

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра динамического моделирования и биомедицинской инженерии

Устройство для параллельной цифровой обработки биологических
сигналов на основе ПЛИС

наименование темы выпускной квалификационной работы

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 4081 группы

направления 12.03.04 «Биотехнические системы и технологии»

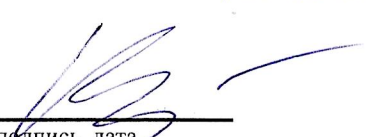
профиль подготовки «Методы и устройства обработки биосигналов»

институт физики

Косолапов Роман Сергеевич

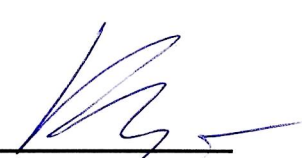
фамилия, имя, отчество

Научный руководитель:
Зав. кафедрой динамического
моделирования и
биомедицинской инженерии,
д.ф.-м.н., профессор

«Отлично»

подпись, дата
20.06.2025

А.С. Караваев

Зав. кафедрой динамического
моделирования и
биомедицинской инженерии,
д.ф.-м.н., профессор


подпись, дата
20.06.2025

А.С. Караваев

Саратов 2025

Введение. Обработка биологических сигналов, таких как электроэнцефалограмма (ЭЭГ), электрокардиограмма (ЭКГ), электромиограмма (ЭМГ), играет ключевую роль в современной медицинской диагностике. С развитием цифровых технологий всё большее значение приобретают аппаратные решения, позволяющие обеспечить минимальные задержки, высокую точность и надёжность обработки данных в реальном времени. Программные подходы, основанные на использовании микропроцессоров или DSP, не всегда могут удовлетворять требованиям быстродействия и энергоэффективности, особенно при работе с многоканальными системами. Поэтому особый интерес вызывает использование программируемых логических интегральных схем (ПЛИС), которые благодаря своей внутренней архитектуре обеспечивают истинный параллелизм вычислений, детерминированную работу и низкое энергопотребление.

Целью данной работы является разработка устройства параллельной цифровой обработки биологических сигналов на основе ПЛИС, включающего восемь КИХ-фильтров, модуль мультиплексирования и Ethernet-интерфейс передачи данных. Представленное устройство позволяет эффективно обрабатывать данные в реальном времени, сохраняя информацию о каждом канале и обеспечивая возможность дальнейшего анализа на персональном компьютере.

Для достижения этой цели были определены следующие задачи:

1. Обзор существующих методов параллельной цифровой обработки биологических сигналов.
2. Обзор и сравнение интерфейсов передачи данных с ПЛИС на персональный компьютер.
3. Блок-схема разрабатываемого устройства для параллельной цифровой обработки биологических сигналов на основе ПЛИС.
4. Разработка интерфейса приема и передачи биологических данных на персональный компьютер.

5. Реализация модуля на ПЛИС, отвечающего за цифровую обработку данных.

6. Разработка программного обеспечения на персональный компьютер для приема и передачи данных с ПЛИС.

7. Тестирование разработанной системы на примере обработки заранее зарегистрированных временных рядов сигнала электроэнцефалограммы.

Объектом исследования является устройство для параллельной цифровой обработки биологических сигналов на основе ПЛИС. Материалы исследования включают современные методы для цифровой обработки данных, методы передачи данных, методы буферизации данных. Для разработки использовалась отладочная плата на Zynq 7010. В ходе работы применялись методы ЦОС, навыки описания цифровых схем на языке Verilog, навыки программирования на языке Python.

Структура выпускной квалификационной работы состоит из введения, заключения, списка использованных источников и основного содержания работы. Основное содержание работы состоит из 7 пунктов, а именно:

1. Обзор существующих методов параллельной цифровой обработки биологических сигналов

2. Обзор и сравнение интерфейсов передачи данных с ПЛИС на персональный компьютер

3. Блок схема разрабатываемого устройства для параллельной цифровой обработки биологических сигналов на основе ПЛИС

4. Разработка интерфейса приема и передачи биологических данных на персональный компьютер

5. Реализация модуля на ПЛИС, отвечающего за цифровую обработку данных

6. Разработка программного обеспечения на персональный компьютер для

приема и передачи данных с ПЛИС

7. Тестирование разработанной системы на примере обработки заранее зарегистрированных временных рядов сигнала электроэнцефалограммы.

Основное содержание работы. В современных медицинских системах всё чаще возникает необходимость высокоскоростной обработки многоканальных биомедицинских сигналов, таких как электроэнцефалограмма (ЭЭГ), электрокардиограмма (ЭКГ) или электромиограмма (ЭМГ). Эти сигналы характеризуются: низкой амплитудой, высоким уровнем шума

Традиционные подходы к обработке, основанные на программных алгоритмах на универсальных процессорах (CPU), сталкиваются с рядом ограничений:

1.Последовательная обработка не позволяет эффективно обрабатывать сотни каналов одновременно.

2.Задержки при выполнении свёрток и преобразований Фурье могут быть критичными для клинической диагностики.

3.Ограниченная масштабируемость — добавление новых каналов резко увеличивает нагрузку на вычислительную систему.

Поэтому всё большее внимание уделяется аппаратным решениям, способным обеспечить истинный параллелизм, минимальные задержки и низкое энергопотребление.

На сегодняшний день наиболее популярны три подхода к параллельной обработке сигналов:

1.Цифровые сигнальные процессоры (DSP). Оптимизированы для выполнения операций ЦОС (цифровая фильтрация, БПФ, корреляция). Поддерживают конвейеризацию и SIMD-инструкции.

2.Графические процессоры (GPU)Предоставляют массовый параллелизм благодаря тысячам ядер. Хорошо подходят для регулярных вычислений.

3.Программируемые логические интегральные схемы (ПЛИС). Реализуют аппаратный параллелизм. Детерминированная работа позволяет использовать их в системах реального времени. Низкое энергопотребление делает ПЛИС особенно перспективными для носимых устройств. [1]

Именно поэтому в данной работе был выбран путь реализации системы на основе ПЛИС, позволяющий сочетать гибкость программного уровня с производительностью аппаратного исполнения. [2]

Для эффективной работы системы необходимо организовать надёжный и быстрый обмен данными между ПЛИС и внешним миром. В рамках диплома были рассмотрены следующие интерфейсы:

1. PCI Express (PCIe): высокая пропускная способность (до нескольких Гигабайт в секунду), используется в мощных медицинских системах, сложность реализации: требует специализированных IP-ядер и хорошего знания протокола.

2. USB 3.0 / USB-C: широко распространённый интерфейс, пропускная способность до 5–10 Гбит/с, подходит для портативных устройств.

3. Ethernet: высокая устойчивость к помехам, возможность удалённого мониторинга через сеть, протокол UDP обеспечивает минимальные накладные расходы, легко интегрируется в существующую IT-инфраструктуру.

4. I2C: низкая скорость (до 400 Кбит/с – 5 Мбит/с), подходит только для малых объёмов данных, используется в микроконтроллерных системах.

5. Wi-Fi и Bluetooth: подходят для беспроводной передачи, Wi-Fi — до 9,6 Гбит/с (Wi-Fi 6), Bluetooth — до 3 Мбит/с (Bluetooth 5.0), подвержены помехам и рассинхронизации.

Выбор в пользу Ethernet был сделан из-за его устойчивости к помехам, возможности организации сетевой передачи и совместимости с IP-сетями. Также он позволяет отправлять данные на удалённый ПК без дополнительных драйверов и поддерживает работу в реальном времени.

Разработанная система состоит из следующих ключевых модулей:

1. Модуль входных данных — получает данные от АЦП, хранит их во временных регистрах и передаёт дальше по цепочке.
2. Модуль цифровой обработки — восемь независимых КИХ-фильтров, каждый из которых обрабатывает свой канал.
3. Модуль буферизации и мультиплексирования — согласует скорости работы блоков, объединяет данные всех каналов в один поток.
4. Модуль передачи данных по Ethernet — отправляет данные по сети с помощью IP-ядра MAC.

Система работает строго по тактовому сигналу (clk) и управляется сигналом rst_n (сброс). Все модули взаимодействуют друг с другом через сигналы достоверности (data_valid) и готовности (data_ready), что позволяет избежать потерь данных и обеспечить детерминированную работу.[3]

Интерфейс приёма и передачи играет ключевую роль в системе. Он состоит из двух частей:

Приёмная часть. Обеспечивает корректное считывание данных от аналого-цифровых преобразователей (АЦП) и их синхронизацию с внутренним тактовым сигналом. Для этого используется 8-канальный входной регистр (adc_data_0–adc_data_7), который принимает 16-битные значения.

При активации сигнала adc_ready данные записываются в соответствующие регистры и направляются в фильтры.[4]

Передающая часть. После фильтрации результаты направляются в мультиплексор, где к каждому 32-битному значению добавляется 4-битный номер канала. Это позволяет легко определить источник данных на стороне ПК.

Сформированный пакет передаётся через Ethernet MAC по протоколу UDP. Такой выбор обусловлен минимальными накладными расходами и возможностью работы в реальном времени.[5]

Все КИХ-фильтры реализованы аппаратно, на уровне железа ПЛИС. Каждый фильтр реализует дискретную свёртку по формуле (1):

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n - k], \quad (1)$$

где $y[n]$ — выходной отсчёт,

$x[n - k]$ — входной сигнал с задержками,

$h[k]$ — заранее заданные коэффициенты фильтрации,

N — количество taps.

Для повышения тактовой частоты и снижения задержек была применена конвейеризация — вычисления разделены на несколько этапов[6]:

1. Запись в линию задержки,
2. Умножение на коэффициенты,
3. Суммирование произведений,
4. Формирование выходного значения.

Кроме того, реализован механизм динамической загрузки коэффициентов через AXI-Lite, что позволяет менять параметры фильтрации без перекомпиляции всей системы.[7]

На стороне персонального компьютера была разработана простая, но эффективная программа на языке Python, которая:

1. Слушает указанный UDP-порт,
2. Принимает данные от ПЛИС,
3. Сохраняет их в бинарный файл для дальнейшего анализа.

4. Программа использует стандартные модули: Socket — для сетевого взаимодействия, sys, os — для управления файлами и выводом, numpy — для численного анализа, matplotlib — для визуализации сигналов.

Для верификации системы было проведено моделирование в Icarus Verilog с использованием искусственных данных, имитирующих ЭЭГ-сигналы. [8] Данные читались из файла eeg_data.txt, где каждая строка представляла собой одно 16-битное значение.

Моделирование показало, что:

1. Тактовый сигнал (clk) стабильный, период — 10 нс (частота — 100

МГц).

2. После активации сигнала сброса (rst_n) система начинает обработку.

3. Входные данные (adc_data_0–adc_data_7) меняются поочерёдно — это говорит о правильном считывании из файла.

4. Фильтры выполняют вычисления правильно — выходные значения отличаются от входных, согласно формуле дискретной свертки, что подтверждает работу алгоритма свёртки.

5. Мультиплексор корректно объединяет данные и добавляет информацию о номере канала.

6. Ethernet-интерфейс работает надёжно, данные успешно отправляются.

Анализ временных диаграмм в GTKWave [9]. Было проанализировано несколько сигналов:

adc_data_0–adc_data_7: входные данные от АЦП.

filtered_data_0–filtered_data_7: выходы КИХ-фильтров.

multiplexed_data: объединённый поток с информацией о канале и результатом фильтрации.

eth_tdata, eth_tvalid: сигналы Ethernet-передачи.

Анализ показал, что все сигналы синхронны, фильтры работают корректно, мультиплексор последовательно выбирает каналы, сетевая передача осуществляется без ошибок.

Заключение. В ходе работы был выполнен обзор на существующие методы параллельной цифровой обработки биологических сигналов, так же выполнен обзор и сравнение интерфейсов передачи данных с ПЛИС на персональный компьютер. Была разработана блок схема устройства для параллельной цифровой обработки биологических сигналов на основе ПЛИС. Разработан интерфейс приема и передачи биологических данных, так же разработан модуль, отвечающий за цифровую обработку данных. Было разработано ПО для персонального компьютера для приема данных с ПЛИС, а

также проведено тестирование с помощью среды Icarus Verilog и визуального анализатора вол GTK Wave.

.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Оппенгейм, А. В. Цифровая обработка сигналов / А. В. Оппенгейм, Р. В. Шафер. — Москва: Техносфера, 2006.
2. Лайонс, Р. Г. Понимание цифровой обработки сигналов / Ричард Г. Лайонс. — Москва: Бином, 2009.
3. Прокис, Дж. Г. Цифровая обработка сигналов: принципы, алгоритмы, применение / Джон Г. Прокис, Димитрис Г. Манолакис. — Москва: Постмаркет, 2003.
4. IEEE Standard for Verilog Hardware Description Language / IEEE Std 1364-2005. — IEEE, 2006. — Режим доступа: <https://ieeexplore.ieee.org/document/4018236>
5. Килтц, М. Д. Цифровое проектирование на языке Verilog / М. Д. Килтц. — Санкт-Петербург: BHV, 2010. — 512 с.
6. Пархи, К. К. VLSI Digital Signal Processing Systems: Design and Implementation / Keshab K. Parhi. — Wiley, 1999.
7. Xilinx. UG937 – Vivado High-Level Synthesis [Электронный ресурс]. Режим доступа: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug937-vivado-high-level-synthesis.pdf
8. Icarus Verilog. Official Documentation [Электронный ресурс]. — Режим доступа: <http://iverilog.icarus.com>
9. GTKWave. User Manual [Электронный ресурс]. — Режим доступа: <http://gtkwave.sourceforge.net/gtkwave.pdf>



20.06.2025