

# ПРОЕКТИРОВАНИЕ И ВЕРИФИКАЦИЯ КОНТРОЛЛЕРОВ SSD-ДИСКОВ

Авдеев Н. А., Глебов С. К., Заливако С. С., Иванюк А. А., Пинчуков Д. А., Семенов В. В.,  
Цибулько Ю. В.

ООО «СК хайникс мемори солишнс Восточная Европа»

Минск, Республика Беларусь

E-mail: {nikolai.avdeev, sergey.glebov, sergey.zalivako, alexander.ivaniuk, dmitry.pinchukov, vladimir.semenov,  
yury.tsybulka}@sk.com

*Потребность в системах хранения данных в настоящее время непрерывно растёт. Направление SSD-дисков в системах хранения данных в последнее десятилетие развивается наиболее интенсивно. С ростом объема хранимых данных на одном диске и скорости передачи данных растет и сложность контроллеров SSD-дисков, реализующих протокол передачи данных с компьютером и обеспечивающих запись и считывание данных с NAND флеш-памяти. В данной работе рассматриваются основные особенности проектирования и верификации современных контроллеров SSD-дисков, на основе опыта разработки таких устройств резидентом ПВТ ООО «СК хайникс мемори солишнс Восточная Европа» (SKHMS E).*

## ВВЕДЕНИЕ

Современный мир невозможно представить без использования информационных технологий и глобальной сети Интернет. Как известно, объём хранимых данных в Интернете экспоненциально растёт и поэтому спрос на системы хранения данных в мире сохранится в будущем. В последнее десятилетие технология твёрдотельных накопителей (Solid State Drive, SSD – далее SSD-диск) активно совершенствуется. SSD-диски имеют ряд преимуществ перед HDD-дисками: более быстрый отклик на действия пользователя, беззвучная работа, компактные размеры, меньшее энергопотребление, отсутствие механических подвижных частей, что способствует большей удароустойчивости. Хранение данных в SSD-диске происходит в 3D NAND флеш-памяти [1], представляющей собой электрически стираемое перезаписываемое постоянное запоминающее устройство. Запоминающая ячейка флеш-памяти основана на МОП-транзисторе с плавающим затвором. Накопитель устроен таким образом, что ячейки памяти располагаются не только в одной плоскости, но и вертикально друг над другом, составляя несколько десятков уровней [2]. В современных накопителях одна такая ячейка способна сохранять до четырех бит данных [1]. Скорость записи данных достигает 3,2 ГБайт/с, скорость чтения – 3,5 ГБайт/с [3].

Компания SK hynix является одним из немногих производителей флеш-накопителей в мире, который может обеспечить полный цикл разработки и производства всех необходимых компонент внутри компании, а именно – разработку и производство 3D NAND флеш-памяти, DRAM, контроллеров и программного обеспечения накопителей, вспомогательного программного обеспечения.

История компании SK берет свое начало в 1983 году. Компания, которая в наши дни известна как третий по продажам производитель полупроводниковых приборов в мире, начинала с разработки и производства динамической памяти. На данный момент компания также разрабатывает и производит NAND флеш-память, КМОП-сенсоры, SSD и мобильные накопители на базе флеш-памяти и имеет множество исследовательских центров и производственных мощностей по всему миру.

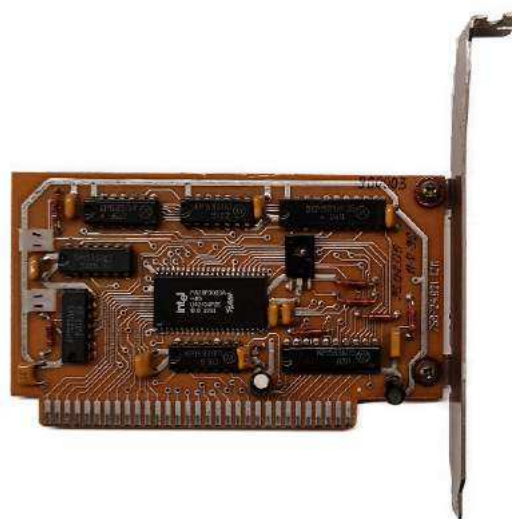


Рис. 1 – SSD-накопитель

Для обеспечения полного цикла разработки накопителей компания создала четыре исследовательских центра, которые расположены в Южной Корее, Соединенных Штатах Америки, Тайване и Беларуси. Отдельно стоит отметить, что появление исследовательского центра в Беларуси не случайно. Еще до прихода SK hynix на белорусский рынок, в стране уже существовала команда специалистов, имеющих опыт в создании

программного обеспечения для накопителей на базе NAND флеш-памяти, а первый накопитель на базе флеш-памяти был разработан и выпущен на территории Беларуси еще в 1996 году (см. рис. 1).

Процесс разработки флеш-накопителя может занять не один год и состоит из множества этапов работ, выполняемых командами четырех исследовательских центров. Белорусский центр принимает участие в таких этапах разработки как, проектирование аппаратуры контроллеров накопителей, формальная и функциональная верификация контроллеров, валидация контроллеров, разработка программного обеспечения накопителей, верификация программного обеспечения накопителей, разработка алгоритмов для последующего их использования в программном или аппаратном обеспечении флеш-накопителей. Кроме этого, белорусский офис отвечает за разработку сервисного программного обеспечения накопителей, ведет исследовательские работы и занимается подготовкой кадров при университетах Беларуси. В частности, белорусское подразделение SK hynix явилось инициатором организации совместной учебной лаборатории с БГУИР.

Как уже упоминалось, процесс разработки флеш-накопителя имеет большую продолжительность в силу его сложности. Контроллер SSD-диска представляет собой систему на кристалле (СнК), содержащую более 10 процессорных ядер и десятки аппаратных ускорителей, а интерфейсы работают на частотах в десятки гигагерц. Как следствие, все перечисленные этапы разработки имеют критическое влияние как на время выхода конечного продукта на рынок, так и на минимизацию рисков ошибок в программном и аппаратном обеспечении. Рассмотрим эти этапы подробнее.

## I. УСТРОЙСТВО SSD-КОНТРОЛЛЕРА

Контроллер SSD-диска (SSD-контроллер) отвечает за управление данными, хранящимися в NAND флеш-памяти и обеспечивает связь с другими устройствами (хостом), будь то компьютер, цифровая камера или мобильный телефон. SSD-контроллер имеет сложную архитектуру СнК, в состав которой входят (см. рис. 2):

- хост-контроллер, включающий интерфейс протокола передачи данных (USB, Ethernet, PCIe, Unipro, CXL);
- процессорная подсистема, обычно состоящая из нескольких вычислительных ядер общего назначения (ARM, CXC, RISC-V);
- подсистема NAND флеш-контроллера;
- контроллеры внешней памяти различных типов (DDR, SDRAM);
- энергонезависимая память и/или средства долговременного хранения конфигурации СнК (EEPROM или флеш-память);

- широкий набор контроллеров вспомогательных периферийных интерфейсов (UART, SPI, I<sup>2</sup>C, GPIO, и т. п.);
- средства управления доменами частот и частотами, средства управления доменами питания, интеллектуальная логика сброса устройств и доменов;
- средства отладки и анализа производительности (DFT, JTAG, а также собственные интерфейсы);
- комплексная многоуровневая система коммуникации элементов в составе СнК (AXI, AHB, APB интерфейсы).

Хост-контроллер обеспечивает связь с внешним устройством по стандартному протоколу. В настоящее время наиболее используемым протоколом для SSD-дисков является NVMe-протокол [4] через PCIe интерфейс, а для мобильных устройств – UFS на базе UniPro [5].

Контроллер флеш-памяти обеспечивает запись и чтение данных в NAND флеш-накопитель, контроль количества циклов чтения/записи, обработку ошибок и прочие операции с данными. Чаще всего для ускорения операций чтения и записи флеш-контроллер является многоканальным и обеспечивает параллельную работу сразу с несколькими флеш-накопителями.

Для хранения данных в контроллере используется DDR-память, где сохраняются данные, полученные от хоста перед записью их во флеш-память, а также данные, вычитанные из флеш-памяти перед отправкой их на хост. Для хранения встроенного программного обеспечения (Firmware, FW) служит SRAM-память и загрузочная энергонезависимая память.

## II. ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ SSD-КОНТРОЛЛЕРА

Задача функциональной верификации заключается в том, чтобы доказать, что модель СнК или сложно-функционального блока (СФ-блока) позволяет успешно выполнять на ней требуемые задачи во всех допустимых условиях и режимах, предусмотренных спецификацией. Функциональная верификация SSD-контроллера разделяется на несколько уровней:

- уровень СФ-блоков (периферийные блоки, коммутаторы и сети, вычислительные процессоры, акселераторы и пр.);
- уровень подсистем (таких как хост и флеш-контроллеры, процессорная подсистема);
- уровень всей СнК.

Верификация на уровне СФ-блоков проводится с использованием классических UVM тестирующих программ [6]. На этом уровне максимально проверяется функционирование блока в соответствии с его спецификацией, получая 100% покрытие кода и полное функциональное покрытие. В большинстве случаев при этом время мо-

делирования всех тестов для СФ-блока не превышает нескольких часов.

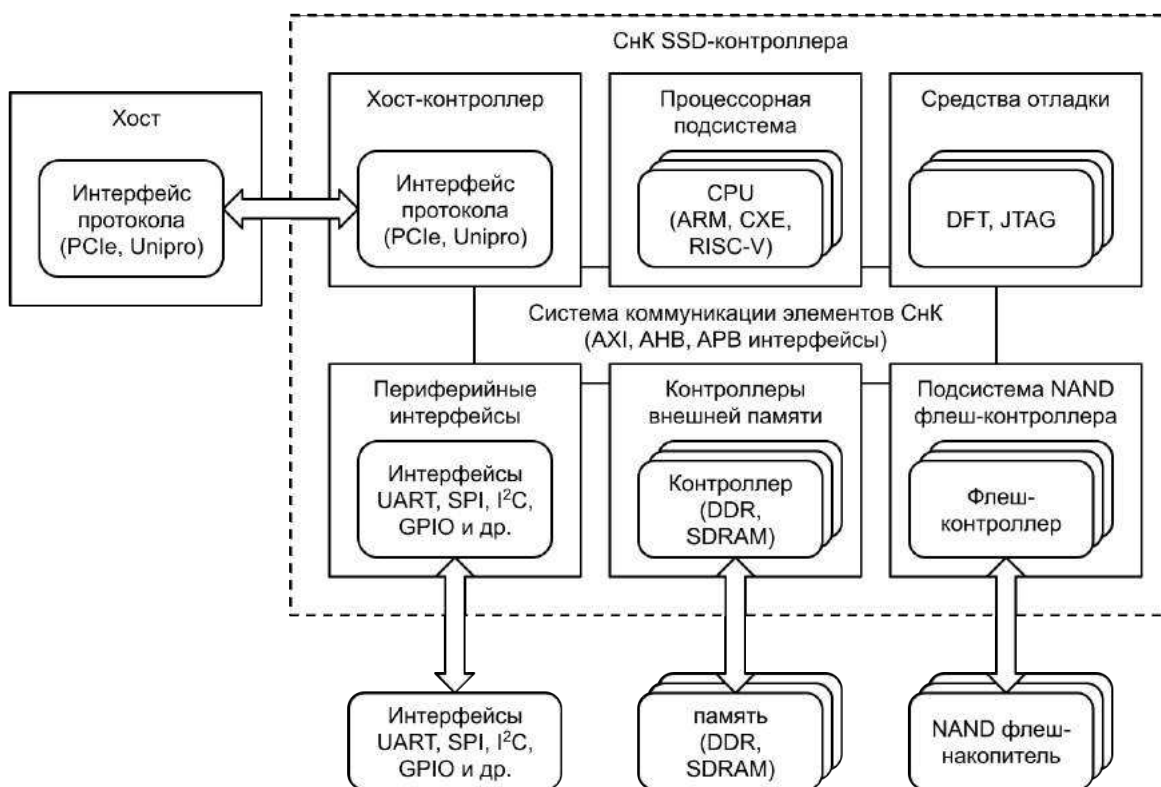


Рис. 2 – Типовая блок-схема СнК SSD-контроллера

При верификации на уровне подсистемы проверяется взаимодействие СФ-блоков между собой, а также работа основных функций, выполняемых подсистемой. Например, на уровне хост-контроллера следует проверить поддерживаемые команды протокола передачи данных, причем акцент делать на командах, которые имеют аппаратную реализацию в подсистеме, а не программную. Также нужно проконтролировать все сценарии взаимодействия встраиваемого ПО (со стороны процессора и других связанных подсистем) и хост-контроллера, проверить доступ к регистрам всех внутренних блоков, а также установку и сбрасывание всех прерываний, обработать сценарии сбросов подсистемы. На данном уровне время моделирования одного теста может занимать несколько часов, что значительно увеличивает требуемые вычислительные ресурсы для проведения ежедневных и еженедельных запусков регрессионного тестирования.

На уровне СнК верифицируется в первую очередь правильность взаимодействия всех блоков и подсистем, проверяется доступ ко всем регистрам в системе, проверяется базовый функционал, а также оценивается быстродействие системы при выполнении основных функций СнК. Особое внимание уделяется динамическим и формальным проверкам переходов между асин-

хронными доменами, подсистемам управления питанием и генераторам частот. Длительность моделирования одного теста на данном уровне варьируется от нескольких часов до нескольких дней, поэтому как только начинают проходить базовые тесты на данном уровне, то сразу приступают к следующему этапу верификации – валидации с помощью FPGA (например, на основе платформы прототипирования HAPS от фирмы Synopsys [18], а также ко-симуляции с помощью программно-аппаратных ускорителей моделирования, таких как Veloce [7], Palladium [8], Zebu [9].

### III. СТАНДАРТИЗАЦИЯ ПРОЦЕССА ВЕРИФИКАЦИИ

Можно выделить несколько основных аспектов, которые нужно учитывать, чтобы процесс функциональной верификации был максимально эффективным.

**Высокая скорость создания тестирующих программ (testbench)**, что достигается использованием стандартной структуры директорий проекта, переиспользование общих исходных кодов (базовые классы, пакеты, агенты, makefile и скрипты для запуска компиляции и симуляции проекта, и прочее), разработка собственных Verification intellectual property (VIP), а так-

же использование генераторов тестового окружения. Немаловажным является обмен знаниями и постоянное обучение инженеров.

**Автоматизация создания и тестирования регистровых моделей и моделей памяти** обеспечивается использованием специализированных генераторов, таких как Semifore CSRCompiler [10], позволяющих по одному исходному описанию регистров системы получать описание регистров для RTL в виде Verilog-кода, для HDL-моделирования в виде UVM регистровой модели, в виде заголовочных файлов для разработки встроенного ПО, а также документацию в различных форматах.

**Верификация каждого отдельного СФ-блока**, входящего в СнК, позволяет достичь 100% покрытия кода, сократить время моделирования и проверить максимальное число тестовых сценариев, что не отменяет проведение верификации на других уровнях: подсистем и СнК.

**Параметризация тестирующей программы (Testbench)** для возможности переиспользования в мульти-блочном режиме (когда тестируются несколько однотипных блоков одновременно). Также, при разработке тестирующей программы, необходимо учитывать возможность переиспользования эталонной модели СФ-блока на более высоком уровне (подсистемы или СнК), для этого она должна разрабатываться с учетом возможности работы в пассивном режиме, т. е. следить только за внешними портами блока и не «подсматривать» за внутренними сигналами блока – верификация «чёрного ящика».

**Максимально возможное документирование кода и поддержание документации в актуальном состоянии.** Для выполнения этой задачи необходимо использование общих шаблонов для документации, это позволяет стандартизировать и упростить процесс ее разработки. Также следует использовать средства для совместной разработки документации, например, Confluence [11], Mediawiki [12]. Использование программ документирования кода, таких как Doxygen [13], Natural Docs [14], WaveDrom [15], позволяет автоматизировать обновление документации.

**Разработка верификационного плана** обязательна. Необходимо унифицировать требования к описанию тестовых сценариев, их созданию и отслеживанию, определить обязательный набор метрик, организовать автоматический сбор всех видов покрытия и статуса выполнения тестов, а также сделать вывод статистики для последующего анализа состояния проекта. Можно выделить следующие основные задачи планирования верификации:

- определить набор свойств (функций) и тестов для верификации на каждом уровне (СФ-блоков, подсистемы, СнК);

- расставить приоритеты покрытия свойств, т. к. выделенного на верификацию времени может быть недостаточно для покрытия всех свойств;
- определить методы и средства верификации для разных групп свойств, при этом следует исключить дублирования верификации одних и тех свойств разными тестами, определиться какие свойства какими средствами могут быть проверены (динамические, статические тесты, FPGA) и на каких уровнях верификации какие из свойств проверять (уровень СФ-блоков, подсистемы, СнК);
- очень важно организовать взаимодействие между инженерами и командами, работающими на разных этапах и уровнях верификации;
- определить критерии успешного окончания верификации.

Критериями завершения верификации могут являться: выполнение всех пунктов верификационного плана, прохождение всех тестов без ошибок, достижение заданного уровня покрытия кода и функционального покрытия, успешное прохождение случайных тестов во время регрессионного тестирования длительное время.

Оценка функционального и кодового покрытия позволяют:

- оценить полноту верификации проекта;
- определить, какие существующие тестовые сценарии можно расширить для повышения покрытия;
- определить, какие дополнительные тесты необходимо разработать для повышения покрытия;
- своевременно определять «избыточные» тесты (которые не увеличивают покрытие) и исключать их;
- удостовериться в том, что тест проверяет необходимое свойство;
- оценить эффективность случайного теста.

**Выполнение регулярного регрессионного тестирования** позволяет своевременно выявить возникновение ошибки в базе проекта, а запрет на внесение изменений в базу проекта без прохождения специальной, обычно достаточно быстрой, регрессии защищает базу исходных кодов проекта от внесения ошибок. Для осуществления автоматического запуска регрессий могут использоваться программы Jenkins [16], BuildBot, Enterprise Manager (Cadence).

**Эффективное использование вычислительных ресурсов**, таких как серверы (ядра), оперативная память и лицензии на САПР и VIP, достигается использованием программ распределения задач в сети удалённых серверов, таких как RTDA, LSF и др.

**Использование Verification intellectual property (VIP)** позволяет сократить сроки ве-

рификации и уменьшить вероятность ошибок в проекте. При верификации SSD-контроллеров в основном используются VIP для протоколов PCIe, NVMe, UniPro, UFS и др. VIP могут включать как набор UVM-агентов, упрощающих разработку тестов, так и сами наборы тестов для выполнения полной проверки верифицируемого блока. Использование покупных VIP связано с рядом особенностей: число доступных лицензий обычно ограничено высокой стоимостью VIP, что может существенно влиять на длительность регрессии; при разработке SSD-контроллеров с поддержкой новых ревизий стандартов протоколов высока вероятность, что сроки обновления VIP не будут согласованы со сроками выполнения проекта и соответствующий VIP может содержать ошибки в реализации новых функций протокола; сроки исправления ошибок в VIP могут варьироваться от нескольких дней до месяцев. В связи с вышесказанным в долгосрочной перспективе целесообразно разрабатывать собственные VIP для верификационных компонентов, особенно для широко используемых в SnK протоколов, таких как UART, SPI, AXI, ANV, APB и прочих. Собственные VIP имеют открытый код, а значит обеспечивают более быстрое выявление и исправление ошибок, по сравнению с покупным VIP. В общем случае лучше выбирать различных разработчиков VIP и соответствующего IP (intellectual property) в SnK, например, если в проекте используется IP PCIe интерфейса от фирмы Synopsys, то лучше отдать предпочтение VIP от другой фирмы.

#### IV. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НАКОПИТЕЛЯ

Одним из основных компонентов флеш-накопителя, управляющим его работой, является программное обеспечение, известное как FTL (Flash Translation Layer).

FTL флеш-накопителя обеспечивает надежное и целостное хранение данных в NAND флеш-памяти путем реализации как базовых элементов FTL, таких как преобразование логических адресов (LBA) в физические адреса, сборщик мусора (Garbage Collection, GC), контроль равномерности износа NAND флеш-памяти (Wear-Leveling, WL) [17], управление поврежденными блоками NAND флеш-памяти, так и более сложных, связанных с алгоритмами коррекции ошибок в хранящихся данных и другие.

Помимо функциональных к современным флеш-накопителям предъявляются жесткие требования по характеристикам производительности, энергопотребления и QoS (Quality of Service). Обеспечение выполнения всех этих требований является достаточно сложной задачей, которая также решается программным обеспечением путем усложнения существующих алгорит-

мов, внедрения новых и балансировкой их совместной работы.

К флеш-накопителю могут предъявляться специфические требования, не оговоренные или не определенные стандартами, например, изменение поведения флеш-накопителя в разных температурных режимах его работы, сбор определенных статистических данных внутри флеш-накопителя с возможностью доступа к ним извне, особые требования по защите информации и ограничению доступа к данным накопителя. Все это также реализуется средствами программного обеспечения.

#### V. ВЕРИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НАКОПИТЕЛЯ

Отдельная команда в SK hynix отвечает за программную верификацию – с использованием симулятора накопителя собственной разработки. Для верификации создаются модели (hardware models), позволяющие моделировать любое оборудование, внедрять ошибки (например, в NAND или RAM) и исполнять код на всех процессорах моделируемого накопителя или их подмножестве – их число может достигать до нескольких десятков для серверных SSD.

Одного симулятора и моделей недостаточно для масштабной верификации, также необходимы инструменты для отладки кода, необходимо автоматизировать управление тестовыми компаниями (это тысячи машин, на которых выполняется моделирование десятков тысяч тестов), также нужно как можно более упростить работу программиста, производящего анализ упавших тестов, т. к. при этом приходится иметь дело с огромным количеством кода – это сотни тысяч строк: встроенное ПО, симулятор, модели, тесты и т. д. Для этого разрабатывается ПО, позволяющее группировать возникшие проблемы и связывать эти группы с проблемами, выявленными ранее.

#### VI. ВАЛИДАЦИЯ ПРОГРАММНОЙ И АППАРАТНОЙ ЧАСТЕЙ НАКОПИТЕЛЯ

Одним из этапов разработки флеш-накопителя является валидация аппаратных модулей SnK, на базе которых построен накопитель. Данный процесс можно условно поделить на две части: валидация на FPGA и валидация непосредственно готового устройства (или его ревизий). На основе технических требований к SnK определяется и утверждается набор тестовых сценариев для всех аппаратных модулей. Эти сценарии реализуются в специализированном встроенном программном обеспечении. Тестовый фреймворк позволяет в автоматическом режиме запускать наборы тестов различных подсистем SnK. Выявленные дефекты на этапе валидации на FPGA исправляются проектировщиками SnK и перепроверяются последую-

щим регрессионным тестированием. Поскольку не весь аппаратный функционал может быть протестирован на FPGA, после изготовления СнК, производится тестирование непосредственно интегральной микросхемы. Это делается с помощью того же (или незначительно модифицированного) специализированного ПО. Дефекты, которые выявляются в готовых СнК исправляются в последующих его ревизиях. Кроме валидационного ПО для тестирования модулей и подсистем СнК используется различный аппаратный и программный инструментарий: логические и протокол-анализаторы, осциллографы, программно-аппаратные эмуляторы такие, как Velocе, тестовые фреймворки и системы как сторонних производителей, так и разрабатываемые внутри компании.

## VII. ИССЛЕДОВАНИЯ В ОБЛАСТИ АППАРАТУРЫ КОНТРОЛЛЕРОВ NAND ФЛЕШ-ПАМЯТИ

В настоящее время в компании проводятся исследования различных структурных элементов контроллеров флеш-памяти. Результаты исследований, как правило, внедряются в продукты SK hynix, а также защищаются с помощью патентов США. Рассмотрим основные направления исследований.

**Аппаратные ускорители.** Существуют аппаратные решения на основе твердотельных накопителей, которые специально предназначены для хранения баз данных [19]. В связи с этим в компании были разработаны решения для аппаратной поддержки различных операций в системах управления базами данных, которые позволяют повысить производительность выполнения данных операций с незначительными аппаратными затратами (менее 1% общей аппаратуры контроллера). Также была предложена система управления памятью контроллера, которая позволяет динамически перераспределять нагрузку с одних типов памяти (например, статические ОЗУ) на другие (например, флеш-память) [20]. Для более детального анализа разрабатываемого аппаратного и программного обеспечения ускорителей были разработаны методы их отладки [21], позволяющие уменьшить зависимость от существующих дорогостоящих решений. В настоящее время одним из направлений исследований является разработка новых архитектур аппаратных ускорителей алгоритмов, реализованных на уровне встроенного ПО [22].

**Безопасность накопителей информации на основе флеш-памяти.** Проблемы безопасности твердотельных накопителей представляют интерес как для исследователей, так и для производителей данного класса устройств [23]. В соответствии с данным направлением в компании был разработан ряд аппаратных реше-

ний для уменьшения уязвимостей в различных структурных элементах контроллера накопителей информации на основе флеш-памяти. Генераторы случайных числовых последовательностей (ГСЧП) являются одним из важнейших элементов протоколов безопасности. В компании был разработан источник случайности общего назначения, основанный на асинхронном триггере и инверторе, а также генератор на базе чтения элементов флеш-памяти без коррекции ошибок [24]. В связи с необходимостью хранения части таблицы соответствия логических и физических адресов на стороне хоста в накопителях информации для мобильных устройств, был представлен метод кодирования и декодирования записей таблицы с помощью физически неклонированных функций (ФНФ) [25]. ФНФ были использованы в процессе улучшения аппаратуры для скремблирования данных, обладающей уязвимостями, которые приводят к более интенсивному износу накопителя информации [27]. Также был предложен аппаратный метод уникальной идентификации интегральных схем флеш-памяти, основанный на сравнении числа единиц при чтении различных страниц.

**Надежность элементов флеш-памяти.** Особенностью накопителей информации на основе NAND флеш является низкая надежность элементов памяти, уменьшающаяся с увеличением разрядности одной ячейки [28]. В настоящее время большинство накопителей на основе флеш-памяти построено на элементах, способных хранить три бита информации (Triple Layer Cell, TLC). Для улучшения надежности элементов памяти был предложен метод скремблирования, позволяющий сделать износ элементов более равномерным [26]. Также надежность элементов памяти обеспечивается путем сбора и анализа диагностической информации на уровне встроенного ПО. Основной проблемой сбора данной информации является многопроцессорность и гетерогенность контроллеров NAND флеш-памяти. Актуальным направлением в области повышения надежности накопителей является проектирование и разработка аппаратных методов самотестирования и диагностики элементов памяти и аппаратуры контроллера [29].

**Анализ потока входных команд контроллера.** Одним из актуальных направлений в развитии архитектуры контроллеров накопителей информации на основе флеш-памяти является анализ входного потока команд (Workload) [30]. В данном направлении был предложен алгоритм анализа и прогнозирования потока команд с целью оптимизации работы контроллера [31]. Также для решения проблемы фрагментации логических и физических адресов памяти был предложен алгоритм кластеризации входных команд [32]. Было представле-

но аппаратно-программное решение, основанное на компактном представлении входного потока команд. Предложенный подход основан на использовании рекуррентных автокодировочных нейронных сетей (Recurrent Autoencoder Neural Networks) и позволяет преобразовать поток команд в вектор, состоящий из вещественных чисел. Данное решение позволяет производить классификацию потоков входных команд с целью самотестирования устройства памяти, а также настройки параметров встроенного ПО для улучшения производительности и надежности. Перспективным направлением в данной области является внедрение методов машинного обучения для анализа входного потока команд с целью реконфигурации аппаратуры контроллера SSD.

**Организация работы с патентными заявками.** Компания SKHMS E с 2016 года подготовила 41 патентную заявку, 23 из которых были опубликованы в открытом доступе, а 18 – находятся на стадии оформления. По 8 заявкам уже получены патентные свидетельства. Процесс патентования технических решений состоит из нескольких этапов:

1. предварительное обсуждение патентной идеи авторов с экспертами;
2. разработка предварительной заявки, которая содержит краткое описание предметной области и основной идеи, а также детальное описание технической реализации;
3. обсуждение предварительной заявки в расширенной экспертной группе;
4. рецензирование заявки патентным бюро в головной компании SK hynix;
5. оформление патентной заявки юридическим бюро в США;
6. подача заявки в патентное бюро США (United States Patent and Trademark Office, USPTO);
7. опубликование заявки в открытом доступе;
8. экспертиза заявки патентным бюро США;
9. оформление патентного свидетельства.

Весь процесс от предложения идеи до получения патентного свидетельства может занимать от 2 до 5 лет.

В компании в процесс разработки патентных заявок вовлечены все департаменты, а также технические писатели. Успешные патентные заявки материально поощряются компанией.

## VIII. Совместная учебная лаборатория в БГУИР

В сентябре 2018 года на факультете компьютерных систем и сетей БГУИР была открыта совместная учебная лаборатория SKHMS E [33]. Территориально лаборатория расположена в аудитории 402 корпуса №5. Возглавляет лабораторию профессор кафедры информатики Иванюк А.А.

Деятельность лаборатории нацелена на привлечение заинтересованных студентов, магистрантов и аспирантов в исследовательские и инженерные проекты в области проектирования контроллеров запоминающих устройств на основе NAND флеш-памяти. В первый год работы лаборатории ведущими специалистами SKHMS E были прочитаны открытые лекции по следующим направлениям: верификация аппаратуры контроллеров SSD, проектирование и отладка встроенного ПО контроллеров SSD, исследование и разработка специализированных алгоритмов, позволяющих улучшать основные характеристики SSD-накопителей, контроллеры SSD на основе микропроцессорных систем.

В 2019 году в лаборатории были прочитаны специализированные курсы по особенностям разработки ПО для моделирования, тестирования и верификации встроенного ПО, по архитектурным особенностям микропроцессоров и их применению в составе контроллеров SSD, теоретический курс по алгебраической теории помехоустойчивых кодов, применяемых в запоминающих устройствах на основе NAND флеш-памяти. Кроме этого проведены практические занятия по реализации базовых алгоритмов управления NAND флеш-памятью с применением отладочных плат на основе ARM-процессоров.

В начале 2020 года были проведены лекционные занятия по основам машинного обучения и возможности его применения в управлении NAND флеш-накопителями. В этом же году был реализован студенческий проект по сборке и запуску полноценного SSD-накопителя с применением платы быстрого прототипирования Cosmos/Cosmos+ FPGA Platform (см. рис. 3) [34]. Данная плата, основу которой составляет программируемая логическая интегральная схема Xilinx FPGA Zynq-7000, предназначена для совместного проектирования как аппаратной, так и программной части контроллера SSD. С целью вовлечения большего числа студентов в процесс проектирования контроллеров SSD были проведены лекции по основам технологий программируемых логических интегральных схем, HDL-языкам описания цифровой аппаратуры и системам ее автоматизированного проектирования, основам проектирования цифровых систем с использованием софт-процессоров. Для реализации студенческих проектов в лаборатории имеются платы быстрого прототипирования Digilent NEXYS4 [35] и Zybo Z7 [36], лицензионный пакет САПР Vivado/Vitis от Xilinx [37].

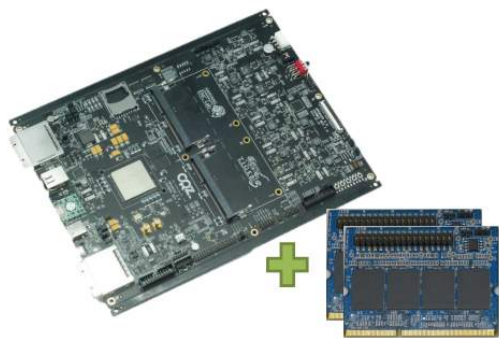


Рис. 3 – Плата быстрого прототипирования Cosmos/Cosmos+ FPGA Platform

В связи с пандемией COVID-19 учебная лаборатория перешла на индивидуальное взаимодействие со студентами, магистрантами и аспирантами. Каждый обучаемый в лаборатории вовлечен в академический исследовательский проект. По мере необходимости работы с оборудованием для слушателей выделен специальный график посещения лаборатории. Список тематических проектов разнообразен и включает в себя такие темы как:

- исследование конфигурируемых возможностей микропроцессорного ядра RISC-V;
- исследование и реализация алгоритмов скремблирования данных для SSD;
- исследование и аппаратно-программная реализация специализированных кодов коррекции пакетных ошибок;
- сравнение различных стратегий управления GC (Garbage Collection) для NAND флеш-памяти;
- исследование и прототипирование новых методов обеспечения защиты хранящихся данных, встроенного ПО и аппаратуры SSD-накопителей от несанкционированного доступа и использования.

Студенты совместной лаборатории учатся работать с англоязычной технической и научной литературой, работать над постановкой и проведением экспериментов, анализом полученных результатов. Помимо этого, студенты приобретают навыки участия в научно-технических конференциях, тренируются излагать свои мысли и результаты в форме тезисов докладов и при выступлениях на соответствующих секциях. Большинство обучаемых в лаборатории используют свои наработки в курсовом и дипломном проектировании, выбирают соответствующие исследовательские направления при обучении в магистратуре и аспирантуре.

В ближайшее время в лаборатории планируется открыть новое направление по верификации аппаратуры контроллеров SSD с изучением языка SystemVerilog и методологии UVM [38].

За время работы лаборатории в ней прошли обучения более 30 студентов с различных специальностей БГУИР. Пять из них уже успешно ра-

ботают на инженерных должностях в SKHMS E, двое из которых продолжают взаимодействие по своим академическим проектам, помогая другим студентам.

В лабораторию приглашаются студенты, магистранты и аспиранты, заинтересованные в изучении принципов функционирования современных запоминающих устройств на основе NAND флеш-памяти и технологий проектирования/верификации аппаратуры и встроенного программного обеспечения контроллеров SSD-дисков.

## IX. ЗАКЛЮЧЕНИЕ

Разработка современных SSD-дисков является комплексной задачей, для успешного решения которой в SK Hynix развиваются сразу несколько научно-технических направлений: развитие технологии изготовления флеш-памяти, стандартизация процесса проектирования и верификации чипов SSD-контроллеров и встраиваемого ПО, непрерывное проведение исследовательской работы по всем связанным направлениям, взаимодействие и поддержка образовательных программ для подготовки высококвалифицированных инженеров.

## X. СПИСОК ЛИТЕРАТУРЫ

1. Evolution of 4D NAND Flash Opens the Era of Terabyte Smartphones [Electronic resource] – Publ. date 25.09.2019. – Mode of access: <https://news.skhynix.com/evolution-of-4d-nand-flash-opens-the-era-of-terabyte-smartphones/>. – Date of access: 20.10.2021.
2. Park, J-W. A 176-Stacked 512 Gb 3b/Cell 3D-NAND Flash with 10.8 Gb/mm<sup>2</sup> Density with a Peripheral Circuit Under Cell Array Architecture / J-W. Park [et al.] // IEEE international solid-state circuits conference: proc. of the Intern. Conf., 13-22.02.2021.
3. SK hynix Gold P31 M.2 NVMe SSD Review: High-Performance, Unprecedented Efficiency [Electronic resource] – Publ. date 26.09.2021. – Mode of access: <https://www.tomshardware.com/reviews/sk-hynix-gold-p31-m2-nvme-ssd-review>. – Date of access: 20.10.2021.
4. NVM Express [Electronic resource] – Mode of access: <https://nvmexpress.org>. – Date of access: 20.10.2021.
5. JEDEC UFS & MIPI UniPro: Enabling Storage for Mobile, Automotive, and Other Applications [Electronic resource] – Mode of access: <https://www.mipi.org/sites/default/files/JEDEC-UFS-MIPI-UniPro-Webinar-Enabling-Storage-Mobile-Automotive.pdf>. – Date of access: 20.10.2021.
6. Rosenberg, S. A Practical Guide to Adopting the Universal Verification Methodology (UVM) / S. Rosenberg, K. Meade. – CA, USA : Cadence Design Systems, 2010. – 275 p.
7. Veloce HW-Assisted Verification System [Electronic resource] – Mode of access: <https://eda.sw.siemens.com/en-US/ic/veloce/>. – Date of access: 20.10.2021.
8. Emulation: High performance hardware and software verification and debug of complex SoCs and Systems [Electronic resource] – Mode of access: <https://www.cadence.com/en-US/home/tools/system-design-and-verification/emulation-and-prototyping/palladium.html>. – Date of access: 20.10.2021.



9. Emulation: Industry's Fastest Emulation Systems [Electronic resource] – Mode of access: <https://www.synopsys.com/verification/emulation.html>. – Date of access: 20.10.2021.
10. What Semifore Does [Electronic resource] – Mode of access: <https://semifore.com/about-semifore/>. – Date of access: 20.10.2021.
11. Confluence [Electronic resource] – Mode of access: <https://www.atlassian.com/software/confluence>. – Date of access: 20.10.2021.
12. MediaWiki: серебряная пуля или швейцарский нож? / С. Фомин // Открытые системы. СУБД, № 3, 2009 [Electronic resource] – Publ. date 30.04.2009. – Mode of access: <https://www.osp.ru/os/2009/03/8162918>. – Date of access: 20.10.2021.
13. Doxygen: Generate documentation from source code [Electronic resource] – Mode of access: <https://www.doxygen.nl/index.html>. – Date of access: 20.10.2021.
14. Natural Docs: Readable Source Code Documentation for 21 Programming Languages [Electronic resource] – Mode of access: <https://www.naturaldocs.org/>. – Date of access: 20.10.2021.
15. WaveDrom: Rendering Beautiful Waveforms from Plain Text / A. Chapyzenka, J. Probell [Electronic resource] – Mode of access: [https://wavedrom.com/images/SNUG2016\\_WaveDrom.pdf](https://wavedrom.com/images/SNUG2016_WaveDrom.pdf). – Date of access: 20.10.2021.
16. Jenkins [Electronic resource] – Mode of access: <https://www.jenkins.io>. – Date of access: 20.10.2021.
17. Ming-Chang, Y. Garbage collection and wear leveling for flash memory: Past and futur / Y. Ming-Chang [et al.] : proc. of the 2014 International Conference on Smart Computing / SMARTCOMP. – 2015. – Pp. 66-73.
18. Гибридная верификация процессоров Baikal: косимуляция с FPGA-платформой прототипирования Synopsys HAPS-80 [Электронный ресурс] / «Навр». – Москва, 2021. – Режим доступа: <https://habr.com/ru/company/baikalelectron/blog/582782/>. – Дата доступа: 20.10.2021.
19. Park, J.-H. SaS: SSD as SQL Database System / J.-H. Park, S. Choi, G. Oh, S.-W. Lee // Proc. VLDB Endowment. – 2021. – Vol. 14, № 9. – pp. 1481–1488.
20. Microprocessor-based system memory manager hardware accelerator [Electronic resource] : pat. app. US20210255956A1 / A. Kuyel. – Publ. date 19.08.2021. – Mode of access: <https://patents.google.com/patent/US20210255956A1>. – Date of access: 20.10.2021.
21. Method of debugging hardware and firmware of data storage [Electronic resource] : pat. US10970442B1 / A. Kuyel. – Publ. date 06.04.2021. – Mode of access: <https://patents.google.com/patent/US10970442B1>.
22. Dally, W. J. Domain-Specific Hardware Accelerators / W. J. Dally, Y. Turakhia, S. Han // Comm. ACM. – 2020. – Vol. 63 № 7. – pp. 48–57.
23. Hassell, J. The importance of SSD security and data encryption [Electronic resource]. – Mode of access: <https://insights.samsung.com/2021/06/10/the-importance-of-ssd-security-and-data-encryption/>. – Samsung, 2021. – Date of access: 20.10.2021.
24. Raw read based physically unclonable function for flash memory [Electronic resource] : pat. app. US20210055912A1 / S. S. Zalivaka, A. A. Ivaniuk – Publ. date 25.02.2021. – Mode of access: <https://patents.google.com/patent/US20210055912A1>.
25. Encoder and decoder using physically unclonable functions [Electronic resource] : pat. app. US20210226772A1 / S. S. Zalivaka, A. A. Ivaniuk – Publ. date 22.07.2021. – Mode of access: <https://patents.google.com/patent/US20210226772A1>.
26. Data scrambler for memory systems and method thereof [Electronic resource] : pat. app. US20210303715A1 / S. S. Zalivaka, A. A. Ivaniuk – Publ. date 30.09.2021. – Mode of access: <https://patents.google.com/patent/US20210303715A1>.
27. Data scramblers with enhanced physical security [Electronic resource] : pat. app. US20210326490A1 / S. S. Zalivaka, A. A. Ivaniuk – Publ. date 21.10.2021. – Mode of access: <https://patents.google.com/patent/US20210326490A1>.
28. Cai, Y. Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques / Y. Cai [et al.] // IEEE Int. Symp. High Perf. Comp. Arch. (HPCA'17) – Austin, USA, 2017. – pp. 49–60.
29. Lai, K. H. Why Next-Generation NAND Flash Requires a Dedicated Test Solution [Electronic resource]. – Mode of access: <https://www3.advantest.com/documents/11348/11d6ba1a-7d3f-4f66-acf3-cd9d1167f924>. – Advantest, 2013. – Date of access: 20.10.2021.
30. Seo, B. IO Workload Characterization Revisited: A Data-Mining Approach / B. Seo [et al.] // IEEE Trans. on Comp. – 2014. – Vol. 63 № 13. – pp. 3026–3038.
31. Workload prediction in memory system and method thereof [Electronic resource] : pat. US10802761B2 / A. Charnevich, S. S. Zalivaka – Publ. date 13.10.2020. – Mode of access: <https://patents.google.com/patent/US10802761B2>.
32. Workload clusterization for memory system and method of executing the same [Electronic resource] : pat. app. US20190361628A1 / Y. Yaromenka, A. Charnevich, J. M. Koo, S. S. Zalivaka – Publ. date 28.11.2019. – Mode of access: <https://patents.google.com/patent/US20190361628A1>.
33. В ВГУИР открылась новая лаборатория в партнерстве с IT-компанией SK hynix memory solutions Eastern Europe [Электронный ресурс]. – Режим доступа: <https://www.bsuir.by/ru/news/101971-v-bguir-otkrylas-novaya-laboratoriya-v-partnerstve-s-it-kompaniy-sk-hynix>. – Дата доступа: 20.10.2021.
34. OPEN-SOURCE SOLID-STATE DRIVE PROJECT FOR RESEARCH AND EDUCATION [Electronic resource]. – Mode of access: <http://openssd.io/>. – Date of access: 20.10.2021.
35. Nexys 4 Artix-7 FPGA Trainer Board [Electronic resource]. – Mode of access: <https://digilent.com/shop/nexys-4-artix-7-fpga-trainer-board/>. – Date of access: 20.10.2021.
36. Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board [Electronic resource]. – Mode of access: <https://digilent.com/shop/zybo-z7-zynq-7000-arm-fpga-soc-development-board/>. – Date of access: 20.10.2021.
37. Vivado ML Overview [Electronic resource]. – Mode of access: <https://www.xilinx.com/products/design-tools/vivado.html>. – Date of access: 20.10.2021.
38. UVM Verification Primer [Electronic resource]. – Mode of access: <https://www.doulos.com/knowhow/systemverilog/uvm/uvm-verification-primer/>. – Date of access: 20.10.2021.