ПРИБОРЫ И ТЕХНИКА ЭКСПЕРИМЕНТА, 2027, № 1

ЛАБОРАТОРНАЯ
ТЕХНИКА

УДК 621.396.6

МОДИФИКАЦИЯ МЕТОДА ДВОЙНОЙ БУФЕРИЗАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ НЕПРЕРЫВНОГО СБОРА ДАННЫХ

©2027 г. С. Ю. Ксенофонтов*, А. В. Купаев

Институт прикладной физики им. А.В. Гапонова-Грехова
Российской академии наук

Россия, 603951, Нижний Новгород, ул. Ульянова, 46

**E-mail: xen@ipfran.ru*

Поступила в редакцию 17.07.2026 г.
После доработки 07.10.2026 г.
Принята к публикации 12.10.2026 г.

Предложена модификация метода двойной буферизации, которая была успешно использована в составе модуля сбора данных и управления скоростной мобильной портативной системы спектральной оптической когерентной томографии. Данный способ промежуточной аппаратной буферизации обеспечил непрерывность скоростного (со скоростью до 305 МБ/с) потока исходных данных через интерфейс USB 3.0 в течение длительного времени (до нескольких часов) в условиях использования мобильных и встраиваемых компьютеров и операционных систем общего назначения.

Ключевые слова: непрерывный сбор данных, двойная буферизация, асинхронный сбор данных, высокоскоростная передача данных

1. ВВЕДЕНИЕ

Методы промежуточной аппаратной буферизации данных, описанные в этой статье, применялись в ходе работ по созданию мобильных портативных систем спектральной оптической когерентной томографии (ОКТ) [1]. ОКТ – это способ визуализации внутренней структуры биотканей. Благодаря высокому разрешению и малой инвазивности ОКТ нашла широкое применение в биомедицинских исследованиях и в клинической диагностике [2].

Необходимость использования предложенных методов была вызвана следующими конструктивными особенностями разрабатываемых спектральных ОКТ-систем.

Основные функции обработки, управления и визуализации в этих ОКТ-системах исполняет стандартный персональный компьютер (ПК). В отличие от стационарных ОКТ-систем, где обычно используются настольные ПК, в составе мобильных портативных ОКТ-систем для этих целей используются ноутбуки или встраиваемые компьютерные модули.

Специально разработанная управляющая пользовательская программа, которая непосредственно исполняет функции обработки, управления и визуализации, в описываемом случае работает в условиях стандартной операционной системы (ОС) общего назначения. Это решение, с одной стороны, обеспечивает удобство в эксплуатации, так как используется привычный для пользователя программный интерфейс и предоставляются широкие возможности использования различных готовых стандартных и специализированных программ для обработки и документирования полученной информации. С другой стороны, это решение продиктовано доступностью драйверов используемых интерфейсных контроллеров для сбора данных и управления.

Для связи программной и аппаратной частей ОКТ-системы был выбран интерфейс USB. Универсальность такого конструктивного решения обеспечивает независимость разрабатываемого программного обеспечения от особенностей архитектуры используемого ПК (мобильного или встраиваемого).

В процессе работы (сканирования) данная ОКТ-система осуществляет непрерывный сбор исходных данных, синтез соответствующего потока изображений и их визуализацию в реальном времени. Эти процедуры должны исполняться с минимальными задержками для обеспечения интерактивности манипуляций пользователя. Стоит отметить высокую вычислительную сложность алгоритмов синтеза спектральных ОКТ-изображений [3–10]. Поэтому процессы сбора данных, управления и визуализации в данном случае происходят на фоне интенсивной параллельной загрузки ресурсов центрального процессора и оперативной памяти ПК [11, 12].

Таким образом, основным требованием к модулю сбора данных и управления спектральной ОКТ-системы является необходимость обеспечения высокоинтенсивного потока исходных данных через интерфейс USB в ПК в условиях нормального функционирования драйверов и сервисов ОС общего назначения и интенсивного фонового использования вычислительных ресурсов. При этом главная задача состоит в минимизации задержек и потерь исходных данных.

2. КОНСТРУКЦИЯ МОДУЛЯ СБОРА ДАННЫХ И УПРАВЛЕНИЯ

Источником исходных данных, из которых в дальнейшем математически вычисляется распределение коэффициента обратного рассеяния в исследуемой ткани, в спектральной

ОКТ является сенсор линейного сканирования. В данном случае – это ключевой элемент спектрометрического узла, который синхронно с поперечным сканированием регистрирует спектральное распределение результата интерференции обратно рассеянного излучения и опорного излучения. Последовательность, в которой количество отсчетов соответствует числу фоточувствительных элементов сенсора линейного сканирования, а каждый из этих отсчетов пропорционален интенсивности излучения, зарегистрированной соответствующим элементом за один период экспозиции, в спектральной ОКТ принято называть исходным А-сканом. Определенная совокупность исходных А-сканов составляет исходные данные двумерного томографического среза (В-скана).

Таким образом, скорость потока исходных данных в спектральной ОКТ-системе определяется техническими характеристиками используемого сенсора линейного сканирования. В различных модификациях ОКТ-систем применялись различные сенсоры. Число элементов в них было от 512 до 2048. Производительность этих сенсоров варьировалась от 20 000 А-сканов/с до 92 000 А-сканов/с. В случае, когда скорость потока исходных данных превышала 25 000 000 Б/с, для связи с ПК приходилось использовать интерфейс USB 3.0.

Ранее для решения таких задач был предложен и разработан модуль сбора данных и управления, описанный в работе [13]. Структурная схема этого модуля представлена на рис. 1.

Рис. 1. Структурная схема модуля сбора данных и управления спектральной ОКТ-системой с интерфейсом USB 3.0

Подробнее назначение и порядок работы элементов схемы (рис. 1) были описаны в работе [13]. В рамках настоящего исследования важны только следующие конструктивные особенности этой схемы.

В схеме на рис. 1 основное формирование связей, управляющих сигналов и коммутация осуществляются за счет применения программируемой логической матрицы серии Cyclone III (FPGA Cyclone III EP3C16F484I7N). Посредством двух межплатных разъемов PLD-40 она соединяется с модулем CYUSB3KIT-003, который содержит чип контроллера USB-устройства CYUSB3014. Таким образом реализуется интерфейс USB 3.0 для связи с управляющим ПК.

Поток исходных данных, который поступает в модуль сбора данных и управления посредством интерфейса LVDS от сенсора линейного сканирования, также направлен в чип FPGA, где он переформатируется для промежуточной буферизации.

В качестве промежуточного буфера в схеме (рис. 1) используются два чипа статической памяти с произвольным доступом CY7C1381KV33-133AXI. Эти микросхемы имеют параллельный интерфейс. В данном случае они использовались в режиме адресации 524 288

четырехбайтных слов. Таким образом, емкость каждой из них составила 2 МБ. В схеме (рис. 1) каждый из этих чипов выполняет функцию FIFO-буфера с одним мультиплексированным параллельным портом ввода-вывода.

Передача данных в USB-интерфейс осуществляется за счет записи в автоматизированный канал DMA контроллера CYUSB3014. Эта процедура осуществляется через 32-разрядную шину GPIF II с тактовой частотой 100 МГц. Стоит отметить, что внутренняя память, обслуживающая канал DMA контроллера CYUSB3014, была сконфигурирована в виде 8 сегментов по 16 кБ каждый. Это было связано с особенностями страничной организации внутренней памяти контроллера CYUSB3014 и с особенностями функционирования драйвера контроллера CYUSB3014 в условиях ОС общего назначения. Экспериментально было установлено, что такой вариант сегментирования обеспечивал максимальную среднюю скорость передачи через USB-интерфейс в ПК. За счет этого гарантировалось, что сразу после сигнала о том, что канал DMA свободен для записи, 16 кБ данных будут записаны в DMA-память с максимально возможной скоростью (без задержек). Поэтому единичные транзакции чтения или записи в промежуточный буфер (CY7C1381KV33-133AXI) всегда осуществлялись порциями по 16 кБ, т.е. читалось или записывалось 4096 32-разрядных слов.

3. СТАНДАРТНЫЙ МЕТОД ДВОЙНОЙ БУФЕРИЗАЦИИ

Особенность функционирования ОС общего назначения, в отличие от функционирования ОС реального времени, заключается в том, что не гарантирован строгий временной лимит реакции системы на программные или аппаратные запросы. Теоретически (а иногда практически) такое время задержки в критических случаях может становиться бесконечным. В случае ОС общего назначения можно говорить только о тех или иных вероятностях задержек, связанных с различными обстоятельствами в виде чрезмерной загрузки вычислительных ресурсов или каналов передачи данных в ходе исполнения различных фоновых задач.

В случае использования интерфейса USB ситуация усугубляется тем, что передача данных осуществляется сравнительно маленькими пакетами (в случае USB 3.0 – пакетами размером 1 кБ), и на каждую такую транзакцию центральный процессор так или иначе отвлекается. Поэтому интенсивные фоновые вычисления могут существенно снизить среднюю скорость передачи данных по USB-каналу в течение существенного промежутка времени. Также на это естественным образом влияет текущая загрузка системной шины ПК другими аппаратными событиями.

Чтобы избежать потерь в непрерывном потоке исходных данных, в таких условиях обычно применяется двойная буферизация [14].

В одном из первых вариантов модуля сбора данных [13] авторами применялся стандартный вариант двойной буферизации. Такой архитектурный подход характерен тем, что используются как минимум два независимых блока памяти. Пока один буфер обрабатывается (запись или чтение), второй буфер доступен для выполнения других операций, что обеспечивает непрерывность потока данных

В работе [13] метод двойной буферизации был реализован следующим образом. Назовем одну из микросхем CY7C1381KV33-133AXI на схеме рис. 1 “буфер А”, а другую “буфер Б”. Тогда с учетом применяемого сегментирования DMA-памяти в контроллере CYUSB3014 такую последовательность действий можно представить в виде блок-схемы, представленной на рис. 2.

Рис. 2. Блок-схема последовательности действий при реализации стандартного метода двойной буферизации, использованного в работе [13]

Алгоритм работы, как показано на рис 2, строится на непрерывной смене ролей двух буферов: буфера А и буфера Б. Система записывает новую порцию данных в буфер А. Одновременно с этим модуль чтения считывает уже готовые и актуальные данные из буфера Б. По завершении операции оба буфера меняются ролями (переключаются): система начинает писать данные в буфер Б, а данные из буфера А отправляются на обработку.

Практическое применение такого метода двойной буферизации показало, что в случае использования высокопроизводительных сенсоров линейного сканирования удовлетворительное (не слишком большое) количество потерь исходных данных при длительном сканировании (несколько десятков секунд) достигается только при использовании мощных ПК настольной или серверной архитектуры, или дорогих ноутбуков игрового класса. Для объяснения причины этого, а также для простоты дальнейшего изложения в качестве примера рассмотрим практический случай использования сенсора линейного сканирования со средней скоростью потока исходных данных

$$V_1 = 3.19488 \cdot 10^8 \text{ Б}/c.$$

Пусть V_2 – это средняя скорость передачи данных в ПК. Принципиальная работоспособность метода двойной буферизации достижима только когда $V_1 < V_2$. В нашем случае чаще всего это условие выполняется. При этом с определенностью можно сказать только, что

$$V_2 < 4 \cdot 10^8 \text{ Б/с}.$$

Это связано с характеристикой шины GPIF II в контроллере CYUSB3014.

Другими словами, в терминах блок-схемы на рис. 2 каждая из транзакций записи 16 кБ осуществляется со скоростью $3.19488 \cdot 10^8$ Б/с. Задержки между транзакциями записи можно

считать нулевыми. Транзакции чтения 16 кБ осуществляются со скоростью $4 \cdot 10^8$ Б/с. Однако задержками между транзакциями чтения пренебречь нельзя, и они иногда достигают критических величин.

В силу изложенных выше причин существуют промежутки времени, когда средняя скорость передачи данных в ПК через USB-интерфейс сильно снижается. Чтобы описать это явление, назовем временем полной латентности канала передачи промежуток времени T_L, в течение которого средняя скорость передачи равняется нулю, несмотря на наличие запроса на передачу. При этом время заполнения буфера

$$T_1 \approx 6.56 \cdot 10^{-3} \text{ с.}$$

Время “опустошения” буфера

$$T_2 > 5.24 \cdot 10^{-3} \text{ с.}$$

Таким образом, чтобы не было потерь данных, необходимо, чтобы сумма промежутков времени полной латентности на каждом из интервалов времени T_1 удовлетворяла условию

$$\sum_{t=0}^{T_1} T_L(t) < (T_1 - T_2) \approx 1.32 \cdot 10^{-3} \text{ с.}$$

Другими словами, в каждом из последовательных промежутков времени T_1 средняя скорость записи не должна превышать среднюю скорость чтения. В течение меньших промежутков времени незначительное превышение возможно, если оно не нарушает предыдущего условия.

Практика показала, что вероятность нарушения этих условий достаточно велика и что отсутствие таких событий при использовании большинства мобильных или встраиваемых платформ невозможно.

Существует очевидный экстенсивный метод решения этой проблемы: либо увеличение емкости используемых чипов статической памяти, либо каскадное увеличение количества используемых чипов в два или более раз, однако такой метод был признан неприемлемым, так как он влечет за собой неоправданную задержку визуализации (более чем на один B-скан) и неоправданное усложнение топологии платы модуля сбора данных и управления.

4. МОДИФИКАЦИЯ МЕТОДА ДВОЙНОЙ БУФЕРИЗАЦИИ

Для решения изложенной выше проблемы и минимизации потерь исходных данных был предложен и разработан алгоритм действий при использовании буферов А и Б, показанный на блок-схеме рис. 3.

Рис. 3. Блок-схема алгоритма использования двух буферов памяти в модифицированном методе двойной буферизации

Как показано на рис. 3, функция чтения-записи в этом методе переключается между буферами А и Б сразу после опустошения того буфера, из которого в данный момент происходит чтение. Проверки этого производятся после каждой транзакции записи 16 кБ.

При таком режиме буферизации в идеальных условиях в каждом из буферов А и Б длительное время будет заполнено не более 16 кБ. Самый “жесткий” сценарий возникновения ситуации с потерей данных будет выглядеть следующим образом. Например, буфер А заполнен на 16 кБ, и активен запрос на его чтение, однако чтения так и не происходит до тех пор, пока буфер Б не заполнился до 2 МБ, т.е. полностью. В некотором приближении можно утверждать, что вероятность такого события в 128 раз меньше, чем вероятность события потери данных, описанного в предыдущем разделе.

Стоит отметить, что вероятность потери данных в текущий интервал времени T_1 в методе на рис. 3 описывается не так однозначно, как в методе на рис. 2. Когда предназначенный для считывания буфер заполнен на 16 кБ, эта вероятность минимальна. Но она растет пропорционально тому, насколько заполнен буфер, предназначенный для считывания. В случае, когда он заполнен полностью, вероятность потерь сравнивается с вероятностью в методе на рис. 2. Таким образом события, вызывающие суммарное увеличение интервалов T_L, но не вызвавшие потери данных, в этом случае обладают накопительным эффектом увеличения вероятности потери данных в дальнейшем, однако при длительном отсутствии критических ситуаций этот эффект постепенно нивелируется.

В любом случае, интегрально вероятность потери данных при использовании метода согласно блок-схеме рис. 3 на порядки меньше, чем при использовании метода согласно блок-схеме рис. 2. Это было подтверждено экспериментально.

Дополнительное положительное свойство метода рис. 3 в сравнении с методом рис. 2 заключается в уменьшении начальной задержки передачи данных.

5. ЗАКЛЮЧЕНИЕ

Описанная в этой работе модификация метода использования буферов для двойной аппаратной буферизации была реализована благодаря использованию особенностей контроллера CYUSB3014 и учету особенностей функционирования сенсоров линейного сканирования. При этом удалось осуществить данную модификацию, не меняя схемного решения радиотехнической платы сбора данных и управления спектральной ОКТ-системы. Изменения были внесены лишь в микрокод, задающий топологию и алгоритмы работы программируемой логической матрицы Cyclone III.

Практические испытания подтвердили устойчивую работоспособность высокоскоростных спектральных ОКТ-систем, использующих представленный здесь метод

двойной буферизации. При этом в составе ОКТ-систем использовались спектрометрические модули с разнообразными сенсорами линейного сканирования и разные мобильные и встраиваемые ПК. Было зафиксировано полное отсутствие событий потери данных в течение нескольких часов непрерывной работы. В этих же условиях при исходном методе буферизации гарантированно фиксировалось несколько событий потерь данных уже в течении 10 с работы.

Этот результат позволяет осуществить разработку семейства компактных мобильных спектральных ОКТ-систем с перспективой их коммерциализации.

ФИНАНСИРОВАНИЕ

Работа выполнена в рамках государственного задания Института прикладной физики им. А.В. Гапонова-Грехова РАН (проект № FFUF-2024-0029).

СПИСОК ЛИТЕРАТУРЫ

1. *Fercher A.F., Hitzenberger C.K., Kamp G., El-Zaiat S.Y.* Optics Communications. 1995. V. 117. № 1-2. P. 43. https://doi.org/10.1016/0030-4018(95)00119-S
2. Optical coherence tomography: Technology and applications. Ed. by W. Drexler, J.G. Fujimoto. Springer, 2015. https://doi.org/10.1007/978-3-319-06419-2
3. *Moiseev A., Ksenofontov S., Sirotkina M. et al.* J. Biophotonics. 2018. V. 11. № 10. P. e201700292. https://doi.org/10.1002/jbio.201700292
4. *Ксенофонтов С.Ю., Терпелов Д.А., Геликонов Г.В., Шилягин П.А., Геликонов В.М.* Изв. вузов. Радиофизика. 2019. Т. 62. № 2. С. 167. https://www.elibrary.ru/item.asp?id=37749323
5. *Геликонов Г.В., Ксенофонтов С.Ю., Шилягин П.А., Геликонов В.М.* Изв. вузов. Радиофизика. 2019. Т. 62. № 3. С. 252. https://www.elibrary.ru/item.asp?id=38480973
6. *Ксенофонтов С.Ю., Шилягин П.А., Терпелов Д.А., Новожилов А.А., Геликонов В.М., Геликонов Г.В.* ПТЭ. 2020. № 1. С. 136. https://doi.org/10.31857/S003281622001005X
7. *Ксенофонтов С.Ю., Моисеев А.А., Маткивский В.А., Шилягин П.А., Василенкова Т.В., Геликонов В.М., Геликонов Г.В.* ПТЭ. 2020. № 5. С. 104. https://doi.org/10.31857/S0032816220040291
8. *Ksenofontov S.Y., Shilyagin P.A., Terpelov D.A., Shabanov D.V., Gelikonov V.M., Gelikonov G.V.* Front. Optoelectron. 2020. V. 13 № 4. P. 393. https://doi.org/10.1007/S12200-019-0951-0
9. *Ksenofontov S.Y., Shilyagin P.A., Gelikonov V.M., Gelikonov G.V.* Photonics. 2023. V. 10 № 7. P. 736. https://doi.org/10.3390/photonics10070736

10. *Ксенофонтов С.Ю., Шилягин П.А., Терпелов Д.А., Шабанов Д.А., Геликонов В.М., Геликонов Г.В.* ПТЭ. 2023. № 6. С. 154. https://doi.org/10.31857/S0032816223050312
11. *Ксенофонтов С.Ю.* ПТЭ. 2019. № 3. С. 17. https://doi.org/10.1134/S0032816219030078
12. *Ксенофонтов С.Ю.* ПТЭ. 2026. № 1. С. 42. https://elibrary.ru/item.asp?id=89332629
13. *Ксенофонтов С.Ю., Купаев А.В., Василенкова Т.В., Терпелов Д.А., Шилягин П.А., Моисеев А.А., Геликонов Г.В.* ПТЭ. 2021. № 5. С. 131. https://doi.org/10.31857/S0032816221040224
14. *Tan X., Shen X.W., Ye X.C. et al.* J. Comput. Sci. Technol. 2018. V. 33. № 1. P. 145. https://doi.org/10.1007/s11390-017-1747-6

ПОДПИСИ К РИСУНКАМ

Рис. 1. Структурная схема модуля сбора данных и управления спектральной ОКТ-системой с интерфейсом USB 3.0

Рис. 2. Блок-схема последовательности действий при реализации стандартного метода двойной буферизации, использованного в работе [13]

Рис. 3. Блок-схема алгоритма использования двух буферов памяти в модифицированном методе двойной буферизации

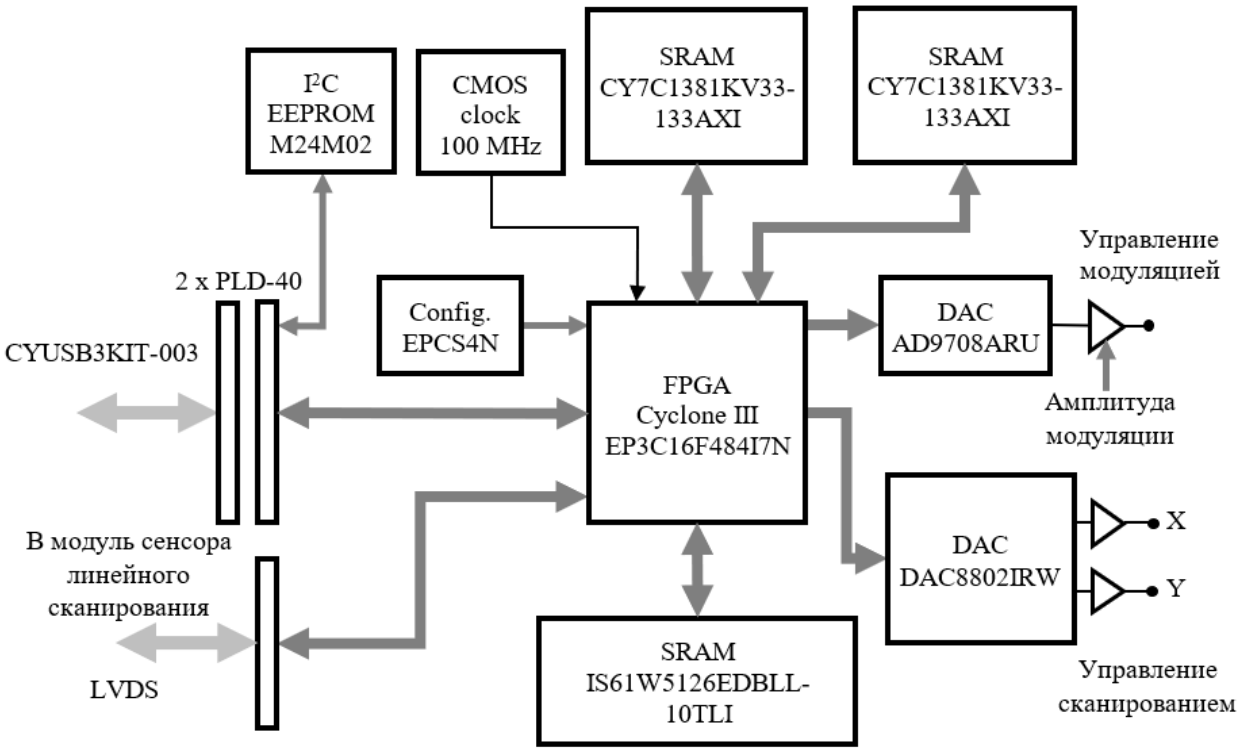

I²C
EEPROM
M24M02
CMOS
clock
100 MHz
SRAM
CY7C1381KV33-
133AXI
SRAM
CY7C1381KV33-
133AXI
Управление
модуляцией
2 x PLD-40
Config.
EPCS4N
DAC
AD9708ARU
CYUSB3KIT-003
FPGA
Cyclone III
EP3C16F484I7N
Амплитуда
модуляции
В модуль сенсора
линейного
сканирования
DAC
DAC8802IRW
X
Y
LVDS
SRAM
IS61W5126EDBLL-
10TLI
Управление
сканированием

Рис. 1

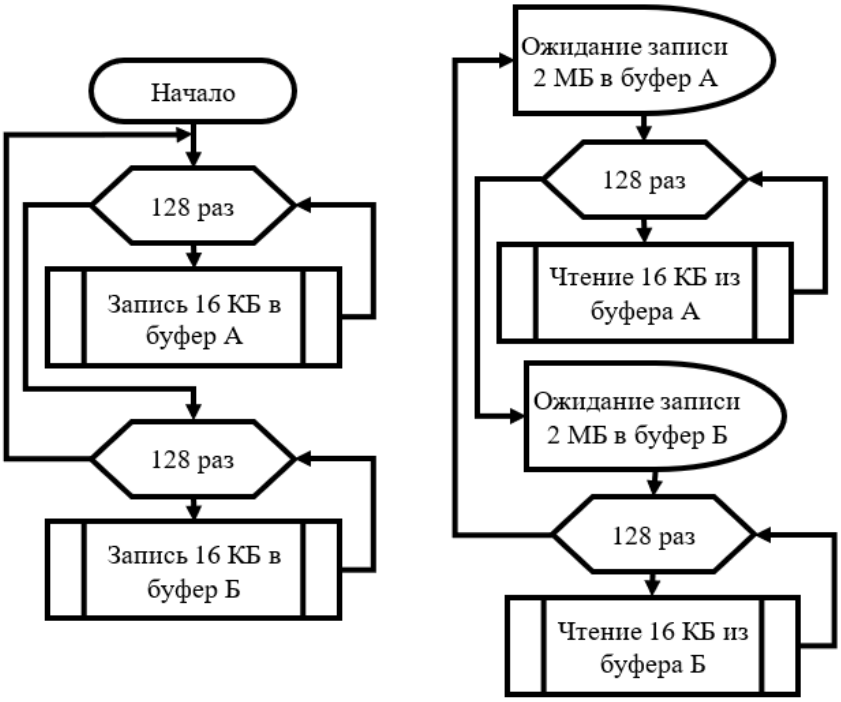

Начало
128 раз
Запись 16 КБ в буфер А
128 раз
Запись 16 КБ в буфер Б
Ожидание записи 2 МБ в буфер А
128 раз
Чтение 16 КБ из буфера А
Ожидание записи 2 МБ в буфер Б
128 раз
Чтение 16 КБ из буфера Б

Рис. 2

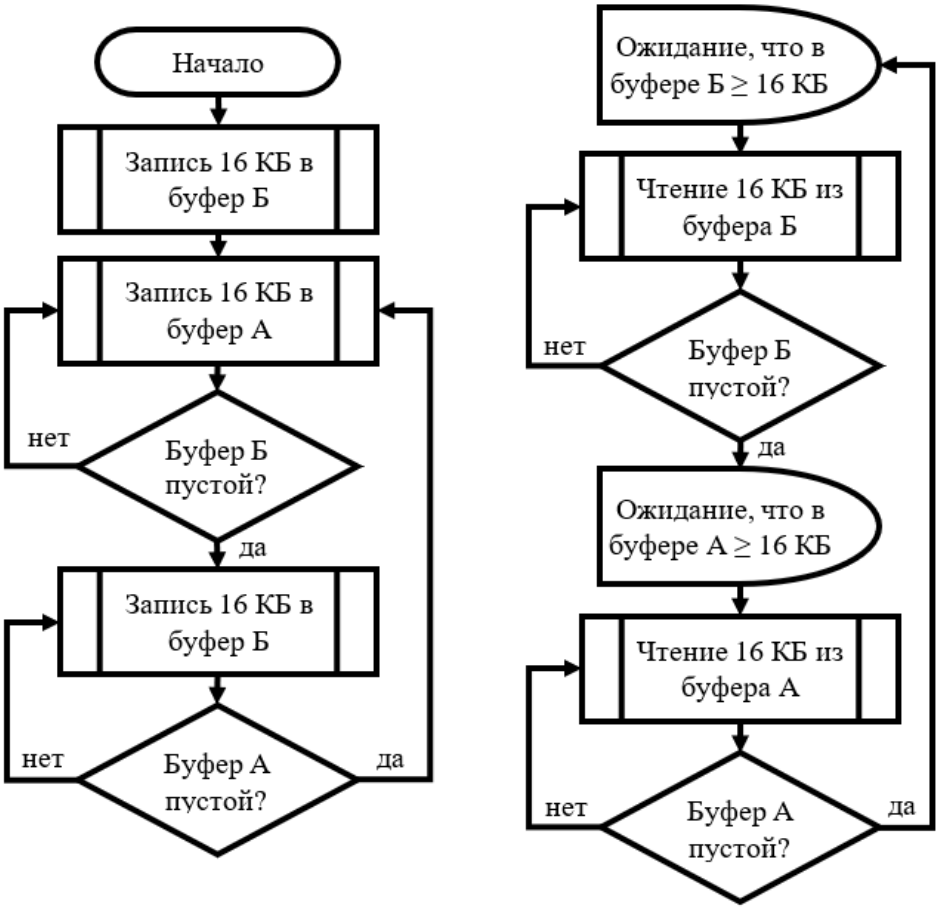

Начало
Запись 16 КБ в буфер Б
Запись 16 КБ в буфер А
нет
Буфер Б пустой?
да
Запись 16 КБ в буфер Б
нет
Буфер А пустой?
да
Ожидание, что в буфере Б ≥ 16 КБ
Чтение 16 КБ из буфера Б
нет
Буфер Б пустой?
да
Ожидание, что в буфере А ≥ 16 КБ
Чтение 16 КБ из буфера А
нет
Буфер А пустой?
да

Рис. 3