

3.2.1 Приоритетность

Если имеется несоответствие между текстом, рисунками и таблицами, приоритетность имеют сначала таблицы, затем рисунки, затем текст.

3.2.2 Списки

Списки по порядку, описывающие последовательность, делаются по следующей форме:

- a)
- b)
- c)

Не упорядоченные списки делаются по форме:

- 1)
- 2)
- 3)

3.2.3 Ключевые слова

Для разграничения различных уровней требований и выборочности используется несколько ключевых слов

3.2.3.1 expected (ожидаемый): Ключевое слово, которое используется для описания поведения устройства или ПО определенных моделей, принятых в текущем Стандарте. Другие модели устройств или ПО могут быть также включены.

3.2.3.2 mandatory (принудительный): Ключевое слово, показывающее пункты, которые должны быть включены как определено анстоящим Стандартом.

3.2.3.3 may (может): Ключевое слово которое обозначает гибкость выбора неподразумеваемого предпочтения.

3.2.3.4 obsolete (устаревший): Ключевое слово, используемое для описания битов, байтов, полей и значений кода которые долгое время не имеют текущей актуальности или функциональности, как минимум в одном из более старых Стандартов. Однако, некоторая доля функциональности может быть отдана элементам, названным «устаревшими», для обеспечения обратной совместимости. Устаревший бит, байт, поле, или команда, никогда не может быть использовано для замены на более новый во всех последующих Стандартах. Устаревшие команды не могут быть использованы хостом. Команды, определенные как устаревшие, в предыдущих стандартах, могут отвергаться устройствами, поддерживающими более новый Стандарт. Однако, если устройство не отвергает устаревшую команду, минимальное что должно сделать устройство в ответ на эту команду – это ее исполнение. Если устаревшие биты, байты, поля, или кодовые значения не включены, их значения должны быть установлены как зарезервированные.

3.2.3.5 optional (не обязательный): Ключевое слово, которое описывает функции, не обязательные для текущего Стандарта. Однако, если любая такая функция определяется Стандартом как внесенная в него, функция должна использоваться как функция, определенная текущим Стандартом.

3.2.3.6 retired (оставленный): Ключевое слово, которое обозначает биты, байты, поля и кодовые значения, которые были определены в предыдущих Стандартах, но не определены в текущем Стандарте и могут быть перенаправлены для использования в других целях в бедеших Стандартах. Если оставленные биты, байты, поля или кодовые значения используются прежде, чем они были переназначены, они должны быть использованы в значении или функциональности, четко их определяющих в предыдущих Стандартах.

3.2.3.7 reserved (зарезервировано): АКлючевое слово, указывающее на резервные биты, байты, слова, поля, и кодовые значения, которые устанавливаются в в стороне от будущей стандартизации. Их использование и интерпретация может быть обозначена в последующих расширениях этого или других Стандартов. Зарезервированный бит, байт, поле или слово должно быть установлено в ноль, или в соответствии с будущим расширением текущего Стандарта. Реципиент не должен проверять зарезервированные биты, байты, слова или поля. Получение зарезервированный значений кода в определенных полях может быть распознано как ошибка в командном параметре и возвращен ответ об отвергнутой команде в регистр ошибок.

3.2.3.8 shall (будет): Ключевое слово, которое обозначает принудительное требование. Разработчики должны включать все такие принудительные требования для обеспечения полной и гарантированной совместимости с другими определенными текущим Стандартом продуктами.

3.2.3.9 should (если): Ключевое слово, обозначающее гибкость выбора при четко ограниченных альтернативах. Эквивалентно фразе «это рекомендовано».

3.2.4 Цифровые обозначения

Числа, которые набраны в нижнем регистре (без “b” и “h”), являются десятичными значениями. Числа, которые сопровождаются простой буквой b в нижнем регистре (к примеру 01b), являются двоичными значениями. Числа, которые сопровождаются набранной в нижнем регистре буквой h (к примеру 3Ah), являются шестнадцатеричными.

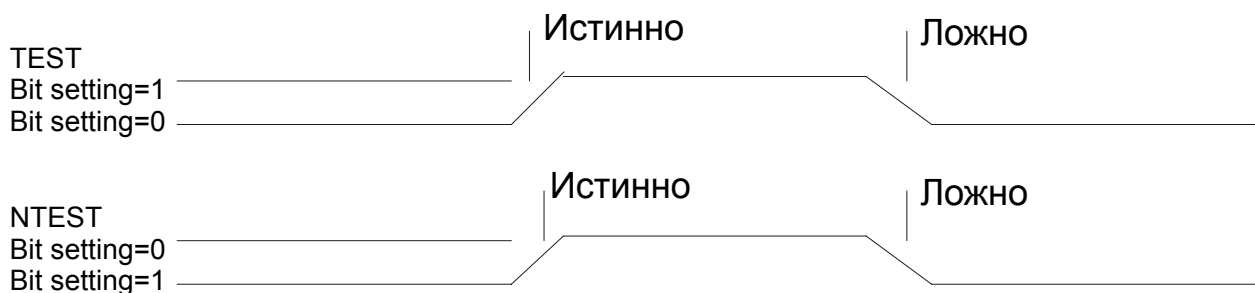
3.2.5 Соглашения о сигналах

Сигнальные имена печатаются всегда в верхнем регистре.

Сигналы подразделяются на сигналы высокой активности и низкой активности. Знак дефиса (-) на конце названия сигнала обозначает что этот сигнал является сигналом низкой активности. Сигнал низкой активности является истинным если сигнал ниже V_{IL} , и ложным, если сигнал выше V_{IH} . Без дефиса на конце имени сигнала обозначается сигнал высокой активности. Сигнал высокой активности является истинным если сигнал выше V_{IH} , и ложным если он ниже V_{IL} . Положительность обозначает, что сигнал совершает активный кругооборот к состоянию истинности. Отрицательность обозначает что сигнал совершает активный кругооборот к состоянию ложности. Освобожденный – обозначает что сигнал в настоящее время не совершает активных действий (см. п. 4.2.1). Некоторые сигналы имеют схематичный отклонения которые помещают сигнал в оба состояния (ложности или истинности), но при этом драйвер сигнала может активно проводить утверждение или отрицание сигнала. Контрольные сигналы которые могут быть использованы для более чем одной взаимоисключающей функции, определяются их названиями, разделенными двоеточием (например DIOW:-STOP).

3.2.6 Соглашения о битах

Названия битов показываются как все заглавные буквы, за исключением случаев когда прописная буква n предшествует названию бита. Если перед названием бита не стоит n, то если БИТ установлен в одно из доступных значений, то значение бита истинно, а если БИТ очищен до нуля, то значение бита ложно. Если названию бита предшествует n, то это означает, что если значение nBIT установлено в ноль, бит истинный, а если значение nBIT установлено в какое-то значение, бит ложный.

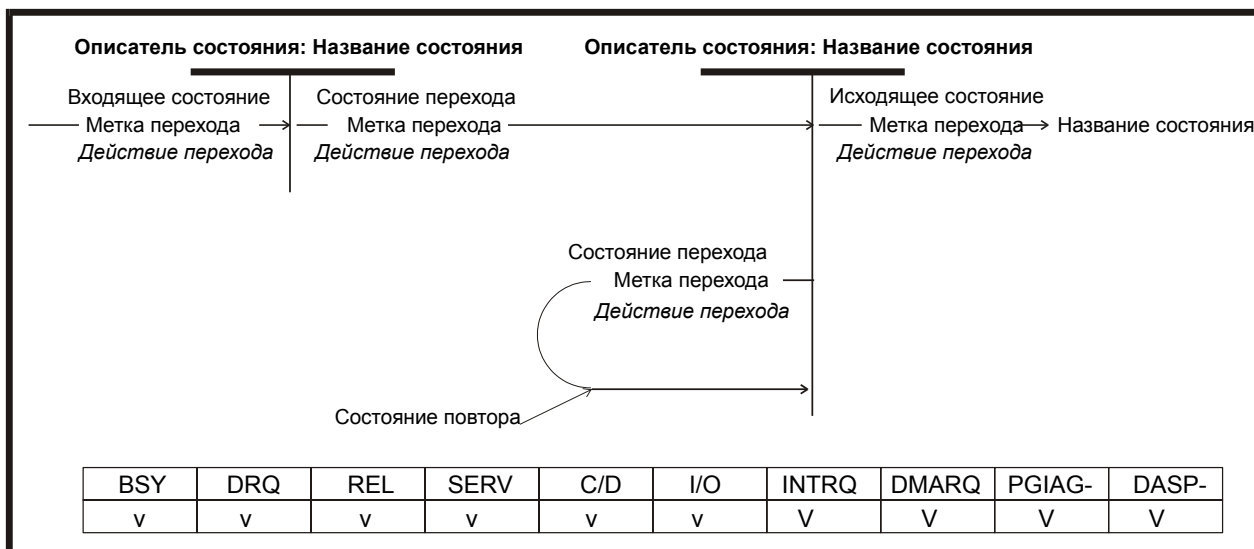


3.2.7 Соглашения о диаграммах состояния

Диаграммы состояния могут быть такими, как показано на рис. 1.

Каждое состояние идентифицируется по описателю состояния и имени состояния. Описатель состояния является уникальным и позволяет разграничить все состояния на всех диаграммах состояний в этом документе. Описатель состояния состоит из множества букв, которые наираются в верхнем регистре в заголовке рисунка, содержащего диаграмму состояния, следующую по уникальному номеру названия состояния. Название состояния – это короткое описание первичный действий, которые будут произведены в состоянии, и одно и то же название состояния может присутствовать в разных диаграммах. Если одна и та же первичная функция присутствует в других состояниях на той же диаграмме состояния, эти функции обозначаются разными буквами на конце

названия состояния. Могут также производиться дополнительные действия, если текущие состояния и акции описываются в тексте-описании состояния.



В диаграммах состояния командного протокола устройства, состояние битов и сигналов, которое изменяется в процессе использования текущей диаграммы состояния, показываются как описатели состояния: имя_состояния, и таблица, которая показывает состояние всех битов и сигналов в текущей диаграмме, как показано ниже:

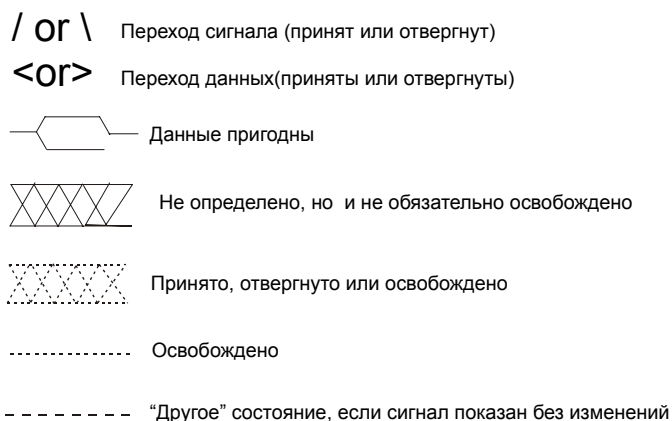
v = значение бита изменено.
 1 = бит установлен в единицу.
 0 = бит очищен в нуль.
 x = состояние бита безразлично.
 V = сигнал изменен.
 A = сигнал утвержден.
 N = сигнал отвергнут.
 R = сигнал освобожден.
 X = сигнал безразличен.

Каждый переход идентифицирован меткой перехода и состоянием перехода. Метка перехода состоит из описателя состояния, причем того состояния, из которого переход следует как описатель состояния будучи уже сделанным. В некоторых случаях, переход ко входу или выходу диаграммы состояния может происходить от одной диаграммы к другой, в зависимости от запущенной команды. В этом случае описатель состояния обозначен как хх. Состояние перехода – это краткое описание состояния или события, которое определяет переход и занимает или может включать действия перехода (показаны курсивом), что означает что переход состоялся. Такое действие полностью описывается в тексте описания перехода. После входа в состояние, все действия которые должны быть запущены в этом состоянии, будут запущены. Если состояние должно быть реинициализировано изнутри, все действия которые должны быть запущены в этом состоянии, будут повторены снова.

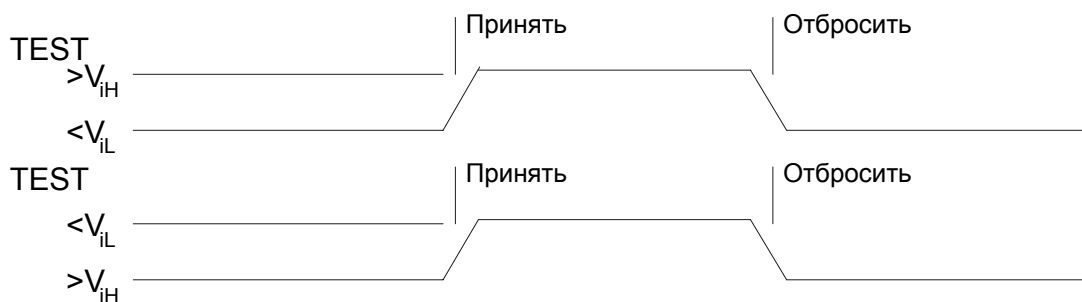
Принимается, что все действия, определенные для состояния, выполняются внутри состояния, и что переходы из состояния в состояние являются мгновенными.

3.2.8 Соглашения об установках времени

В диаграммах тайминга используются определенные символы. Эти символы и их текущие значения приведены ниже.



Все сигналы, показывающиеся с принятыми состояниями, сталкиваются с вершиной страницы. Отвергнутые состояния показываются в нижней части страницы, в противовес принятым состояниям. Интерфейс использует смесь негативных и позитивных сигналов для управления и данных. Понятия «принятый» и «отвергнутый» используются последовательно и являются независимыми от электрических характеристик. На всех временных диаграммах, нижняя линия указывает отвергнутые, а верхняя линия указывает принятые сигналы. Следующая иллюстрация показывает представление сигнала, названного TEST,двигающегося от отвергнутого к принятому и обратно к отвергнутому, на основании полярности сигнала.



3.2.9 Запрос байта для операций передачи

Данные передаются поблочно с использованием или PIO, или DMA протоколов. Протокол PIO используется когда бит BSY обнулен, а бит DRQ взведен в единицу. Эти передачи обучно являются 16-битными, но компакт-флеш может использовать 8-битную передачу. Данные передаются блоками по одному или более байту, такой блок известен как DRQ-блок. Передача данных по протоколу DMA происходит когда хост принимает DMACK- а в ответ устройство принимает DMARQ. Передача DMA – всегда 16-битная. Каждый принятый хостом DMACK- определяет DMA data burst (порцию данных DMA)⁵. Порция данных DMA состоит из двух или более байт. Принятый DRQ блок или порция данных DMA содержит n байт информации, байты обозначены как Байт (0) – и дальше до Байта (n – 1), где Байт (0) является первым байтом в блоке данных, а Байт (n – 1) – последним. В таблице 1 показан порядок байт в случае, если блок данных передается по протоколам PIO или DMA в 16-битном виде. На таблице 2 показан блок данных при передаче по 8-битному PIO-протоколу.

Таблица 1. Порядок байт

	DD 15	DD 14	DD 13	DD 12	DD 11	DD 10	DD 09	DD 08	DD 07	DD 06	DD 05	DD 04	DD 03	DD 02	DD 01	DD 00
Первая передача	Байт (1)								Байт (0)							
Вторая передача	Байт (3)								Байт (2)							
.....																
Последняя передача	Байт (n – 1)								Байт (n – 2)							

⁵ DMA data burst можно перевести как «Порция данных DMA». В этом документе я буду использовать этот термин [Прим. переводчика]

Таблица 2. Порядок байт

	DD 07	DD 06	DD 05	DD 04	DD 03	DD 02	DD 01	DD 00
Первая передача	Байт (0)							
Вторая передача	Байт (1)							
.....								
Последняя передача	Байт (n – 1)							

ПРИМЕЧАНИЕ – Описание выше определяет описание данных по интерфейсу. Хост-системы и/или хост-адаптеры могут работать с иным порядком байт, например, с таким, каоторый организован в их памяти. Некоторые параметры определяются как строки символов ASCII. Поля данных ASCII могут содержать только значения от 20h до 7Eh. Например, строка "Copyright", содержит первым символом «С», следующим «о» и так далее до конца. Если нужно передать все поля этого слова, порядок передачи будет следующим:

- 1 символ ("С") из битов DD (15:8) первого слова,
- 2 символ ("о") из битов DD (7:0) первого слова,
- 3 символ ("р") из битов DD (15:8) второго слова,
- 4 символ ("у") из битов DD (7:0) второго слова,
- 5 символ ("г") из битов DD (15:8) третьего слова,
- 6 символ ("и") из битов DD (7:0) третьего слова,
- 7 символ ("g") из битов DD (15:8) четвертого слова,
- 8 символ ("h") из битов DD (7:0) четвертого слова,
- 9 символ ("t") из битов DD (15:8) пятого слова,
- 10 символ ("пробел") из битов DD (7:0) пятого слова,

и так далее.