

# ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ФЛЭШ-ПАМЯТИ NAND

**ВЛАДИМИР ЛОПАТКИН**

*В статье рассматривается зависимость скорости записи и чтения флэш-памяти NAND от последовательности обращений к памяти и предшествовавшей записанной последовательности. Надеемся, этот материал поможет пользователям при выборе тестов для проверки требований к среде для хранения информации. Для лучшего понимания этой статьи необходимо знать принцип работы программы для очистки памяти.*

## ВВЕДЕНИЕ

В отличие от накопителей на жестких дисках, флэш-память NAND не всегда в состоянии выполнять операции за миллисекунды или менее того. Главной причиной такой неспособности является работа высокоприоритетной программы для очистки памяти, нагрузкой которой трудно прогнозировать. Таким образом, операции в режиме жесткого реального времени всегда требуют кэша большого объема в операционной системе.

Несмотря на то, что накопители на жестких дисках реализуют четко установленное соответствие (mapping) между физическими адресами памяти и логическими адресами, которые использует операционная система, эта функция отсутствует у флэш-памяти NAND. При осуществлении нескольких операций записи по одному и тому же логическому адресу данные каждый раз записываются по другому физическому адресу. Взаимосвязью между логическими и физическими адресами управляет микропрограмма с помощью таблиц,

которые тоже хранятся во флэш-памяти NAND. Данные этого типа, записанные на носитель при проведении испытания, не влияют на скорость выполнения.

Те контроллеры с функцией сжатия данных, которые применялись на заре появления флэш-памяти NAND, теперь не используются, благодаря чему можно обойтись без генерации псевдослучайных данных, требующей большой вычислительной мощности, что влияет на измерение скорости.

## СКОРОСТЬ ЗАПИСИ

На рисунке 1 показана зависимость скорости записи от типа доступа и предыдущих операций. В рассматриваемом случае объем данных номинальной емкости записывается на носитель семь раз; при этом в каждом цикле меняется метод доступа. Из рисунка 1 видно, как записываются данные на носитель без использования кэша большого объема.

### Участок 1: произвольная запись 4 КиБ<sup>1</sup>

В начале участка 1 память очищена: либо ее никогда прежде не использова-

ли для записи, либо из нее полностью удалили все данные («безопасно очистили»). Следовательно, любым логическим адресам не назначаются физические адреса – в таблицах преобразований отсутствуют записи. Среда хранения заполняется произвольно 4-КиБ блоками. Каждый логический адрес записывается только единожды. Скорость записи существенно ограничена из-за малого объема данных и больших затрат времени на запись таблиц распределения.

### Участок 2: произвольная запись 4 КиБ, та же адресная последовательность

На участке 2 доступы на право записи из участка 1 в точности повторяются: последовательность записанных адресов идентична. Видно, что скорость записи несколько уменьшилась. Это произошло потому, что таблицы отображения полностью заполнены, и каждую старую запись требуется объявить недействительной для каждого нового записанного адреса. Из-за последовательности идентичных адресов на программу чистки памяти не накла-



Рис. 1. Скорость записи

<sup>1</sup> В русскоязычной транскрипции — 1 КиБ (Кибитайт). Эта единица, сравнительно недавно введенная МЭК, призвана избавить от неоднозначности. В соответствии со стандартами МЭК, 1 Кбайт (Килобайт) = 1000 байт, а 1 КиБ = 1024 байт.

дываются дополнительные требования – при заполнении каждого нового блока флэш-памяти автоматически освобождается другой блок. Часто в техническом описании скорость на этом участке определяется как «непрерывная произвольная запись». Однако это маркетинговый показатель, который не дает представления о минимальной скорости записи, достигаемой только при максимальной нагрузке на программу очистки. Таким образом, следует исходить из особенностей каждого конкретного приложения.

### Участок 3: произвольная запись 4 КиБ, новая адресная последовательность

После заполнения среды хранения дважды одной и той же последовательностью произвольных доступов на запись адресная последовательность меняется в начале участка 3. Даже после записи данных небольшого объема скорость значительно падает из-за выделения дополнительных ресурсов для контроля над процессом (Overprovisioning, OP), и прежде неактивная программа очистки памяти стала работать на полную нагрузку.

Блоки больше не освобождаются автоматически при изменении адресной последовательности, и программа очистки памяти должна постоянно копировать данные, чтобы освободить блоки для новых данных. Поскольку каждый адрес по-прежнему описывается только единожды для каждого участка, скорость записи растет экспоненциально в течение короткого времени в конце участка, поскольку (при сокращении объема, когда по-прежнему недостает некоторого количества адресов) теперь при заполнении каждого нового блока снова начинают автоматически освобождаться блоки. После записи последнего блока этого участка все зарезервированные под нужды контроллера блоки автоматически освобождаются.

### Участок 4: последовательная запись 128 КиБ

В начале участка 4 последовательность произвольных адресов переключается на последовательность из последовательных адресов. Кроме того, объем данных из расчета на доступ увеличивается с 4 КиБ до 128 КиБ. В результате немедленно возрастает пропускная способность за счет большей эффективности передачи между хостом и средой хранения.

После выделения дополнительных ресурсов (OP) скорость записи падает почти до той же величины, что и на участке 3, т.к. все старые данные записывались с произвольными адресами, а нагрузка на программу чистки памяти возросла.

Поскольку эта программа по возможности сортирует адреса при перемещении данных, быстродействие увеличивается раньше при уменьшении количества адресов и автоматическом высвобождении блоков.

### Участок 5: последовательная запись 128 КиБ, второй проход

На участке 4 среда хранения заполнялась последовательно. Количество адресов данных в блоках флэш-памяти теперь монотонно растет в каждом блоке. Все блоки, которые использовались для резервирования, снова освобождаются, поскольку каждый адрес снова был в точности записан. На участке 5 среда хранения заполняется последовательно, как на участке 4.

Поскольку последовательность адресов та же, что и на предыдущем участке, и каждый блок флэш-памяти записан, каждый последующий блок освобождается автоматически. Благодаря большому объему данных, приходящихся на один доступ на запись хоста, передача информации между хостом и накопителем данных является очень эффективной. Программа очистки памяти не загружена, а контроль над таблицами распределения создает небольшую дополнительную нагрузку при последовательном доступе на запись. На этом участке достигается максимальная скорость записи в накопитель.

Только у пустой среды хранения (после обработки или безопасного удаления) скорость может немного вырасти.

### Участки 6–7: произвольная запись 4 КиБ по произвольному адресу

В начале участка 6 снова выбирают последовательности произвольных адресов. В отличие от участков 1–3, где каждый адрес записывался только один раз, все адресное пространство теперь открыто для каждого нового адреса. Таким образом, на обоих участках, с одной стороны, не все логические адреса требуется перезаписать, а, с другой стороны, несколько адресов

записываются несколько раз. После заполнения исходного резерва (OP) программа очистки памяти начинает работать на полную нагрузку. Поскольку блок флэш-памяти никогда автоматически не освобождается, эта программа постоянно работает с полной нагрузкой. В таком рабочем состоянии осуществляется действительно непрерывная произвольная запись с минимальной скоростью. Однако на практике это состояние едва наблюдается, главным образом, потому что число циклов записи и стирания среды хранения очень быстро достигает установленной спецификацией величины.

### КЭШ ПСЕВДО-SLC

Чтобы увеличить скорость записи, в промышленных системах хранения данных стал применяться кэш псевдо-SLC – технология для приложений потребительской электроники. Ее суть заключается в конфигурации части массива NAND в качестве памяти SLC, в одной ячейке которой хранится только один бит. Скорость записи и чтения у этой памяти очень велика.

Поскольку это запоминающее устройство не является узкоспециализированным, как SLC-память, его называют псевдо-SLC (pSLC). Такой кэш можно использовать для запоминающих устройств всех типов, в которых хранится несколько бит в одной ячейке (MLC, TLC, QLC). На рисунке 2 показана типичная скорость записи среды хранения с кэшем псевдо-SLC.

При быстром заполнении кэша pSLC скорость существенно падает, поскольку для последующего доступа для записи в среду хранения требуется освободить pSLC, переместив находящиеся данные из кэша в память TLC. Использование кэша pSLC увеличивает скорость записи, особенно если среда хранения не занята доступами на чтение или запись между записью данных большого объема. Эти интервалы бездействия используются средой для перемещения данных из кэша в область TLC.

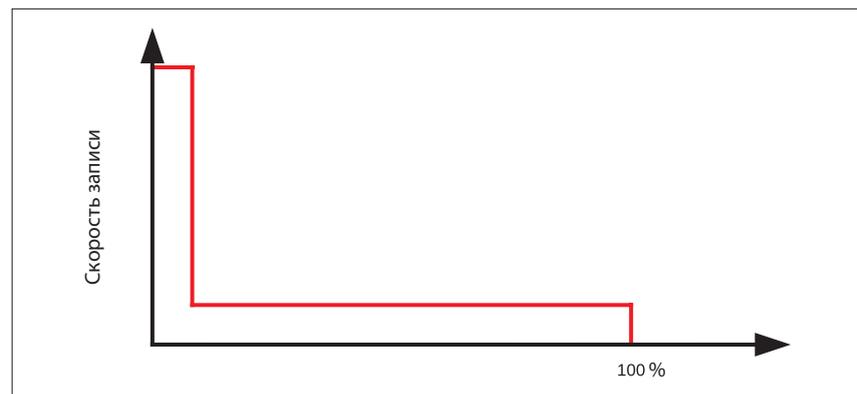


Рис. 2. Последовательная запись с использованием кэша pSLC

Перечислим еще два преимущества кэша рSLC.

- Этот кэш повышает устойчивость к сбоям питания, т. к. они не могут повредить уже имеющиеся данные при записи в рSLC благодаря тому, что в нем хранится только один бит на ячейку. При сбое питания во время перемещения данных в память TLC они по-прежнему находятся в кэше.
- Он улучшает защиту данных в случае появления поврежденных блоков, особенно при использовании с памятью 3D NAND, а также позволяет уменьшить сложность решений и смягчить требования RAM к контроллеру.

При классификации флэш-памяти NAND необходимо учитывать наличие кэша рSLC, т. к. непрерывная запись данных большого объема может существенно уменьшить скорость записи. В качестве подходящего теста рекомендуется последовательно и без прерываний заполнять среду хранения до 100% (см. рис. 2).

#### ДИНАМИЧЕСКИЙ КЭШ рSLC

Технология динамического кэша рSLC стала применяться и в промышленных решениях для хранения данных. В отличие от статического кэша рSLC, характеристика которого показана на рисунке 2, до 100% флэш-памяти NAND используются динамически в качестве рSLC-кэша в зависимости от степени заполнения

среды хранения, что вызывает осложнения по следующим причинам.

- Скорость записи среды хранения зависит не только от количества данных, записанных без прерываний, но и от степени заполнения памяти. В результате скорость записи становится непредсказуемой.
- Динамическое изменение конфигурации блоков флэш-памяти в виде рSLC или TLC-памяти ухудшает надежность приложений, особенно в промышленном диапазоне рабочей температуры. Допускается разве что одно изменение из рSLC в MLC-память, но оно осуществляется только в пределах 1% разрешенных рSLC-циклов.
- Все производители динамических NAND-накопителей переходят на использование статического кэша после нескольких циклов программы и стирания. До этого наилучшие результаты достигались при проведении коротких испытаний на скорость, в которых емкость полностью не заполнялась. Спустя небольшое время эксплуатации скорость среды хранения начинает постоянно уменьшаться. Если это обстоятельство не учитывается, в реальных условиях эксплуатации могут возникнуть проблемы. На рисунке 3 показана характеристика TLC-устройства хранения с динамическим кэшем в начале срока службы (сплошная линия) и по прошествии как минимум 10% циклов (пунктирная линия).

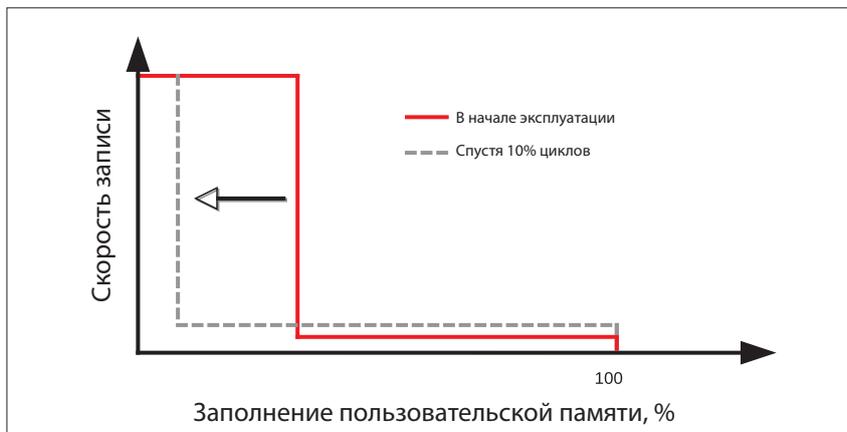


Рис. 3. Внезапное уменьшение скорости записи носителя с динамическим рSLC-кэшем

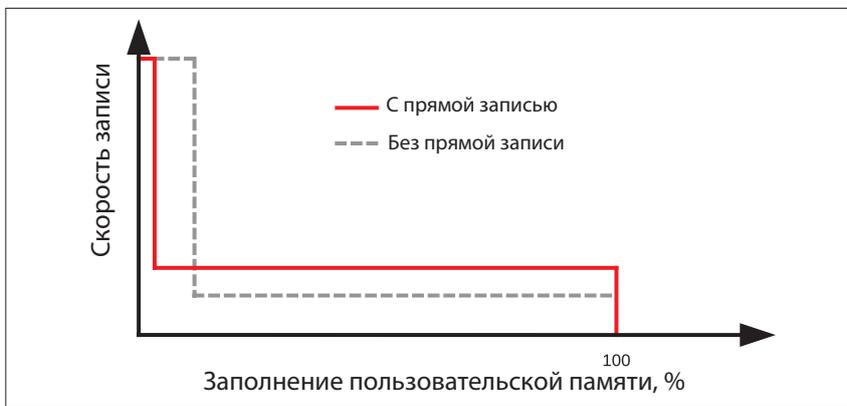


Рис. 4. Переход на прямую запись при обнаружении последовательной записи

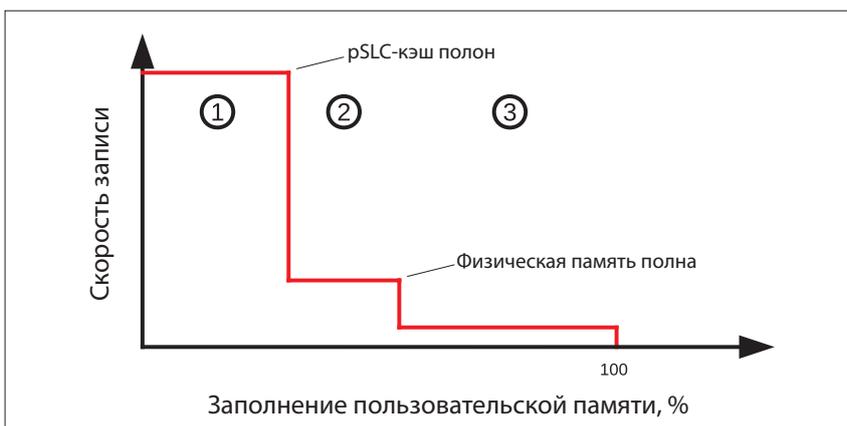


Рис. 5. Прямая запись с динамическим рSLC-кэшем

#### ПРЯМАЯ ЗАПИСЬ

Т. н. «прямая запись» осуществляется в обход рSLC-кэша. Она используется в следующих случаях.

- Если непрерывный доступ на запись к среде хранения осуществляется чаще всего последовательно, можно предположить, что через непродолжительное время израсходуется весь объем рSLC-кэша. В этом случае некоторые запоминающие устройства, применяемые в промышленных приложениях, не используют рSLC-кэш и осуществляют запись непосредственно в TLC-память. В результате уменьшается износ этого кэша, и возрастает скорость записи в случае данных очень большого объема, поскольку они не должны записываться дважды.

Недостатком такого способа записи, однако, является то, что устройство в меньшей мере защищено от сбоев питания. Поскольку данные большого объема, как правило, используются только на этапе установки операционной системы, этот недостаток считается приемлемым. Преимущество же

в том, что установка осуществляется быстрее. На рисунке 4 показана разница между скоростями последовательной записи в двух рассматриваемых случаях.

– Прямая запись применяется в комбинации с динамическим рSLC-кэшем с целью в максимальной мере замедлить ликвидацию и сокращение объема увеличенного рSLC-кэша (при испытании скорости). На рисунке 5 иллюстрируется один такой случай. Среда хранения последовательно заполняется и имеет динамический рSLC-кэш, на чью долю приходится около  $\frac{3}{4}$  физической NAND-памяти в начале теста, т.е.  $\frac{1}{4}$  номинальной емкости TLC NAND. Если в конце раздела 1 рSLC-кэш полностью заполнится, среда хранения в разделе 2 переходит на медленную прямую запись. После вычитания рSLC-кэша  $\frac{1}{4}$  номинальной емкости остается в качестве TLC-памяти для прямой записи. Если эта емкость тоже заполнится, истощится физическая память.

- $\frac{3}{4}$  физической TLC-емкости используются как SLC-память, но она может хранить данные объемом только в  $\frac{1}{4}$  от номинальной емкости;
- поскольку  $\frac{1}{4}$  физической емкости по-прежнему используется как TLC-память, для хранения можно задействовать  $\frac{1}{4}$  от номинальной емкости.

Теперь среда для хранения в разделе 3 должна преобразовать динамическую часть рSLC-кэша в TLC-память, чтобы снова освободить физическую память. Скорость записи при этом, соответственно, низкая.

### СКОРОСТЬ ЧТЕНИЯ

Скорость чтения не зависит от того большого множества факторов, что скорость записи. Она зависит только от того, какой тип доступа был записан и как он читается. Данные в разделе 1

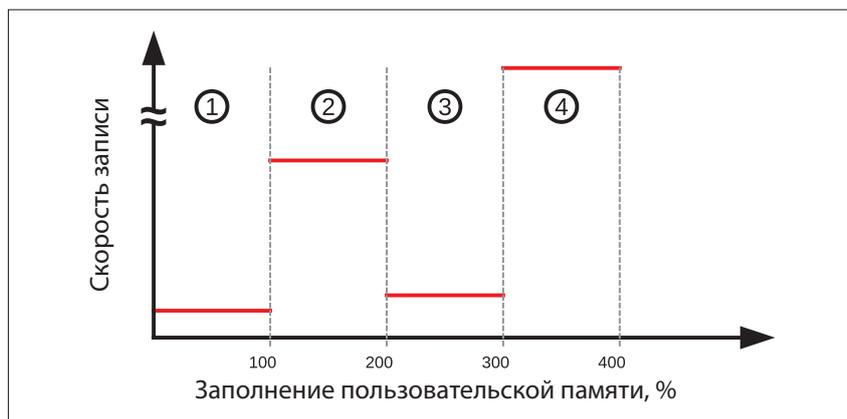


Рис. 6. Скорость чтения

были записаны произвольно и снова прочтены произвольно, но в другом порядке (см. рис. 6).

В рассматриваемом случае обеспечивается наименьшая скорость чтения, поскольку очень много усилий затрачивается на поиски в таблицах распределения, а эффективность передачи данных между средой хранения и хостом низка из-за того, что размер пакета составляет 4 КиБ.

В разделе 2 данные, записанные произвольным образом, читаются последовательно. Скорость чтения намного выше, т.к. данные объемом 128 КиБ теперь передаются хосту при каждом доступе, благодаря чему возрастает эффективность. И в этом случае затрачивается немало ресурсов на поиски в таблицах распределения.

До раздела 3 запись в среду хранения осуществлялась последовательно. Доступы на чтение в этом разделе были произвольными. Скорость чтения, соответственно, низка, поскольку размер пакета равен 4 КиБ. Однако последовательная запись упрощает и ускоряет поиск в таблицах отображения по сравнению с разделом 1.

В разделе 4 в среде хранения достигается максимальная скорость чтения. Последовательно записанные данные теперь читаются последовательно

в 128-КиБ пакетах. Поиск в таблицах распределения осуществляется очень просто, передача в хост эффективна, а внутренний доступ к флэш-памяти NAND может выполняться как чтение с упреждением.

У среды хранения с рSLC-кэшем более высокая скорость наблюдается для некоторых адресов памяти в четырех разделах, а именно в тех случаях, если эти адреса находятся в рSLC-кэше.

### ВЫВОДЫ

Для корректного сравнения скоростей в технических описаниях флэш-памяти NAND необходимо исходить из идентичных условий проведения испытаний. Если измерение не осуществлялось повторно или не охватило все пространство логических адресов, указанная в спецификации скорость может оказаться значительно выше фактически достижимой. При определении производительности флэш-памяти NAND или возможности использовать ее в критичных по времени приложениях рекомендуется тщательно выполнить симуляцию максимальных пиковых нагрузок, последовательности обращений к памяти в конкретном приложении и протестировать полученную модель в течение продолжительного времени. ▬