка функции. CMD6 mode 1 используется для переключения функциональности карты.

Переключение на новый функционал делается следующим образом:

- Установка бита Mode в значение 1.
- Выбор только одной функции в каждой из функциональных групп. Выбор функции по умолчанию делается установкой функции в 0x0. Рекомендуется указать 0xF (никак не влиять) для всех выбранных функций, кроме функций, которые нужно поменять. Выбор 0xF сохранит текущую функцию для функциональной группы.
- Когда функция не может быть переключена из-за того, что она занята, карта вернет текущий номер функции (не вернет 0xF), другие функции в других группах все еще могут быть переключены.

В ответ на функцию установки функция переключения вернет 3 статуса:

- Функции, которые поддерживаются каждой функциональной группой.
- Функция, которая являются результатом команды переключения. В случае неправильного выбора одной функции или нескольких, все устанавливаемые значения игнорируются и не осуществляются никакие изменения (идентично случаю, где хост выбирает 0xF для всех функциональных групп). Ответ на все недопустимый выбор функции будет 0xF.
- Максимальное потребление тока для выбранных функций. Если одна из выбранных функций неправильная, возвращенное значение будет 0.

Таблица 4-9. Доступные функции.

Срез аргумента/ номер группы:	**[23:20]/6**	**[19:16]/5**	**[15:12]/4**	**[11:8]/3**	**[7:4]/2**	**[3:0]/1**
Имя функции:	**зарезервировано**	**зарезервировано**	**зарезервировано**	**зарезервировано**	**Система команд**	**Режим доступа**
0x0	По умолчанию (Ver. 1.01)					
0x1	зарезервировано	зарезервировано	зарезервировано	зарезервировано	Для eC	High-Speed
0x2	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x3	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x4	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x5	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x6	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x7	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x8	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0x9	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0xA	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0xB	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0xC	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0xD	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано	зарезервировано
0xE	зарезервировано	зарезервировано	зарезервировано	зарезервировано	Специфика вендора	зарезервировано
0xF	Никакого влияния					

4.3.10.4 Статус функции переключения (Switch Function Status). Статус функции переключения это возвращаемый блок данных, который содержит функцию и информацию о текущем потреблении тока. Длина блока предопределена на 512 бит, и использовать команду SET_BLK_LEN нет необходимости. Таблица 4-10 описывает структуру данных статуса.

Биты статуса в ответе содержат информацию о функциональной группе. Максимальное потребление тока будет использоваться только для новой функции, добавляемой этой командой. В этом случае значения VDD_R_CURR_MIN, VDD_W_CURR_MIN, VDD_R_CURR_MAX и VDD_W_CURR_MAX в регистре CSD предоставляют потребление тока, когда все функции установлены в состояние по умолчанию, и могут использоваться хостом, совместимым со спецификацией 1.01.

Таблица 4-10. Структура данных статуса.

Биты	Описание	Ширина
[511:496]	Максимальное потребление тока (0: ошибка, 1: 1 мА, 2: 2 мА, ..., 65535: 65535 мА) под функцией, показанной битами [399:376]. Напряжение для вычисления потребления определяется командой ACMD41 (для карты памяти SD) или CMD5 (карта SDIO). Максимальный потребляемый ток показывает общее потребление тока (от части памяти), если эти функции подключены. Хост должен проверить максимальное потребление тока и проверить, сможет ли он предоставить такой необходимый ток перед работой Mode 1. Максимальный потребляемый ток является средним током в течение 1 секунды.	16
[495:480]	Информация по функциональной группе 6. Если бит установлен, то соответствующая функция поддерживается.	16
[479:464]	Информация по функциональной группе 5. Если бит установлен, то соответствующая функция поддерживается.	16
[463:448]	Информация по функциональной группе 4. Если бит установлен, то соответствующая функция поддерживается.	16
[447:432]	Информация по функциональной группе 3. Если бит установлен, то соответствующая функция поддерживается.	16
[431:416]	Информация по функциональной группе 2. Если бит установлен, то соответствующая функция поддерживается.	16
[415:400]	Информация по функциональной группе 1. Если бит установлен, то соответствующая функция поддерживается.	16
[399:396]	Mode 0: эта функция может быть переключена в группе 6. Mode 1: функция, которая является результатом команды переключения в группе 6. 0xF показывает ошибку установки функции с аргументом.	4
[395:392]	Mode 0: эта функция может быть переключена в группе 5. Mode 1: функция, которая является результатом команды переключения в группе 5. 0xF показывает ошибку установки функции с аргументом.	4
[391:388]	Mode 0: эта функция может быть переключена в группе 4. Mode 1: функция, которая является результатом команды переключения в группе 4. 0xF показывает ошибку установки функции с аргументом.	4
[387:384]	Mode 0: эта функция может быть переключена в группе 3. Mode 1: функция, которая является результатом команды переключения в группе 3. 0xF показывает ошибку установки функции с аргументом.	4
[383:380]	Mode 0: эта функция может быть переключена в группе 2. Mode 1: функция, которая является результатом команды переключения в группе 2. 0xF показывает ошибку установки функции с аргументом.	4
[379:376]	Mode 0: эта функция может быть переключена в группе 1. Mode 1: функция, которая является результатом команды переключения в группе 1. 0xF показывает ошибку установки функции с аргументом.	4
[375:368]	Версия структуры данных. 00h: определены биты [511:376] 01h: определены биты [511:272] 02h..FFh: зарезервировано	8
[367:352]	Статус занятости функций в группе 6. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[351:336]	Статус занятости функций в группе 5. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[335:320]	Статус занятости функций в группе 4. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[319:304]	Статус занятости функций в группе 3. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[303:288]	Статус занятости функций в группе 2. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[287:272]	Статус занятости функций в группе 1. Если бит [i] установлен, то функция [i] занята (busy). Это поле может быть прочитано в Mode 0 и Mode 1.	16
[271:0]	Зарезервировано (все биты в лог. 0).	272

4.3.10.4.1 Состояние занятости для функций. Каждый бит [367-272] показывает статус занятости (busy status) соответствующих функций; 0 показывает готовность, и 1 показывает занятость. Когда статус показывает занятость, хост не должен менять соответствующую функцию. Команда переключения mode 1 может применяться только к функциям в состоянии готовности.

Если функция не смогла переключиться в режиме работы mode 1 и вернула текущий номер функции в ответе, то эта функция считается занятой. Работа в mode 1 может повлиять на поведение функции. Работа в mode 0 должна использоваться для проверки статуса занятости функции, потому что это не влияет на поведение функции, особенно для функции группы 2, как это описано ниже.

Функциональная группа 2															
303	302	301	300	299	298	297	296	295	294	293	292	291	290	289	288
0	VS(1)	0	0	0	0	0	0	0	0	0	0	0	0	eC(2)	0

Рис. 4-5. Busy Status системы команд.

Примечания к рис. 4-5:

(1) Статус занятости для "Vendor Specific". 0: состояние готовности (ready), 1: состояние занятости (busy).
(2) Статус занятости для "eC". 0: состояние готовности (ready), 1: состояние занятости (busy).

4.3.10.4.2 Версия структуры данных (Data Structure Version). Показывает эффективные поля бит в статусе функции переключения (Switch Function Status). Карты могут установить либо 00h, либо 01h. Когда это поле установлено в 01, эффективна индикация занятости статуса.

Таблица 4-11. Версия структуры данных.

Версия структуры данных	Поля структуры данных статуса
00h	Определены биты [511:376]
01h	Определены биты [511:272]
02h..FFh	Зарезервировано

4.3.10.4.3 Таблица функций команды Switch. Таблицы 4-12, 4-13 и 4-14 показывают возможные комбинации функции переключения. "Argument" показывает 4-битный код, заданный в аргументе команды переключения (биты 23-0). "Busy Status" показывает занятость функции, как определено ниже. "Status Code" показывает 4-битный код структуры данных статуса, биты 399-376.

Таблица 4-12. Код статуса Mode 0 для поддерживаемой функциональной группы.

Аргумент	Статус занятости	Код статуса	Комментарий
0	Не имеет значения	0	Статус показывает функцию по умолчанию, которая всегда поддерживается.
Поддерживаемая функция	Ready (в состоянии готовности)	=аргументу	Статус показывает, что функция, указанная в аргументе, поддерживается и может быть подключена.
	Busy (в состоянии занятости)	Текущий выбор	Статус показывает, что функция, указанная в аргументе, поддерживается, но не может быть переключена, потому что эта функция занята.
Не поддерживаемая функция	Не имеет значения	Fh	Статус показывает, что функция, указанная в аргументе, не поддерживается.
Fh		Текущий выбор	Статус показывает текущую выбранную функцию.

Таблица 4-13. Код статуса Mode 1 для поддерживаемой функциональной группы.

Аргумент	Статус занятости	Код статуса	Комментарий
0	Не имеет значения	0	Функция по умолчанию всегда может быть подключена.
Поддерживаемая функция	Ready (в состоянии готовности)	=аргументу	Статус показывает ту же функцию, что и номер, указанный в аргументе. Это показывает успешное переключение функции.
	Busy (в состоянии занятости)	Текущий выбор	Переключение функции отменено, и статус показывает текущую выбранную функцию.
Не поддерживаемая функция	Не имеет значения	Fh	Если одна из функциональных групп показывает код ошибки (Fh), то запросы на переключение всех функций отменены и остальные данные в структуре данных статуса должны быть проигнорированы.
Fh		Текущий выбор	Статус показывает текущую выбранную функцию.

Таблица 4-14. Код статуса Mode 0 и Mode 1 для не поддерживаемой функциональной группы.

Аргумент	Статус занятости	Код статуса	Комментарий
0	Не имеет значения	0	Статус всегда показывает 0.
1h..Eh		Fh	Статус всегда показывает Fh.
Fh		0	Статус всегда показывает 0.

4.3.10.5 Взаимосвязь между данными CMD6 и другими командами.

4.3.10.6 Пример использования переключения функции (Switch Function).

4.3.10.7 Пример проверки.

В упрощенной спецификации (этот документ) описание секций 4.3.10.5, 4.3.10.6, 4.3.10.7 опущено.

4.3.11 Режим High-Speed (скорость интерфейса 25 MB/sec). Хотя Rev 1.01 карт SD memory поддерживает скорость интерфейса до 12.5 MB/sec, поддержка скорости 25 MB/sec нужна для увеличения производительности, потому что хосту приходится работать в носителями данных, у которых постоянно растет объем памяти.

Чтобы достичь скорости интерфейса 25 MB/sec, тактовая частота увеличивается до 50 МГц, и параметры времени сигналов CLK/CMD/DAT и схемотехника должна быть совместимой со стандартом Physical Layer Specification Version 1.01.

После включения питания карта памяти SD находится в режиме скорости по умолчанию, и с помощью команды переключения функций (Switch Function, CMD6) карты памяти Version 1.10 и более новой версии могут быть переключены в режим высокой скорости (High-Speed). Функция High-Speed это функция в группе режима доступа (access mode, см. таблицу 4-9). Поддержка High-Speed является опциональной.

Поскольку невозможно управлять двумя картами или большим количеством карт в случае, когда каждая из карт имеет разные параметры интерфейса (по умолчанию и High-Speed) и чтобы удовлетворить разные требования по интервалам времени сигналов, хост в любой момент времени должен управлять только одной картой. Сигнал CLK/CMD/DAT должны быть подключены 1-в-1 между хостом и картой.

4.3.12 Система команд

Команды SD CMD34-37, CMD50 и CMD57 зарезервированы для расширения системы команд SD через команду переключения. Переключение между различными функциями функциональной группы системы команд, поменяет интерпретацию и связанные транзакции шины (например, команды без передачи данных, single block read, multiple block write, и т. д.) этих команд. Поддержка системы команд является опциональной.

- Когда выбран "стандартный набор команд" (функция по умолчанию 0x0), эти команды не будут распознаны картой и будут считаться недопустимыми (как определено в стандарте Version 1.01 SD Physical Layer Specification).
- Когда выбран набор "vendor specific" (функция 0xE), поведение этих команд будет специфично для вендора. Такое поведение не определено в этом стандарте, и может меняться у разных вендоров.
- Когда выбран набор "mobile e-commerce" (функция 0x1), поведение этих команд покрывается SD Specifications Part A1: Mobile Commerce Extension Specification.

Когда используется одно из этих расширений, требуется особое внимание для правильного выбора функции набора команд, иначе команда хоста может быть интерпретирована некорректно. Все другие команды карт памяти SD (не зарезервированные под команды переключения) всегда доступны и будут выполняться, как описано в этом документе, независимо от выбранного в настоящий момент набора команд.

4.3.13 Команда отправки состояния интерфейса (CMD8). CMD8 (Send Interface Condition Command) определена для инициализации карт памяти SD, совместимых со стандартом Physical Specification Version 2.00. CMD8 допустима, когда карта находится в состоянии ожидания (Idle state). Эта команда имеет 2 функции.

- Проверка напряжения: проверяет, работает ли карта от напряжения питания, предоставленного хостом.
- Разрешение расширения существующей команды и ответа (response): возобновление CMD8 разрешает расширить новый функционал для некоторых существующих команд путем переопределения зарезервированных бит. ACMD41 расширена для поддержки инициализации карт памяти SD High Capacity (SDHC). Таблица 4-15 показывает формат CMD8.

Таблица 4-15. Формат команды CMD8.

Позиции бит	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Ширина (в битах)	1	1	6	20	4	8	7	1
Значение поля	0	1	001000	00000h	x	x	x	1
Описание	start bit	Бит передачи	Индекс команды	Зарезервированные биты	Предоставленное напряжение (VHS)	Маска проверки	CRC7	end bit

VHS	Определение значения
0000b	Не определено
0001b	2.7V-3.6V
0010b	Зарезервировано для диапазона низких напряжений (Low Voltage Range)
0100b	Зарезервировано
1000b	
Другие значения	Не определено

Когда карта находится в Idle state, хост должен выдать команду CMD8 перед ACMD41. В аргументе "voltage supplied" устанавливается напряжение, предоставляемое хостом, и "check pattern" устанавливается в любую 8-битную маску проверки.

Карта проверяет, может ли она работать от напряжения питания хоста. Карта, которая принимает предоставленное напряжение, вернет ответ R7 (response 7). В этом ответе карта эхом отправляет обратно и диапазон напряжений, и маску проверки, установленные в аргументе команды. Если карта не поддерживает предоставленное хостом напряжение, то карта не возвращает ответ и остается в Idle state.

Указание по применению: рекомендуется для маски проверки (check pattern) значение 10101010b.

Дальнейшая часть этой секции в упрощенной спецификации не приводится.

4.3.14 Функциональное отличие команд High Capacity SD Memory Card (SDHC). Команды доступа к памяти включают команды блочного чтения (CMD17, CMD18), блочной записи (CMD24, CMD25) и блочной очистки (CMD32, CMD33).

Ниже приведены функциональные отличия команд доступа к памяти между картами SD стандартной емкости (Standard Capacity) и высокой емкости (High Capacity):

• Аргумент команды. В картах High Capacity 32-разрядный аргумент команд доступа к памяти использует адрес памяти в блочном формате (т. е. адрес означает не номер байта, а номер блока). Длина блока фиксирована, и составляет 512 байт. В картах стандартного размера 32-битный аргумент команды доступа к памяти использует адрес в байтовом формате. Длина блока определяется CMD16, например:

(a) Аргумент 0001h в байтовом адресе означает "байт номер 0001h" для карт Standard Capacity, и "блок номер 0001h" для карт High Capacity.
(b) Аргумент 0200h в байтовом адресе означает "байт номер 0200h" для карт Standard Capacity, и "блок номер 0200h" для карт High Capacity.

• Частичный доступ (Partial Access) и не выровненный доступ (Misalign Access). Частичный доступ и не выровненный доступ (с пересечением границы блока) запрещен для карт High Capacity, так как используется адрес блока. Доступ предоставляется только на основе адреса блока.

• Установка длины блока. Когда используются команды чтения или записи в режиме адреса блока, 512-байтная фиксированная длина блока используется независимо от длины блока, установленного командой CMD16. Установка длины блока не влияет на команды доступа к памяти. CMD42 не классифицируется как команда доступа к памяти. Размер блока данных должен быть указан командой CMD16, и длина блока может быть установлена до 512 байт. Установка длины блока больше чем 512 байт приведет к установке бита ошибки BLOCK_LEN_ERROR, независимо от емкости карты.

• Группа, защищенная от записи (Write Protected Group). Карты памяти SD High Capacity не поддерживают группы, защищенные от записи. Выдача команд CMD28, CMD29 и CMD30 генерирует ошибку ILLEGAL_COMMAND.

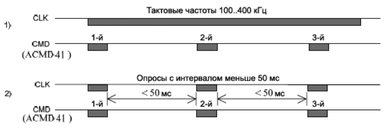
4.4 Управление тактированием. Тактовый сигнал карт памяти SD может использоваться хостом перевода карт в режим пониженного потребления энергии, или для управления потоком данных - чтобы избежать условий недогрузки (under-run) или перегрузки (over-run) данных на шине. Хосту разрешено понизить тактовую частоту, или совсем её отключить. Например, в случае, когда у хоста буфер данных 512 байт, и он хочет передавать данные карты при записи блоками по 1 килобайту. Таким образом, чтобы сохранить постоянную передачу данных, с точки зрения карты поток нужно останавливать после первых 512 байт. Тогда хост заполнит свой внутренний буфер другой порцией 512 байт. После того, как в хосте готова для записи вторая половина блока, хост продолжит передачу данных в карту путем перезапуска подачи тактов. В таком способе работы карта не распознает какой-либо перерыв в передаче данных. Здесь требуется учесть хостом несколько ограничений:

• Частота шины может быть изменена в любое время (с ограничением на максимальную частоту передачи данных и частоту при идентификации, как это определено в документации стандарта).

• Исключение из вышеупомянутого составляет ACMD41 (SD_APP_OP_COND). После выдачи ACMD41 хостом должны быть выполнены следующие процедуры 1) или 2), пока карта не придет в состояние готовности.

1) Подразумевает выдачу постоянных тактов с частотой в диапазоне 100..400 кГц.

Если хост хочет остановить тактирование, он опрашивает бит занятости (busy bit) командой ACMD41 с интервалами как минимум 50 мс.


Тактовые частоты 100..400 кГц
1)
CLK
CMD
(ACMD41)
1-й
2-й
3-й
Опросы с интервалом меньше 50 мс
2)
< 50 мс
< 50 мс

• Имеется очевидное требование, что тактовая частота должна выдаваться все время, когда карта выводит данные или токены ответа (response). По окончанию последней транзакции по шине SD хосту требуется предоставить 8 тактовых циклов для карты, чтобы завершить операцию перед выключением тактов. Ниже приведен список различных транзакций по шине:

- Команда без ответа. 8 тактов подается хостом после завершающего бита команды.
- Команда с ответом карты (response). 8 тактов после завершающего бита ответа карты.
- Транзакция чтения данных. 8 тактов по завершении последнего бита последнего блока данных.
- Транзакция записи данных. 8 тактов после токена статуса CRC.

• Хосту разрешено отключить такты у карты, которая занята внутренней операцией (busy). Карта будет завершать операцию программирования независимо от тактов хоста. Однако хост должен предоставить тактовый перепад для карты, чтобы выключить её сигнал занятости. Без этого тактового перепада (за исключением случая, когда карта ранее была отключена командой отмена выборки CMD7) сигнал DAT будет всегда в лог. 0.

4.5 Cyclic Redundancy Code (CRC, контрольная сумма). CRC предназначена для защиты команд, ответов и передач данных карты SD от ошибок передачи по шине карты памяти. Одна CRC генерируется для каждой команды, и проверяется для каждого ответа по линии CMD. Для данных блока генерируется одна CRC на передаваемый блок. CRC генерируется и проверяется по алгоритмам, описанным ниже.

• CRC7. Проверка на CRC7 используется для всех команд, всех ответов кроме R3, и для регистров CSD и CID. CRC7 это 7-битное значение, и оно вычисляется по полиному $G(x) = x^7 + x^3 + 1$. Алгоритм вычисления:

$M(x) = (\text{первый бит}) * x^n + (\text{второй бит}) * x^{n-1} +\ldots+ (\text{последний бит}) * x^0$
$CRC[6\ldots0] = \text{Остаток} [(M(x) * x^7)/G(x)]$

Первый бит это самый левый бит соответствующей строки бит (эта строка является потоком бит команды, ответа response, данных регистров CID или CSD). Степень n полинома это количество бит, защищаемых CRC минус 1. Количество защищаемых бит равно 40 для команд и ответов (n = 39) и 120 для регистров CSD и CID (n = 119).

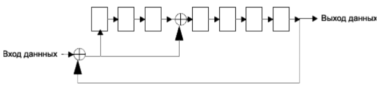


Рис. 4-6. Блок генератора/проверки CRC7.

Примеры вычисления CRC7. Секция CRC команды/ответа выделена жирным шрифтом.

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

• CRC16. В случае использования одной линии DAT применяется CRC16 для защиты полезной нагрузки в режиме блочной передачи. Контрольная сумма CRC это 16-битное значение и оно вычисляется по полиному $G(x) = x^{16} + x^{12} + x^5 + 1$. Алгоритм вычисления:

$M(x) = (\text{первый бит}) * x^n + (\text{второй бит}) * x^{n-1} +\ldots+ (\text{последний бит}) * x^0$
$CRC[15\ldots0] = \text{Остаток} [(M(x) * x^{16})/G(x)]$

Первый бит это первый бит данных соответствующего блока. Степень n полинома обозначает номер бита в блоке данных минус 1 (например n = 4095 для длины блока 512 байт). Генератор G(x) использует стандартный полином CCITT. Код имеет минимальную дистанцию d=4 и используется для полезной нагрузки до 2048 байт (n меньше или равно 16383).

Тот же самый метод CRC16, который используется в режиме передачи по одному сигналу DAT ("узкая" шина), используется и в режиме "широкой" шины (4 сигнала DAT). В режиме широкой шины CRC16 вычисляется и проверяется для каждой линии по отдельности.

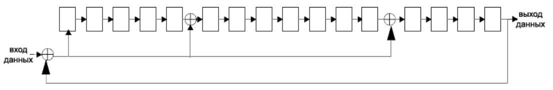


Рис. 4-7. Блок генератора/проверки CRC16.

Пример вычисления CRC16: 512 байт, где все данные равны 0xFF, дает CRC16 = 0x7FA1.

4.6 Условия возникновения ошибки

4.6.1 CRC и недопустимая команда. Все команды защищены битами CRC (cyclic redundancy check, контрольная сумма). Если не совпала CRC адресованной карты, карта не отвечает, и команда не выполнится. The card does not change its state, and COM_CRC_ERROR bit is set in the status register.

Подобным образом, если была принята недопустимая команда, карта не поменяет свое состояние, не выдаст response и установит бит ошибки ILLEGAL_COMMAND в регистре статуса. На диаграммах состояний показаны только безошибочные ветки (см. рис. 4-1 и рис. 4-3).

Таблица 4-28 содержит полное описание переходов между состояниями.

Возможны разные виды недопустимой команды:

- Команды, которые принадлежат классам, не поддерживаемых этой картой (например, команда записи поступила на карте read only).
- Команды, которые нельзя использовать в текущем состоянии (например CMD2 в состоянии передачи Transfer State).
- Команды, которые не определены (например CMD5).

4.6.2 Условия таймаута чтения, записи и очистки (Read, Write and Erase Timeout Conditions). Карта должна выполнить команду в заданном промежутке времени, или сдаться и вернуть сообщение об ошибке. Если хост не получил никакой ответ в течение заданного таймаута, то он должен предположить, что произошла ошибка, и должен попытаться восстановить рабочее состояние (например сбросить карту командой CMD0, или передернуть питание, и т. п.).

4.6.2.1 Read (чтение). Для карт стандартного объема (Standard Capacity SD Memory Card) времена, после которых возникает условие таймаута операций чтения, составят (независимо от карты) либо в 100 раз больший интервал, чем типичное время доступа для этих операций, либо меньше или равно 100 мс (меньшее из этих двух величин времени). Время доступа чтения (read access time) определено как сумма двух времен, полученных в параметрах TAAC и NSAC регистра CSD (см. Главу 5.3). В случае одиночной операции чтения эти параметры карты определяют типичную задержку между последним битом (end-бит) команды чтения и начальным битом (start-бит) блока данных. В случае операции множественного чтения (Multiple Block Read) они также определяют типичную задержку между конечным битом блока данных и начальным битом следующего блока данных.

Карты повышенной емкости (High Capacity SD Memory Card, SDHC) показывают параметры TAAC и NSAC как фиксированные величины. Хост должен использовать таймаут 100 мс (минимум) для операций одиночного и множественного чтения вместо использования TAAC и NSAC.

4.6.2.2 Write (запись). Для карт стандартной емкости (Standard Capacity SD Memory Card)времена, после которых возникает условие таймаута операций записи, составят (независимо от карты) время либо в 100 раз больше, чем типичные времена программирования для этих операций, либо 250 мс (меньшее из этих двух величин времени). Поле R2W_FACTOR в регистре CSD используется для вычисления типичного времени программирования блока, получаемого умножением read access time на этот множитель. Это относится ко всем командам записи (например SET_WRITE_PROTECT, CLR_WRITE_PROTECT, PROGRAM_CSD и команды блочной записи).

Карты повышенной емкости (High Capacity SD Memory Card, SDHC) показывают R2W_FACTOR как фиксированное значение. Максимальная длительность занятости определена как 250 мс для всех операций записи. Хост должен использовать таймаут 250 мс (минимум) для операций одиночной и множественной записи вместо использования множителя R2W_FACTOR.

4.6.2.3 Erase (очистка). Если карта поддерживает параметры для вычисления таймаута очистки в статусе карты (SD Status), то хост должен использовать их для определения таймаута erase (см. главу 4.10.2). Если карта не поддерживает эти параметры, таймаут erase может быть оценен по задержке записи блока. Длительность выполнения команды очистки может быть оценена по количеству записываемых блоков (WRITE_BL) для очистки, умноженному на 250 мс.

4.7 Команды

4.7.1 Типы команд. Имеется 4 вида команд, определенных для управления картой памяти SD:

- Широковещательные команды без response (Broadcast command, сокращенно bc) - функция broadcast будет работать для всех карт только в том случае, когда все сигналы CMD карт параллельно подключены к хосту. Если же сигналы CMD раздельные, то каждая карта получит команду отдельно от других.
- Широковещательные команды с response (Broadcast commands with response, сокращенно bcr) - ответ приходит от всех карт одновременно. Поскольку на картах SD нет режима для выхода с открытым стоком, этот тип команд должен использоваться только если все сигналы CMD отделены друг от друга - команды будут приниматься каждой картой по отдельности, и карты будут по отдельности отвечать на каждую полученную команду.
- Адресованные (точка-точка) команды без передачи данных по сигналам DAT (Addressed commands, сокращенно ac).
- Адресованные (точка-точка) команды с передачей данных по сигналам DAT (Addressed data transfer commands, сокращенно adtc).

Все команды и ответы на них посылаются через сигнал CMD карты памяти SD. Передача команды всегда начинается с левого (старшего, MSB) бита строки бит, соответствующей кодовому слову команды.

4.7.2 Формат команды. У всех команд длина кодового слова фиксирована и составляет 48 бит, и необходимое время для её передачи равно 1.92 мкс на тактовой частоте 25 МГц и 0.96 мкс на 50 МГц.

Таблица 4-16. Формат строки бит команды.

Позиции бит	47	46	[45:40]	[39:8]	[7:1]	0
Ширина (в битах)	1	1	6	32	7	1
Значение поля	0	1	x	x	x	1
Описание	start bit	Бит передачи	Индекс команды	Аргумент	CRC7	end bit

Все команды начинаются со start-бита (всегда равен лог. 0), за которым идет бит направления передачи (для хоста лог. 1). Следующие 6 бит показывают индекс команды, это значение будет интерпретировано как двоичное число (от 0 до 63). Некоторые команды требуют аргумента (например адрес), который кодируется 32 битами. Значение, обозначенное как 'x' в таблице 4-16, соответствует изменяемой величине, зависящей от команды. Все команды защищены CRC (см. Главу 4.5 для определения CRC7). Каждое кодовое слово команды завершается end-битом (всегда равен лог. 1). Все команды и их аргументы перечислены в таблицах 4-18 .. 4-27.

4.7.3 Классы команд. Набор системы команд SD Memory Card поделен на несколько классов (см. таблицу 4-17). Каждый класс поддерживает определенный набор функций карты.

Таблица 4-17 определяет установку Card Command Classes (CCC) от команд, поддерживаемых картой. Бит CCC, который соответствует поддерживаемому номеру команды, устанавливается в 1. Класс в CCC включает обязательные команды, которые всегда устанавливают этот бит в лог. 1. Карты со специальными функциями могут нуждаться в поддержке некоторых опциональных команд. Например, Combo Card должна поддерживать CMD5.

Классы Class 0, Class 2, Class 4, Class 5, Class 7 и Class 8 являются обязательными, и они должны поддерживаться всеми картами памяти SD. Поддержка других классов опциональна (не является обязательной). Поддерживаемые классы команд (CCC) кодируются как параметр в регистре специфичных данных карты (card specific data, CSD) каждой карты, предоставляя хосту информацию о том, как получить доступ к карте.

Таблица 4-17. Классы команд карты (Card Command Classes, CCC).

	CCC	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Поддерживаемые команды	**Описание класса**(5)	B	res	BR	res	BW	E	WP	LC	APP	I/O	SW	res
CMD0	Обязательно	+											
CMD2	Обязательно	+											
CMD3	Обязательно	+											
CMD4	Обязательно	+											
CMD5	Опционально										+		
CMD6(2)	Обязательно											+	
CMD7	Обязательно	+											
CMD8(3)	Обязательно	+											
CMD9	Обязательно	+											
CMD10	Обязательно	+											
CMD12	Обязательно	+											
CMD13	Обязательно	+											
CMD15	Обязательно	+											
CMD16	Обязательно			+		+			+				
CMD17	Обязательно			+									
CMD18	Обязательно			+									
CMD24	Обязательно(1)					+							
CMD25	Обязательно(1)					+							
CMD27	Обязательно(1)					+							
CMD28	Опционально							+					
CMD29	Опционально							+					
CMD30	Опционально							+					
CMD32	Обязательно(1)						+						
CMD33	Обязательно(1)						+						
CMD34..CMD37(2)	Опционально											+	
CMD38	Обязательно(1)						+						
CMD42	Обязательно(4)								+				
CMD50(2)	Опционально											+	
CMD52	Опционально										+		
CMD53	Опционально										+		
CMD55	Обязательно									+			
CMD56	Обязательно									+			
CMD57(2)	Опционально											+	

Примечания к таблице 4-17:

(1) Команды, связанные с записью, являются обязательными только для карт записываемого типа (OTP и R/W).
(2) Эта команда была определена в стандарте Version 1.10.
(3) Эта команда была определена в стандарте Version 2.00.
(4) Эта команда опциональна в Version 1.01 и 1.10, и обязательна начиная с Version 2.00.
(5) Мнемоника для описания класса:
B - Basic, базовый класс команд
res - зарезервировано
BR - Block Read
BW - Block Write
E - Erase
WP - Write Protection
LC - Lock Card
APP - Application Specific
I/O - I/O mode
SW - Switch

4.7.4 Подробное описание команд. В таблицах, приведенных далее, дано детальное описание всех команд шины SD Memory Card. Ответы (response) R1..R3, R6 определены в Главе 4.9. Регистры CID, CSD и DSR определены в Главе 5. Карта должна игнорировать stuff-биты и зарезервированные биты в аргументе.

Таблица 4-18. Базовые команды (Basic Commands, Class 0).

CMD Index	**Тип**	**Аргумент**	**Resp.**	**Аббревиатура**	**Описание**
CMD0	bc	[31:0] stuff-биты (все в 0)	-	GO_IDLE_STATE	Программный сброс карты (Software reset).
CMD1	Зарезервировано				
CMD2	bcr	[31:0] stuff-биты (все в 0)	R2	ALL_SEND_CID	Запрашивает у любой карты послать номер CID через сигнал CMD (ответит любая карта, которая подключена к хосту).
CMD3	bcr	[31:0] stuff-биты (все в 0)	R6	SEND_RELATIVE_ADDR	Запрашивает карту опубликовать новый относительный адрес (Relative Card Address, RCA).
CMD4	bc	[31:16] DSR [15:0] stuff-биты (все в 0)	-	SET_DSR	Программирует DSP у всех карт.
CMD5	Зарезервировано для использования в I/O mode (см. "SDIO Card Specification")				
CMD7	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	R1b	SELECT_CARD/ DESELECT_CARD	Команда переключает карту между состояниями Standby и Transfer или между состоянием "программирование" и "отключено". В обоих случаях карта выбирается по своему RCA-адресу, и становится не выбранной по любому другому адресу; адрес 0 отменяет выбор всех карт. Когда RCA равен 0, хост может выполнить следующие действия: - Использование другого номера RCA чтобы выполнить отмену карты. - Повторная отправка CMD3 для смены номера RCA с номером RCA, не равным 0, и затем использование CMD7 с номером RCA=0 для отмены выбора карты.
CMD8	bcr	[31:12] зарезервированы [11:8] напряжение питания (VHS) [7:0] маска проверки (check pattern)	R7	SEND_IF_COND	Карте посылается информация об интерфейсе, включающая напряжение питания, и запрашивается у карты ответ - может ли она работать от этого напряжения. Зарезервированные биты должны быть заполнены нулями.
CMD9	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	R2	SEND_CSD	Чтение регистра CSD. Адресованная карта пошлет свои специфические данные (Card Specific Data, CSD) через сигнал CMD.
CMD10	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	R2	SEND_CID	Чтение регистра CID. Адресованная карта пошлет свои идентификационные данные (Card Identification Data, CID) через сигнал CMD.
CMD11	Зарезервировано				
CMD12	ac	[31::0] stuff-биты (все в 0)	R1b	STOP_TRANSMISSION	Принудительная остановка передачи данных.
CMD13	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	R1	SEND_STATUS	Запрашивает выбранную карту послать её регистр статуса.
CMD14	Зарезервировано				
CMD15	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	-	GO_INACTIVE_STATE	Переводит адресованную карту в неактивное состояние (Inactive State). Эта команда используется, когда хост явно хочет деактивировать карту. Зарезервированные биты должны быть установлены в 0.

Таблица 4-19. Блочные команды чтения (Block-Oriented Read Commands, Class 2).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD16	ac	[31:0] длина блока	R1	SET_BLOCKLEN	В случае карты стандартной емкости (Standard Capacity) эта команда меняет размер длины блока (в байтах) последующих блочных команд (чтения, записи, блокировки). Длина блока по умолчанию составляет 512 байт. Установка длины блока допустима только тогда, когда разрешены операции частичного доступа в регистре CSD. В случае карт High Capacity (SDHC) длина блока, установленная командой CMD16, никак не влияет на поведение команд чтения и записи, длина блока для этих карт фиксирована на размере 512 байт. CMD16 эффективна для команд LOCK_UNLOCK с картами обоих емкостей (стандартной емкости и SDHC). В обоих случаях, если длина блока установлена этой командой больше 512 байт, то карта установит бит ошибки BLOCK_LEN_ERROR.
CMD17	adtc	[31:0] адрес данных(2)	R1	READ_SINGLE_BLOCK	В случае карты стандартной емкости (Standard Capacity) эта команда читает блок данных того размера (в байтах), какой был установлен командой SET_BLOCKLEN(1). В случае карт High Capacity (SDHC) длина блока фиксирована на 512 байтах, независимо от длины блока, установленной командой SET_BLOCKLEN.
CMD18	adtc	[31:0] адрес данных(2)	R1	READ_MULTIPLE_BLOCK	Постоянное чтение друг за другом блоков, пока этот процесс не будет остановлен командой STOP_TRANSMISSION хоста. Длина одного блока будет такой же, как в описании команды READ_SINGLE_BLOCK.
CMD19.. CMD23	Зарезервировано				

Примечания к таблице 4-19:

(1) Передаваемые данные не должны пересекать границу физического блока, за исключением случая, когда установлен бит READ_BLK_MISALIGN в регистре CSD.
(2) Адрес данных байтовый для карт обычной емкости (Standard Capacity SD Memory Card, которые встречаются все реже и реже) и блочный (адрес это номер блока размером 512 байт) для карт повышенной емкости (High Capacity SD Memory Card, SDHC).

Таблица 4-20. Блочные команды записи (Block-Oriented Write Commands, Class 4).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD16	ac	[31:0] длина блока	R1	SET_BLOCKLEN	См. описание этой команды в таблице 4-19.
CMD24	adtc	[31:0] адрес данных(2)	R1	WRITE_BLOCK	В случае карты стандартной емкости (Standard Capacity) эта команда записывает блок размера, установленного командой SET_BLOCKLEN(1). В случае карт повышенной емкости (High Capacity, SDHC) длина блока остается фиксированной на 512 байтах, независимо от команды SET_BLOCKLEN.
CMD25	adtc	[31:0] адрес данных(2)	R1	WRITE_MULTIPLE_BLOCK	Постоянная запись друг за другом блоков, пока не будет отправлена STOP_TRANSMISSION. Длина блока такая же, как указана в описании команды WRITE_BLOCK.
CMD26	Зарезервировано для производителя				
CMD27	adtc	[31:0] stuff-биты (все в 0)	R1	PROGRAM_CSD	Записывает программируемые биты регистра CSD.

Примечания к таблице 4-20:

(1) Передаваемые данные не должны пересекать границу физического блока, за исключением случая, когда установлен бит WRITE_BLK_MISALIGN в регистре CSD. В случае, когда частичная запись блоков не поддерживается, длина блока равна длине блока по умолчанию (дается в регистре CSD).
(2) Адрес данных байтовый для карт обычной емкости (Standard Capacity SD Memory Card, которые встречаются все реже и реже) и блочный (адрес это номер блока размером 512 байт) для карт повышенной емкости (High Capacity SD Memory Card, SDHC).

Таблица 4-21. Команды для защиты блоков от записи (Block Oriented Write Protection Commands, Class 6).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD28	ac	[31:0] адрес данных(2)	R1b	SET_WRITE_PROT	Если карта поддерживает функции защиты от записи, эта команда устанавливает бит защиты от записи для адресованной группы. Свойства защиты от записи закодированы в специфических данных карты (регистр CSD, поле WP_GRP_SIZE). Карты High Capacity (SDHC) не поддерживают эту команду.
CMD29	ac	[31:0] адрес данных(2)	R1b	CLR_WRITE_PROT	Если карта поддерживает функции защиты от записи, эта команда очищает бит защиты от записи для адресованной группы. Карты High Capacity (SDHC) не поддерживают эту команду.
CMD30	adtc	[31:0] адрес защищенных от записи данных(2)	R1	SEND_WRITE_PROT	Если карта поддерживает функции защиты от записи, эта команда запросит у карты послать состояние бит, определяющих защиту от записи(1). Карты High Capacity (SDHC) не поддерживают эту команду.
CMD31	Зарезервировано				

1) 32 бита, устанавливающие защиту от записи (write protection bits, WP биты) представляют 32 группы блоков, защищенных от записи. Каждая группа определена по начальному и конечному адресу. За 32 битами WP идет 16-битная CRC, передаваемая в формате полезной нагрузки по сигналам данных DAT. Последний (наименее значащий, LSB) бит WP соответствует первой адресованной группе. Если адреса последних адресованных групп выходят за допустимый диапазон, то соответствующие биты WP должны быть установлены в 0.

2) Адрес данных на карте стандартной емкости (Standard Capacity SD Memory Card) байтовый. Карты повышенной емкости (High Capacity SD Memory Card, SDHC) не поддерживают защиту блоков от записи.

Таблица 4-22. Команды очистки данных (Erase Commands, Class 5).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD32	ac	[31:0] адрес данных(1)	R1	ERASE_WR_BLK_START_ADDR	Устанавливает адрес первого записываемого блока, который должен быть очищен.
CMD33	ac	[31:0] адрес данных(1)	R1	ERASE_WR_BLK_END_ADDR	Устанавливает адрес последнего записываемого блока, который должен быть очищен.
CMD38	ac	[31:0] stuff-биты (все в 0)	R1b	ERASE	Стирает все ранее выбранные записываемые блоки.
CMD39	Зарезервировано				
CMD40					Недопустимая команда для карт памяти SD - зарезервировано для MultiMediaCard I/O mode.
CMD41	Зарезервировано				

Примечание к таблице 4-22: (1) адрес данных байтовый для карт стандартной емкости (Standard Capacity SD Memory Card) и блочный (в номерах блоков по 512 байт) для карт повышенной емкости (High Capacity SD Memory Card, SDHC).

Таблица 4-23. Команды блокировки карты (Lock Card, Class 7).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD16	ac	[31:0] длина блока	R1	SET_BLOCKLEN	См. описание этой команды в таблице 4-19.
CMD42	adtc	[31:0] зарезервированные биты (все должны быть в 0)	R1	LOCK_UNLOCK	Используется для установки/снятия пароля или блокировки/разблокировки карты. Размер блока данных определяется командой SET_BLOCK_LEN. Зарезервированные биты в аргументе и структуре Lock Card Data должны быть установлены в 0.
CMD43.. CMD49, CMD51	Зарезервировано				

Таблица 4-24. Команды, относящиеся к особенностям приложения (Application-specific Commands, Class 8).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD55	ac	[31:16] RCA [15:0] stuff-биты (все в 0)	R1	APP_CMD	Определяет для карты следующую команду как команду специфичную для приложения (ACMD), а не как обычную команду (CMD).
CMD56	adtc	[31:1] stuff-биты (все в 0) [0] RD/WR	R1	GEN_CMD	Используется либо для передачи блока данных в карту, либо для получения блока данных от карты для обычных команд и для команд, специфичных для приложения. В случае карты памяти Standard Capacity размер блока данных должен быть определен командой SET_BLOCK_LEN. Для карт High Capacity (SDHC) размер блока данных для этой команды фиксирован на 512 байтах. Хост устанавливает RD/WR=1 для чтения данных из карты, и RD/WR=0 для записи данных в карту.
CMD58, CMD59	Зарезервировано				
CMD60.. CMD63	Зарезервировано для производителя карт.				

Все команды, специфичные для приложения (определенные в таблице 4-24), будут поддерживаться, если разрешен Class 8 (обязательное условие для SD Memory Card).

Таблица 4-25. Команды режима ввода/вывода (I/O Mode Commands, Class 9).

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD52.. CMD54	Зарезервировано для I/O Mode (см. "SDIO Card Specification").				

Все команды, зарезервированные для использования в будущем, должны иметь длину кодового слова 48 бит, как и их ответы response (если ответ на команду предусмотрен).

Команды ACMD. Для расширения системы команд придумана категория команд, специфичных для приложения (application-specific commands). Следующая таблица описывает все application-specific команды, поддерживаемые/зарезервированные для карт памяти SD. Всем application-specific командам обязательно должна предшествовать команда APP_CMD (CMD55), за которой идут команды, которые будут интерпретироваться картой по-другому. Эти application-specific команды получили в своей мнемонике префикс A (команды обозначаются как ACMDxx, где xx индекс команды), и всем shall be preceded with APP_CMD command (CMD55).

Таблица 4-26. Application Specific Commands, используемые/зарезервированные для SD Memory Card.

CMD Index	SPI	Аргумент	Resp.	Аббревиатура	Описание
ACMD6	ac	[31:2] stuff-биты [1:0] ширина шины	R1	SET_BUS_WIDTH	Определяет ширину шины данных DAT (00b означает 1 бит, 10b означает 4 бита) для передачи данных. Разрешенные ширины шины данных даны в регистре SCR.
ACMD13	adtc	[31:0] stuff-биты	R2	SD_STATUS	Отправка SD Status. Поля битов статуса показаны в таблице 4-37.
ACMD17	Зарезервировано				
ACMD18	--	--	--	--	Зарезервировано для приложений безопасности SD(1).
ACMD19.. ACMD21	Зарезервировано				
ACMD22	adtc	[31:0] stuff-биты	R1	SEND_NUM_WR_BLOCKS	Отправка количества успешно записанных (без ошибок) блоков. Отвечает блоком данных из 32 бит + CRC. Если WRITE_BL_PARTIAL=0, то единица записываемых данных всегда 512 байт. Если WRITE_BL_PARTIAL=1, то длина блока будет такой, как была длина блока в выполненной команде записи блока.
ACMD23	ac	[31:23] stuff-биты [22:0] количество блоков	R1	SET_WR_BLK_ERASE_COUNT	Устанавливает количество записываемых блоков, которые должны быть предварительно очищены перед записью (для ускорения использования команды Multiple Block Write). Установка по умолчанию 1 (один записываемый блок)(2).
ACMD24	Зарезервировано				
ACMD25, ACMD26, ACMD38	--	--	--	--	Зарезервировано для приложений безопасности SD(1).
ACMD39.. ACMD40	Зарезервировано				
ACMD41	bcr	[31] зарезервированный бит [30] HCS (OCR[30]) [29:24] зарезервированные биты [23:0] Окно напряжения V_{DD} (OCR[23:0])	R3	SD_SEND_OP_COND	Отправляет хосту информацию о поддержке емкости и запрашивает адресованную карту отправить свой регистр OCR в ответе, переданном через сигнал CMD. HCS эффективен, когда карта принимает команду SEND_IF_COND. Зарезервированные биты должны быть установлены в 0. Бит CCS назначается на бит OCR[30].
ACMD42	ac	[31:1] stuff-биты [0] set_cd	R1	SET_CLR_CARD_DETECT	Задает подключить (1) / отключить (0) резистор 50 кОм pull-up на выводе выборки карты CS/DAT3 (вывод 1).
ACMD43.. ACMD49	--	--	--	--	Зарезервировано для приложений безопасности SD(1).
ACMD51	adtc	[31:0] stuff-биты	R1	SEND_SCR	Читает регистр конфигурации карты (SCR).

Примечания к таблице 4-26:

(1) См. "SD Specifications Part3 Security Specification" для подробного описания функций безопасности карт SD (SD Security Features).
(2) Команда STOP_TRAN (CMD12) должна использоваться для остановки всех многоблочных передач записи (Write Multiple Block), независимо от того, использовалась или нет функция предварительной очистки (preerase, ACMD23).

Таблица 4-27. Команды переключения функций (Switch Function Commands, Class 10). Это было добавлено в Version 1.10.

CMD Index	Тип	Аргумент	Resp.	Аббревиатура	Описание
CMD6	adtc	[31] Mode 0: функция проверки 1: функция переключения [30:24] зарезервировано (все 0) [23:20] зарезервировано для функциональной группы 6 (0h.. Fh) [19:16] зарезервировано для функциональной группы 5 (0h..Fh) [15:12] зарезервировано для функциональной группы 4 (0h..Fh) [11:8] зарезервировано для функциональной группы 3 (0h..Fh) [7:4] функциональная группа 2 для системы команд [3:0] функциональная группа 1 для режима доступа	R1	SWITCH_FUNC	Проверяет переключаемую функцию (Mode 0) и переключает функцию карты (Mode 1). См. главу 4.3.10.
CMD34.. CMD37, CMD50, CMD57	Зарезервировано для каждой системы команд, устанавливаемой командой переключения функции (CMD6). Подробнее это описано в спецификации для каждой системы команд.				

4.8 Таблица переходов состояний карты (Card State Transition). В таблице 4-28 определены переходы между состояниями карты в зависимости от принятой команды.

Таблица 4-28. Card State Transition.

	Текущее состояние									
	idle	ready	ident	stby	tran	data	rcv	prg	dis	ina
Триггер изменения состояния	следующее состояние									
Независимо от класса										
"Операция завершена"	-	-	-	-	-	-	-	tran	stdby	-
Class 0										
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	-
CMD2	-	ident	-	-	-	-	-	-	-	-
CMD3	-	-	stdby	stdby	-	-	-	-	-	-
CMD4	-	-	-	stdby	-	-	-	-	-	-
CMD7, карта адресована	-	-	-	tran	-	-	-	-	prg	-
CMD7, карта не адресована	-	-	-	stdby	stby	stby	-	dis	-	-
CMD8	idle	-	-	-	-	-	-	-	-	-
CMD9	-	-	-	stdby	-	-	-	-	-	-
CMD10	-	-	-	stdby	-	-	-	-	-	-
CMD12	-	-	-	-	-	tran	prg	-	-	-
CMD13	-	-	-	stby	tran	data	rcv	prg	dis	-
CMD15	-	-	-	ina	ina	ina	ina	ina	ina	-
Class 2										
CMD16	-	-	-	-	tran	-	-	-	-	-
CMD17	-	-	-	-	data	-	-	-	-	-
CMD18	-	-	-	-	data	-	-	-	-	-
Class 4										
CMD16	-	-	-	-	tran	-	-	-	-	-
CMD24	-	-	-	-	rcv	-	-	-	-	-
CMD25	-	-	-	-	rcv	-	-	-	-	-
CMD27	-	-	-	-	rcv	-	-	-	-	-
Class 6										
CMD28	-	-	-	-	prg	-	-	-	-	-
CMD29	-	-	-	-	prg	-	-	-	-	-
CMD30	-	-	-	-	data	-	-	-	-	-
Class 5										
CMD32	-	-	-	-	tran	-	-	-	-	-
CMD33	-	-	-	-	tran	-	-	-	-	-
CMD38	-	-	-	-	prg	-	-	-	-	-
Class 7										
CMD42	-	-	-	-	rcv	-	-	-	-	-
Class 8										
CMD55	idle	-	-	stdby	tran	data	rcv	prg	dis	-
CMD56; RD/WR = 0	-	-	-	-	rcv	-	-	-	-	-
CMD56; RD/WR = 1	-	-	-	-	data	-	-	-	-	-
ACMD6	-	-	-	-	tran	-	-	-	-	-
ACMD13	-	-	-	-	data	-	-	-	-	-
ACMD22	-	-	-	-	data	-	-	-	-	-
ACMD23	-	-	-	-	tran	-	-	-	-	-
ACMD18, ACMD25, ACMD26, ACMD38, ACMD43, ACMD44, ACMD45, ACMD46, ACMD47, ACMD48, ACMD49	См. стандарт "SD Part3 Security Specification" для получения информации о функциях безопасности.									
ACMD41, проверка OCR ок и карта находится в состоянии готовности	ready	-	-	-	-	-	-	-	-	-
ACMD41, проверка OCR ок и карта находится в состоянии занятости(2)	idle	-	-	-	-	-	-	-	-	-
ACMD41, проверка OCR неудачная	ina	-	-	-	-	-	-	-	-	-
ACMD41, режим опроса	idle	-	-	-	-	-	-	-	-	-
ACMD42	-	-	-	-	tran	-	-	-	-	-
ACMD51	-	-	-	-	data	-	-	-	-	-
Class 9										
CMD52..CMD54	См. стандарт "SDIO Card Specification".									
Class 10(1)										
CMD6	-	-	-	-	data	-	-	-	-	-
CMD34..CMD37, CMD50, CMD57	См. спецификации по каждой функции.									
Class 11										
CMD41, CMD43...CMD49, CMD58, CMD59	Зарезервировано									
CMD60..CMD63	Зарезервировано для производителя карт									

Примечания к таблице 4-28:

(1). Class 10 команд был определен в стандарте Version 1.10.
(2). Карта вернет состояние занятости (busy) в следующих случаях:
- карта выполняет внутренний процесс инициализации.
- карта типа High Capacity SD Memory (SDHC), и хост не поддерживает карты High Capacity.

Переходы между состояниями карты памяти SD для application-specific команд были даны выше под классом Class 8.

4.9 Ответы карты на команду (Response). Все ответы карты посылаются через сигнал CMD (это не относится к режиму подключения карты через SPI [2]). Передача потока бит ответа всегда начинается с левого (старшего) бита соответствующего ответа на кодовое слово. Длина потока бит ответа зависит от типа ответа. Начальный бит является стартовым (он всегда равен 0), за ним идет бит, обозначающий направление передачи (для карты он равен 0). Значения, показанные как X в таблицах ниже, показывают изменяемое поле бит. Все типы ответа, кроме R3, защищены контрольной суммой CRC7. Каждое кодовое слово команды заканчивается битом окончания, end-бит (всегда равен 1).

Имеется 5 типов ответа для SD Memory Card. Карты типа SDIO Card поддерживают дополнительные типы ответа, обозначенные как R4 и R5. За подробностями по этим типам ответов и соответствующим командам см. спецификацию по картам SDIO. Ниже приведено описание ответов различного типа.

4.9.1 R1 (обычный ответ на команду). Длина этот типа ответа составляет 48 бит (6 байт). Биты 45:40 показывают индекс команды, на которую выполнен ответ, это значение интерпретируется как двоичное число (значение от 0 до 63). Состояние карты (статус) кодируется в 32 битах (описание статуса карты см. во врезке "Card Status"). Обратите внимание, что если вовлечена передача данных в карту, то может появиться сигнал занятости на линии данных после передачи каждого блока данных (касается native-подключения карты). Хост должен проверять занятость после передачи каждого блока.

Таблица 4-29. Response R1.

Позиции бит	47	46	[45:40]	[39:8]	[7:1]	0
Ширина (в битах)	1	1	6	32	7	1
Значение поля	0	0	x	x	x	1
Описание	start bit	Бит передачи	Индекс команды	Статус карты	CRC7	end bit

4.9.2 R1b. Ответ R1b идентичен ответу R1 с опциональным сигналом занятости, передаваемым по линии данных выхода карты. Карта может оказаться занятой внутренней операцией после получения команд, в зависимости от её состояния перед приемом команды. Хост должен проверять занятость карты на каждом ответе карты.

4.9.3 R2 (содержимое регистров CID, CSD). Длина этого ответа составляет 136 бит. Содержимое регистра CID (идентификатор карты) посылается в ответ на команды CMD2 и CMD10. Содержимое регистра CSD (параметры карты) передается в ответ на команду CMD9. Передаются только биты [127...1] регистров CID и CSD, зарезервированный бит [0] этих регистров заменяется конечным битом ответа.

Таблица 4-30. Response R2.

Позиции бит	135	134	[133:128]	[127:1]	0
Ширина (в битах)	1	1	6	127	1
Значение поля	0	0	111111	x	1
Описание	start bit	Бит передачи	Зарезервировано	Содержание регистр CID или CSD, включая внутреннюю CRC7	end bit

4.9.4 R3 (регистр OCR). Длина этого ответа составляет 48 бит. Содержимое регистра OCR (operation conditions register) посылается в ответ на команду ACMD41.

Таблица 4-31. Response R3.

Позиции бит	47	46	[45:40]	[39:8]	[7:1]	0
Ширина (в битах)	1	1	6	32	7	1
Значение поля	0	0	111111	x	1111111	1
Описание	start bit	Бит передачи	Зарезервировано	Регистр OCR	Зарезервировано	end bit

4.9.5 R6 (Published RCA). Для карты, подключенной через SPI, этот тип ответа не используется, потому что в этом режиме подключения команда CMD3 не поддерживается.

Длина ответа R6 составляет 48 бит. Биты 45:40 показывают индекс команды, на которую произведен ответ - в этом случае индекс будет равен 000011 (вместе с битом 5 битов статуса это будет означать = CMD3). 16 MSB старших бит в поле аргумента (Argument) используются для передачи номера Published RCA.

Таблица 4-32. Response R6.

Позиции бит	47	46	[45:40]	[39:8] Поле аргумента		[7:1]	0
Ширина (в битах)	1	1	6	16	16	7	1
Значение поля	0	0	x	x	x	x	1
Описание	start bit	Бит передачи	Индекс команды (000011)	Новый опубликованный RCA [31:15] карты	[15:0] биты статуса карты: 23, 22, 19, 12:0 (см. табл. 4-35)	CRC7	end bit

4.9.6 R7 (Card interface condition). Это состояние интерфейса карты, длина ответа 48 бит. Поддерживаемое напряжение карты посылается в ответ на команду CMD8. Биты 19..16 показывают диапазон рабочих напряжений, которые может поддерживать карта. Карта, которая приняла предоставленное напряжение, вернет ответ R7. В этом ответе карта эхом передаст обратно диапазон напряжения и маску битов проверки (check pattern), переданный в аргументе команды.

Таблица 4-33. Response R7.

Позиции бит	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Ширина (в битах)	1	1	6	20	4	8	7	1
Значение поля	0	0	001000	00000h	X	X	X	1
Описание	start bit	Бит передачи	Индекс команды	Зарезервировано	Voltage Accepted	Обратное эхо check pattern	CRC7	end bit

В таблице 4-34 показан формат поля 'voltage accepted' в R7.

Таблица 4-34. Поле Voltage Accepted в ответе R7.

Voltage Accepted	Значения напряжения
0000b	Не определено
0001b	2.7 - 3.6V
0010b	Зарезервировано для диапазона низких напряжений питания
0100b 1000b	Зарезервировано
Другие значения	Не определено

4.10 Два вида информации статуса SD Memory Card. Карты памяти SD поддерживают следующие 2 поля статуса:

Card Status: информация об ошибке и статусе выполненной команды, показанная в response.
SD Status: поле расширенного статуса из 512 бит, которая поддерживает специальные функции SD Memory Card и будущие Application-Specific функции.

4.10.1 Card Status. Ответ формата R1 содержит 32-битное поле, которое называется card status (состояние карты). Это поле предназначено для передачи информации о состоянии карты (которая может быть сохранена в локальном регистре статуса) к хосту. Если не указано нечто иное, записи в статусе всегда относятся к ранее выданной команде. Таблица 4-35 определяет различные записи статуса. Поля типа (Type) и условия очистки (Clear Condition) обозначены сокращениями следующим образом:

Type (тип)

E: бит ошибки (Error).
S: бит статуса (Status).
R: детектирована и установлена информация для актуального ответа на команду.
X: детектирована и установлена информация во время выполнения команды. Хост может получить статус выдачей команды с ответом R1.

Clear Condition (условие очистки)

A: информация соответствует текущему состоянию карты.
B: всегда относится к предыдущей команде. Прием допустимой команды очистит это (с задержкой на одну команду).
C: очищено чтением.

Таблица 4-35. Card Status.

Биты	Идентификатор	Type	Значение	Описание	Clear Condition
31	OUT_OF_RANGE	E R X	0: нет ошибки, 1: ошибка	Аргумент команды вышел за пределы допустимого диапазона для этой карты.	C
30	ADDRESS_ERROR	E R X	0: нет ошибки, 1: ошибка	Ошибочно выровненный адрес, который не соответствует длине блока, который использовался в команде.	C
29	BLOCK_LEN_ERROR	E R X	0: нет ошибки, 1: ошибка	Длина передаваемого блока не допустима для этой карты, или количество передаваемых байт не соответствует длине блока.	C
28	ERASE_SEQ_ERROR	E R	0: нет ошибки, 1: ошибка	Произошла ошибка в последовательности команд стирания.	C
27	ERASE_PARAM	E R X	0: нет ошибки, 1: ошибка	Произошел недопустимый выбор записываемых блоков для стирания.	C
26	WP_VIOLATION	E R X	0: не защищена, 1: защищена	Устанавливается, когда хост попытался выполнить запись в защищенный блок либо на временно защищенную от записи, либо на постоянно защищенную от записи карту.	C
25	CARD_IS_LOCKED	S X	0: карта не заблокирована, 1: карта заблокирована	Если установлен, то сигнализирует, что карта заблокирована хостом.	A
24	LOCK_UNLOCK_FAILED	E R X	0: нет ошибки, 1: ошибка	Устанавливается, когда была детектирована ошибка пароля в команде блокировки/разблокировки карты.	C
23	COM_CRC_ERROR	E R	0: нет ошибки, 1: ошибка	Ошибка CRC предыдущей команды.	B
22	ILLEGAL_COMMAND	E R	0: нет ошибки, 1: ошибка	Команда недопустима для текущего состояния карты.	B
21	CARD_ECC_FAILED	E R X	0: успешное завершение, 1: ошибка	Была применена внутренняя ECC, но произошла ошибка для корректных данных.	C
20	CC_ERROR	E R X	0: нет ошибки, 1: ошибка	Ошибка внутреннего контроллера карты.	C
19	ERROR	E R X	0: нет ошибки, 1: ошибка	Во время выполнения операции произошла общая или неизвестная ошибка.	C
18 17	Зарезервировано				
16	CSD_OVERWRITE	E R X	0: нет ошибки, 1: ошибка	Произошла одна из следующих ошибок: - Секция только для чтения CSD не соответствует содержимому карты. - Попытка реверса копирования (копирование в место источника), или ошибка защиты от записи.	C
15	WP_ERASE_SKIP	E R X	0: нет защиты от записи, 1: защита от записи	Устанавливается, когда была очищена только часть адресного пространства - из-за наличия защищенных от записи блоков, или очищалась карта, временно или постоянно защищенная от записи.	C
14	CARD_ECC_DISABLED	S X	0: разрешено, 1: запрещено	Была выполнена команда без внутреннего ECC.	A
13	ERASE_RESET	S R	0: очищено, 1: установлено	Была очищена последовательность стирания перед выполнением, потому что была принята команда выхода из последовательности стирания.	C
12:9	CURRENT_STATE	S X	0: idle 1: ready 2: ident 3: stby 4: tran 5: data 6: rcv 7: prg 8: dis 9..14:зарезервировано 15: зарезервировано для режима I/O	Состояние карты, когда принимается команда. Если выполнение команды приводит к изменению состояния карты, это увидит хост в ответ на следующую команду. Эти 4 бита интерпретируются как двоичное число со значением в диапазоне от 0 до 15.	B
8	READY_FOR_DATA	S X	0: нет готовности, 1: готовность к данным	Соответствует сигнализации по шине, что буфер пуст.	A
7:6	Зарезервировано				
5	APP_CMD	S R	0: запрещено, 1: разрешено	Карта ожидает ACMD, или показывается, что команда была интерпретирована как ACMD.	C
4	Зарезервировано для карты SD I/O.				
3	AKE_SEQ_ERROR (специфика приложения SD Memory Card)	E R	0: нет ошибки, 1: ошибка	Ошибка в последовательности аутентификации.	C
2	Зарезервировано для команд, специфичных для приложения.				
1:0	Зарезервировано для режима тестирования производителя.				

Для каждой команды, на которую приходит ответ R1, в следующей таблице определены биты в статусе, на которые влияет эта команда. Символ 'x' в таблице обозначает, что бит ошибки/статуса может быть установлен в ответ на соответствующую команду.

Таблица 4-36. Взаимосвязь между полями статуса карты и командами.

CMD#	Формат ответа R1, номера бит статуса.																					
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12:9	8	5
3(1)									x	x			x							x		
6(2)	x						x		x	x	x	x	x	x	x					x		
7					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
12	x	x				x	x		x	x	x	x	x	x	x			x		x		
13	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	
16			x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
17	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
18	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
24	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
25	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
26					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
27					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
28	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
29	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
30	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
32	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
33	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
38				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
55					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
56					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ACMD6	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD13					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD22					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD23					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD51					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x

Примечания к таблице 4-36:

(1) Ответ на команду CMD3 это ответ R6, который включает в себя только биты 23, 22, 19 и 12:9 из информации Card Status.
(2) Эта команда была определена в версии 1.10.

4.10.2 SD Status. SD Status содержит биты статуса, которые относятся к проприетарным функциям карт памяти SD, и они могут быть полезны для будущего использования в зависимости от приложения. Размер SD Status - один блок данных из 512 бит. Содержимое этого регистра передается хосту через шину данных (DAT) вместе с 16-битной CRC. SD Status отправляется к хосту по шине DAT как ответ на команду ACMD13 (CMD55, которая следует с CMD13). ACMD13 может быть отправлена в карту только в состоянии tran_state (карта выбрана). Ниже описана структура SD Status.

Для типа (Type) и условий очистки (Clear Condition) используются те же сокращения, какие были показаны во врезке "Card Status" выше.

Таблица 4-37. SD Status.

Биты	Идентификатор	Type	Значение	Описание	Clear Condition
511:510	DAT_BUS_WIDTH	S R	00: 1 (по умолчанию), 01: зарезервировано, 10: 4 бита, 11: зарезервировано	Показывает текущую определенную ширину шины, как это было задано командой SET_BUS_WIDTH.	A
509	SECURED_MODE	S R	0: не в режиме защиты, 1: в режиме защиты	Показывает, находится ли карта в защищенном режиме (см. SD Security Specification).	A
508:496	Зарезервировано				
495:480	SD_CARD_TYPE	S R	00xxh: карта SD Memory, как это определено в стандарте Ver1.01-2.00. В настоящее время определены следующие карты: 0000 это обычные карты SD RD/RW, 0001 это карты SD ROM.	В будущем 8 младших бит будет использовано для определения различных вариаций карт SD Memory (каждый бит определит разные типы SD). Старшие 8 бит будут использоваться для определения карт SD, которые не соответствуют текущей спецификации SD Physical Layer.	A
479:448	SIZE_OF_PROTECTED_AREA	S R	Размер защищенной области.	См. описание ниже.	A
447:440	SPEED_CLASS	S R	Класс скорости карты.	См. описание ниже.	A
439:432	PERFORMANCE_MOVE	S R	Быстродействие при перемещении данных в шагах по 1 мегабайту/сек.	См. описание ниже.	A
431:428	AU_SIZE	S R	Размер AU (единица данных, allocation unit).	См. описание ниже.	A
427:424	Зарезервировано				
423:408	ERASE_SIZE	S R	Количество AU для стирания за 1 операцию.	См. описание ниже.	A
407:402	ERASE_TIMEOUT	S R	Значение таймаута для очистки областей, указанных через UNIT_OF_ERASE_AU.	См. описание ниже.	A
401:400	ERASE_OFFSET	S R	Фиксированное значение смещения, добавляемое для времени стирания.	См. описание ниже.	A
399:312	Зарезервировано				
311:0	Зарезервировано для производителя.				

SIZE_OF_PROTECTED_AREA. Установка этого поля отличается для стандартных карт (Standard Capacity Card) и карт большой емкости (High Capacity Card). В случае Standard Capacity Card емкость защищенной области вычисляется следующим образом:

```
Protected Area (размер защищенной области) = SIZE_OF_PROTECTED_AREA_* MULT * BLOCK_LEN
```

Здесь SIZE_OF_PROTECTED_AREA указывается в единицах MULT*BLOCK_LEN.

В случае High Capacity Card емкость защищенной области указывается в этом поле так:

```
Protected Area (размер защищенной области) = SIZE_OF_PROTECTED_AREA
```

Здесь SIZE_OF_PROTECTED_AREA указывается в байтах.

SPEED_CLASS. Это 8-битное поле показывает класс скорости (Speed Class), и значение можно вычислить через $P_w/2$.

Таблица 4-38. Поле кода Speed Class.

SPEED_CLASS	Значения
00h	Class 0
01h	Class 2
02h	Class 4
03h	Class 6
04h..FFh	Зарезервировано

PERFORMANCE_MOVE. Это 8-битное поле показывает P_m, и это значение может быть установлено с шагом 1 [MB/sec]. Если карта не использует move RU, то P_m должно считаться бесконечностью. Установка в FFh означает бесконечность.

Таблица 4-39. Поле Performance Move.

PERFORMANCE_MOVE	Значения
00h	Не определено
01h	1 MB/сек
02h	2 MB/сек
...	...
FEh	254 MB/сек
FFh	Бесконечность

AU_SIZE. Это 4-битное поле показывает размер AU, и это значение может быть выбрано в степенях 2, умноженное на 16 KB.

Таблица 4-40. Поле AU_SIZE.

AU_SIZE	Значения
0h	Не определено
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB
8h	2 MB
9h	4MB
Ah..Fh	Зарезервировано

Максимальный размер AU зависит от емкости карты, что определено в таблице 4-41. Карта может установить любой размер AU между размером RU и максимальным размером AU.

Таблица 4-41. Максимальный размер AU.

Емкость	16..64 MB	128..256 MB	512 MB	1..32 GB
Максимальный размер AU	512 KB	1 MB	2 MB	4 MB

Указания по применению: хост должен использовать максимальный размер AU (4 MB), чтобы определить размер буфера хоста. Хост может обрабатывать несколько AU, скомбинированных в один юнит.

ERASE_SIZE. Это 16-битное поле показывает N_{ERASE}. Когда стерто N_{ERASE} AU, значение таймаута указывается через ERASE_TIMEOUT (см. ниже ERASE_TIMEOUT). Хост должен определить правильное количество AU для стирания, чтобы хост мог корректно отобразить состояние прогресса операции стирания. Если это поле установлено в 0, то вычисление таймаута стирания не поддерживается.

Таблица 4-42. Поле размера стирания (Erase Size).

ERASE_SIZE	Значения
0000h	Вычисление таймаута стирания не поддерживается.
0001h	1 AU
0002h	2 AU
0003h	3 AU
...	...
FFFFh	65535 AU

ERASE_TIMEOUT. Это 6-битное поле показывает время T_{ERASE}, и это значение означает таймаут стирания от смещения, когда стирается несколько AU, как задано ERASE_SIZE. Диапазон ERASE_TIMEOUT можно определить до 63 секунд, и производитель карты может выбрать любую комбинацию ERASE_SIZE и ERASE_TIMEOUT в зависимости от реализации карты. Как только был определен ERASE_TIMEOUT, он определяет ERASE_SIZE. Хост может определить таймаут для любого количества стираемых AU. Для вычисления таймаута стирания см. пункт 4.14 даташита [8]. Если поле ERASE_SIZE установлено в 0, тр это поле должно быть установлено в 0.

Таблица 4-43. Поле Erase Timeout.

ERASE_TIMEOUT	Значения
0	Вычисление таймаута стирания не поддерживается.
1	1 секунда
2	2 секунды
3	3 секунды
...	...
63	63 секунды

ERASE_OFFSET. Это 2-битное поле показывает смещение T_{OFFSET}, и можно выбрать одно из 4 значений. Смещение стирания подстраивает строку путем параллельного перемещения на верхнюю сторону. Подробнее см. рис. 4-11 и выражение (6) в пункте 4.14 даташита [8]. Это поле бесполезно, если поля ERASE_SIZE и ERASE_TIMEOUT установлены в 0.

Таблица 4-44. Поле Erase Offset.

ERASE_OFFSET	Значения
0	0 секунд
1	1 секунда
2	2 секунды
3	3 секунды

4.11 Деление массива памяти на разделы (Memory Array Partitioning). *В упрощенной спецификации описание этой секции опущено.*

4.12 Интервалы времени шины (Timings). *В упрощенной спецификации описание этой секции опущено.*

4.13 Спецификация классов скорости (Speed Class Specification). Классы скорости делят карты по производительности - насколько быстро хост может работать с данными карты. Класс скорости обозначается номером (Speed Class number), и он предоставляет метод вычисления производительности. Спецификация разрешает хосту поддерживать приложения AV (audio-video), чтобы выполнять запись на карту памяти SD в реальном времени. Следующие секции описывают спецификацию Speed Class для карты. См. указания по применению для получения примера реализации хоста.

4.13.1 Allocation Unit (AU). Область данных пользователя (User Area) поделена на блоки данных, которые называются "Allocation Unit" (сокращенно AU), см. рис. 4-8). Каждая карта имеет свой собственный фиксированный размер, AU Size (S_{AU}), и максимальный AU Size определяется емкостью карты. Хост должен управлять областями данных порциями размером в AU. Несколько AU из AU1 не должно использоваться для записи в реальном времени, потому что здесь может содержаться системная информация. Приложение AV (audio-video) должно начать запись с первого полного AU, на котором могут записываться только пользовательские данные. Обратите внимание, что эта спецификация не относится к защищенной области данных (Protected Area).

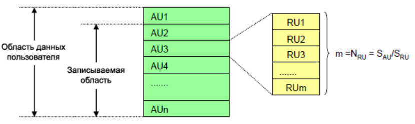


Рис. 4-8. Определение Allocation Unit (AU).

4.13.2 Recording Unit (RU). Каждый AU делится на элементы, которые называются единицами записи "Recording Unit (RU)" (см. рис. 4-8). Размер этой единицы RU Size (S_{RU}) должен быть кратен размеру кластера (один или несколько кластеров), как это указано в спецификации на файловую систему карты. Количество RU в одном AU (N_{RU}) вычисляется из частного S_{AU}/S_{RU}.

4.13.3 Производительность при записи (Write Performance). *В упрощенной спецификации описание этой секции опущено.*

4.13.4 Производительность при чтении (Read Performance). *В упрощенной спецификации описание этой секции опущено.*

4.13.5 Определение кривой производительности (Performance Curve Definition). *В упрощенной спецификации опущена часть этой секции.*

Соотношение, используемое RU(r), определено следующим образом:

$$r = \frac{Nu}{N_{RU}}, \qquad Nu = r*N_{RU}$$

Диапазон r от 0 до 1. (1 – r) означает соотношение свободных RU, r=0 означает, что все RU свободны. r=1 означает, что все RU используются, и производительность в этом месте показывает ноль.

4.13.6 Определение Speed Class. Рис. 4-9 показывает 3 кривых производительности. Pw показывает производительность r=0, и Pm определяет форму кривой. Все кривые производительности сходятся в точке (1, 0). Таким образом, имеются незначительные различия в производительности, когда r близок к 1. Эти 3 кривые делят производительности на 4 класса скорости: Class 0, Class 2, Class4 и Class 6. Class 0 не предоставляет гарантии совместимости со стандартом Speed Class Specification. Он не сообщает о параметрах производительности, даже если карты могут достичь производительности высших классов скорости. Class 0 также покрывает все старые (legacy) виды карт SD, чтобы они были представлены в этой спецификации. Классы, определенные как приложения AV, такие как запись MPEG2, могут поддерживать устройства карт SD. Производительность карты Speed Class 2 должна быть выше кривой 2. Это определено для качества изображения стандартного TV, т. е. удовлетворяются его требования приблизительно 2 мегабайта/сек. Производительность Speed Class 4 будет выше, чем кривая производительности 4. Speed Class 4 определен как соответствующий классу качества картинки HD video; удовлетворяется требование производительности 4 мегабайта/сек. Высшие классы могут быть добавлены в будущем, если это необходимо. Важно, чтобы хост всегда принимал карты, удовлетворяющие производительности минимального класса скорости.

Speed Class определяется на основе производительности интерфейса шины SD, хотя кривая производительности вычисляется только по конечной полученной производительности из анализа в 4.13.3. Частота тактов и размер RU определены на условиях измерений для классов скорости (см. секцию 4.13.8).

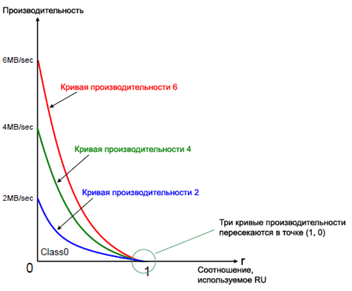


Рис. 4-9. Кривые производительности классов скорости карт.

Указание по применению: для удобства пользователей старых (legacy) карт хост должен попробовать использовать карту меньшей производительности, чем ожидалось, и попытаться при необходимости выполнить запись на неё. Когда режим работы предоставляется только для карт определенного класса скорости, один из других режимов должен представлять скорость менее высокого Speed Class включая Class 0.

4.13.7 Что следует учесть для вставки обновления FAT перед записью. *В упрощенной спецификации описание этой секции опущено.*

4.13.8 Условия и требования измерений для Speed Class. *В упрощенной спецификации описание этой секции опущено.*

4.14 Вычисление таймаута для очистки (Erase Timeout Calculation). В этой части предоставлено руководство для ситуации, когда очистка данных происходит долго, и нужен метод для вычисления значения таймаута.

4.14.1 Единица стирания (Erase Unit). Стандарт Speed Class Specification определяет новую единицу обслуживания AU (Allocation Unit). Вычисление таймаута определяется на базе AU. Карта памяти SD поддерживает очистку блока, но если блок занимает часть AU, то очистка занимает больше времени (partial erase AU). В этом случае хост должен добавить 250 мс к результату таймаута, рассчитанного на базе AU. Когда начало и конец блоков находятся в частично стираемых AU, нужно добавить 500 мс.

4.14.2 Анализ характеристик времени стирания. Рис. 4-10 показывает пример характеристик стирания данных (erase) - взаимосвязь времени и количества стираемых AU. Время стирания вычисляется из очистки указанного количества AU одной командой стирания. С предположением, что стирание выполняется на базе AU, характеристики стирания можно аппроксимировать прямой линией. Линия A на рисунке 4-10 показывает пример характеристики.

Красная линия показывает значение таймаута, которое должен использовать хост. Значение таймаута может быть определено линией A. Если таймаут очистки меньше 1 секунды, то хост должен использовать в качестве таймаута 1 секунду. Если таймаут больше 1 секунды, то хост должен использовать величину, определяемую линией A.

Параметры регистра N_{ERASE}, T_{ERASE} и T_{OFFSET} определяют форму линии. T_{ERASE} показывает таймаут для стирания N_{ERASE} блоков AU со смещения T_{OFFSET}. T_{ERASE} и N_{ERASE} определяют наклон линии. T_{OFFSET} перемещает линию параллельно верхней стороне. Производитель карты должен определить эти параметры так, что линия была всегда больше, чем время стирания любого AU.

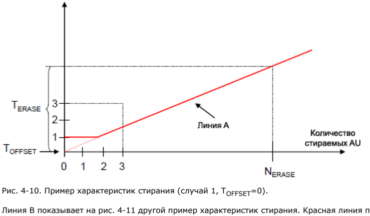


Рис. 4-10. Пример характеристик стирания (случай 1, T_{OFFSET}=0).

Линия B показывает на рис. 4-11 другой пример характеристик стирания. Красная линия показывает значение таймаута, которое хост должен использовать. Поскольку таймаут больше 1 секунды, то красная линия совпадает с линией B.

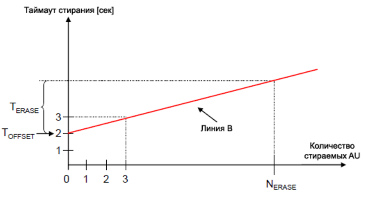


Рис. 4-11. Пример характеристик стирания (случай 2, T_{OFFSET}=2).

4.14.3 Метод стирания больших областей памяти. Вычисленный метод таймаута для многих AU может быть слишком большим в сравнении с реальным временем очистки. Вычисление таймаута стирания не точное, потому что вычисленный таймаут включает отклонение (погрешность времени стирания). Погрешность накапливается на AU, и результат вычисления таймаута для большого количества AU будет включать большие погрешности. Такие вычисления могут быть бессмысленными, потому что диапазон допуска может быть порядка минут. Таким образом, за один раз следует стирать малое количество AU. Это позволит хосту вычислить малый таймаут с меньшими ошибками.

Указание по применению: когда стирается большая область данных, хост должен поделить её на меньшие области, выбранные по границам AU, и стирать из друг за другом с малым таймаутом. Это может занять несколько больше времени для стирания большой области, но хост получит возможность информировать пользователя о процессе стирания, и пользователь при желании сможет остановить процесс стирания в любой момент.

4.14.4 Вычисление таймаута стирания с использованием параметров регистров. Таймаут стирания (Erase Timeout) X блоков AU может быть вычислен по следующему выражению:

$$\text{Erase Timeout XAU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \qquad \text{(Выражение 6)}$$

Таймаут вычисляется по следующим шагам:

(1) Вычислить выражение 6.
(2) Если результат на шаге (1) меньше 1 секунды, то таймаут принимается за 1 секунду.
(3) К результату (2) нужно добавить 250 мс для каждого частично стираемого AU. Когда начало и конец блоков также находятся в частично стираемых AU, то к результату (2) нужно добавить 500 мс.

[**5. Регистры карты**]

В программном интерфейсе карты определены 6 регистров: **OCR**, **CID**, **CSD**, **RCA**, **DSR** и **SCR**. К ним можно получить доступ соответствующими командами (см. Главу 4.7). Регистры OCR, CID, CSD и SCR несут информацию, специфичную для карты/содержимого, в то время как регистры RCA и DSR являются регистрами конфигурации, где хранятся актуальные параметры конфигурации.

Чтобы позволить дальнейшее расширение, карты должны вернуть 0 в зарезервированных битах регистров.

5.1 Регистр OCR. 32-битный регистр рабочих условий хранит профиль напряжения питания VDD карты. Дополнительно этот регистр включает в себе информационные биты статуса. Один бит статуса устанавливается, если завершена процедура включения карты (power up). Другой бит показывает статус емкости карты после того, как установится бит статуса power up. Регистр OCR реализуется внутренней аппаратурой карты.

Профиль VDD регистра OCR включает следующее: бит 7 был недавно определен для карт, поддерживающих двойное напряжение питания (Dual Voltage Card), и по умолчанию этот бит установлен в 0. Если Dual Voltage Card не получила команду CMD8, то 7 регистра OCR в response покажет 0, и Dual Voltage Card, которая приняла CMD8, установит этот бит в 1.

2 дополнительных бита статуса OCR: бит 31 это бит статуса power up, он устанавливается после того, как завершится процедура включения карты. Бит 30 это бит статуса емкости карты, он установится в 1, если карта повышенной емкости (High Capacity SD Memory Card, SDHC), а 0 покажет карту стандартной емкости (Standard Capacity SD Memory Card). Бит статуса емкости достоверен после завершения процедуры power up (когда бит статуса power up установился в 1). Хост должен прочитать этот бит статуса для определения емкости карты (обычная она или SDHC).

Таблица 5-1. Определение регистра OCR.

№ бит в OCR	Определение полей OCR
31	Бит статуса включения карты (busy)(2)
30	Состояние емкости карты (Card Capacity Status, CCS)(1)
29..24	Зарезервировано
23	3.5V..3.6V
22	3.4V..3.5V
21	3.3V..3.4V
20	3.2V..3.3V
19	3.1V..3.2V
18	3.0V..3.1V
17	2.9V..3.0V
16	2.8V..2.9V
15	2.7V..2.8V
14..8	Зарезервировано
7	Зарезервировано для диапазона низких напряжений (Low Voltage Range)
6..0	Зарезервировано

Примечания к таблице 5-1:

(1) Этот бит допустим только когда карта установила бит статуса включения питания (power up status bit).
(2) Этот бит устанавливается в лог. 0, если карта не завершила процедуру своего включения (power up routine).

Поддерживаемый диапазон напряжений кодируется, как показано в таблице 5-1. Диапазон напряжений не поддерживается, если значение соответствующего бита установлено в 0. Пока карта занята, соответствующий бит (31) установлен в 0.

5.2 Регистр CID. Регистр Card IDentification (CID, регистр идентификации карты) имеет ширину 128 бит. Он содержит идентификационную информацию карты, используемую на фазе идентификации. Каждая индивидуально читаемая/записываемая (Read/Write, RW) карта должна иметь уникальный идентификационный номер. Структура регистра CID определена в параграфах под таблицей 5-2.

Таблица 5-2. Поля регистра CID.

Наименование поля	Поле	Ширина	Срез CID
Manufacturer ID (идентификатор производителя)	MID	8	[127:120]
OEM/Application ID (идентификатор приложения)	OID	16	[119:104]
Product name (название продукта)	PNM	40	[103:64]
Product revision (ревизия продукта)	PRV	8	[63:56]
Серийный номер продукта	PSN	32	[55:24]
Зарезервировано	-	4	[23:20]
Дата производства	MDT	12	[19:8]
Контрольная сумма CRC7	CRC	7	[7:1]
Не используется, здесь всегда лог. 1.	-	1	[0]

MID. 8-битное двоичное число, идентифицирующее производителя карты (Manufacturer ID).

OID. 2-символьная строка ASCII, идентифицирующая карту OEM и/или содержимое карты (когда она используется как носитель дистрибуции ROM или FLASH).

Примечание: данные MID и OID управляются, определяются и выделяются для производителя SD Memory Card организацией SD-3C, LLC. Эта процедура устанавливается для гарантирования уникальности содержимого регистра CID. Организация SD-3C, LLC лицензирует компании, которые хотят производить и/или продавать карты памяти SD, что включает, но не ограничивается картами с памятью FLASH, ROM, OTP, RAM и картами SDIO Combo. SD-3C, LLC это организация с ограниченной ответственностью, организованная компаниями Matsushita Electric Industrial Co. Ltd., SanDisk Corporation и Toshiba Corporation.

PNM. 5-символьная ASCII строка имени производителя.

PRV. Ревизия продукта, составленная из двух цифр в двоично-десятичном формате (Binary Coded Decimal, BCD), по 4 бита на каждую цифру, представляя номер ревизии как "n.m". "n" это старший значащий ниббл, и "m" младший значащий ниббл. Например, двоичное значение поля PRV для ревизии продукта 6.2 будет равно 0110 0010b.

PSN. 32-битное двоичное число серийного номера карты.

MDT. Дата производства, составленная из 2 шестнадцатеричных цифр. Одна цифра из 8 бит представляет год (y), и другие 4 бита представляют месяц (m). В поле m (биты [11:8]) содержится двоичный код месяца, где 1 соответствует январю. Поле y (биты [19:12]) это код года, где 0 соответствует 2000 году. Например, двоичное значение поля даты производства "Апрель 2001" будет равен 00000001 0100b.

CRC. Контрольная сумма CRC7 (7 бит). Это контрольная сумма содержимого CID, которая вычислен по алгоритму, описанному в Главе 4.5.

5.3 Регистр CSD. Это регистр данных, относящихся к карте (Card-Specific Data, сокращенно CSD), и он предоставляет информацию которая нужна для организации доступа к содержимому карты. CSD определяет формат данных, тип коррекции ошибки, максимальное время доступа, может ли использоваться регистр DSR и т. д. Программируемая часть регистра (части, помеченные W или E, см. ниже) может быть изменена командой CMD27. Типы частей регистра в таблице ниже кодируются следующим образом: R читаемая часть (буква R от слова readable), W(1) записываемая 1 раз, W записываемая многократно (буква W от слова writable).

5.3.1 CSD_STRUCTURE. Структуры регистра CSD отличаются в зависимости от версии физической спецификации (Physical Specification Version) и емкости карты. Поле CSD_STRUCTURE в регистре CSD показывает версию структуры. Таблица 5-3 показывает номер версии, связанной структуры CSD.

Таблица 5-3. Структура регистра CSD.

CSD_STRUCTURE	Версия структуры CSD	Допустимо для Physical Specification Version/Card Capacity карт памяти SD
0	CSD Version 1.0	Version 1.01-1.10 Version 2.00/Standard Capacity
1	CSD Version 2.0	Version 2.00/High Capacity
2, 3	Зарезервировано	

5.3.2 Регистр CSD (CSD Version 1.0)

Таблица 5-4. Поля регистра CSD (CSD Version 1.0).

Наименование поля	Поле	Ширина	Значение	Тип	Срез CSD
Структура CSD	CSD_STRUCTURE	2	00b	R	[127:126]
Зарезервировано	-	6	000000b	R	[125:120]
Время доступа чтения 1	TAAC	8	xxh	R	[119:112]
Время доступа чтения 2 в тактах CLK (NSAC*100)	NSAC	8	xxh	R	[111:104]
Макс. скорость передачи данных	TRAN_SPEED	8	32h или 5Ah	R	[103:96]
Классы команд карты (card command classes)	CCC	12	01x110110101b	R	[95:84]
Макс. длина блока чтения данных	READ_BL_LEN	4	xh	R	[83:80]
Разрешено ли частичное чтение блоков	READ_BL_PARTIAL	1	1b	R	[79]
Разрешены ли не выровненные блоки при записи	WRITE_BLK_MISALIGN	1	xb	R	[78]
Разрешены ли не выровененные блоки при чтении	READ_BLK_MISALIGN	1	xb	R	[77]
Реализован ли DSR	DSR_IMP	1	xb	R	[76]
Зарезервировано	-	2	00b	R	[75:74]
Размер памяти устройства	C_SIZE	12	xxxh	R	[73:62]
Макс. потребляемый ток при чтении и мин. VDD	VDD_R_CURR_MIN	3	xxxb	R	[61:59]
Макс. потребляемый ток при чтении и макс. VDD	VDD_R_CURR_MAX	3	xxxb	R	[58:56]
Макс. потребляемый ток при записи и мин. VDD	VDD_W_CURR_MIN	3	xxxb	R	[55:53]
Макс. потребляемый ток при записи и макс. VDD	VDD_W_CURR_MAX	3	xxxb	R	[52:50]
Множитель размера устройства	C_SIZE_MULT	3	xxxb	R	[49:47]
Разрешено ли стирание одиночного блока	ERASE_BLK_EN	1	xb	R	[46]
Размер стираемого сектора	SECTOR_SIZE	7	xxxxxxxb	R	[45:39]
Размер группы, защищенной от записи	WP_GRP_SIZE	7	xxxxxxxb	R	[38:32]
Разрешена ли защита групп от записи	WP_GRP_ENABLE	1	xb	R	[31]
Зарезервировано (не используется)		2	00b	R	[30:29]
Множитель скорости записи	R2W_FACTOR	3	xxxb	R	[28:26]
Макс. длина блока записи	WRITE_BL_LEN	4	xxxxb	R	[25:22]
Разрешена ли частичная запись блоков	WRITE_BL_PARTIAL	1	xb	R	[21]
Зарезервировано	-	5	00000b	R	[20:16]
Группа формата файла	FILE_FORMAT_GRP	1	xb	R/W(1)	[15]
Флаг копирования (OTP)	COPY	1	xb	R/W(1)	[14]
Постоянная защита от записи	PERM_WRITE_PROTECT	1	xb	R/W(1)	[13]
Временная защита от записи	TMP_WRITE_PROTECT	1	xb	R/W	[12]
Формат файла	FILE_FORMAT	2	xxb	R/W(1)	[11:10]
Зарезервировано		2	00b	R/W	[9:8]
Контрольная сумма CRC7	CRC	7	xxxxxxxb	R/W	[7:1]
Не используется, здесь всегда лог. 1	-	1	1b	-	[0]

В следующих секциях описаны поля CSD и относящиеся к ним типы данных. Если не указано явно нечто другое, все строки бит интерпретируются как двоичные числа, начинающиеся с левого (старшего, MSB) бита.

TAAC. Определяет асинхронную часть времени доступа к данным.

Таблица 5-5: Определение времени доступа TAAC.

Позиция бита TAAC	Код
7	Зарезервировано
6:3	Значение времени: 0 = зарезервировано 1 = 1.0 2 = 1.2 3 = 1.3 4 = 1.5 5 = 2.0 6 = 2.5 7 = 3.0 8 = 3.5 9 = 4.0 A = 4.5 B = 5.0 C = 5.5 D = 6.0 E = 7.0 F = 8.0
2:0	Единица времени: 0 = 1 нс 1 = 10 нс 2 = 100 нс 3 = 1 мкс 4 = 10 мкс 5 = 100 мкс 6 = 1 мс 7 = 10 мс

NSAC. Определяет наихудший случай для фактора времени доступа, зависящего от тактовой частоты. Единица NSAC составляет 100 периодов тактовой частоты. Таким образом, максимальное значение для части, зависящей от частоты времени доступа составляет 25.5 k периодов тактов.

Общее время доступа NAC будет суммой TAAC и NSAC. Оно должно быть вычислено хостом для действительно использующейся тактовой частоты. Время доступа на чтение должно интерпретироваться как типичная задержка появления первого бита данных блока или потока бит.

TRAN_SPEED. Следующая таблица определяет максимальную скорость передачи на одну линию данных:

Таблица 5-6. Определение максимальной скорости передачи данных (Maximum Data Transfer Rate).

Позиция бита TRAN_SPEED	Код
7	Зарезервировано
6:3	Значение скорости: 0 = зарезервировано 1 = 1.0 2 = 1.2 3 = 1.3 4 = 1.5 5 = 2.0 6 = 2.5 7 = 3.0 8 = 3.5 9 = 4.0 A = 4.5 B = 5.0 C = 5.5 D = 6.0 E = 7.0 F = 8.0
2:0	Единица скорости передачи: 0 = 100 кбит/сек 1 = 1 мегабит/сек 2 = 10 мегабит/сек 3 = 100 мегабит/сек 4..7 = зарезервировано

Обратите внимание, что для имеющихся в настоящее время карт памяти SD, это поле всегда должно быть равно 0_0110_010b (032h), что эквивалентно 25 МГц - обязательная максимальная рабочая частота SD Memory Card.

В режиме высокой скорости (High-Speed) это поле должно всегда быть равно 0_1011_010b (05Ah), что эквивалентно 50 МГц, и когда режим интервалов времени возвращается к интервалам по умолчанию командами CMD6 или CMD0, это значение будет 032h.

CCC. Набор команд SD Memory Card поделен на отдельные группы (классы команд). Регистр классов команд (card command class, сокращенно CCC) определяет, какие классы команд поддерживаются этой картой. Значение 1 в бите CCC означает, что поддерживается соответствующий класс команд. Для определений классов команд см. таблицу 4-17.

Таблица 5-7. Поддерживаемые классы команд.

Бита CCC	Поддерживаемый класс команд
0	Class 0
1	Class 1
...	...
11	Class 11

READ_BL_LEN. Максимальная читаемая длина блока вычисляется как 2READ_BL_LEN. Максимальная длина блока может быть таким образом в диапазоне 512..2048 байт (подробности см. в Главе 4.11). Обратите внимание, что в SD Memory Card значение WRITE_BL_LEN всегда равно значению READ_BL_LEN.

Таблица 5-8. Длина блока данных.

READ_BL_LEN	Длина блока
0..8	Зарезервировано
9	2^9 = 512 байт
10	2^{10} = 1024 байт
11	2^{11} = 2048 байт
12..15	Зарезервировано

READ_BL_PARTIAL (всегда = 1 в картах памяти SD). Частичное чтение блока (Partial Block Read) для SD Memory Card всегда разрешено. Это означает, что могут также использоваться блоки меньшего размера. Минимальный размер блока будет равен 1 байту.

WRITE_BLK_MISALIGN. Определяет, может ли логический блок данных, записываемый одной командой, распространяться через границу между соседними физическими блоками, начинаясь и/или заканчиваясь при этом внутри физического блока, не на его границе. Размер блока памяти для записи определяется в поле WRITE_BL_LEN. WRITE_BLK_MISALIGN=0 сигнализирует о том, что пересекать физические границы блока недопустимо (нельзя использовать при записи данные, не выровненные по физической границе блока). WRITE_BLK_MISALIGN=1 сигнализирует, что можно пересекать физические границы блока при записи.

READ_BLK_MISALIGN. Определяет, может ли логический блок данных, читаемый одной командой, распространяться через границу между соседними физическими блоками, начинаясь и/или заканчиваясь при этом внутри физического блока, не на его границе. Размер блока памяти для чтения определяется в поле READ_BL_LEN. READ_BLK_MISALIGN=0 сигнализирует о том, что пересекать физические границы блока недопустимо (нельзя использовать при чтении данные, не выровненные по физической границе блока). READ_BLK_MISALIGN=1 сигнализирует, что можно пересекать физические границы блока при чтении.

DSR_IMP. Определяет, интегрирован ли в карте конфигурируемый драйвер сигнала шины. Если этот бит установлен, то в карте должен быть реализован регистр стадии драйвера (driver stage register, сокращенно DSR, см. Главу 5.5).

Таблица 5-9. Кодовая таблица реализации DSR.

DSR_IMP	Тип DSP
0	DSR не реализован
1	DSR реализован

C_SIZE. Этот параметр используется для вычисления емкости области данных пользователя карты (user data card capacity). Область данных пользователя не включает защищенную область безопасности (security protected area). Емкость карты вычисляется из данных полей C_SIZE, C_SIZE_MULT и READ_BL_LEN следующим образом:

Емкость памяти = BLOCKNR * BLOCK_LEN

Здесь:

BLOCKNR = (C_SIZE+1) * MULT
$\text{MULT} = 2^{\text{C\_SIZE\_MULT+2}}$ (C_SIZE_MULT < 8)
$\text{BLOCK\_LEN} = 2^{\text{READ\_BL\_LEN}}$ (READ_BL_LEN < 12)

Чтобы указать карту на 2 гигабайта, BLOCK_LEN должен быть 1024 байта. Таким образом, максимальная емкость, которую можно закодировать, составляет 4096*512*1024 = 2 гигабайта. Например: карта на 32 мегабайта с длиной блока BLOCK_LEN = 512 может кодироваться с C_SIZE_MULT = 3 и C_SIZE = 2000.

VDD_R_CURR_MIN, **VDD_W_CURR_MIN**. Минимальные значения для токов потребления чтения и записи при минимальном напряжении питания VDD кодируются следующим образом:

Таблица 5-10. Минимальное потребление тока от VDD.

VDD_R_CURR_MIN VDD_W_CURR_MIN	Код для текущего потребляемого тока и VDD
2:0	0 = 0.5 mA 1 = 1 mA 2 = 5 mA 3 = 10 mA 4 = 25 mA 5 = 35 mA 6 = 60 mA 7 = 100 mA

VDD_R_CURR_MAX, **VDD_W_CURR_MAX**. Максимальные значения для токов чтения и записи при максимальном напряжении питания VDD кодируются следующим образом:

Таблица 5-11. Максимальное потребление тока от VDD.

VDD_R_CURR_MAX VDD_W_CURR_MAX	Код для текущего потребляемого тока и VDD
2:0	0 = 1 mA 1 = 5 mA 2 = 10 mA 3 = 25 mA 4 = 35 mA 5 = 45 mA 6 = 80 mA 7 = 200 mA

C_SIZE_MULT. Этот параметр используется для кодирования множителя MULT, используемого при вычисления емкости карты (см. описание поля C_SIZE). Множитель MULT определяется как 2C_SIZE_MULT+2.

Таблица 5-12. Множитель для емкости памяти устройства.

C_SIZE_MULT	MULT
0	2^2 = 4
1	2^3 = 8
2	2^4 = 16
3	2^5 = 32
4	2^6 = 64
5	2^7 = 128
6	2^8 = 256
7	2^9 = 512

ERASE_BLK_EN. Поле ERASE_BLK_EN задает гранулярность размера стираемого блока данных. Операция стирания (erase) может очистить либо один, либо несколько блоков по 512 байт, либо один или большее количество элементов данных (или секторов) размером SECTOR_SIZE (см. определение ниже).

Если ERASE_BLK_EN=0, то хост может стереть один или несколько элементов данных размером SECTOR_SIZE. Стирание начнется с начала сектора, соответствующего начальному адресу стирания, и будет продолжаться до конца сектора, который содержит конечный адрес стирания. Например, если SECTOR_SIZE=31, и хост установил начальный адрес стирания (Erase Start Address) равным 5, и конечный адрес стирания (Erase End Address) в 40, то будут стерты физические блоки от 0 до 63, как это показано на рис. 5-1.

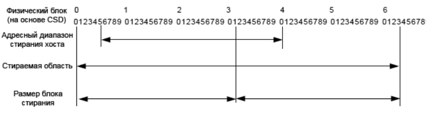


Рис. 5-1. Пример стирания при ERASE_BLK_EN = 0.

Если ERASE_BLK_EN=1, то хост может стереть один или несколько элементов данных по 512 байт. Будут стерты данные всех блоков, которые содержат данные от начального адреса стирания до конечного адреса стирания. Например, если хост установил Erase Start Address = 5 и Erase End Address = 40, то будут стерты физические блоки от 5 до 40, как это показано на рис. 5-2.

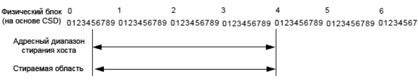


Рис. 5-2. Пример стирания при ERASE_BLK_EN = 1.

SECTOR_SIZE. Размер стираемого сектора. Это 7-битное двоичное значение, определяющее количество записываемых блоков (см. WRITE_BL_LEN). Действительный размер вычисляется увеличением этого числа на 1. Значение 0 означает 1 записываемый блок, 127 означает 128 записываемых блоков.

WP_GRP_SIZE. Размер группы, защищенной от записи. Это 7-битное двоичное значение, определяющее количество стираемых секторов (см. SECTOR_SIZE). Действительный размер вычисляется увеличением этого числа на 1. Значение 0 означает 1 стираемый сектор, 127 означает 128 стираемых секторов.

WP_GRP_ENABLE. Значение 0 означает, что защита группы от записи невозможна.

R2W_FACTOR. Определяет типичное время программирования блока как множитель времени доступа чтения. В следующей таблице определен формат этого поля.

Таблица 5-13. R2W_FACTOR.

R2W_FACTOR	Множитель для времени доступа чтения
0	1
1	2 (запись происходит в 2 раза медленнее, чем чтение)
2	4
3	8
4	16
5	32
6, 7	Зарезервировано

WRITE_BL_LEN. Максимальная длина записываемого блока, вычисляемая как 2WRITE_BL_LEN. Максимальная длина блока может быть, таким образом в диапазоне 512..2048 байт. Всегда поддерживается длина записываемого блока (Write Block Length) размером 512 байт. Обратите внимание, что в SD Memory Card параметр WRITE_BL_LEN всегда равен READ_BL_LEN.

Таблица 5-14. Длина записываемого блока данных.

WRITE_BL_LEN	Длина блока
0..8	Зарезервировано
9	2^9 = 512 байт
10	2^{10} = 1024 байт
11	2^{11} = 2048 байт
12..15	Зарезервировано

WRITE_BL_PARTIAL. Определяет, могут ли в командах записи блока использоваться блоки частичного размера (partial block size). WRITE_BL_PARTIAL=0 означает, что в блочно-ориентированных операциях записи могут использоваться только блоки размера WRITE_BL_LEN или размера, кратного WRITE_BL_LEN, в единицах по 512 байт. WRITE_BL_PARTIAL=1 означает, что также могут использоваться и блоки меньшего размера, при этом минимальный размер блока может быть 1 байт.

FILE_FORMAT_GRP. Показывает форматы файла выбранной группы. Это поле всегда только для чтения для карт ROM. Использование этого поля показано в таблице 5-15 (см. FILE_FORMAT).

COPY. Определяет, является содержимое карты оригиналом (0) или это копия оригинала (1). Бит COPY для устройств OTP и MTP, продаваемых конечным потребителям, устанавливается в 1, показывая тем самым, что содержимое карты является копией. Бит COPY является однократно программируемым битом (one time programmable, OTP).

PERM_WRITE_PROTECT. Задает постоянную защиту всего содержимого карты от записи или стирания (все команды записи и стирания для этой карты будут постоянно запрещены). Значение по умолчанию 0, т. е. нет постоянной защиты от записи.

TMP_WRITE_PROTECT. Временно защищает все содержимое карты от перезаписи или стирания (все команды записи и стирания для этой карты будут временно запрещены). Этот бит может быть установлен или сброшен. Значение по умолчанию 0, т. е. защита от записи отключена.

FILE_FORMAT. Показывает формат файлов на карте. Для ROM это поле только для чтения. Определены следующие форматы:

Таблица 5-15. Форматы файла.

FILE_FORMAT_GRP	FILE_FORMAT	Тип
0	0	Файловая система наподобие жесткого диска, где имеются таблица разделов (partition table).
0	1	DOS FAT (FAT12, как на floppy-диске), где имеется только загрузочный сектор (boot sector) и нет таблицы разделов
0	2	Universal File Format
0	3	Другой / неизвестный формат
1	0..3	Зарезервировано

Более подробное описание см. в спецификации файловой системы карт памяти SD (SD Memory Card File System).

CRC. Поле CRC несет контрольную сумму содержимого регистра CSD. Она вычисляется по алгоритму, описанному в Главе 4.5. Контрольная сумма заново вычисляется хостом, когда делается любая модификация CSD. Значение по умолчанию соответствует первоначальному содержимому CSD.

5.3.3 Регистр CSD (CSD Version 2.0). Таблица 5-16 показывает определение содержимого CSD для карт повышенной емкости High Capacity SD Memory Card, SDHC (CSD Version 2.0). В последующих секциях описываются поля CSD и соответствующие типы данных для High Capacity SD Memory Card (SDHC).

CSD Version 2.0 применяется только для High Capacity SD Memory Card (SDHC). Имя поля в круглых скобках устанавливается в фиксированное значение, и показывает, что хосту необязательно обращаться к этим полям. Эти фиксированные значения позволяют хосту, который к ним обращается, сохранить совместимость со спецификацией CSD Version 1.0. Поле Cell Type (тип ячейки) кодируется следующим образом: R читаемое (readable), W(1) записываемое (writable) один раз, W записываемое многократно.

Таблица 5-16. Поля регистра CSD (CSD Version 2.0). Поля такие же, как у структуры CSD Version 1.0.

Наименование поля	Поле	Ширина	Значение	Тип	Срез CSD
Структура CSD	CSD_STRUCTURE	2	00b	R	[127:126]
Зарезервировано	-	6	000000b	R	[125:120]
Время доступа чтения 1	TAAC	8	xxh	R	[119:112]
Время доступа чтения 2 в тактах CLK (NSAC*100)	NSAC	8	xxh	R	[111:104]
Макс. скорость передачи данных	TRAN_SPEED	8	32h или 5Ah	R	[103:96]
Классы команд карты (card command classes)	CCC	12	01x110110101b	R	[95:84]
Макс. длина блока чтения данных	READ_BL_LEN	4	xh	R	[83:80]
Разрешено ли частичное чтение блоков	READ_BL_PARTIAL	1	1b	R	[79]
Разрешены ли не выровненные блоки при записи	WRITE_BLK_MISALIGN	1	xb	R	[78]
Разрешены ли не выровененные блоки при чтении	READ_BLK_MISALIGN	1	xb	R	[77]
Реализован ли DSR	DSR_IMP	1	xb	R	[76]
Зарезервировано	-	2	00b	R	[75:74]
Размер памяти устройства	C_SIZE	12	xxxh	R	[73:62]
Макс. потребляемый ток при чтении и мин. VDD	VDD_R_CURR_MIN	3	xxxb	R	[61:59]
Макс. потребляемый ток при чтении и макс. VDD	VDD_R_CURR_MAX	3	xxxb	R	[58:56]
Макс. потребляемый ток при записи и мин. VDD	VDD_W_CURR_MIN	3	xxxb	R	[55:53]
Макс. потребляемый ток при записи и макс. VDD	VDD_W_CURR_MAX	3	xxxb	R	[52:50]
Множитель размера устройства	C_SIZE_MULT	3	xxxb	R	[49:47]
Разрешено ли стирание одиночного блока	ERASE_BLK_EN	1	xb	R	[46]
Размер стираемого сектора	SECTOR_SIZE	7	xxxxxxxb	R	[45:39]
Размер группы, защищенной от записи	WP_GRP_SIZE	7	xxxxxxxb	R	[38:32]
Разрешена ли защита групп от записи	WP_GRP_ENABLE	1	xb	R	[31]
Зарезервировано (не используется)		2	00b	R	[30:29]
Множитель скорости записи	R2W_FACTOR	3	xxxb	R	[28:26]
Макс. длина блока записи	WRITE_BL_LEN	4	xxxxb	R	[25:22]
Разрешена ли частичная запись блоков	WRITE_BL_PARTIAL	1	xb	R	[21]
Зарезервировано	-	5	00000b	R	[20:16]
Группа формата файла	FILE_FORMAT_GRP	1	xb	R/W(1)	[15]
Флаг копирования (OTP)	COPY	1	xb	R/W(1)	[14]
Постоянная защита от записи	PERM_WRITE_PROTECT	1	xb	R/W(1)	[13]
Временная защита от записи	TMP_WRITE_PROTECT	1	xb	R/W	[12]
Формат файла	FILE_FORMAT	2	xxb	R/W(1)	[11:10]
Зарезервировано		2	00b	R/W	[9:8]
Контрольная сумма CRC7	CRC	7	xxxxxxxb	R/W	[7:1]
Не используется, здесь всегда лог. 1	-	1	1b	-	[0]

TAAC. Значение этого поля фиксировано на 0Eh, что показывает 1 миллисекунду. Хост не должен использовать TAAC, NSAC и R2W_FACTOR для вычисления таймаута, и должен использовать фиксированный таймаут для операций чтения и записи (см. главу 4.6.2).

NSAC. Значение этого поля фиксировано на 00h. NSAC не должно использоваться для вычисления значений таймаута.

TRAN_SPEED. Определение этого поля то же самое, что и для CSD Version 1.0.

CCC. Определение этого поля то же самое, что и для CSD Version 1.0.

READ_BL_LEN. Значение этого поля фиксировано на 9h, показывая тем самым длину блока чтения 512 байт.

READ_BL_PARTIAL. Значение этого поля фиксировано на 0, показывая запрет partial block read (к данным может быть доступ только блоками).

WRITE_BLK_MISALIGN. Значение этого поля фиксировано на 0, показывая запрет записи с пересечением физических границ блока для карт High Capacity SD Memory Card (SDHC).

READ_BLK_MISALIGN. Значение этого поля фиксировано на 0, показывая запрет чтения с пересечением физических границ блока для карт High Capacity SD Memory Card (SDHC).

DSR_IMP. Определение этого поля то же самое, что и для CSD Version 1.0.

C_SIZE. Это поле расширено до 22 бит, и может показывать емкость до 2 терабайт (соответствует максимальному значению 32-битного блочного адреса). Этот параметр используется для вычисления размера области данных пользователя карты памяти SD (не включая защищенную область). Емкость данных пользователя вычисляется по параметру C_SIZE следующим образом:

Емкость памяти = (C_SIZE+1) * 512 килобайт

В качестве максимальной емкости Physical Layer Specification Version 2.00 задано 32 гигабайта, старшие 6 бит этого поля должны быть установлены в 0.

ERASE_BLK_EN. Значение этого поля фиксировано на 1. Это означает, что хост может очистить один или несколько блоков памяти по 512 байт.

SECTOR_SIZE. Значение этого поля фиксировано на 7Fh, что показывает 64 килобайта. Это значение не связано с операцией стирания (erase). Карты Version 2.00 показывают границу памяти по AU, и это поле не должно использоваться.

WP_GRP_SIZE. Значение этого поля фиксировано на 00h. High Capacity SD Memory Card (SDHC) не поддерживают группы с защитой от записи.

WP_GRP_ENABLE. Значение этого поля фиксировано на 0. High Capacity SD Memory Card (SDHC) не поддерживают группы с защитой от записи.

R2W_FACTOR. Значение этого поля фиксировано на 2h, что показывает множитель на 4. Таймаут записи может быть вычислен умножением времени доступа чтения на R2W_FACTOR. Однако хост не должен использовать этот множитель, и должен использовать 250 мс для таймаута записи (см. главу 4.6.2).

WRITE_BL_LEN. Значение этого поля фиксировано на 9h, что показывает длину блока записи 512 байт.

WRITE_BL_PARTIAL. Значение этого поля фиксировано на 0, что показывает запрет partial block read, для чтения можно использовать элементы только размера, кратного размеру блока.

FILE_FORMAT_GRP. Значение этого поля фиксировано на 0. Хост не должен использовать это поле.

COPY. Определение этого поля то же самое, что и для CSD Version 1.0.

PERM_WRITE_PROTECT. Определение этого поля то же самое, что и для CSD Version 1.0.

TMP_WRITE_PROTECT. Определение этого поля то же самое, что и для CSD Version 1.0.

FILE_FORMAT. Определение этого поля то же самое, что и для CSD Version 1.0.

CRC. Определение этого поля то же самое, что и для CSD Version 1.0.

5.4 Регистр RCA. Записываемый 16-битный регистр относительного адреса карты (relative card address, сокращенно RCA) несет в себе адрес карты, который публикуемый картой во время идентификации карты. Этот адрес используется для адресованной коммуникации между хостом и картой после процедуры идентификации карты. Значение по умолчанию в регистре RCA равно 0x0000. Значение 0x0000 зарезервировано для установки всех карт командой CMD7 в состояние приостановки (Stand-by State).

Примечание: в режиме подключения карты через SPI регистр RCA недоступен и не используется.

5.5 Регистр DSR (опционально). 16-битный регистр стадии драйвера (driver stage register, сокращенно DSR) подробно описан в Главе 6.5. Он может опционально использоваться для улучшения производительности шины при расширенных условиях работы (в зависимости от таких параметров шины, как длина её сигналов, скорость передачи или количество подключенных карт). Регистр CSD переносит информацию об использовании регистра DSR (есть он или нет). Значение по умолчанию регистра DSR равно 0x404.

5.6 Регистр SCR. В дополнение к регистру CSD, имеется еще и другой конфигурационный регистр, который назвали SD CARD Configuration Register (сокращенно SCR). SCR предоставляет информацию о специальных функциях карты памяти SD, которые были сконфигурированы для данной карты. Размер регистра SCR составляет 64 бита. Этот регистр должен быть установлен на заводе производителем SD Memory Card. Ниже в таблице описано содержимое регистра SCR.

Таблица 5-17. Поля регистра SCR.

Описание	Поле	Ширина	Тип	Срез SCR
Структура SCR	SCR_STRUCTURE	4	R	[63:60]
Версия спецификации SD Memory Card	SD_SPEC	4	R	[59:56]
Состояние данных после стирания	DATA_STAT_AFTER_ERASE	1	R	[55]
Поддержка SD Security	SD_SECURITY	3	R	[54:52]
Поддерживаемые варианты ширины шины данных DAT	SD_BUS_WIDTHS	4	R	[51:48]
Зарезервировано	-	16	R	[47:32]
Зарезервировано для производителя карты	-	32	R	[31:0]

Таблица 5-18. Версия структуры регистра SCR.

SCR_STRUCTURE	Версия структуры SCR	Physical Specification Version карт памяти SD
0	SCR Version 1.0	Version 1.01-2.00
1..15	Зарезервировано	

SD_SPEC. Описывает версию стандарта Physical Layer Specification, поддерживаемую этой картой.

Таблица 5-19. Physical Layer Specification Version.

SD_SPEC	Номер для Physical Specification Version
0	Version 1.0-1.01
1	Version 1.10
2	Version 2.00
3..15	Зарезервировано

DATA_STAT_AFTER_ERASE. Определяет состояние данных карты после очистки, либо 0, либо 1 (это зависит от технологии, примененной вендором карты).

SD_SECURITY. Описывает версию спецификации безопасности (Security Specification Version), поддерживаемую картой.

Таблица 5-20. Поддерживаемый алгоритм безопасности SD (Security Algorithm).

SD_SECURITY	Security Specification Version
0	Нет поддержки безопасности
1	Не используется
2	Version 1.01
3	Version 2.00
4..7	Зарезервировано

Обратите внимание, что обязательным требованием для обычной записываемой карты памяти SD является поддержка протокола безопасности (Security Protocol). Для карт памяти SD типа ROM (Read Only, только для чтения) и типа OTP (One Time Programmable, память с возможностью только однократного программирования), функция безопасности является опциональной (необязательной). В случае карты SD стандартной емкости (Standard Capacity SD Memory Card) это поле должно быть установлено в значение 2 (соответствует Version 1.01). В случае карт повышенной емкости High Capacity SD Memory Card (SDHC), это поле должно быть установлено в 3 (Version 2.00).

SD_BUS_WIDTHS. Описывает все возможные выборы ширины шины DAT, поддерживаемые этой картой.

Таблица 5-21. Поддерживаемые картой памяти SD варианты Bus Width.

SD_BUS_WIDTHS	Поддерживаемые варианты ширины шины данных
Бит 0	1 бит (только DAT0)
Бит 1	Зарезервировано
Бит 2	4 бита (DAT3..DAT0)
Бит 3	Зарезервировано

Поскольку карта памяти SD обязана поддерживать как минимум 2 режима ширины шины 1-битная и 4-битная, то любая карта SD должна устанавливать как минимум биты 0 и 2 (SD_BUS_WIDTH="0101").

[**6. Аппаратура интерфейса SD Memory Card**]

В упрощенной спецификации (этот документ) приведена только часть секции 6 полного стандарта.

6.1 Горячая установка и извлечение карты. *В упрощенной спецификации описание этой секции опущено.*

6.2 Детектирование карты (установка/извлечение). *В упрощенной спецификации описание этой секции опущено.*

6.3 Защита по питанию (установка/извлечение). *В упрощенной спецификации описание этой секции опущено.*

6.4 Схема питания. Схема питания карты памяти SD поддерживается локально для каждой подключенной карты и мастера шины. Примеры организации схемы питания можно посмотреть в статье [3].

6.4.1 Включение питания (Power Up).

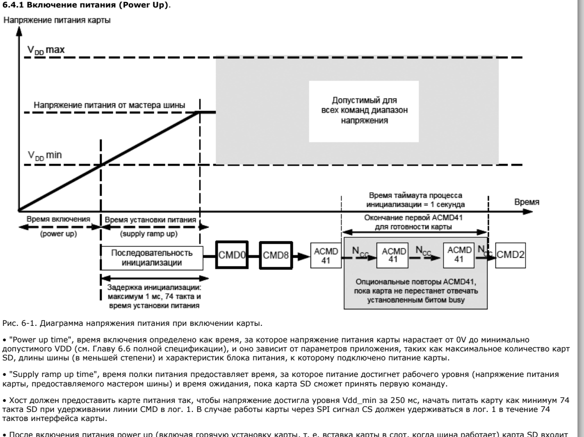

Напряжение питания карты
V_{DD} max
Допустимый для всех команд диапазон напряжения
Напряжение питания от мастера шины
V_{DD} min
Время таймаута процесса инициализации = 1 секунда
Время
Время включения (power up)
Время установки питания (supply ramp up)
Окончание первой ACMD41 для готовности карты
Последовательность инициализации
CMD0
CMD8
ACMD 41
N_{CC}
ACMD 41
N_{CC}
ACMD 41
N_{CC}
CMD2
Задержка инициализации: максимум 1 мс, 74 такта и время установки питания
Опциональные повторы ACMD41, пока карта не перестанет отвечать установленным битом busy

Рис. 6-1. Диаграмма напряжения питания при включении карты.

- "Power up time", время включения определено как время, за которое напряжение питания карты нарастает от 0V до минимально допустимого VDD (см. Главу 6.6 полной спецификации), и оно зависит от параметров приложения, таких как максимальное количество карт SD, длины шины (в меньшей степени) и характеристик блока питания, к которому подключено питание карты.

- "Supply ramp up time", время полки питания предоставляет время, за которое питание достигнет рабочего уровня (напряжение питания карты, предоставляемого мастером шины) и время ожидания, пока карта SD сможет принять первую команду.

- Хост должен предоставить карте питания так, чтобы напряжение достигла уровня Vdd_min за 250 мс, начать питать карту как минимум 74 такта SD при удерживании линии CMD в лог. 1. В случае работы карты через SPI сигнал CS должен удерживаться в лог. 1 в течение 74 тактов интерфейса карты.

- После включения питания power up (включая горячую установку карты, т. е. вставка карты в слот, когда шина работает) карта SD входит в состояние ожидания (idle state). В случае хоста SD с традиционным подключением карты команда CMD0 не требуется. В случае хоста с подключением через SPI команда CMD0 должна быть первой (после перевода выборки CS в лог. 0), чтобы карта перешла в режим интерфейса SPI (подробнее см. Главу 7 [2]).

- CMD8 была добавлена как новая в Physical Layer Specification Version 2.00, чтобы поддержать несколько диапазонов напряжений, и она используется для проверки, поддерживает ли карта указанный диапазон напряжений питания. Версия спецификации 2.00 хоста должна выдать CMD8 и проверить напряжение перед инициализацией карты. Хост, который не поддерживает CMD8, должен предоставить диапазон "высокого" (порядка 3.3V) напряжения питания.

- ACMD41 является командой синхронизации, чтобы договориться о рабочем диапазоне напряжения питания, и опрашивать карты, пока они находятся в процессе включения (power-up sequence). В случае, когда к хосту подключено несколько карт, хост должен проверить, что все карты удовлетворяют предоставленному напряжению питания. Иначе хост должен по одной выбирать и инициализировать карты.

6.4.2 Цикл выключения (Power Down) и включения питания карты (Power Cycle).

- Когда хост выключает питание, у карты должно быть понижено питание VDD до уровня менее 0.5V для минимального периода в 1 миллисекунду. Во время выключения питания сигналы DAT, CMD и CLK должны быть отключены или переведены в лог. 0 хостом, чтобы избежать паразитного питания карты через сигнальные выводы.

- Если хосту нужно поменять рабочее напряжение, то требуется передернуть питание карты (power cycle). "Передернуть" означает, что питание карты должно быть снято (выключено) и затем включено обратно. Передергивание питания также требуется, чтобы получить доступ к картам, которые уже находятся в неактивном состоянии (Inactive State). Чтобы выполнить выключение и включение питания карты, хост должен следовать описанию выключения питания карты (power down) перед тем, как восстановить питание карты (power up). Например, уровень напряжения питания карты VDD должно быть снижено до значения меньше 0.5V на время как минимум 1 миллисекунда.

6.5 Программируемый выходной драйвер карты (опционально). *В упрощенной спецификации эта секция опущена.*

6.6 Рабочие условия шины (Bus Operating Conditions). *В упрощенной спецификации эта секция опущена.*

6.7 Параметры времени сигналов шины по умолчанию (Bus Timing default). *В упрощенной спецификации эта секция опущена.*

6.8 Параметры времени сигналов шины в режиме высокой скорости (Bus Timing High-Speed Mode). *В упрощенной спецификации эта секция опущена.*

[**7. Режим SPI**]

Перевод этой секции приведен в статье [2].

8 Механические спецификации SD Memory Card. *В упрощенной спецификации эта глава опущена.*

[**Приложение A**]

A.1 Коннектор карты. *В упрощенной спецификации эта часть описания опущена.* См. также цоколевки карт в статье [3].

A.2 Дополнительная связанная документация

- Спецификация карт памяти miniSD (miniSD Memory Card Specification)
- Спецификация карт памяти microSD (microSD Memory Card Specification)
- Дополнительные замечания по физической спецификации (Supplementary Notes for SD Physical Specification)
- Дополнительные замечания для спецификации карт miniSD (Supplementary Notes for miniSD Memory Card Specification)
- Указания по применению, связанные с физической спецификацией карт SD (Application Notes Relating to SD Physical Specification)
- Указания по применению, связанные с физической спецификацией карт miniSD (Application Notes Relating to miniSD Physical Specification)
- Руководство по реализации класса скорости (Speed Class Implementation Guideline)
- Спецификации SD часть 2 - описание файловой системы (SD Specifications Part2 File System Specification)
- Спецификации SD часть 3 - безопасность (SD Specifications Part3 Security Specification)

[**Приложение B. Словарик**]

AU Allocation Unit, элементарный блок, выделенный под данные.

block некоторое количество байт, в основном это базовая передаваемая порция данных (data transfer unit).

broadcast команда, отправляемая всем картам, подключенным к одной шине SD. Правда, к одной шине SD обычно подключается только одна карта, множественное подключение нескольких карт SD к одной шине встречается редко.

Blocklen длина блока, установленная командой CMD16.

CID регистр идентификационного номера карты (аббревиатура от Card IDentification).

CLK тактовый сигнал.

CMD сигнал для передачи команды от ответа на неё (SD bus command, если расширено, то CMDXX).

CRC Cyclic Redundancy Check, контрольная сумма для защиты целостности данных.

CSD регистр данных, относящихся к карте (аббревиатура от Card Specific Data).

DAT сигнал (сигналы DAT0, DAT1, ...) данных.

DSR регистр уровня драйвера (аббревиатура от Driver Stage Register).

ECC код коррекции ошибки (аббревиатура от Error Correction Code).

FLASH тип многократно программируемой энергонезависимой памяти (non volatile memory).

group группа, некоторое количество секторов, объединенные в операции стирания (erase) и области данных, защищенных от записи.

LOW, **HIGH** двоичные состояния интерфейса, привязанные к состояниям битов регистра и уровням напряжения на сигналах.

MSB, **LSB** сокращения от Most Significant Bit (старший значащий бит) и Least Significant Bit (младший значащий бит).

MTP многократно программируемая память (аббревиатура от Multiple Time Programmable).

N_{ERASE} рекомендуемое количество AU для очистки за одну операцию стирания (erase).

NSAC определяет наихудший случай для времени доступа к данным, зависящем от скорости следования тактов.

OCR регистр состояния операций (Operation Conditions Register).

open-drain открытый сток, режим работы логики интерфейса. Внешний резистор, подключенный к плюсу источника тока, используется для подачи уровня HIGH на логический сигнал, и внутренний транзистор своим стоком подтягивает этот сигнал к уровню LOW.

OTP однократно программируемая (One Time Programmable) память.

payload полезная нагрузка, данные.

P_m Performance of Move (производительность при перемещении данных).

P_r Performance of Read (производительность при чтении данных).

P_w Performance of Write (производительность при записи данных).

RU Recording Unit, записываемый блок.

push-pull режим работы логического интерфейса, когда комплементарная пара транзисторов (или двухтактный выход) используется для перевода логического сигнала в уровни HIGH или LOW.

RCA регистр относительного адреса карты (Relative Card Address).

ROM память только для чтения (Read Only Memory).

sector базовая единица данных для чтения, записи и стирания.

stuff bit заполняющие биты, чтобы обеспечить фиксированную длину фреймов для команд и ответов на команду.

SPI Serial Peripheral Interface, последовательный периферийный интерфейс. Подключение карты через SPI подробно описано в статьях [,].

TAAC фактор времени, определяющий время доступа к данных.

tag маркер, используемый для выбора групп или секторов для стирания.

TBD To Be Determined, т. е. должно быть определено (в будущем).

T_{ERASE} значение таймаута, используемое для стирания нескольких AU, как указано в ERASE_SIZE.

T_{fw} время записи файловой системы FAT (write time).

T_{fr} время чтения файловой системы FAT (read time).

three-state driver драйвер управления логическим сигналом, который может иметь 3 состояния: HIGH, LOW и высокое сопротивления. Последнее означает, что интерфейс никак не влияет на уровень сигнала, т. е. отключен от него.

token токен, кодовое слово, представляющее команду.

T_{OFFSET} смещение времени, используемое для вычисления таймаута стирания.

VDD + напряжения питания.

VSS общий провод напряжения питания, минус, земля.

VCA диапазон напряжений питания и логических уровней, принимаемых картой (Voltage Card Accepted).

VHS диапазон напряжений, предоставленных хостом (Voltage Host Supplied).

[**Ссылки**]

1. SD_Physical_Layer_Spec.pdf - SD Specifications Part 1 Physical Layer Simplified Specification Version 2.00 September 25, 2006.
2. Подключение карт SD через SPI (упрощенное описание стандарта).
3. Как использовать карты памяти MMC/SDC.
4. Blackfin Removable Storage Interface.

Добавить комментарий

Имя (обязательное)

E-Mail (обязательное)

Сайт

Осталось: 1000 символов

☐ Подписаться на уведомления о новых комментариях

Обновить

Отправить

JComments

TOP OF PAGE