|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** | | |
| [**Институт перспективных технологий и индустриального программирования**](https://www.mirea.ru/docs/250117) | | |
| **Кафедра наноэлектроники** | | |

|  |  |
| --- | --- |
| **ОТЧЕТ** | |
| **О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ** | |
|  | |
| **Тема: «Модуль сбора и обработки данных от КМОП биосенсоров»** | |
| Студент группы ЭНБО-01-21 | Матевосов И.В. |
| Руководитель практики  *доцент, к.т.н.* | Певцов Е.Ф. |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_» \_\_\_\_\_\_202\_\_ г. |  |
|  |  |  |
| «Отчет утвержден.  Допущен к защите.» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_202\_\_ г. |  |

Москва, 2025

Оглавление

[Введение 3](#_Toc198500936)

[1. Литературный обзор 4](#_Toc198500937)

[1.1 Обзор современных тенденций в области биосенсорных систем и их интеграции с FPGA 4](#_Toc198500938)

[1.2 Принципы работы CMOS-биосенсоров 5](#_Toc198500939)

[1.3 Примеры использования в медицине, экологии и промышленности 7](#_Toc198500940)

[1.4 Общие требования к системам сбора данных от биосенсоров 14](#_Toc198500941)

[1.5 Области применения FPGA 15](#_Toc198500942)

[1.6 Реализация программ на FPGA: языки и инструменты 17](#_Toc198500943)

[1.7 Обзор средств разработки и отладки для FPGA​ 18](#_Toc198500944)

[1.8 Принципы работы SPI и его применение в биосенсорных системах 19](#_Toc198500945)

[1.9 Анализ публикаций и проектов, реализующих сбор и обработку данных от биосенсоров на FPGA 21](#_Toc198500946)

[1.10 Проблемы, возникающие при разработке систем сбора и обработки данных 23](#_Toc198500947)

[1.11 Перспективы и направления дальнейших исследований 25](#_Toc198500948)

[1.12 Заключение 26](#_Toc198500949)

[2. Экспериментальная часть 28](#_Toc198500950)

[Введение 28](#_Toc198500951)

[2.1 Постановка задачи 28](#_Toc198500952)

[2.2 Выбор инструментов и сред разработки 29](#_Toc198500953)

[2.3 Этапы разработки системы 33](#_Toc198500954)

[2.4 Верификация и моделирование 37](#_Toc198500955)

[3. Результаты и обсуждение 40](#_Toc198500956)

[Заключение 42](#_Toc198500957)

[4. Экономический раздел 44](#_Toc198500958)

[4.1 Организация и планирование работ по теме 44](#_Toc198500959)

[4.2 Расчёт стоимости проведения работ 1](#_Toc198500960)

[4.3 Статья «Сырье и материалы» 1](#_Toc198500961)

[4.4 Статья «Основная заработная плата» 4](#_Toc198500962)

[4.5 Статья «Дополнительная заработная плата» 7](#_Toc198500963)

[4.6 Статья «Страховые взносы» 7](#_Toc198500964)

[4.7 Статья «Амортизация» 8](#_Toc198500965)

[4.8 Статья «Прочие расходы» 10](#_Toc198500966)

[Список использованной литературы 16](#_Toc198500967)

# Введение

В последние годы наблюдается заметный рост интереса к разработке систем, способных осуществлять сбор и обработку данных от биосенсоров, в том числе на базе CMOS-технологии. CMOS-биосенсоры находят широкое применение в медицине, экологии, пищевой промышленности и других областях, где требуется оперативное и точное получение информации о биологических процессах [1]. При этом актуальной становится задача интеграции таких сенсоров с современными цифровыми устройствами, обеспечивающими высокую производительность и гибкость в обработке информации.

Использование FPGA для реализации модулей сбора и предварительной обработки данных приобретает всё большую популярность. FPGA позволяют создавать специализированные аппаратные решения, способные в режиме реального времени обрабатывать большие объемы информации и выполнять параллельные вычисления, что особенно важно для систем, работающих с биосенсорными данными. В свою очередь, язык описания аппаратуры Verilog является одним из наиболее перспективных и широко используемых средств для разработки цифровых систем на FPGA. Он позволяет эффективно моделировать сложные логические структуры и оптимизировать обмен данными, например, посредством интерфейса SPI.

Цель данного ВКР литературного обзора заключается в систематическом анализе существующих научных публикаций и технических решений, связанных с интеграцией CMOS-биосенсоров в системы на базе FPGA. В обзоре рассматриваются ключевые аспекты разработки модулей сбора и обработки данных, вопросы реализации цифровой логики на языке Verilog, а также особенности реализации SPI-интерфейса для обмена данными с периферийными устройствами. Особое внимание уделяется выявлению существующих проблем и ограничений в данной области, а также перспективам дальнейших исследований и возможностям оптимизации систем обработки биосенсорных данных. Актуальность рассматриваемой темы обусловлена необходимостью создания высокопроизводительных и гибких систем обработки биосенсорной информации, способных удовлетворить современные требования в различных прикладных областях.

# Литературный обзор

## 1.1 Обзор современных тенденций в области биосенсорных систем и их интеграции с FPGA

За последние годы наблюдается значительный рост интереса к биосенсорным системам, что обусловлено развитием технологий микро- и наноэлектроники, а также повышенными требованиями к точности и быстродействию в медицине, экологии, биотехнологиях и других прикладных областях. Одной из ключевых характеристик современных биосенсоров является использование CMOS-технологии, которая позволяет создавать компактные, энергоэффективные и высокочувствительные устройства. Современные CMOS-биосенсоры находят применение для мониторинга физиологических параметров, диагностики заболеваний и даже в системах безопасности, где требуется непрерывное измерение биомаркеров (биомаркер – это характеристика, которая используется в качестве индикатора состояния организма).

В данной области наблюдаются несколько ключевых тенденций:

1. Миниатюризация и интеграция функций.

Современные биосенсоры стремятся к уменьшению габаритов при сохранении высокой чувствительности [2,3,4]. Это достигается за счёт оптимизации архитектуры пикселей и внедрения дополнительных функций непосредственно в кристалл, таких как предварительная цифровая обработка сигнала. Такая интеграция снижает нагрузку на последующую обработку данных и позволяет обеспечить более высокую точность измерений.

1. Повышение скорости обработки данных.

Значительные объемы данных, генерируемые биосенсорами, требуют применения высокопроизводительных вычислительных решений. FPGA становятся идеальным выбором в этом контексте, поскольку они обеспечивают возможность параллельной обработки, гибкость в реализации специализированных алгоритмов и низкую задержку обработки сигналов. Благодаря возможности настраивать архитектуру под конкретные задачи, FPGA позволяют эффективно решать проблемы обработки сигналов в режиме реального времени.

1. Интеграция с высокоскоростными интерфейсами передачи данных.

Для передачи данных от биосенсоров к месту дальнейшей обработки используются различные протоколы. В современных системах наблюдается переход от традиционных интерфейсов, таких как UART, к более высокоскоростным и надёжным интерфейсам, например, SPI. Такой переход обусловлен необходимостью обеспечить стабильную и быструю передачу больших объемов данных, что особенно важно в системах реального времени.

1. Модульность и масштабируемость систем.

Современные решения по интеграции биосенсоров с FPGA характеризуются модульным подходом, когда отдельные блоки системы – от фронтенда сбора данных до цифрового процессора – разрабатываются как взаимозаменяемые модули. Это позволяет не только адаптировать систему под специфические требования конкретного приложения, но и масштабировать её в зависимости от требуемой точности и объёма обрабатываемых данных.

1. Применение алгоритмов интеллектуальной обработки сигналов.

В связи с ростом вычислительной мощности FPGA активно внедряются алгоритмы обработки сигналов, включая фильтрацию, анализ производных и методы обработки изображений.

## 1.2 Принципы работы CMOS-биосенсоров

CMOS‑биосенсоры представляют собой интегрированные устройства, разработанные с использованием стандартной CMOS-технологии, что позволяет объединять в одном чипе как чувствительный элемент, так и схемы обработки сигнала. Данное решение обеспечивает высокую компактность, низкое энергопотребление и возможность массового производства, что особенно важно для устройств, предназначенных для мониторинга биологических параметров.

Основной задачей биосенсора является обнаружение специфических биологических или химических молекул, а затем преобразование этого события в измеримый электрический сигнал. Для этого используются чувствительные поверхности, зачастую модифицированные биоселективными пленками (например, антителами или ферментами). При связывании целевых молекул изменяются параметры поверхностного заряда, проводимости или емкости, что фиксируется встроенными транзисторами CMOS‑структуры.

В случае оптических CMOS‑биосенсоров используется явление фотоэлектрического эффекта: фотодиоды, интегрированные в матрицу, преобразуют свет, который может генерироваться или модулироваться биосенсорным элементом, в электрический ток. Это позволяет создавать системы, в которых оптические сигналы (например, интенсивность флуоресценции) служат индикатором реакции между целевыми молекулами и сенсорной поверхностью.

Одним из ключевых достоинств CMOS‑технологии является возможность интегрировать аналоговые усилители, аналого-цифровые преобразователи и цифровые схемы обработки непосредственно с сенсорными элементами (на одном чипе) [5]. Это обеспечивает немедленную обработку сигнала в режиме реального времени и позволяет снизить уровень шума, возникающего при передаче данных внешним устройствам.

Существует несколько классификаций CMOS‑биосенсоров, различающихся по принципу трансдукции биосигнала [6,7]:

1. **Электрохимические CMOS‑биосенсоры** основаны на измерении изменений электрохимических параметров (например, тока, напряжения или импеданса) при взаимодействии целевых биомолекул с электродной поверхностью, которая может быть функционализирована селективными биохимическими агентами [8]. Преимущество данного подхода заключается в высокой чувствительности и возможности реализации низкоразмерных интегрированных систем для мониторинга физиологических параметров.
2. **Оптические CMOS‑биосенсоры:** при оптических сенсорах используется принцип преобразования оптического сигнала в электрический. Например, в таких устройствах измеряется изменение интенсивности или спектрального состава света, который проходит через или отражается от биосенсорной поверхности, где происходит специфическое связывание. Встроенные фотодиоды, реализующие фотоэлектрический эффект, позволяют преобразовывать световой сигнал в ток, который затем усиливается и обрабатывается цифровой схемой.

**3. Импедансные и емкостные CMOS‑биосенсоры:** в этих устройствах измеряется изменение импеданса или емкости сенсорного элемента в результате биологических процессов (например, адсорбции клеток или молекул). Такие датчики зачастую используются для неинвазивного контроля биологических процессов и характеризуются простотой конструкции и возможностью высокой интеграции на одном чипе.

**4. Механические CMOS‑биосенсоры (на основе MEMS-технологий):** хотя они встречаются реже, интеграция микро электромеханических систем (MEMS) с CMOS-технологией позволяет создавать сенсоры, в которых биологическое событие приводит к механическим изменениям (например, изгибу или колебаниям микроэлементов). Изменения этих параметров можно преобразовать в электрический сигнал с помощью встроенных CMOS‑схем.

## 1.3 Примеры использования в медицине, экологии и промышленности

CMOS-биосенсоры находят широкое применение в различных отраслях благодаря своей высокой чувствительности, миниатюризации и возможности интеграции с цифровыми системами [9,10]. Рассмотрим примеры их использования в медицине, экологии и промышленности [11,12,13].​

CMOS-биосенсоры активно применяются в медицинской диагностике, особенно для быстрого выявления инфекционных заболеваний. Например, электрохимические биосенсоры используются для детектирования вируса SARS-CoV-2 благодаря их высокой чувствительности и возможности миниатюризации, что делает их пригодными для тестирования в местах оказания медицинской помощи (POC) [14]. Одной из ключевых целей статьи «Электрохимические биосенсоры для определения SARS-CoV-2» является анализ современных подходов к диагностике вируса COVID-19 и рассмотрение возможностей электрохимических биосенсоров как альтернативы существующим методам. Авторы акцентируют внимание на ограничениях традиционных тестов, таких как количественная ПЦР в реальном времени (qRT-PCR), серологические и антигенные тесты, и подчёркивают необходимость разработки новых решений, способных обеспечить быструю, надёжную и экономически доступную диагностику в условиях ограниченного доступа к лабораторному оборудованию. В рамках обзора рассматриваются физико-химические принципы действия биосенсоров, особенности конструкций и конкретные примеры реализованных сенсорных платформ. Основной задачей работы является демонстрация потенциала электрохимических биосенсоров в контексте борьбы с пандемией и в перспективе — расширения их применения в области вирусной диагностики.

Физический принцип работы электрохимических биосенсоров основан на регистрации изменений электрических характеристик, возникающих в результате специфического взаимодействия между биологическим распознающим элементом и целевой молекулой. В качестве чувствительного элемента может выступать, например, одноцепочечная ДНК, антитело или аптамер, которые избирательно связываются с фрагментом вирусной РНК или с белками вирусной оболочки. Это взаимодействие вызывает изменение электрических параметров на поверхности сенсора, что фиксируется электрохимическим методом. При этом возможно использование различных электрохимических подходов: от импедансной спектроскопии до вольтамперометрии. В случае сенсоров, основанных на полевых транзисторах (например, графеновых FET), происходит изменение поверхностного потенциала, влияющее на проводимость канала транзистора. Это позволяет фиксировать факт связывания биомолекулы с высокой чувствительностью.

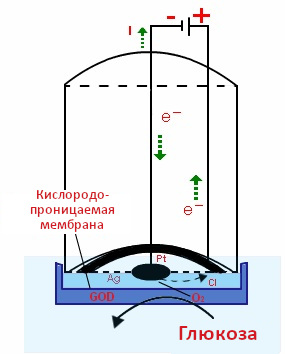


Рисунок 1 – Амперометрический биосенсор [16].

На рис.1 изображено автономное встроенное устройство, основанное на величине тока, протекающего между счетным и рабочим электродом, который стимулируется окислительно-восстановительной реакцией на рабочем электроде.

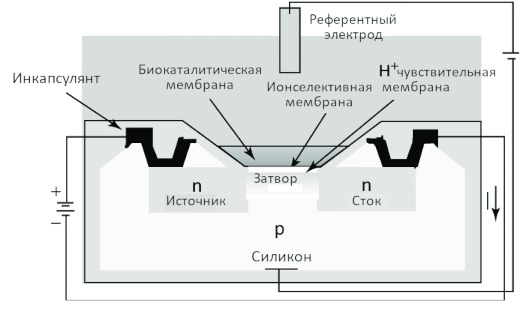


Рисунок 2 – Ионно-чувствительный потенциометрический биосенсор на основе FET [16].

Биосенсор типа, представленного на рис.2 обеспечивает  логарифмический ответ посредством высокого энергетического диапазона. Они состоят из двух электродов, обладающих огромной чувствительностью и прочностью. Они позволяют распознавать аналиты на стадиях, ранее достижимых только с помощью высокоэффективной жидкостной хроматографии, жидкостной хроматографии/масс-спектроскопии и без точной подготовки модели.

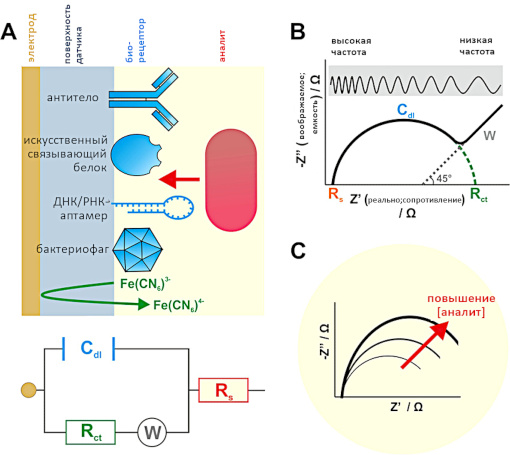


Рисунок 3 – Импедиметрический биосенсор [16].

(А) – Послойная конструкция датчика на рис.3 обычно включает поверхность электрода, функционализированную (например, с помощью полимера или самособирающегося монослоя) для обеспечения возможности присоединения биорецепторов, включая антитела, полуантитела, искусственные связывающие белки, аптамеры нуклеиновых кислот и бактериофаги. В большинстве систем на основе импеданса используются медиаторы электронов, например, ферри/ферроцианид [Fe(CN6)3/4] для контроля сопротивления переносу заряда. Схема Рэндлса иллюстрирует компоненты системы: емкость двойного слоя (Cdl), сопротивление переноса заряда (Rct), сопротивление раствора (Rs) и импеданс Варбурга (W) (W наблюдается только в некоторых системах на низкой частоте). (B) График Найквиста, показывающий особенности схемы Рэндлса. (C) Изменения импеданса, возникающие в результате взаимодействия аналита с поверхностью, пропорциональны концентрации аналита.

Физические биосенсоры делятся на два типа: пьезоэлектрические биосенсоры и термометрические биосенсоры.

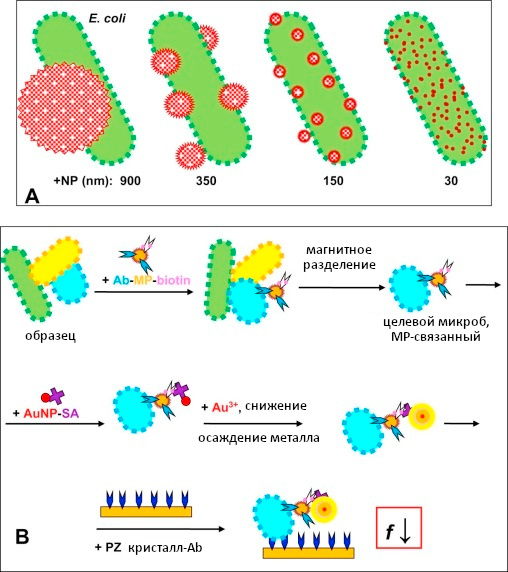


Рисунок 4 – Пьезоэлектрический биосенсор [16].

Эти датчики представляют собой набор аналитических устройств, которые работают по закону записи аффинного взаимодействия. Пьезоэлектрическая платформа — это сенсорный элемент, работающий по закону преобразования колебаний за счет скачка коллектора на поверхности пьезоэлектрического кристалла.

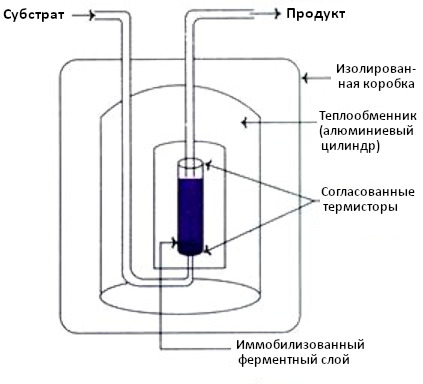


Рисунок 5 – Термометрический биосенсор [16].

Термометрический биосенсор используется для измерения или оценки уровня холестерина в сыворотке крови. Когда холестерин окисляется с помощью фермента холестерин-оксидазы, выделяется тепло, которое можно рассчитать. Аналогичным образом с помощью этих биосенсоров можно оценить уровень глюкозы, мочевины, мочевой кислоты и пенициллина G.

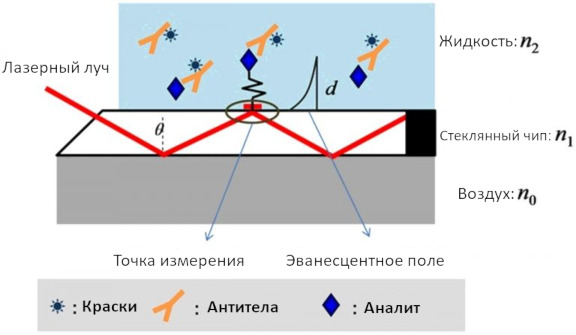


Рисунок 6 – Оптический биосенсор [16].

Оптический биосенсор (рис.6) — это устройство, использующее оптический принцип измерения. В них используется волоконная оптика, а также оптоэлектронные преобразователи. Термин «оптрод» представляет собой сжатие двух терминов — «оптический» и «электрод». В качестве преобразующих элементов в этих сенсорах в основном используются антитела и ферменты.

Оптические биосенсоры позволяют проводить безопасное неэлектрическое бесконтактное зондирование оборудования. Дополнительным преимуществом является то, что они часто не нуждаются в эталонных датчиках, поскольку сравнительный сигнал может быть получен с помощью источника света, аналогичного датчику отбора проб.

Таким образом, одно из главных преимуществ электрохимических биосенсоров заключается в их высокой чувствительности при сравнительно невысокой стоимости и простоте конструкции. В условиях необходимости массового тестирования, особенно вне лабораторных центров, это качество становится особенно актуальным. Такие сенсоры хорошо адаптируются к формату «point-of-care» — диагностики на месте оказания медицинской помощи, где требуется оперативность и мобильность. Благодаря возможности миниатюризации и интеграции с мобильными устройствами, например со смартфоном, они могут стать основой для дешёвых и доступных экспресс-тестов.

Дополнительным преимуществом является гибкость выбора биомишеней: биосенсоры могут быть настроены как на детекцию вирусной РНК, так и на специфические белки (например, S или N белки вируса), а также на целые вирусные частицы. Это особенно важно в условиях мутационной изменчивости вируса, когда одновременное определение нескольких мишеней может повысить надёжность диагностики и минимизировать вероятность ложноотрицательных результатов.

Кроме того, оптомеханические биосенсоры на основе фотонных и плазмонных волноводов находят применение в высокоточной медицинской диагностике заболеваний на ранних стадиях, требующих обнаружения малых концентраций биомаркеров.

В экологическом мониторинге CMOS-биосенсоры используются для контроля качества воздуха и воды. Они способны обнаруживать загрязнители в реальном времени, что позволяет оперативно реагировать на изменения в окружающей среде. Такие сенсоры могут быть интегрированы в портативные устройства для полевых измерений, обеспечивая высокую точность и чувствительность.​

В промышленности CMOS-биосенсоры применяются для контроля качества продукции и мониторинга технологических процессов. Например, в пищевой промышленности они используются для определения свежести продуктов и обнаружения патогенов. В химической промышленности такие сенсоры помогают в контроле концентрации различных веществ, обеспечивая безопасность и эффективность производственных процессов.

## 1.4 Общие требования к системам сбора данных от биосенсоров

Современные системы сбора и обработки данных от биосенсоров предъявляют высокие требования к точности, надежности и эффективности работы. Одним из ключевых аспектов является обеспечение высокой чувствительности и точности измерений, что особенно важно в медицинской диагностике и экологическом мониторинге. Для достижения этого необходимо использовать высококачественные биосенсоры, способные детектировать низкие концентрации биомолекул, а также применять методы фильтрации и обработки сигналов для минимизации влияния шумов и помех.​

Важным требованием является также минимизация задержки обработки данных, особенно в системах реального времени, где своевременность принятия решений может быть критичной. Это достигается за счет оптимизации алгоритмов обработки данных и использования высокоскоростных интерфейсов передачи информации. Кроме того, системы должны быть энергоэффективными, что особенно актуально для автономных устройств, работающих от батарей. Для этого разрабатываются энергосберегающие режимы работы и оптимизированные схемы питания [15]. ​

Масштабируемость и модульность системы также являются важными требованиями, позволяющими адаптировать систему под различные условия эксплуатации и расширять её функциональные возможности без значительных изменений в архитектуре. Это достигается за счет использования модульных компонентов и стандартизированных интерфейсов. Надежность и устойчивость к отказам обеспечиваются через применение избыточных компонентов, регулярное техническое обслуживание и мониторинг состояния системы.​

Совместимость с различными интерфейсами передачи данных, такими как SPI, I2C и UART, позволяет интегрировать биосенсоры в различные системы и устройства, расширяя области их применения. Наконец, обеспечение безопасности и защиты данных, включая шифрование, аутентификацию и контроль доступа, является необходимым условием для защиты конфиденциальной информации и предотвращения несанкционированного доступа.

FPGA (Field-Programmable Gate Array) играют ключевую роль в обеспечении систем реального времени и параллельной обработки данных благодаря своей архитектурной гибкости и способности к аппаратной реализации алгоритмов [16].

FPGA позволяют реализовывать алгоритмы с высокой степенью параллелизма, что значительно ускоряет обработку данных по сравнению с традиционными процессорами. Например, при обработке изображений FPGA могут выполнять множество операций одновременно, что критично для систем технического зрения в реальном времени.

Современные FPGA поддерживают динамическую частичную перепрограммируемость, позволяя изменять части конфигурации устройства во время работы. Это особенно полезно для многозадачных систем, где необходимо быстро переключаться между различными алгоритмами обработки данных.

Низкая задержка и высокая пропускная способность обеспечивают минимальную задержку при обработке данных, что критично для приложений, требующих быстрого реагирования, таких как системы обнаружения частиц в физике высоких энергий.

## 1.5 Области применения FPGA

Использование FPGA позволяет эффективно обрабатывать зрительные данные в реальном времени, что важно для робототехники и автоматизированных систем.

В представленной работе [17] исследуется эффективность использования программируемых логических интегральных схем (ПЛИС) в системах технического зрения (СТЗ), функционирующих в режиме реального времени. Основная цель исследования заключается в оценке целесообразности внедрения ПЛИС в качестве аппаратного ускорителя первичной обработки зрительных данных. Авторы стремятся проанализировать, насколько ощутимое преимущество дают эти схемы по сравнению с традиционными универсальными процессорами при выполнении вычислительно сложных алгоритмов обработки изображений, таких как расчет гистограммы и алгоритм выделения границ по методу Канни.

Актуальность рассматриваемой тематики обусловлена возрастающим объёмом зрительной информации, обрабатываемой в современных технических системах, и необходимостью соблюдения строгих временных ограничений при её анализе. Это особенно критично в приложениях, связанных с робототехникой, системами автономной навигации и интеллектуальным видеонаблюдением. При работе в режиме реального времени универсальные процессоры сталкиваются с ограничениями по пропускной способности и задержкам, связанным с последовательным выполнением инструкций и взаимодействием с памятью. В таких условиях использование ПЛИС представляется перспективным направлением, способным обеспечить параллельную, специализированную и низкозатратную по времени реализацию ключевых этапов обработки данных.

С технической точки зрения ПЛИС представляют собой конфигурируемые логические структуры, включающие логические блоки, программируемые соединения, а также встроенную память и интерфейсы для внешнего обмена данными. В рассматриваемом исследовании использовалась ПЛИС семейства Cyclone IV GX (модель EP4CGX150DF31), интегрированная в экспериментальный набор с поддержкой Gigabit Ethernet, DDR2 SDRAM и интерфейсом PCI Express. Такой набор позволяет обеспечить получение данных непосредственно от видеокамер и их обработку в режиме реального времени без существенной задержки на этапах передачи и предварительной обработки.

Алгоритмы, реализованные на ПЛИС, включали расчет гистограммы изображения и алгоритм Канни. В обоих случаях удалось добиться значительного выигрыша по времени по сравнению с реализациями на универсальном процессоре Intel Quad Core i7 2.5GHz. Так, среднее время выполнения алгоритма расчета гистограммы на ПЛИС составило 99 мкс, против 317 мкс на процессоре. Для алгоритма Канни средние значения составили 168 мкс и 968 мкс соответственно. Это демонстрирует более чем пятикратное преимущество аппаратной реализации на ПЛИС в условиях ограниченного временного бюджета.

Следует отметить, что реализация алгоритмов на ПЛИС требует значительно больших трудозатрат по сравнению с реализациями на языках высокого уровня. Алгоритмы на C++ потребовали порядка сотен строк кода, тогда как эквивалентная реализация на языке Verilog включала до нескольких тысяч строк. Это обусловлено необходимостью точного описания архитектуры, организации потоков данных и взаимодействия модулей. Тем не менее, высокая производительность и возможность параллельной обработки делают ПЛИС незаменимым элементом в тех случаях, когда приоритетом является минимизация задержек и максимизация пропускной способности системы.

Проведенное исследование убедительно демонстрирует потенциал ПЛИС как аппаратной платформы для реализации алгоритмов первичной обработки изображений. Использование гибридных архитектур, объединяющих универсальные многоядерные процессоры и ПЛИС, представляется наиболее рациональным подходом. В перспективе авторы предполагают расширить набор реализуемых алгоритмов и выполнить дополнительное сравнение с производительностью графических процессоров (GPU), что позволит уточнить область наилучшего применения ПЛИС в задачах технического зрения.

## 1.6 Реализация программ на FPGA: языки и инструменты

​Verilog и SystemVerilog являются основными языками описания аппаратуры (HDL), широко применяемыми в проектировании цифровых схем, включая реализацию на ПЛИС (FPGA). Verilog, разработанный в 1984 году, стал стандартом IEEE в 1995 году и получил широкое распространение благодаря своей простоте и эффективности. SystemVerilog, в свою очередь, был представлен в 2005 году как расширение Verilog, добавляя множество новых возможностей для описания и верификации аппаратуры.​

Одним из ключевых отличий SystemVerilog является введение новых типов данных, таких как logic, bit, byte, shortint, int и longint, которые обеспечивают более точное управление представлением данных и их поведением в различных условиях. Кроме того, SystemVerilog поддерживает структуры (struct), объединения (union), перечисления (enum) и интерфейсы (interface), что способствует более модульному и читаемому коду. Эти возможности особенно полезны при проектировании сложных систем с большим количеством взаимодействующих компонентов.

SystemVerilog также вводит новые процедурные блоки, такие как always\_comb, always\_ff и always\_latch, которые обеспечивают более явное и точное описание комбинационной и последовательной логики. Это позволяет избежать некоторых распространенных ошибок, связанных с использованием традиционного блока always в Verilog. Дополнительно, SystemVerilog включает в себя расширенные возможности для верификации, включая утверждения (assertions), покрытия (coverage) и объектно-ориентированное программирование, что делает его мощным инструментом для разработки и тестирования сложных цифровых систем.​

Несмотря на все преимущества SystemVerilog, Verilog остается популярным выбором для многих проектов, особенно тех, которые требуют простоты и меньших ресурсов. Verilog обладает более простой синтаксисом и меньшей сложностью, что делает его подходящим для обучения и реализации менее сложных проектов. Однако, по мере увеличения сложности и требований к проекту, преимущества SystemVerilog становятся более очевидными, особенно в контексте современных требований к верификации и модульности

## 1.7 Обзор средств разработки и отладки для FPGA​

​Разработка и отладка проектов на программируемых пользователем вентильных матрицах (FPGA) требуют использования специализированных инструментов, обеспечивающих эффективное проектирование, моделирование, синтез и тестирование цифровых схем. Среди наиболее распространенных средств разработки можно выделить Xilinx Vivado, Intel Quartus Prime и Lattice Diamond [18]. Эти интегрированные среды проектирования предоставляют полный набор инструментов для создания и оптимизации FPGA-дизайнов, включая редакторы схем, средства для синтеза и размещения логики, а также симуляторы для верификации работы проекта.

В процессе разработки особое внимание уделяется отладке, которая позволяет выявлять и устранять ошибки на различных этапах проектирования. Традиционные методы отладки включают использование внешних измерительных приборов, таких как осциллографы и логические анализаторы, для мониторинга сигналов на выводах FPGA. Однако эти методы ограничены в возможностях наблюдения за внутренними сигналами устройства.

Для более глубокой отладки разработчики используют встроенные логические анализаторы, такие как Xilinx Integrated Logic Analyzer (ILA) и Intel Signal Tap Logic Analyzer [19]. Эти инструменты позволяют в реальном времени захватывать и анализировать внутренние сигналы FPGA без необходимости использования внешнего оборудования, что значительно упрощает процесс верификации и ускоряет выявление проблем. ​

Современные тенденции в разработке FPGA включают использование облачных платформ для синтеза и эмуляции, что позволяет повысить производительность и гибкость разработки. Кроме того, активно развиваются инструменты высокоуровневого синтеза (HLS), позволяющие описывать аппаратные модули на языках высокого уровня, таких как C/C++, с последующей автоматической трансляцией в HDL-код. Это способствует сокращению времени разработки и упрощает процесс проектирования сложных систем.

## 1.8 Принципы работы SPI и его применение в биосенсорных системах

​Интерфейс последовательной периферии (SPI) представляет собой синхронный последовательный протокол обмена данными, широко применяемый в системах цифровой электроники. Он обеспечивает высокоскоростную передачу данных между микроконтроллерами и периферийными устройствами, такими как датчики, памяти и дисплеи. Благодаря своей простой архитектуре и эффективности, SPI стал популярным выбором для многих встроенных систем, включая биосенсорные устройства.

Принцип работы SPI основан на использовании четырёх основных линий: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (Serial Clock) и SS (Slave Select). Мастер-устройство инициирует передачу, генерируя тактовые импульсы на линии SCLK и активируя нужное периферийное устройство через линию SS. Данные передаются по линиям MOSI и MISO, обеспечивая двунаправленную связь. Такая структура позволяет достигать высокой скорости обмена данными и минимальной задержки, что особенно важно в приложениях, требующих оперативной обработки информации.

В контексте биосенсорных систем, SPI играет ключевую роль в обеспечении эффективной связи между сенсорами и управляющими микроконтроллерами или процессорами. Многие биосенсорные устройства, такие как электрокардиографические (ЭКГ) или фотоплетизмографические (ФПГ) сенсоры, генерируют аналоговые сигналы, которые необходимо преобразовать в цифровую форму для последующей обработки. Аналого-цифровые преобразователи (АЦП), интегрированные в такие сенсоры, часто используют SPI для передачи оцифрованных данных. Например, микросхема ADS1298 от Texas Instruments, предназначенная для многоканального ЭКГ-мониторинга, использует SPI-интерфейс для передачи данных о сердечной активности в реальном времени.

Преимущества использования SPI в биосенсорных системах включают высокую скорость передачи данных, что позволяет оперативно реагировать на изменения физиологических параметров. Кроме того, SPI обеспечивает низкое энергопотребление, что критично для портативных и носимых медицинских устройств, работающих от батарей. Простота реализации и широкая поддержка со стороны различных микроконтроллеров и процессоров делают SPI удобным выбором для разработчиков биосенсорных решений.

Интерфейсы передачи данных играют ключевую роль в обеспечении эффективной и надежной связи между различными компонентами электронных систем. Наиболее распространенными интерфейсами кроме SPI являются I²C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter) и USB (Universal Serial Bus).

I²C — это синхронный последовательный интерфейс, использующий всего две линии: SDA (Serial Data Line) и SCL (Serial Clock Line). Он поддерживает адресацию до 127 устройств на одной шине, что упрощает подключение множества компонентов. I²C широко применяется в системах с ограниченным пространством и ресурсами, таких как датчики, часы реального времени и EEPROM. Однако его скорость передачи данных ниже по сравнению с SPI, и он более чувствителен к помехам, что может ограничивать его применение в высокоскоростных или критичных к надежности системах.

UART — это асинхронный последовательный интерфейс, использующий две линии: TX (передача) и RX (прием). Он не требует тактового сигнала, что упрощает схему подключения, но требует согласования скорости передачи данных между устройствами. UART широко используется для связи между микроконтроллерами и периферийными устройствами, такими как модули Bluetooth и GPS. Однако его скорость передачи данных ограничена, и он поддерживает только полудуплексную связь, что может быть недостаточно для приложений, требующих высокой пропускной способности.

USB — это универсальный последовательный интерфейс, обеспечивающий высокоскоростную передачу данных и питание устройств через один кабель. Он широко используется для подключения периферийных устройств к компьютерам и микроконтроллерам. USB поддерживает горячую замену устройств и обеспечивает высокую пропускную способность, что делает его подходящим для приложений, требующих передачи больших объемов данных. Однако его реализация требует более сложного аппаратного и программного обеспечения, что может быть избыточным для простых встроенных систем.

## 1.9 Анализ публикаций и проектов, реализующих сбор и обработку данных от биосенсоров на FPGA

В последние годы наблюдается активное развитие проектов и исследований, направленных на реализацию сбора и обработки данных от биосенсоров с использованием программируемых логических интегральных схем (ПЛИС, или FPGA). Такие решения находят применение в медицине, экологии и промышленности, обеспечивая высокую производительность, низкое энергопотребление и возможность обработки данных в реальном времени.

Одним из примеров является использование встроенных аналогово-цифровых преобразователей (АЦП) в ПЛИС семейства MAX10 от Intel [20]. Статья «Практика использования встроенного АЦП в ПЛИС семейства MAX10» посвящена изучению возможностей и особенностей применения встроенного аналого-цифрового преобразователя (АЦП), интегрированного в ПЛИС (программируемые логические интегральные схемы) семейства MAX10 производства Intel (Altera). В центре внимания находится аппаратный модуль АЦП, его функциональные характеристики, структурная организация, параметры и режимы работы. Основная цель материала — предоставить разработчику подробные справочные сведения о встроенном АЦП в MAX10, включая схемотехнические аспекты проектирования, рекомендации по применению, а также принципы настройки IP-ядер для интеграции АЦП в цифровые проекты.

В ПЛИС семейства MAX10 встроен аппаратный АЦП типа SAR (Successive Approximation Register), обеспечивающий 12-битную разрядность и частоту преобразования до 1 Мвыб/с. В зависимости от модели ПЛИС возможно наличие одного или двух независимых модулей АЦП. Устройства классифицируются на два типа по схеме питания: Single Supply (одно напряжение 3,3 В) и Dual Supply (1,2 В для цифровой части и 2,5 В для аналоговой).

Поддерживаются два типа аналоговых входов: выделенные (dedicated) и универсальные (dual-function), которые могут быть использованы также как GPIO. Кроме того, каждый модуль АЦП снабжён встроенным температурным датчиком (TSD), подключённым к отдельному каналу с номером 17. Особенностью архитектуры является возможность включения предделителя на некоторых входах, позволяющего измерять напряжения выше опорного путём предварительного деления.

Для реализации функций АЦП используется IP-ядро Modular ADC Core Intel FPGA IP (или Dual ADC Core, если используется два модуля). Ядро позволяет управлять конфигурацией преобразования, задавать порядок переключения каналов (sequencing), настраивать частоту, источник опорного напряжения (внутренний или внешний), а также устанавливать пороговые значения, при достижении которых возможно формирование сигнала прерывания.

1. IP-ядро поддерживает четыре конфигурации:
2. Сохранение выборок во внутренней памяти.
3. То же, с поддержкой порогового уведомления.
4. Сохранение во внешнюю память.
5. Только управление АЦП, без хранения результатов.

Важным аспектом работы с аналоговой частью ПЛИС является корректная разводка и защита цепей. Производитель рекомендует подключать опорное напряжение и аналоговые входы к специальным выводам (REFGND), обеспечивать минимальное сопротивление заземления, избегать паразитной емкости и индуктивности. На аналоговых входах рекомендуется размещать пассивные или активные фильтры нижних частот (ФНЧ), рассчитанные исходя из параметров источника сигнала. Указывается, что при несоблюдении этих условий возможно значительное ухудшение характеристик преобразования — THD, SINAD, DNL, INL.

Для калибровки встроенного температурного датчика применяется таблица перевода выходного кода АЦП в температуру. Диапазон измерения охватывает от –40 до +125°C, при этом частота выборок ограничена 50 квыб/с, а для повышения точности используется метод бегущего среднего по 64 точкам.

Внедрение программируемых логических интегральных схем (FPGA) в системы сбора и обработки данных от биосенсоров демонстрирует значительный потенциал в различных отраслях, включая медицину, экологию и промышленность.

В диссертационном исследовании А.Ю. Ануфриенко предложен метод детектирования событий на конечных устройствах для систем Интернета вещей (IoT), позволяющий сократить объем передаваемых данных и обеспечить быстрое параллельное детектирование событий [21]. Разработанная модель системы IoT учитывает базовые устройства, пропускную способность каналов связи и задержки обработки данных на устройствах, что позволяет достоверно определить оптимальный диапазон параметров для построения IoT-системы. Практическая значимость полученных результатов заключается в сокращении объемов передаваемой информации в IoT-системах, особенно в системах промышленного интернета вещей.

## 1.10 Проблемы, возникающие при разработке систем сбора и обработки данных

Разработка высокопроизводительных систем сбора и обработки данных от биосенсоров на базе FPGA сопряжена с рядом технических вызовов, связанных в первую очередь с необходимостью обеспечить надежную синхронизацию и корректную передачу данных. Одной из основных проблем является обеспечение синхронизации между различными тактовыми доменами системы. В современных цифровых устройствах даже малейшие отклонения во времени могут привести к возникновению метастабильности, что, в свою очередь, отражается на корректности получаемых данных. При реализации на FPGA часто используется встроенная логика синхронизации, включающая каскадные регистры, которые позволяют минимизировать риск возникновения ошибок, связанных с временными расхождениями. Однако, при работе с высокоскоростными интерфейсами передачи данных, такими как SPI или UART, даже незначительные задержки могут оказать существенное влияние на качество обработки сигналов.

Кроме того, значительное влияние на передачу данных оказывают внешние факторы, такие как электромагнитные помехи и шум, которые могут возникать как в проводной среде, так и в условиях работы в промышленной или медицинской среде. При передаче данных по последовательным интерфейсам зачастую возникают проблемы, связанные с вибрациями сигнала, отражениями и ложными переключениями, что требует применения специальных схем коррекции и усиления.

Проблемы синхронизации и передачи данных в системах на FPGA, предназначенных для обработки биосенсорных сигналов, требуют подхода, учитывающего как аппаратные, так и программные аспекты системы. Решение этих проблем включает в себя не только использование современных средств синхронизации и корректировки сигналов, но и тщательный выбор интерфейса передачи данных, оптимизированного для конкретных условий эксплуатации. Эти вопросы остаются предметом активных исследований, что свидетельствует о высокой актуальности данной темы для дальнейшего развития биосенсорных технологий.

Одним из основных ограничений является ограниченное количество логических элементов и блоков памяти в FPGA. При разработке сложных систем с множеством функций и интерфейсов может возникнуть нехватка ресурсов для реализации всех необходимых компонентов. Для решения этой проблемы рекомендуется оптимизировать архитектуру проекта, используя модульный подход и повторное использование компонентов. Кроме того, применение специализированных IP-ядер и использование встроенных блоков, таких как DSP и блоки памяти, позволяет снизить нагрузку на общие ресурсы FPGA.​

Еще одним важным аспектом является управление энергопотреблением. В системах, особенно предназначенных для портативных или энерго-ограниченных приложений, избыточное энергопотребление может привести к перегреву и снижению надежности. Для минимизации энергопотребления следует применять техники, такие как динамическое управление тактовой частотой, отключение неиспользуемых блоков и оптимизация логики для сокращения переключений. Также важно учитывать выбор технологии FPGA, так как современные чипы предлагают улучшенные характеристики по энергопотреблению.​

Кроме того, при проектировании систем на FPGA необходимо учитывать временные ограничения и синхронизацию между различными компонентами. Нарушение временных характеристик может привести к некорректной работе системы. Для обеспечения надежной работы рекомендуется использовать статический временной анализ и задавать соответствующие временные ограничения в проекте. Это позволяет выявить критические пути и оптимизировать их для соблюдения временных требований. Применение таких методов обеспечивает стабильную и корректную работу системы в различных условиях эксплуатации.

## 1.11 Перспективы и направления дальнейших исследований

​Развитие технологий биосенсоров и программируемых логических интегральных схем (FPGA) демонстрирует значительный потенциал в области медицинской диагностики, мониторинга здоровья и других приложений. Современные тенденции указывают на интеграцию этих технологий для создания более эффективных и точных систем.​

Биосенсоры становятся все более миниатюрными, чувствительными и специфичными. Разработка новых материалов и методов трансдукции позволяет создавать устройства, способные обнаруживать широкий спектр биомаркеров с высокой точностью. Особое внимание уделяется созданию портативных и носимых биосенсоров, которые могут использоваться для непрерывного мониторинга физиологических параметров в реальном времени.​

FPGA предлагают гибкость и высокую производительность, что делает их идеальными для обработки данных от биосенсоров. Благодаря параллельной архитектуре, FPGA способны обрабатывать большие объемы данных с минимальной задержкой, что особенно важно для приложений, требующих быстрого отклика. Кроме того, возможность перепрограммирования позволяет адаптировать систему под различные типы биосенсоров и алгоритмы обработки данных.​

Интеграция биосенсоров с FPGA открывает новые возможности для создания автономных систем мониторинга здоровья, которые могут использоваться как в клинических условиях, так и в домашних условиях. Такие системы могут обеспечивать непрерывный сбор и анализ данных, передавая результаты в облачные хранилища для дальнейшего анализа и принятия решений.​

Таким образом, синергия между биосенсорными технологиями и программируемыми логическими интегральными схемами (FPGA) открывает широкие перспективы для создания высокоэффективных систем мониторинга и диагностики. Биосенсоры обеспечивают точный и непрерывный сбор физиологических данных, в то время как FPGA предлагают гибкость и высокую производительность для их обработки в реальном времени.​

Однако разработка таких систем сопряжена с рядом вызовов, включая ограниченные ресурсы FPGA, сложности в синхронизации и передаче данных, а также необходимость обеспечения энергоэффективности. Для преодоления этих препятствий рекомендуется:​

* Оптимизировать архитектуру проектов, используя модульный подход и специализированные IP-ядра.​
* Применять техники управления энергопотреблением, такие как динамическое масштабирование частоты и отключение неиспользуемых блоков.​
* Использовать статический временной анализ для обеспечения надежной синхронизации компонентов.​

Для разработчиков и исследователей важно также учитывать выбор подходящих интерфейсов передачи данных (SPI, I²C, UART, USB), исходя из требований конкретного приложения. Кроме того, интеграция современных биосенсоров с FPGA требует тщательной проработки протоколов взаимодействия и обеспечения совместимости между различными компонентами системы.​

В перспективе, дальнейшие исследования в области биосенсоров и FPGA будут способствовать созданию более компактных, энергоэффективных и интеллектуальных систем, способных адаптироваться к индивидуальным потребностям пользователей. Развитие стандартов и протоколов, а также внедрение новых материалов и технологий, будут играть ключевую роль в этом процессе.

## 1.12 Заключение

Таким образом, проведённый анализ научных публикаций позволил не только охарактеризовать текущее состояние исследований в области CMOS-базированных биосенсоров, но и выявить основные технологические и архитектурные особенности систем, применяемых для сбора и обработки данных с таких сенсоров.

Учитывая рассмотренные подходы и требования к точности, скорости и помехоустойчивости, в рамках данной выпускной квалификационной работы была поставлена задача разработать специализированный модуль для сбора и обработки данных с CMOS-биосенсоров. Разрабатываемый модуль должен обеспечивать приём, временное хранение, а также сериализацию данных от аналогово-цифрового преобразователя (АЦП), последующее управление передачей данных по интерфейсу SPI и формирование сигнала управления внешними устройствами (например, сигналов выбора микросхем или сброса). В качестве обрабатываемых данных предполагается 16-битный выход АЦП. Обработка данных будет заключаться в нахождении сдвига напряжения. Помимо приёма и буферизации, модуль должен поддерживать работу в составе более крупной цифровой системы, что требует использования тестируемых и верифицированных логических компонентов. Таким образом, дальнейшая часть работы будет сосредоточена на проектировании, моделировании и проверке функциональности данного модуля.

На основании вышеизложенной информации, были сформулированы следующие задачи, выполнение которых поспособствует достижению цели выпускной квалификационной работы:

1. Разработать код поведенческого описания модуля.

2. Разработать код тестового окружения для моделирования работы устройства

3. Проверить работоспособности модуля на основе отладочной платы с ИС программируемой логики.

# 

# Экспериментальная часть

## Введение

Целью прохождения преддипломной практики являлась разработка цифровой системы сбора и обработки данных с КМОП-биосенсоров, реализованной на языке описания аппаратуры Verilog. Основной задачей было создание и верификация модуля, способного принимать данные от аналогово-цифрового преобразователя (АЦП) через интерфейс SPI, выполнять математическую обработку сигнала в режиме реального времени, производить логарифмическое преобразование, а также формировать результат для последующей передачи или анализа.

В ходе практики особое внимание было уделено вопросу преобразования сигнала биосенсора в логарифмическую шкалу. Такая форма представления данных позволяет наиболее эффективно отслеживать динамику изменения порогового напряжения на выходе чувствительного элемента, связанного с концентрацией исследуемого вещества.

Проект охватывает полный цикл цифровой обработки сигнала: от моделирования SPI-протокола и буферизации с помощью FIFO, до визуализации результатов с применением Python и библиотеки matplotlib. Дополнительно в ходе работы были освоены средства симуляции (Icarus Verilog), анализа временных диаграмм (GTKWave), работы с системой контроля версий Git и оформлением проекта в виде репозитория GitHub.

Практика стала неотъемлемой частью подготовки к выполнению выпускной квалификационной работы и позволила на практике применить полученные знания в области цифровой схемотехники, микросистемной электроники и обработки данных.

## 2.1 Постановка задачи

В современных микросистемах, ориентированных на биомедицинские измерения, особое внимание уделяется разработке компактных и энергоэффективных узлов сбора и обработки данных. Одним из таких примеров являются КМОП-биосенсоры, которые преобразуют воздействие внешней среды (например, изменение ионной концентрации) в электрический сигнал, чаще всего представленный в виде малых токов или напряжений. Эти сигналы требуют последующей цифровой обработки, включающей преобразование, фильтрацию и анализ отклонений.

В рамках преддипломной практики ставилась задача разработать модуль сбора и обработки данных с КМОП-биосенсоров, пригодный для реализации на программируемой логике (ПЛИС), и выполняющий следующие функции:

* Приём данных от АЦП по SPI-интерфейсу.
* Математическая обработка в реальном времени.
* Оценка сдвига логарифмической функции.
* Буферизация данных с использованием FIFO.
* Визуализация результатов.

АЦП подключается к внешнему аналоговому выходу биосенсора и передаёт оцифрованные значения через последовательный интерфейс. Изначально работа велась с 12-битным потоком, затем формат был расширен до 16 бит для увеличения точности. Сигнал, принятый по SPI, преобразуется в напряжение в милливольтах, затем подаётся на блок аппроксимации логарифмической функции. В частности, реализовано приближённое вычисление натурального логарифма на базе кусочно-линейной интерполяции. Это позволяет отслеживать экспоненциальные изменения, характерные для физики МОП-структур. Важной целью обработки является оценка сдвига значений ln(V) между соседними измерениями:

Такой сдвиг может отражать изменение порогового напряжения транзистора, связанное с воздействием биологического агента.

Результаты обработки хранятся в буфере FIFO, откуда могут быть считаны либо для передачи в ПК, либо для дальнейшего использования в системе. FIFO реализован в 16-битном формате и управляется сигналами wr\_en и rd\_en.

С помощью внешнего скрипта на Python, эмулирующего UART-приём данных, реализовано построение графика логарифмического сдвига Δx. Это позволяет продемонстрировать поведение модели в виде кривой, имитирующей реальный отклик биосенсора. Система охватывает полный цифровой путь от аналогового сигнала до визуального представления данных, и может быть масштабирована под конкретные задачи анализа встраиваемых сенсорных устройств.

## 2.2 Выбор инструментов и сред разработки

В процессе выполнения практики было принято решение использовать инструменты, ориентированные на разработку и моделирование цифровых систем на языке описания аппаратуры Verilog, а также на обработку и визуализацию данных, полученных в результате моделирования.

В качестве основного языка проектирования цифровой логики был выбран Verilog HDL (Hardware Description Language). Язык позволяет описывать параллельную структуру аппаратных модулей, синтезируемых на программируемых логических интегральных схемах (ПЛИС). Verilog активно применяется в промышленной и исследовательской практике, поддерживается большинством САПР, и предоставляет необходимые средства для описания интерфейсов, автоматов состояний, регистров, буферов и арифметических блоков.

В качестве симулятора был использован Icarus Verilog — свободно распространяемая система для компиляции Verilog-кода и запуска временного моделирования. Она позволяет запускать тестбенчи, выполнять отладку логики и генерировать файлы временных диаграмм.



Рисунок 1 – Логотип программного обеспечения Icarus Verilog.

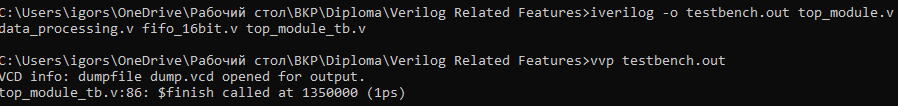


Рисунок 2 – Пример запуска симуляции с помощью команды iverilog.

На рис. 2 показаны команды, исполняемые в ходе симуляции работы модулей, написанных на Verilog. -o testbench.out – вывод результатов и название файла вывода, дальше аргументы – названия файлов, которые содержат модули, использующиеся в коде. Команда vvp выдаёт результат в виде сигналов, которые затем можно визуально отобразить.

Для анализа поведения сигналов во времени использовалась программа GTKWave, читающая файлы формата .vcd, сгенерированные симулятором. В GTKWave удобно отслеживать фронты тактовых импульсов, сигналы чтения и записи, переходы между состояниями в регистрах и буферах. GTKWave предоставила ключевые подтверждения корректной работы SPI-интерфейса, FIFO и блока обработки данных.

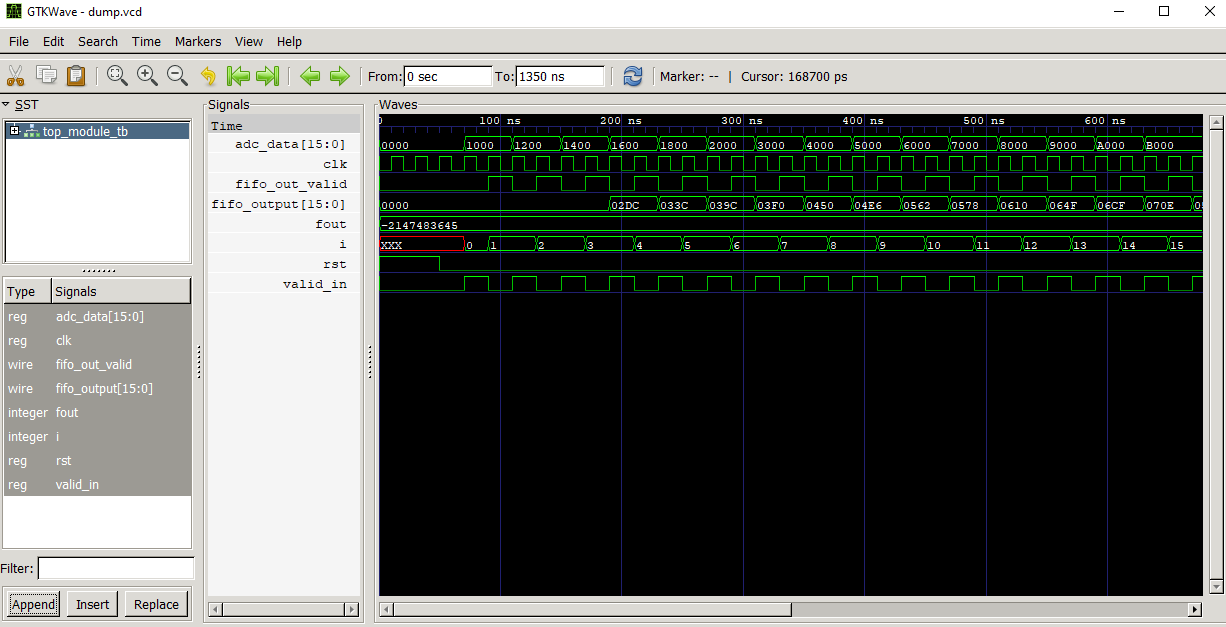


Рисунок 3 – Рабочий интерфейс программы GTKWave, на временной диаграмме результаты выполнения testbench.out.

Разработка кода велась в редакторе VS Code, настроенном с поддержкой синтаксиса Verilog и автозапуска симуляций. Использование расширений (например, Verilog-HDL/SystemVerilog) упростило отладку и повысило читаемость кода.

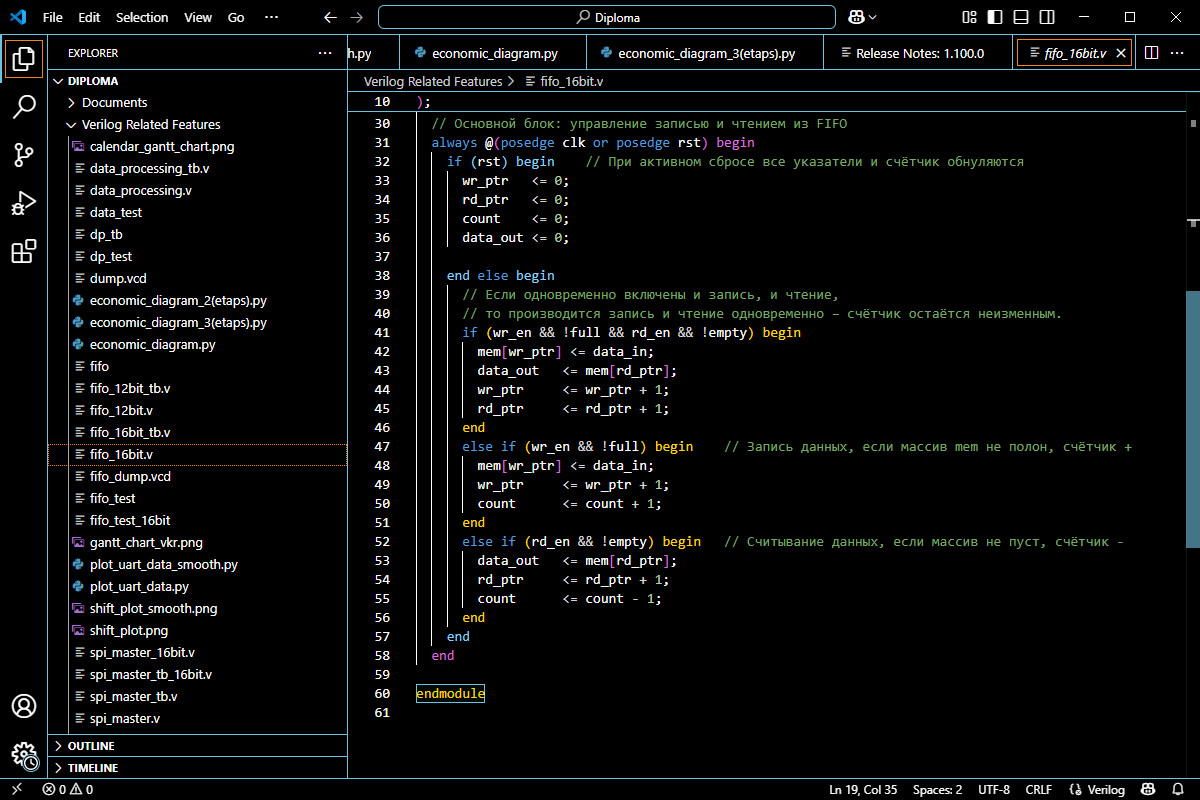


Рисунок 4 – Интерфейс редактора кода Visual Studio Code.

Для обеспечения сохранности проекта и контроля версий использовалась система управления версиями Git. Все ключевые файлы проекта (модули, тестбенчи, результаты) хранились в репозитории GitHub. Актуальная ссылка на репозиторий: https://github.com/SuNGviN/Diploma. Это обеспечило удобный способ отката к предыдущим версиям, а также продемонстрировало владение инструментами командной разработки.

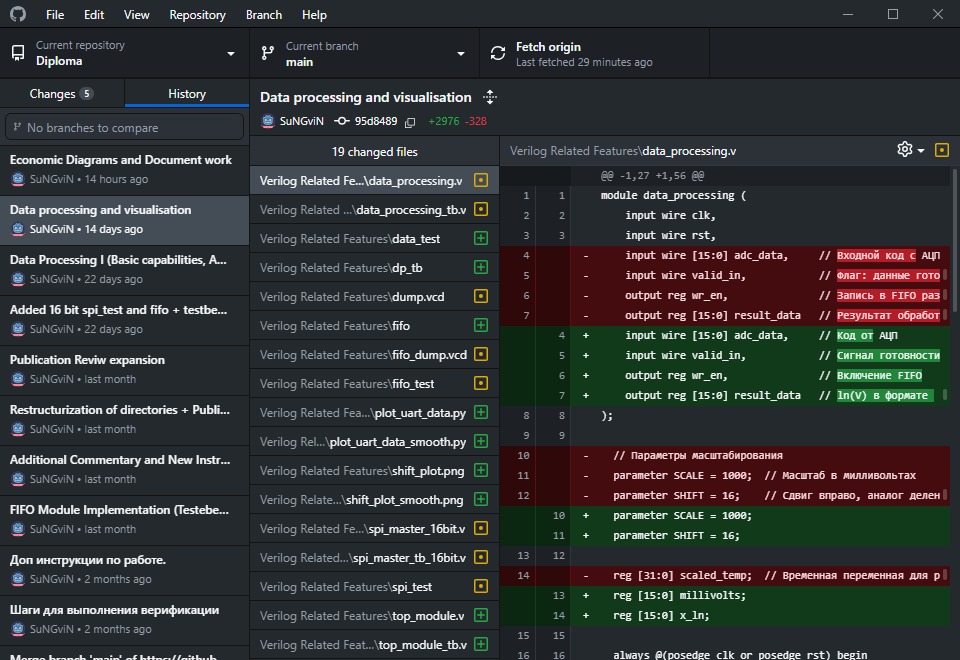


Рисунок 5 – Интерфейс приложения GitHub, версии для рабочего стола.

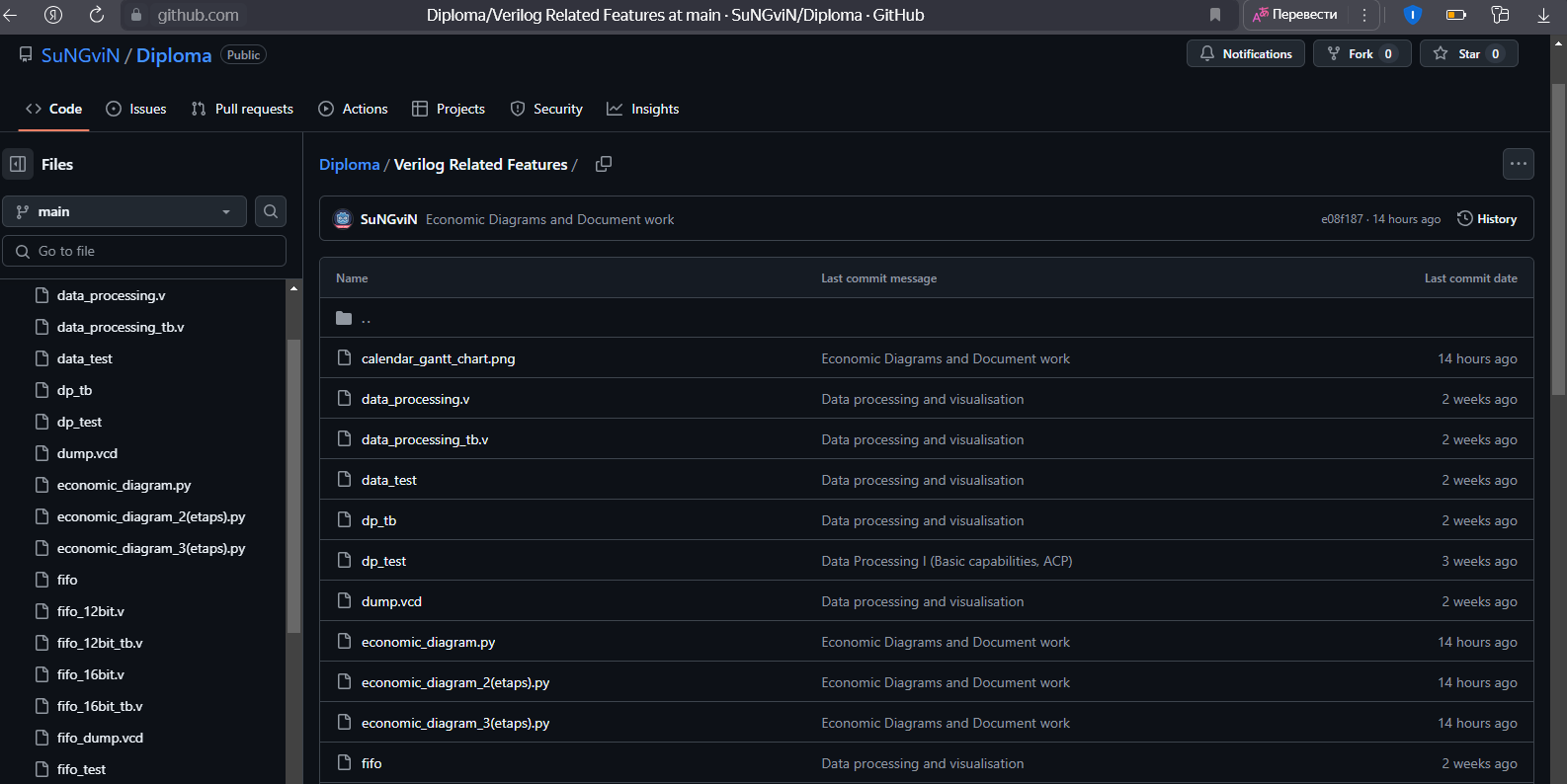


Рисунок 6 –Интерфейс веб-версии GitHub с репозиторием дипломной работы.

На этапе визуализации полученных цифровых результатов применялись Python 3.11 и библиотека matplotlib. Данные, полученные из Verilog (моделируемая UART-передача), сохранялись в текстовый файл и далее использовались для построения графиков логарифмического сдвига Δx.

# 

## 2.3 Этапы разработки системы

На первом этапе был создан модуль, реализующий приём данных с внешнего АЦП по интерфейсу SPI (Serial Peripheral Interface). Этот протокол широко используется для синхронного обмена данными между микросхемами. Изначально модуль работал с 12-битными словами, затем был модифицирован под 16-битный формат, более характерный для высокоточных АЦП.

Ключевые компоненты SPI-модуля:

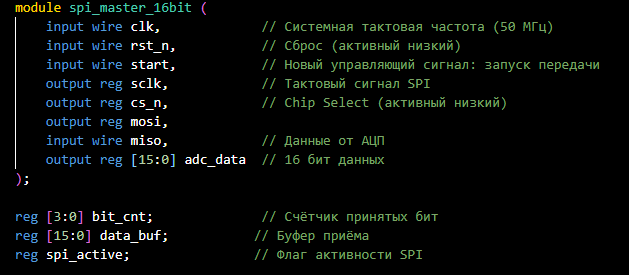


Рисунок 7 – Входные и выходные сигналы модуля spi\_master.

Сигналами являются sclk, cs\_n, mosi, miso. В модуле также присутствует счётчик битов и буфер приёма входных битов из miso.

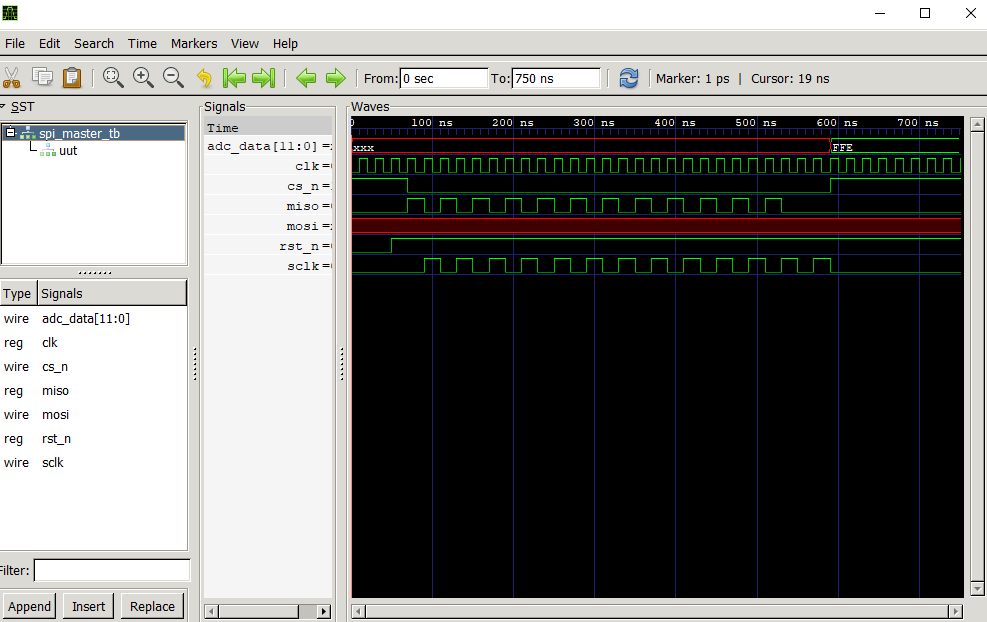


Рисунок 8 – Временная диаграмма работы модуля spi\_master.

Модуль был протестирован с помощью Verilog-тестбенча, имитирующего передачу данных. Диаграммы GTKWave подтвердили корректность работы счётчика, захвата битов и формирования готового результата.

Следующим этапом стал модуль FIFO (First In, First Out), предназначенный для временного хранения результатов обработки. Буфер обеспечивает раздельное управление чтением и записью, позволяет сгладить разницу в скорости поступления и обработки данных.

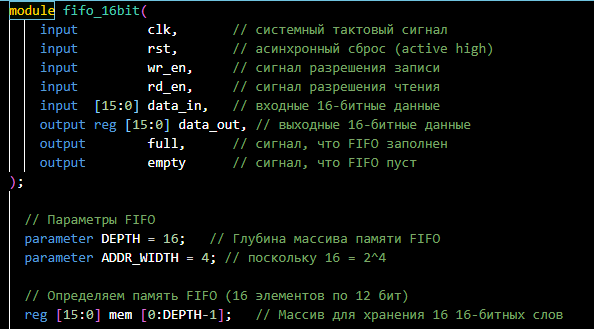


Рисунок 9 – Внутренние параметры модуля fifo.

Особенности реализации модуля: ширина данных 16 бит, управляющими сигналами являются wr\_en, rd\_en, full, empty, чтение данных происходит только, когда empty = 0.

Модуль также прошёл тестирование с записью и считыванием нескольких значений. GTKWave отразил работу указателей записи/чтения и заполнение буфера.

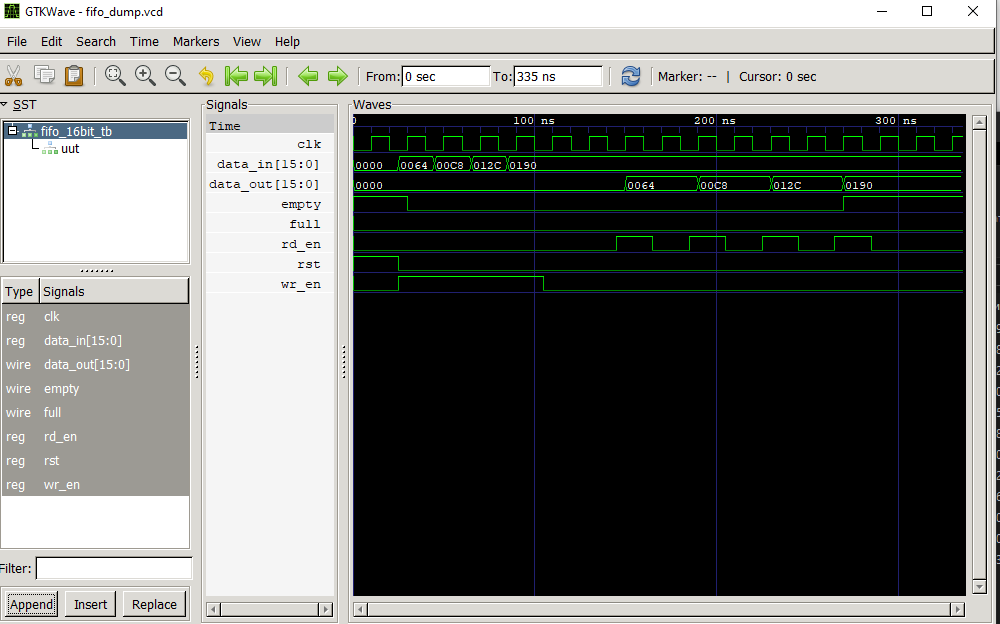


Рисунок 10 – Чтение поступающих данных и запись буфер fifo.

Центральной частью системы стал модуль, выполняющий перевод кода АЦП в физическое значение напряжения и последующую аппроксимацию логарифма этого значения. Работа велась в формате фиксированной запятой (8.8 – 8 бит отводится под целую часть числа, 8 – под часть после запятой), что позволило избежать необходимости в аппаратной арифметике с плавающей точкой.

Обработка включает:

* Перевод кода в милливольты:

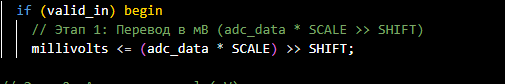


Рисунок 11 – Перевод бинарного кода из adc\_data в милливольты. SCALE = 1000, SHIFT = 16.

* Аппроксимацию ln(V) через кусочно-линейные зависимости (вычисление с коэффициентами a, b в разных интервалах значений)

Вычисление сдвига:

Также была реализована логика автоматического хранения предыдущего значения x\_prev, что позволило рассчитывать сдвиг между измерениями в реальном времени.

Для объединения всех компонентов был создан модуль top\_module.v, включающий:

* SPI-приёмник
* Модуль обработки
* FIFO-буфер

Сигналы были согласованы между модулями:

adc\_data → обработка

result\_data → FIFO

fifo\_output → UART / ПК

Также был добавлен механизм автоматического чтения FIFO, имитирующий последовательную передачу данных наружу.

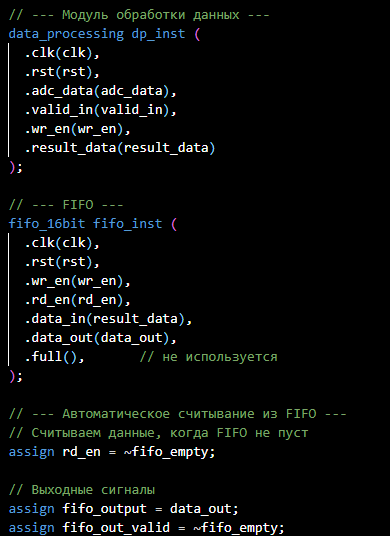


Рисунок 12 – Инстанцирование модулей fifo и data\_processing внутри top\_module, вывод данных, если буфер fifo не пуст.

На заключительном этапе был создан симуляционный интерфейс вывода, данные из FIFO автоматически записывались в файл uart\_output.txt, где каждое значение представляло собой логарифмический сдвиг Δx. Далее использовался Python-скрипт для чтения файла и построения графика.

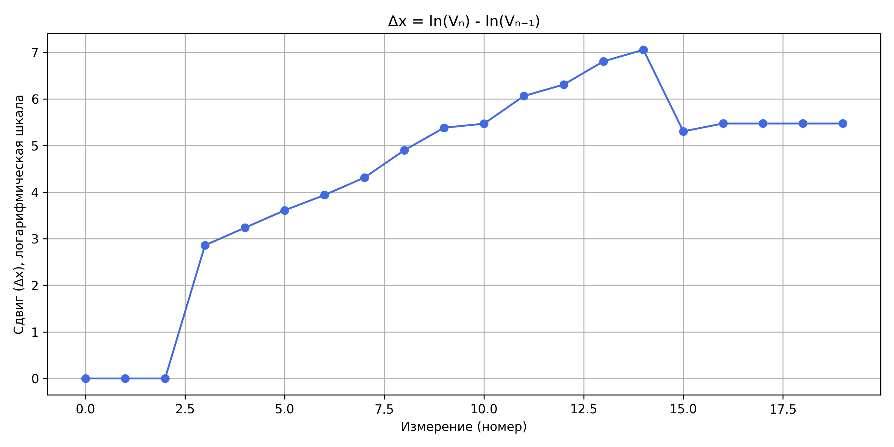


Рисунок 13 – Результат сбора и обработки информации.

На итоговом графике видна фоновая стабильность сигнала, резкий рост при увеличении напряжения на участке измерений №4–14, плато (насыщение) и последующий спад сигнала.

## 2.4 Верификация и моделирование

На каждом этапе разработки всех основных модулей системы была проведена поэтапная верификация каждого блока с использованием инструментов симуляции и анализа временных диаграмм. Основная цель верификации — убедиться в корректной работе цифровой логики, соответствии поведения ожиданиям и адекватности получаемых результатов.

Сначала верификация производилась на уровне SPI-модуля. С помощью Verilog-тестбенча в линию miso подавалась имитированная 16-битная последовательность, например 1010101010101010. Наблюдая в GTKWave сигналы cs\_n, sclk и состояние счётчика, удалось убедиться, что каждый бит корректно принимается и сдвигается в буфере.

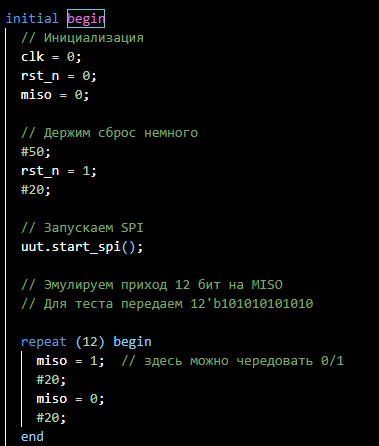


Рисунок 14 – Тестбенч для верификации SPI модуля.

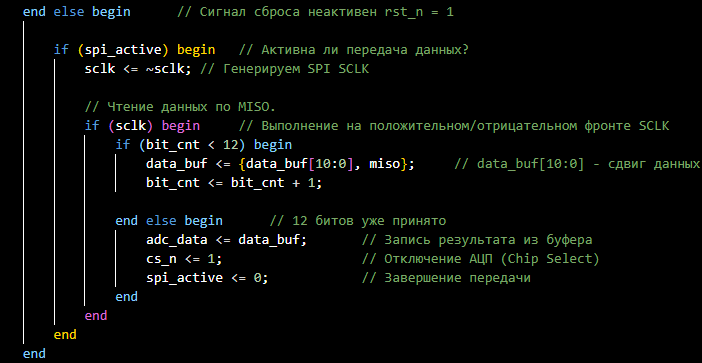


Рисунок 15 – Процесс чтения данных по miso.

В момент окончания приёма формировался сигнал adc\_data, фиксирующий полное 16-битное значение (см. рис. 16). Сигнал cs\_n автоматически переходил в высокий уровень, завершая транзакцию (см. рис. 17).

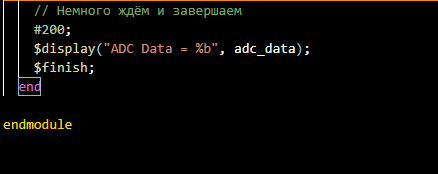


Рисунок 16 – Ожидание 200 нс, и вывод данных с помощью $display.

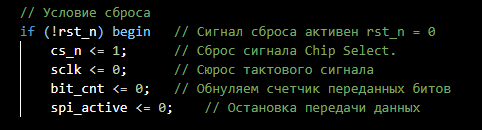


Рисунок 17 – Завершение процесса передачи данных в spi\_module.

Следующим этапом стало моделирование FIFO, обеспечивающего буферизацию между асинхронными потоками записи и чтения. В отдельном тестбенче выполнялась запись и считывание четырёх различных значений. С помощью GTKWave были проверены нарастание сигнала full при заполнении буфера, появление empty = 0 при записи, правильная последовательность выдачи данных при чтении

Особое внимание уделялось проверке расчётов внутри модуля обработки. В качестве теста использовались значения АЦП: 0x4000, 0x8000, 0xC000, 0xFFFF, соответствующие напряжениям примерно от 0.4 В до 1.6 В.

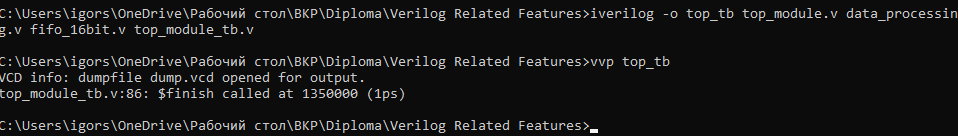


Рисунок 18 – Команды, необходимые для исполнения верификации top\_module.

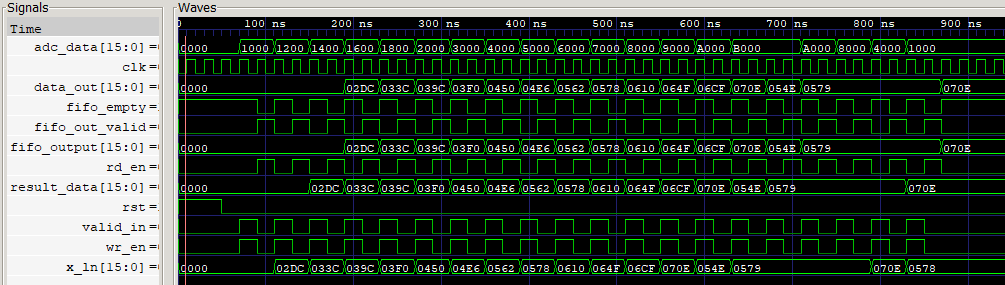


Рисунок 19 – Временная диаграмма из GTKWave модуля top\_module.

В GTKWave отслеживались сигналы:

millivolts — правильность масштабирования

x\_ln — значение логарифма

result\_data — сдвиг Δx

wr\_en — момент готовности нового результата

Поскольку в рамках практики не осуществлялась работа с реальной ПЛИС, было принято решение смоделировать передачу результата через UART. Для этого при каждом fifo\_out\_valid = 1 значение fifo\_output записывалось в файл uart\_output.txt, запись осуществлялась средствами Verilog ($fdisplay), каждая строка файла соответствовала одному измерению Δx

Для построения графика был разработан скрипт на Python с использованием библиотеки matplotlib. На основе содержимого uart\_output.txt создавался график зависимости логарифмического сдвига Δx от номера измерения.

# Результаты и обсуждение

Для тестирования обработки и визуализации была использована последовательность из 20 входных значений adc\_data. Поведение графика отражает типичный отклик биосенсора при воздействии внешнего агента (например, ионов, молекул или температуры).

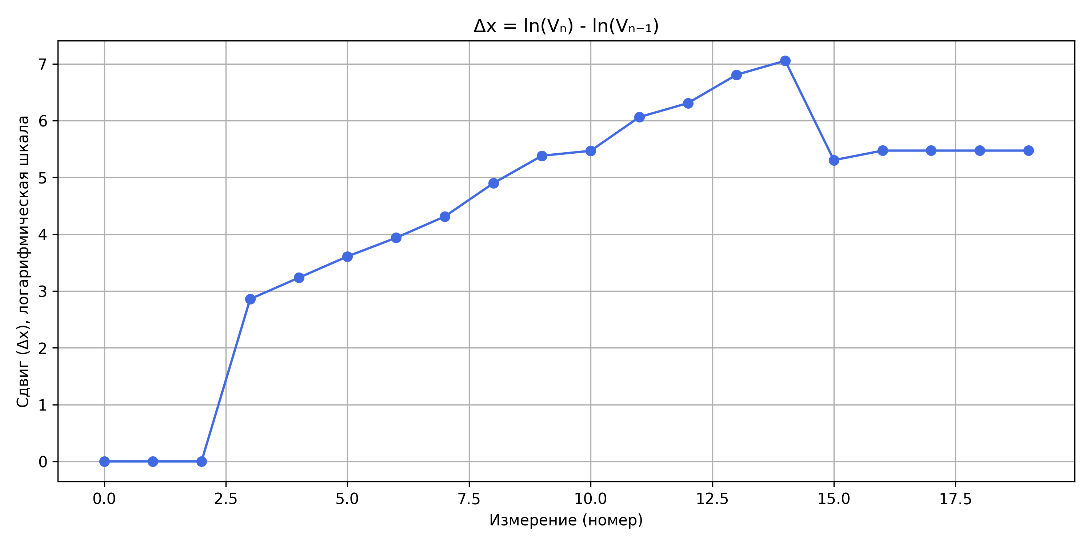


Рисунок 20 – Зависимость значений сдвига от № измерений.

Построенный график показывает изменение логарифма входного напряжения по сравнению с предыдущим измерением:

Когда значения Δx близки к нулю, сигнал стабилен, входное напряжение не изменяется, затем наблюдается быстрый рост Δx, соответствующий экспоненциальному подъёму напряжения. После насыщения логарифм выравнивается, изменения Δx становятся минимальными.

Такой график легко интерпретируется, является устойчивым к шумам и может служить показателем изменения физико-химических параметров среды.

Достоинства выбранного подхода:

* Логарифмическое преобразование усиливает малые изменения на низких уровнях сигнала.
* Переход от абсолютных значений к дифференциальному анализу (Δx) позволяет отсекать постоянную составляющую
* Модуль может быть использован с любым типом биосенсоров, выдающих аналоговый отклик

Формат хранения результатов в FIFO и организация передачи по UART (или через лог-файл) делает систему легко встраиваемой в реальное микропроцессорное или ПЛИС-решение. Возможные пути масштабирования:

1. Подключение нескольких каналов
2. Применение фильтрации
3. Построение диаграмм в реальном времени

# Заключение

В ходе преддипломной практики была выполнена полноценная разработка цифровой подсистемы сбора и обработки данных с КМОП-биосенсоров, реализованной на языке описания аппаратуры Verilog. Система охватывает все ключевые этапы обработки сигнала — от приёма оцифрованных данных по SPI-интерфейсу до расчёта логарифмического сдвига и формирования результирующего потока информации.

В процессе работы были успешно достигнуты следующие результаты:

* Освоены принципы работы интерфейса SPI и реализован приём 16-битных данных от внешнего источника (АЦП);
* Создан модуль буферизации (FIFO), обеспечивающий временное хранение результатов;
* Разработан математический модуль обработки, включающий преобразование данных в милливольты, аппроксимацию натурального логарифма и расчёт его сдвига между последовательными измерениями;
* Интегрирован верхнеуровневый модуль top\_module, объединяющий все элементы системы в единую архитектуру;
* Проведено тестирование с помощью симулятора Icarus Verilog и анализ временных диаграмм в GTKWave;
* Организован вывод результатов обработки в файл, эмулирующий передачу по UART;
* Выполнена визуализация логарифмического сдвига Δx с использованием Python и библиотеки matplotlib;

Полученные графики подтверждают корректность работы системы и возможность анализа динамики отклика биосенсора.

В процессе практики были также освоены современные средства командной и индивидуальной инженерной работы: система контроля версий Git, работа с GitHub, среда Visual Studio Code, а также приёмы структурирования проекта и подготовки отчётной документации.

Полученные навыки и программно-аппаратные наработки послужили основой для выполнения выпускной квалификационной работы и продемонстрировали готовность к решению инженерных задач в области цифровой обработки сигналов и микросистемной электроники.

# 4. Экономический раздел

## 4.1 Организация и планирование работ по теме

Состав задействованных в работе участников представлен на схеме.

Консультант

Буханцева С.Н.

Разработчик – Матевосов И.В.

Руководитель – Певцов Е.Ф.

* 1. Организация работ:

В таблице 1 приведён состав этапов работы с участвующими в разработке исполнителями.

Таблица 1

Этапы разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название этапа | Исполнитель | Трудоемкость,  чел/дни | Продолжительность работ, дни |
| 1 | Формализация технических требований и согласование спецификации | Руководитель | 5 | 5 |
| Консультант | 1 |
| Разработчик | 5 |
| 2 | Разработка функциональной концепции модуля | Руководитель | 7 | 7 |
| Разработчик | 7 |
| 3 | Эскизный проект: |  |  | 16 |
| 3.1 | Анализ исходных данных и требований | Разработчик | 9 |
| 3.2 | Постановка задачи | Консультант | 1 |
| 3.3 | Разработка общего описания алгоритма функционирования | Руководитель | 2 |
| Разработчик | 7 |
| 4 | Технический проект: |  |  | 15 |
| 4.1 | Определение формы представления входных и выходных данных | Руководитель | 2 |
| Разработчик | 5 |
| 4.2 | Разработка логической архитектуры и взаимодействия с ПЛИС | Руководитель | 2 |
| Разработчик | 10 |
| 5 | Рабочий проект: |  |  | 47 |
| 5.1 | Программирование и отладка программы | Разработчик | 24 |
| 5.2 | Испытание программы | Разработчик | 4 |
| 5.3 | Корректировка программы по результатам испытаний | Разработчик | 5 |
| 5.4 | Подготовка технической документации на программный продукт | Консультант | 1 |
| Разработчик | 7 |
| 5.5 | Сдача готового продукта и внедрение | Руководитель | 2 |
| Разработчик | 7 |
| Итого | | | | 90 |

В таблице 2 приведён состав этапов проектирования и разработки решения, планирование сроков выполняется с учетом: шестидневной рабочей (учебной) недели руководителя ВКР, консультанта по экономической части и студента, выходных и праздничных дней, а также состава участвующих в разработке исполнителей.

Таблица 2

Календарный план выполнения проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Дата**  **начала** | **Дата окончания** | **Количество рабочих дней** | **Исполнители** |
| **1 Формирование требований к проекту** | | | | |
| 1.1 Обследование объекта и обоснование необходимости создания проекта | 10.02.2025 | 11.02.2025 | 2 | Руководитель ПЭО  Руководитель ВКР  Разработчик проекта |
| 1.2 Формирование требований пользователя к проекту | 12.02.2025 | 12.02.2025 | 1 | Руководитель ПЭО  Руководитель ВКР  Разработчик проекта |
| 1.3 Оформление отчета о выполненной работе | 13.02.2025 | 13.02.2025 | 1 | Разработчик проекта |
| **2. Разработка концепции проекта** | | | | |
| 2.1 Изучение объекта | 14.02.2025 | 15.02.2025 | 2 | Разработчик проекта |
| 2.2 Проведение необходимых исследовательских работ | 17.02.2025 | 18.02.2025 | 2 | Разработчик проекта |
| 2.3 Разработка вариантов концепции ИТ-решения, выбор и согласование варианта, удовлетворяющего требованиям | 19.02.2025 | 20.02.2025 | 2 | Руководитель ПЭО  Руководитель ВКР  Разработчик проекта |
| 2.4 Оценка рисков проекта | 21.02.2025 | 21.02.2025 | 1 | Разработчик проекта |
| 2.5 Оформление отчета о выполненной работе | 22.02.2025 | 22.02.2025 | 1 | Разработчик проекта |
| **3. Техническое задание** | | | | |
| 3.1 Разработка и утверждение технического задания на ИТ-решение | 24.02.2025 | 26.02.2025 | 2 | Руководитель ПЭО  Разработчик проекта |
| 3.2. Разработка документации | 27.02.2025 | 28.02.2025 | 2 | Разработчик проекта |
| **4. Эскизный (пилотный) проект** | | | | |
| 4.1 Разработка предварительных проектных решений | 01.03.2025 | 03.03.2025 | 2 | Разработчик проекта |
| 4.2 Разработка программного кода ИТ-решения | 04.03.2025 | 10.03.2025 | 5 | Разработчик проекта |
| 4.3 Разработка документации на решение и его части с помощью GitHub Desktop | 11.03.2025 | 12.03.2025 | 2 | Разработчик проекта |
| **5. Технический проект** | | | | |
| 5.1 Разработка итоговых проектных решений | 13.03.2025 | 17.03.2025 | 4 | Руководитель ПЭО  Разработчик проекта |
| 5.2 Разработка итоговой архитектуры ИТ-решения | 18.03.2025 | 19.03.2025 | 2 | Разработчик проекта |
| 5.3 Доработка программного кода ИТ-решения | 20.03.2025 | 21.04.2025 | 31 | Разработчик проекта |
| 5.4 Разработка документации на решение и его части | 22.04.2025 | 23.04.2025 | 2 | Разработчик проекта |
| **6. Рабочая документация** | | | | |
| 6.1 Разработка рабочей документации на решение | 24.04.2025 | 26.04.2025 | 3 | Разработчик проекта |
| 6.2 Формирование комплекта рабочей документации на решение | 27.04.2025 | 28.04.2025 | 2 | Разработчик проекта |
| 6.3 Расчёт затрат на проведение работ | 29.04.2025 | 30.04.2025 | 2 | Разработчик проекта  Консультант по экономической части |

1.2 График проведения работ:

Календарный график исполнения работы, который соответствует таблице 1, представлен на рисунке 1. Из рисунка 1 так же видно, что общий срок разработки составит 90 дней.

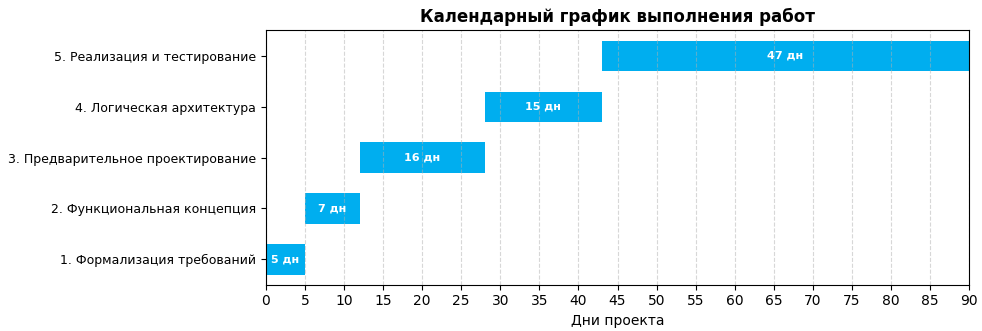


Рисунок 1. Календарный график выполнения работ

Календарный график исполнения проекта в соответствии с таблицей 1 представлен на рисунке 1.

На рисунке 2 представлена диаграмма Ганта,которая широко используется для визуализации хода выполнения задач, планирования ресурсов, графика рабочего времени и других данных, которые представляются набором временных интервалов [1-3]. Она необходима для соблюдения сроков выполнения работ, и, соответственно, анализа хода проектирования и/или разработки ИТ-решения в рамках ВКР.

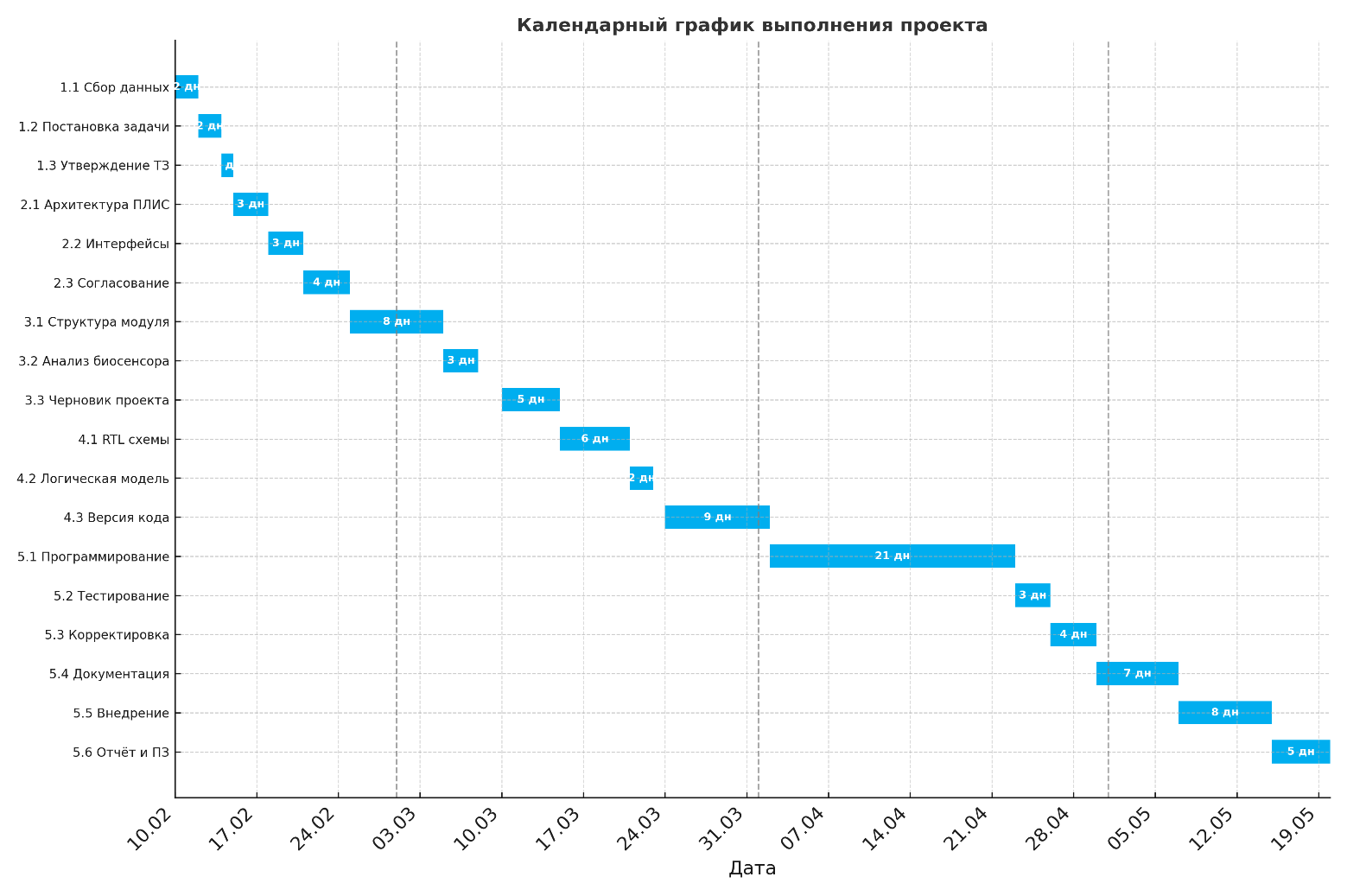


Рисунок 2. Диаграмма Ганта, отображающая процесс выполнения ВКР.

## 4.2 Расчёт стоимости проведения работ

Себестоимость работ (проектирования и/или разработки ИТ-решения) складывается из затрат по следующим статьям:

* 1 статья «Сырье и материалы» + ТЗР (15%) от ∑ итого по материалам
* статья «Основная заработная плата»
* статья «Дополнительная заработная плата» 20-30% от основной заработной платы
* статья «Страховые взносы» - 30% от ФОТ, а также 0,2% ставка за травматизм
* статья «Амортизация»
* статья «Прочие расходы»

В выпускной квалификационной работе объем затрат на НИР (проектирования и разработки решения) был вычислен методом калькулирования.

## 4.3 Статья «Сырье и материалы»

К этой статье относится: стоимость материалов, покупных изделий, полуфабрикатов, комплектующих изделий и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ.

При описании стоимости расходов на «Сырье и материалы» будем считать, что предприятие (университет) в своей учетной политике придерживается лимита стоимости основных средств в 100 тыс. руб. А это значит, что имущество со сроком полезного использования более 12 месяцев, но стоимостью ниже установленного лимита можно относить к расходам [4,5].

В стоимость материальных затрат включаются транспортно-заготовительные расходы, которые возьмем на уровне 5-15 % стоимости затрат по статье (таблица 3). В эту статью включаются также затраты на оформление комплекта документов.

Таблица 3

Стоимость материалов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ пп** | **Наименование**  **материалов** | **Единицы измерения** | **Кол-во** | **Цена за единицу (руб)** | **Стоимость (руб)** |
| **1** | **2** | **3** | **4** | **5** | **6** |
| 1 | Переходник USB-Blaster для программирования ПЛИС | Шт | 1 | 1800 | 1800 |
| 2 | Макетная плата (для отладки подключения биосенсоров) | Шт | 1 | 450 | 450 |
| 3 | Провода соединительные (Dupont кабели) | Набор | 1 | 250 | 250 |
| 4 | Блок питания 5 В / 2.5А | Шт | 1 | 800 | 800 |
| 5 | USB-флешка 8 ГБ для загрузки прошивки | Шт | 1 | 339 | 339 |
| 6 | Картридж для принтера | Шт | 1 | 2989 | 2989 |
| 7 | Канцелярский набор | Шт | 1 | 480 | 480 |
| 8 | Комплект документации и распечаток | Комплект | 1 | 300 | 300 |
| **Итого материалов** | | | | | **7 408,0** |
| **Транспортно-заготовительные расходы** | | | | | **1 111,2** |
| **Итого** | | | | | **8 519,2** |

Затраты на комплектующие и изготовление программно-аппаратного комплекса учитаны в отдельной таблице (табл. 4).

Таблица 4

Стоимость комплектующих

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Наименование**  **элемента** | **Единицы**  **измерения** | **Кол-во** | **Цена за**  **единицу (руб.)** | **Стоимость**  **(руб.)** |
| 1 | ПЛИС Intel Cyclone V (DE10-Nano или аналогичный модуль) | Шт | 1 | 13500 | 13500 |
| 2 | Сенсорный макет (имитатор сигнала от КМОП-биосенсора) | шт | 1 | 2500 | 2500 |
| Итого комплектующих | | | | | 16 000,0 |
| Транспортно-заготовительные расходы | | | | | 2 400,0 |
| **Итого:** | | | | | **18 400,0** |

Таким образом общая сумма затрат по статье «Сырье и материалы» составит 26 919,2 руб.

## 4.4 Статья «Основная заработная плата»

К статье «Основная заработная плата» относится оплата труда научных работников, инженерно-технических работников и рабочих, непосредственно занятых выполнением конкретной работы, а также заработная плата работников внештатного состава, привлекаемых к ее выполнению. Расчет заработной платы проводится исходя из численности различных категорий исполнителей, трудоемкости отдельных видов выполняемых ими работ и их тарифной ставки.

Согласно Производственному календарю на 2025 год, годовой фонд рабочего времени при *шестидневной рабочей неделе* составит 299 рабочих дней [6].

Следовательно, дневная тарифная ставка (ТС) для месячного оклада (ОК) по каждому работнику может быть рассчитана следующим образом:

где:

ОК — месячный оклад участника проекта;

12 — число месяцев в году;

НРВ — годовой фонд рабочего времени при сорокачасовой рабочей неделе.

Рассчитаем тарифную ставку для месячного оклада (ОК) по каждому работнику.

Оклад разработчика = 15 000 руб

Оклад консультанта = 110 000 руб

Оклад руководителя = 130 000 руб

Расчёт дневной ставки:

В таблице 5 приведен расчет заработной платы исполнителей. Так как руководитель и консультант - преподаватели университета, то расчет дневной тарифной ставки проводим по формуле (1). Количество рабочих дней для каждой роли определяется на основе таблице 1 с учетом участия в выполнении работ.

Таблица 5

Расчет основной заработанной платы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ пп** | **Наименование этапа** | **Исполнитель (должность)** | **Мес. оклад (руб)** | **Трудоемкость (чел/дни)** | **Оплата за день (руб)** | **Оплата за этап (руб)** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | ТЗ | Руководитель | 130 000 | 5 | 5217 | 26085 |
| Консультант | 110 000 | 1 | 4414 | 4214 |
| Разработчик | 15 000 | 5 | 602 | 3010 |
| 2 | ТП | Руководитель | 130 000 | 7 | 5217 | 36519 |
|  |  | Разработчик | 15 000 | 7 | 602 | 4214 |
| 3 | Эскизный проект | Руководитель | 130 000 | 2 | 5217 | 10434 |
| Консультант | 110 000 | 1 | 4414 | 4214 |
| Разработчик | 15 000 | 16 | 602 | 9632 |
| 4 | Технический проект | Руководитель | 130 000 | 4 | 5217 | 20868 |
| Разработчик | 15 000 | 15 | 602 | 9030 |
| 5 | Рабочий проект | Руководитель | 130 000 | 2 | 5217 | 10434 |
| Консультант | 110 000 | 1 | 4414 | 4214 |
| Разработчик | 15 000 | 47 | 602 | 28294 |
| **Итого** | | | | | | **171 162** |

## 4.5 Статья «Дополнительная заработная плата»

На эту статью относятся выплаты, предусмотренные законодательством о труде за неотработанное по уважительным причинам время; оплата очередных и дополнительных отпусков; времени, связанного с выполнением государственных и общественных обязанностей; выплата вознаграждения за выслугу лет и т.п.

В среднем расходы по данной статье составляют 20-30% от суммы основной заработной платы.

(3)

ДЗП = 0,2 × 117 162 = 34 232,4 руб.

В процессе определения сметы затрат вводится понятие «фонд оплаты труда», представляющий собой сумму основной (ОЗП) и дополнительной заработной платы (ДЗП). Фонд оплаты труда (ФОТ) используется при расчете взносов в социальный фонд.

(4)

ФОТ = ОЗП + ДЗП = 171 162,00 + 34 232,40 = 205 394,40руб.

Для данного проекта фонд оплаты труда составит 205 394,4 руб.

## 4.6 Статья «Страховые взносы»

В соответствии со статьей 425 НК РФ в 2025 году для лиц, которые производят выплаты и вознаграждения физическим лицам (за исключением плательщиков, для которых установлены пониженные тарифы страховых взносов), установлен Единый тариф страховых взносов 30 % [9] по направлениям:

* обязательное пенсионное страхование (ОПС),
* обязательное социальное страхование на случай временной нетрудоспособности и в связи с материнством (ОСС),
* обязательное медицинское страхование (ОМС) [8].

Также в расчёте необходимо учесть ставку взносов на травматизм, которая может составлять от 0,2% до 8,5% [10,11,12]. Для РТУ МИРЭА данная ставка составляет 0,2% [12].

(5)

СВ = 0,302 \* 205 394,40 = 62 028,11

## 4.7 Статья «Амортизация»

Основные средства в структуре себестоимости учитываются только через амортизацию. Т.е. затраты на закупку оборудования или оборудование, на котором выполнялась научная работа (проект, выпускная квалификационная работа), учитываются через определение амортизационных отчислений.

Амортизация — отчисления части стоимости основных фондов (например, стоимости оборудования) для возмещения их износа. Амортизация включается в издержки производства. Размер амортизационных отчислений определяется на основе установленных норм и балансовой стоимости основных фондов, на которые начисляется амортизация [13].

Будем считать, что предприятие (университет) в своей учетной политике придерживается лимита стоимости основных средств в 100 тыс. руб. А это значит, что объекты стоимостью до этого предела не относятся к амортизируемому имуществу [5].

В рамках определения стоимости работ по проекту используется линейный способ вычисления амортизационных отчислений.

Амортизационные отчисления вычисляются по формуле:

(6)

где:

А - месячная сумма амортизационных отчислений,

С - первоначальная стоимость объекта,

Т – срок полезного использования в месяцах.

В таблице 6 представлен расчет амортизационных отчислений по использованному во время разработки проекта оборудованию.

Таблица 6

Амортизационные отчисления

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование оборудования** | **Первоначаль-ная стоимость объекта, руб.** | **Срок полезного использования, мес.** | **Месячная сумма аморти-зационных отчислений** | **Период эксплу-атации в месяцах** | **Сумма, руб.** |
| Ноутбук | 100 000 | 36 | 2 777,78 | 3 | 8 333,34 |
| Сервер | 132 290 | 36 | 3674,72 | 3 | 11 024,16 |
| Итого: | | | | | 19 357,50 |

## 4.8 Статья «Прочие расходы»

К прочим расходам прежде всего отнесены накладные расходы. Это расходы на содержание и ремонт зданий, сооружений, оборудования, инвентаря. Это затраты, сопутствующие основному производству, но не связанные с ним напрямую, не входящие в стоимость труда и материалов.

Сумма данных расходов определяется процентом от суммы основной заработной платы (ОЗП) и на разных предприятиях, в зависимости от их структуры, технологического процесса и системы управления находится в широком диапазоне — от 100 до 130%.

(7)

НР=1,3×171162= 222 510,60руб.

​

Вышеперечисленные статьи затрат и результаты расчётов по ним обобщаются в таблице 7.

Таблица 7

Полная себестоимость работ

|  |  |  |  |
| --- | --- | --- | --- |
| **№ пп** | **Номенклатура статей расходов** | **Затраты (руб.)** | **Доля затрат, %** |
| **1** | **2** | **3** | **4** |
| 1 | Сырье и материалы | 26 919,20 | 5,7 |
| 2 | Основная заработная плата | 171 162,00 | 36,3 |
| 3 | Дополнительная заработная плата | 34 232,40 | 7,3 |
| 4 | Страховые взносы | 62 028,11 | 13,2 |
| 5 | Амортизация | 19 357,51 | 4,1 |
| 6 | Прочие расходы | 222 510.60 | 47,2 |
| **Итого** | | **536 209,81** | **100,0** |

Для визуализации долевого состава статей затрат в общей себестоимости представим круговую диаграмму на рисунке 3.



Рисунок 3. Структура затрат по работе

Для дальнейшей реализации продукта проведём расчёт договорной цены.

|  |  |
| --- | --- |
|  | (8) |

Норма прибыли составляет 10-30% от стоимости разработки. Прибыль будет равна:

П=20% × 536 209,81=107 241,96руб.

Рассчитаем налог на добавленную стоимость (НДС) в размере 20%:

(9)

НДС = (536209,81+107241,96) х 20/100 =128 690,76руб.

Таким образом, договорная цена будет представлять собой:

ДЦ = С + П + НДС = 536 209,81 + 107 241,96 + 128 690,76= 772 142,53руб.

В рамках данного экономического раздела было выполнено планирование работ по теме «Модуль сбора и обработки данных с КМОП биосенсоров», был проведен расчет стоимости затрат и определена договорная цена.

**Список источников и литературы**

1. Диаграмма Ганта [Электронный ресурс]. — 2025. URL: <https://v8.1c.ru/platforma/diagramma-ganta/>?
2. GanttPRO: Диаграмма Ганта для управления проектами [Электронный ресурс]. — 2025. URL: <https://ganttpro.com/ru/>
3. Диграмма Ганта [Электронный ресурс]. — 2025. URL: <https://www.unisender.com/ru/blog/idei/chto-takoe-diagramma-ganta-i-kak-ee-sdelat/>
4. Стандарт ФСБУ 6/2020 «Основные средства» <https://www.kontur-extern.ru/info/45930-uchet_osnovnyh_sredstv_po_fsbu_6_2020>
5. НК РФ Статья 254. Материальные расходы, Федеральный закон от 05.08.2000 N 117-ФЗ (ред. от 13.12.2024) [Электронный ресурс]. — 2024. URL: <https://www.consultant.ru/document/cons_doc_LAW_28165/0644a51c8d171aad7127867a97d0749ec20be875/>
6. Производственный календарь на 2025 г. [Электронный ресурс]. — 2025. URL: <https://nalog-nalog.ru/proizvodstvennyj_kalendar/2025-6/>
7. Производственный календарь на 2025 г. [Электронный ресурс]. — 2025. URL: <https://static.consultant.ru/obj/file/calendar/calendar_2025.pdf>
8. НК РФ Статья 425. Тарифы страховых взносов. Федеральный закон от 05.08.2000 N 117-ФЗ (ред. от 13.12.2024) [Электронный ресурс]. — 2025. URL: <https://www.consultant.ru/document/cons_doc_LAW_28165/a3f603ffd57b1431ed51e1693ba710093347235d/>
9. Единый тариф (единые размеры) страховых взносов на ОПС, ОСС, ОМС для плательщиков, производящих выплаты физическим лицам, на 2025 г. [Электронный ресурс]. — 2025. URL: <https://www.consultant.ru/document/cons_doc_LAW_93256/97a8d65271700b79d87a7edf33832b5f3ce0fd18/>
10. Федеральный закон от 24.07.1998 N 125-ФЗ (ред. от 29.05.2024) "Об обязательном социальном страховании от несчастных случаев на производстве и профессиональных заболеваний" [Электронный ресурс]. — 2024. URL: <https://www.consultant.ru/document/cons_doc_LAW_19559/>
11. Федеральный закон от 26.10.2024 N 352-ФЗ "О страховых тарифах на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний на 2025 год и на плановый период 2026 и 2027 годов". [Электронный ресурс]. — 2025. URL: <https://www.consultant.ru/document/cons_doc_LAW_489049/>
12. Приказ Минтруда России от 30.12.2016 N 851н (ред. от 29.08.2024) "Об утверждении Классификации видов экономической деятельности по классам профессионального риска" (Зарегистрировано в Минюсте России 18.01.2017 N 45279) [Электронный ресурс]. — 2025. URL: <https://www.consultant.ru/document/cons_doc_LAW_211247/>
13. НК РФ Статья 256. Амортизируемое имущество. Федеральный закон от 05.08.2000 N 117-ФЗ (ред. от 13.12.2024) [Электронный ресурс]. — 2024. URL: <https://www.consultant.ru/document/cons_doc_LAW_28165/df53ee1751d3e93dbf8c0d34076675da18a2fd06/>
14. Методические указания по выполнению экономической части выпускной квалификационной работы бакалавра. : методические указания / И. В. Шацкая, И. В. Белоусова .— М. : РТУ МИРЭА , 2024 [Электронный ресурс]. — 2024. URL: <https://ibc.mirea.ru/books/SHARE/55177>
15. Экономика предприятия. : учеб. пособие для вузов / И. А. Назарова, А. С. Вихрова .— СПб. : Лань , 2023 .— 71 с. [Электронный ресурс]. — 2023. URL: <https://ibc.mirea.ru/books/BOOKS/56300>
16. Ценообразование в сфере информационных технологий: практикум / А. С. Вихрова .— М. : РТУ МИРЭА , 2024 [Электронный ресурс]. — 2024. URL: <https://ibc.mirea.ru/books/SHARE/55220>
17. Экономика предприятия: учебное пособие / А. С. Вихрова. — М.: РТУ МИРЭА, 2021. — Электрон. опт. диск (ISO). [Электронный ресурс]. — 2021. URL: <https://ibc.mirea.ru/books/SHARE/4196>

# Список использованной литературы

1. E.O. Polat, M.M. Cetin, A.F. Tabak, E. Bilget Güven, B.O. Uysal, T. Arsan, А. Каббани, Х. Хамед, С.Б. Гюль, Технологии преобразователей для биосенсоров и их носимых приложений, Biosensors (Базель) 12 (2022) 385.
2. Д. Верма, К.Р. Сингх, А.К. Ядав, В. Наяк, Дж. Сингх, П.Р. Соланки, Р.П. Сингх, Интернет вещей (IoT) в наноинтегрированных носимых биосенсорных устройствахдля приложений здравоохранения, Biosens Bioelectron X 11 (2022) 100153, https://doi. org/10.1016/j.biosx.2022.100153.
3. К.Р. Сингх, В. Наяк, Дж. Сингх, Р.П. Сингх, Носимые датчики с поддержкой нанотехнологий для интернета вещей (IoT), c. 304 (2021) 130614, https://doi.org /10.1016/j.matlet.2021.130614.
4. К.Р.Б. Сингх, С. Рати, Г. Нагпуре, Дж. Сингх, Р.П. Сингх, Умные и развивающиеся биосенсоры на основе наноматериалов для обнаружения SARS-CoV-2, с. 307(2022) 131092, <https://doi.org/10.1016/j.matlet.2021.131092>.
5. Г. Гопал, М. Кумават, Т. Варма, Недавние достижения в области биосенсоров на основе TFET. Техника адресации и результаты: обзор, в: 2024: стр. 119-145. <https://doi.org/10.1007/978-981-97-3048-3_7>.
6. М. Джаваид, А. Халим, С. Раб, Р.Пратап Сингх, Р. Суман, Сенсоры для повседневной жизни: Обзор International 2 (2021), https://doi.org/10.1016/j.sintl.2021.100121
7. Биосенсор, в: Сборник химической терминологии ИЮПАК, Международный союз теоретической и прикладной химии (IUPAC), Исследовательский парк Triangle Park,Северная Каролина, 2014. <https://doi.org/10.1351/goldbook.B00663>.
8. Варнакави. Нареш, Н. Ли, Обзор биосенсоров и последних разработок. Биосенсоры с поддержкой наноструктурированных материалов, сенсоры 21 (2021) 1109, <https://doi.org/10.3390/s21041109>.
9. К.Р. Сингх, Р.П. Сингх, Полезность нанобиосенсоров в анализе окружающей среды и мониторинге систем, в: 2021: стр. 229-246. <https://doi.org/10.1007/978-3-030-63245-8_11>.
10. Д. Бхатия, С. Пол, Т. Ачарджи, С.С. Рамачайри, Биосенсоры и их широкое распространение; воздействие на здоровье человека, Sensors International 5 (2024) 100257, https://doi.org /10.1016/j.sintl.2023.100257.
11. М.А. Ислам, П.К. Датта, Х. Майлер, структуры СБИС для секвенирования ДНК-Обзор, Биоинженерия 7 (2020) 49, <https://doi.org/10.3390/bioengineering7020049>.
12. Р.Р. Сингх, А. Маникам, С. Аязян, А. Хассиби, Д. Шахрджерди, с поддержкой СБИС. Массивы для секвенирования ДНК, 2011 IEEE 54th International Midwest Symposium по теме схемы и системы (MWSCAS), IEEE, 2011, стр. 1-4, https://doi.org/10.1109 /MWSCAS.2011.6026444.
13. З. Чжан, Дж. Ху, С. Линь, Дж. Ву, Ф. Ся, Х. Лу, биосенсоры на полевых транзисторах для мониторинга здравоохранения, междисциплинарная медицина 2 (2024), https://doi.org /10.1002/INMD.20240032.
14. Андрианова М.С., Панова О.С., Титов А.А., Комарова Н.В., Кузнецов А.Е. Электрохимические биосенсоры для определения SARS-CoV-2 // Вестн. Моск. ун-та. Сер. 2. Химия. 2023. Т. 64. № 5. С. 407–440.
15. Cheng, W., Zheng, Z., Yang, J., Chen, M., Yao, Q., Chen, Y., et al. (2019). The visible light-driven and self-powered photoelectrochemical biosensor for organophosphate pesticides detection based on nitrogen doped carbon quantum dots for the signal amplification. Electraochim. Acta 296, 627–636. doi:10.1016/j.electacta.2018.11.086
16. Источник изображения: URL: <https://www.biotechnologies.ru/catalog/_Chto_takoe_biosensor.html?ysclid=m9i6i4o6y7112854108> (дата обращения: 12.04.2025)
17. Муратулы М., Байдельдинов М.У. Использование FPGA в индустрии: Перспективы и изменения // Universum: технические науки : электрон. научн. журн. 2024. 4(121). URL: <https://7universum.com/ru/tech/archive/item/17353>
18. Держановский А.С., Соколов С.М. Обработка зрительных данных в системе технического зрения реального времени с использованием ПЛИС // Препринты ИПМ им. М.В.Келдыша. 2016. № 126. 16 с. doi:10.20948/prepr-2016-126 URL: <http://library.keldysh.ru/preprint.asp?id=2016-126>
19. Андреев А. Е., Силкин И. М., Шафран Ю. В. Высокоуровневые средства разработки для FPGA // Современные научные исследования и инновации. – Июнь, 2012 [Электронный ресурс]. – URL: <http://web.snauka.ru/issues/2012/06/14365> (дата обращения 17.03.25).
20. LegUp: High-Level Synthesis for FPGA-Based Processor/Accelerator Systems [Электронныйресурс] / A. Canis [и др.] // University of Toronto, Altera Toronto. – 2011. – URL: http://legup.eecg.utoronto.ca/fpga60-legup.pdf (дата обращения 12.03.25).
21. Редькин П. Практика использования встроенного АЦП в ПЛИС семейства MAX10 Часть 1. Справочная информация по АЦП ПЛИС MAX10 // Современная электроника № 2 2021
22. Ануфриенко А.Ю. Исследование протоколов для беспроводных IoT устройств и методов обработки информации: дис. … канд. тех. наук: 02.23.04 / НИУ Высшая школа экономики. – М., 2022. – 138 с.