

Анатолій Русланович Гайдай¹, Ірина Анатоліївна Клименко²

¹аспірант

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» (Київ, Україна)

E-mail: tolya.hei@gmail.com. ORCID: <http://orcid.org/0000-0001-9330-414X>

²доктор технічних наук, професор, професор кафедри обчислювальної техніки

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» (Київ, Україна)

E-mail: klymenko.iryana@iit.kpi.ua. ORCID: <http://orcid.org/0000-0001-5345-8806>

ВИСОКОПРОДУКТИВНА АПАРАТНО-ПРОГРАМНА СИСТЕМА ДЛЯ ГРАНИЧНОЇ ОБРОБКИ СЕНСОРНИХ ПОТОКІВ У СИСТЕМАХ ІoT НА ПЛАТФОРМІ FPGA-SoC

У статті розглянуто розроблення високопродуктивної апаратно-програмної системи багаторівневої фільтрації поточкових сенсорних даних на платформі FPGA-SoC з інтегрованим процесорним ядром ARM під управлінням операційної системи Linux. Система орієнтована на граничну обробку даних на edge-рівні системи IoT і забезпечує мінімальні та детерміновані затримки. Реалізовано апаратний фільтраційний модуль у вигляді трисступеневої конвеєрної архітектури, що поєднує медіанний фільтр, фільтр ковзного середнього та експоненційне згладжування з використанням фіксованого формату обчислень. Високопродуктивний обмін даними між вбудованим процесором ARM і логікою FPGA здійснюється через інтерфейс Avalon-MM. Експериментальні дослідження підтвердили пропускну здатність на рівні одного семплу за такт, внутрішню латентність на рівні трьох тактів, а також скорочення часу обробки на 26 % порівняно з програмною реалізацією на ARM без апаратного прискорення на FPGA. Запропонований підхід забезпечує ефективну, масштабовану та надійну обробку сенсорних потоків у режимі реального часу.

Ключові слова: FPGA; фільтрація даних; ARM; IoT.

Рис.: 2. Бібл.: 6.

Актуальність теми дослідження. Стрімке зростання кількості IoT-пристроїв призводить до збільшення обсягів поточкових сенсорних даних, які надходять у режимі реального часу та часто містять імпульсні викиди, високочастотний шум і нестабільність тренду. У промислових, медичних та розподілених системах моніторингу такі спотворення безпосередньо впливають на точність контролю, коректність виявлення аномалій і кількість хибних спрацювань. Типові програмні підходи до фільтрації сенсорних вимірювань, які реалізовані на вбудованих процесорних ядрах ARM (Advanced RISC Machine), при збільшенні обсягу та паралельності потоків призводять до зростання часу обробки, появи недетермінованих затримок і обмеженої масштабованості. Альтернативні рішення, засновані на централізованій хмарній обробці, хоча й дозволяють використовувати значні обчислювальні ресурси, додають мережеву латентність, підвищують навантаження на канал зв'язку та знижують автономність вузлів. У сучасних розподілених IoT-системах моніторингу особливої актуальності набуває задача ефективною фільтрації та очищення поточкових сенсорних даних на edge-рівні. Перенесення попередньої обробки на edge-рівень є критично важливим для зменшення латентності, зниження мережевого навантаження та підвищення автономності вузлів у разі нестабільного зв'язку з хмарою. Застосування систем на кристалі (SoC, System on a Chip), що поєднують програмовану логіку FPGA (Field-Programmable Gate Array) та вбудовані процесорні ядра ARM, дозволяє реалізувати масивний паралелізм, конвеєрну обробку та апаратну детермінованість часових характеристик. Це робить платформи FPGA-SoC перспективним і актуальним технологічним рішенням для побудови систем багаторівневої фільтрації та очищення сенсорних потоків у режимі реального часу.

Постановка проблеми. У сучасних розподілених системах моніторингу в IoT ефективна фільтрація та очищення поточкових сенсорних даних на edge-рівні є критично важливими для забезпечення коректності прийняття рішень і роботи систем у режимі реального часу. Водночас поєднання високих вимог до якості даних і жорстких часових обмежень робить цю задачу складною та недостатньо розв'язаною, особливо за умов зростання кількості сенсорних каналів і частоти надходження вимірювань.

З огляду на це існує суперечність між вимогами сучасних застосувань IoT до якості даних і часових характеристик обробки та можливостями типових підходів до фільтрації. З одного боку, сенсорні потоки потребують одночасного пригнічення різних типів завад (імпульсних викидів, випадкових флуктуацій, дрейфу тренду) із мінімальною затримкою та без втрати динаміки корисного сигналу. З іншого боку, програмна реалізація багаторівневої фільтрації на процесорі ARM при зростанні навантаження (частоти семплів, кількості каналів, паралельних потоків) призводить до збільшення часу обробки, недетермінованих затримок і обмеженої масштабованості, а централізована хмарна обробка додає мережеву латентність і залежність від каналу зв'язку.

Таким чином, науково-практична проблема полягає в розробленні та реалізації високопродуктивного апаратно-програмного підходу до багаторівневої фільтрації поточкових сенсорних даних на платформі FPGA-SoC, який забезпечує детерміновані часові характеристики обробки, малу та передбачувану затримку, високу пропускну здатність і масштабованість для обробки мультиканальних сенсорних потоків у реальному часі за умов обмежених обчислювальних та енергетичних ресурсів edge-вузлів.

Розв'язання цієї проблеми дозволить забезпечити ефективне очищення та стабілізацію поточкових сенсорних даних безпосередньо на edge-рівні, підвищити їхню достовірність і придатність для подальшого аналізу, зменшити обсяг передаваних до мережі та хмарних сервісів даних, а також підвищити ефективність функціонування систем реального часу у промислових, медичних та розподілених сценаріях IoT.

Аналіз останніх досліджень і публікацій. У сучасних системах IoT спостерігається стрімке зростання обсягів даних, які надходять від сенсорів у режимі реального часу та характеризуються нерівномірністю, шумом і надлишковістю. У таких умовах традиційні підходи, що базуються на центральній обробці у хмарі або застосуванні мікроконтролерів (MCU, Microcontroller Unit), часто не забезпечують необхідної швидкодії, затримки та масштабованості. У цьому контексті FPGA набувають особливого значення завдяки своїй здатності виконувати масивний паралелізм, конвеєрну обробку та апаратну реконфігурацію. Розглянуті наукові роботи демонструють широкий спектр застосувань FPGA у IoT-середовищах, підкреслюючи їхню ефективність у реалізації фільтраційних алгоритмів та обробці поточкових даних безпосередньо на edge-рівні.

У статті [1] наголошується, що однією з ключових причин використання FPGA в IoT є необхідність локальної обробки нестабільних та зашумлених сенсорних даних. Автори підкреслюють, що завдяки апаратній природі FPGA можливо реалізовувати конвеєрні архітектури для цифрової обробки сигналів – зокрема ковзні вікна, нелінійні та лінійні фільтри, методи згладжування та попередні перетворення.

Однією з центральних тез роботи є порівняльний аналіз швидкодії: FPGA забезпечують частоти обробки, що сягають сотень мегагерц, і здатні працювати одночасно з кількома потоками даних. Це значно перевершує можливості традиційних MCU та одноплатних комп'ютерів, де фільтрація виконується послідовно.

Особливу увагу автори приділяють SoC, таким як Cyclone V чи Xilinx Zynq, що поєднують процесор ARM та логіку FPGA. Такий підхід дозволяє розділяти задачі:

- апаратна логіка виконує важку фільтрацію та попередню обробку;
- вбудована операційна система (ОС) Linux керує мережею, адаптує параметри фільтрів та виконує логічний контроль.

Це формує основу для багаторівневої, гнучкої та адаптивної системи обробки даних у реальному часі.

У статті [2] акцент зроблено на питаннях енергоспоживання - одному з найважливіших аспектів пристроїв IoT. Автори демонструють, що поєднання *soft-core* процесорів (CPU), на кшталт Nios II, MicroBlaze, з апаратними блоками дозволяє гнучко управляти режимами роботи FPGA.

Серед ключових висновків авторів:

- динамічна зміна тактової частоти, часткове вимкнення логіки та перерозподіл потоків дозволяє зменшити енергоспоживання без втрати точності;
- soft-core процесори працюють під керуванням RTOS та можуть змінювати параметри фільтрів «на льоту», реагуючи на зміни вхідного потоку або енергетичні обмеження;
- експерименти авторів показують значну економію енергії при мінімальних витратах продуктивності.

Ця робота демонструє синергію між програмною та апаратною частиною: фільтри реалізуються у ПЛІС для досягнення максимальної швидкодії, тоді як soft-core CPU відповідає за контроль та адаптацію режимів роботи у відповідь на енергетичні умови. Це робить статтю важливою у контексті побудови енергоефективних фільтраційних рішень.

У статті [3] розглянуто фундаментальні засади побудови сучасних IoT-систем, де значна частина обчислень переноситься на периферію мережі. Автори підкреслюють, що централізована хмарна обробка не здатна забезпечити вимоги сучасних застосувань – зокрема низьку затримку, локальну аналітику та автономність.

Серед ключових положень автори виділяють:

- більшість сенсорних даних супроводжуються шумом та містять надлишкові вимірювання, тому фільтрація має виконуватися на edge-рівні;
- edge-вузли повинні забезпечувати обробку в реальному часі, мінімальну латентність і безперервність роботи у разі втрати хмарного зв'язку з хмарним середовищем;
- FPGA розглядаються, як оптимальні платформи для виконання фільтрації, класифікації, стискання та попередньої аналітики завдяки низькій затримці та високому паралелізму.

Багато актуальних наукових праць присвячено дослідженню моделей взаємодії між сенсорами, шлюзами та хмарною інфраструктурою, методам розподіленої аналітики та енергетичної оптимізації. У цих роботах показано, що FPGA є особливо ефективними на edge-рівні завдяки низькій затримці та підтримці паралельної обробки поточкових даних у реальному часі.

У статті [4] наведено широкий огляд сучасних напрямів використання FPGA у різних сферах IoT. Автори відзначають, що FPGA перетворилися з суто інструментів DSP на критичний компонент edge- та fog-обчислень.

У роботі виділено такі ключові переваги FPGA:

- можливість створення масивно паралельних фільтрів та адаптивних обчислювальних блоків;
- здатність працювати при високій частоті надходження даних;
- висока стійкість та продуктивність при роботі з багатьма потоками одночасно.

Також автори в роботі [4] аналізують платформи SoC, де процесор ARM відповідає за мережу й конфігурацію параметрів, а FPGA – за обчислювально інтенсивні операції: фільтрацію, агрегацію, виявлення аномалій, криптографію, обробку відео та радіосигналів. У статті відзначаються й виклики: складність розробки коду HDL, потреба в спеціалізованих інструментах та відсутність стандартизованих модулів для IoT. Попри це, автори роблять висновок, що FPGA стають стандартною платформою для фільтрації в системах IoT та обробки даних у реальному часі.

У статті [5] наведено приклад використання FPGA у високоточних медичних вузлах в системах IoT. Основна увага приділена задачам попереднього скринінгу та прогнозування за поточковими медичними даними, що часто містять шум й потребують локальної обробки.

Серед основних висновків дослідження можна виділити такі:

- FPGA забезпечують фільтрацію, нормалізацію та кореляційний аналіз сигналів (серцевий ритм, біометрія тощо) у реальному часі;

- апаратна реалізація ковзних фільтрів та алгоритмів виявлення змін зменшує затримку, що критично для медичних застосувань;
- застосовано різні методи енергетичної оптимізації: частотне масштабування, відключення блоків, розподіл обчислень між CPU та FPGA.

Проведений огляд літератури свідчить, що використання FPGA для реалізації попередньої обробки даних на граничному рівні інфраструктури IoT дає змогу не лише зменшити енергоспоживання порівняно з традиційними SoC, а й забезпечити паралельну обробку даних, зменшення затримки та прискорення виконання обчислень.

Виділення недосліджених частин загальної проблеми. Аналіз наявних досліджень свідчить, що, попри значну кількість робіт, присвячених застосуванню FPGA в системах IoT, недостатньо уваги приділено практичній реалізації засобів фільтрації та очищення поточкових сенсорних даних на платформах FPGA-SoC. У таких системах, окрім задач пришвидшення обробки, постають питання ефективного обміну даними між процесорним ядром та програмованою логікою FPGA. Зокрема, залишаються недостатньо дослідженими питання побудови апаратних конвеєрів цифрової фільтрації з детермінованою затримкою, що є критично важливим для систем реального часу. Особливої уваги заслуговує обробка даних на FPGA у фіксованому форматі, який, на відміну від плаваючої коми, дозволяє підвищити продуктивність обробки сенсорних вимірювань, що за природою є дробовими, шляхом спрощення апаратних операцій. Крім того, обмежено висвітлені аспекти організації ефективної програмно-апаратної взаємодії між вбудованими процесорними ядрами та логікою FPGA для потокової обробки даних у реальному часі. Це стосується керування станами готовності, синхронізації обчислювальних модулів, а також масштабування рішень для багатоканальних сенсорних потоків. Зазначені обмеження обумовлюють необхідність подальших досліджень, спрямованих на розробку універсальних, масштабованих та енергоефективних edge-рішень для очищення й стабілізації сенсорних даних у сучасних системах IoT.

Мета дослідження полягає у розробленні та реалізації високопродуктивної апаратно-програмної системи на платформі FPGA-SoC з інтеграцією вбудованого процесора ARM під управлінням Linux для багаторівневої фільтрації сенсорних вимірювань на edge-рівні IoT-систем, яка забезпечує паралельну обробку потоків та мінімальні затримки. Досягнення мети передбачає підвищення продуктивності локальної обробки, зменшення навантаження на мережу та хмарні ресурси, а також масштабованість системи для обробки мультिकанальних потоків.

Для досягнення мети необхідно розв'язати наступні завдання:

1. Розробити апаратну реалізацію трирівневого фільтраційного модуля на FPGA, інтегруючи медіанний фільтр, фільтр ковзного середнього та метод експоненційного згладжування, що забезпечує паралельну та прискорену обробку поточкових даних.

2. Розробити програмно-апаратний механізм взаємодії вбудованого процесорного ядра ARM та логіки FPGA через інтерфейс Avalon-MM (Avalon Memory Mapped Interface) для високопродуктивного обміну сенсорними потоками у режимі реального часу, що дозволяє зменшити затримки, витрати логічних ресурсів та масштабування системи для мультिकанальної обробки даних.

Виклад основного матеріалу. Гіпотезою дослідження є припущення, що апаратна реалізація системи для трирівневої фільтрації сенсорних потоків на FPGA, яка поєднує медіанний фільтр, фільтр ковзного середнього та метод експоненціального згладжування, забезпечить не лише ефективніше зменшення рівня шумів і підвищення стабільності поточкових даних, а й суттєве прискорення їх локальної обробки порівняно з програмною реалізацією на вбудованих процесорах ARM архітектури (наприклад, Cortex-A у SoC, Raspberry Pi та аналогічних платформах). Це досягається завдяки комбінації конвеєрного паралелізму та паралелізму на рівні обробки даних, а також детермінованій пропускну здатності апаратної логіки FPGA.

Для підтвердження гіпотези виконано експериментальне дослідження на платформі FPGA-SoC, що поєднує логіку FPGA та вбудований ARM-процесор. Гіпотеза передбачає також розробку високопродуктивного інтерфейсу обміну даними між конвеєром на FPGA та процесором ARM, ціллю якого є мінімальні затримки обміну даними та висока пропускна здатність. Експериментальна перевірка включає розробку та тестування апаратно-програмної системи для граничної обробки сенсорних потоків, що дозволяє оцінити ефективність апаратної фільтрації в умовах реального часу та підтвердити доцільність її використання в IoT-системах.

Дослідження організовано у вигляді послідовності взаємопов'язаних етапів, які забезпечують комплексну оцінку ефективності апаратної реалізації фільтрації:

1. Проектування апаратної частини, що передбачає реалізацію фільтраційного модуля на FPGA у вигляді триступеневого конвеєра з фіксованим форматом обчислень, а також виконання синтезу, компіляції та оптимізації HDL-модулів з метою забезпечення стабільної роботи системи на частотах, достатніх для обробки даних у режимі реального часу.

2. Розроблення та реалізація програмно-апаратного механізму обміну поточковими даними між вбудованим ARM-процесором Cortex-A9 та логікою FPGA на платформі DE1-SoC з використанням шини Avalon-MM і механізмів ядра ОС Linux, що забезпечує ініціалізацію апаратних модулів, передавання сенсорних потоків і зчитування результатів фільтрації у режимі реального часу.

3. Інтеграція апаратних модулів фільтрації з програмною підсистемою керування, що дозволяє реалізувати безперервну обробку поточкових даних у режимі реального часу.

4. Проведення експериментальних досліджень, спрямованих на порівняльну оцінку апаратної та програмної реалізації фільтрації за показниками рівня шумів, стабільності сигналів, затримки обробки та пропускної здатності системи.

Триступеневий фільтраційний конвеєр реалізує на FPGA метод багаторівневої фільтрації сенсорних вимірювань, який детально описано в роботі [6]. Трирівневий фільтр складається із *медіанного фільтра*, що усуває імпульсні та одиничні аномальні значення; фільтру *ковзного середнього*, що згладжує короточасні коливання та нестабільності та застосування методу *експоненціального згладжування* для стабілізації потоку сенсорних даних та формування трендів для адаптації до динаміки зміни сигналів в динамічних середовищах.

Очікувана наукова новизна дослідження полягає в демонстрації ефективності апаратної реалізації багаторівневої фільтрації сенсорних потоків на платформі FPGA-SoC, що поєднує вбудоване процесорне ядро ARM Cortex-A9 та програмну логіку FPGA. Використання внутрішньої шини Avalon-MM і реалізація обміну даними на рівні модуля ядра ОС Linux дозволяє досягти низької затримки та високої пропускної здатності локальної обробки потоків, а конвеєрна апаратна обробка фільтрів забезпечує паралельність і детермінованість обчислень. Це створює умови для підвищення продуктивності системи моніторингу на edge-рівні, зменшення шумів і стабілізації поточкових даних порівняно з програмною реалізацією на ARM.

Для реалізації трирівневого фільтраційного модуля для обробки сенсорних потоків на edge-рівні в дослідженні використовувалася платформа DE1-SoC (5CSEMA5F31C6), що поєднує двоядерний вбудований процесор ARM Cortex-A9 і логічну матрицю FPGA Cyclone V.

Для підвищення продуктивності обробки даних на FPGA були використані вбудовані апаратні блоки: PLL для генерації стабільних тактових сигналів, вбудовані помножувачі та блоки DSP для швидкої реалізації обчислювально інтенсивних алгоритмів фільтрації, а також модулі FIFO для буферизації даних у поточкових процесах.

Програмна частина була реалізована на вбудованому процесорному ядрі ARM Cortex-A9 під керуванням ОС Linux. Основними функціями програмного забезпечення є ініціалізація FPGA-модулів, передавання сенсорних потоків і зчитування результатів фільтрації в режимі реального часу. Програмне забезпечення реалізовано у вигляді модулів ядра, що дозволяє обійти користувацький простір і знизити латентність. Обмін даними

між процесором і FPGA організовано через внутрішню шину Avalon-MM, що забезпечує низьку затримку та високу пропускну здатність локальної обробки потоків даних.

Для реалізації та тестування системи були використані такі програмні інструменти:

- Intel Quartus Web – синтез HDL та конфігурація FPGA;
- Platform Designer (Qsys) – побудова та налаштування взаємодії між ARM-процесором і логікою FPGA;
- ModelSim – моделювання та верифікація апаратних фільтрів;
- Verilog – мова опису апаратури для проектування та верифікації апаратних модулів;
- ОС Linux на ARM – виконання програм керування та обміну даними;
- GCC та мова C – реалізація тестових програм для перевірки роботи фільтраційного конвеєра.

Для експериментів були використані сенсорні дані з різними видами відхилень, що включали імпульсні піки, різкі стрибки та шумні потокові дані, які імітують реальні умови роботи сенсорної мережі.

Програмний інтерфейс та обробка даних

Метою цього етапу є реалізація ефективного механізму обміну даними між вбудованим процесором ARM Cortex-A9 та апаратною логікою FPGA на платформі SoC під керуванням ОС Linux.

Вхідні дані: вхідні пакети даних що надходять по мережі Ethernet

Вихідні дані: вихідний файл CSV, який містить послідовний номер, сире значення сигналу та відфільтроване значення.

Крок 1: Ініціалізація та налаштування файлів.

1.1. Перевірка наявності необхідних аргументів командного рядка.

1.2. Відкриття вихідного файлу для запису. Запис заголовка CSV: "n, raw, filt\n".

Крок 2: Відображення апаратних регістрів у пам'ять.

2.1. Відкриття спеціального пристрою /dev/mem із правами на читання та запис.

2.2. Виклик функції *mmap* для відображення фізичної адреси бази (LW_BRIDGE_BASE) у віртуальний адресний простір процесу.

2.3. Обчислення віртуальної базової адреси апаратного фільтра (regs) шляхом додавання зміщення фільтра (FILTER_OFFSET) до відображеної бази мосту.

Крок 3: Цикл обробки даних.

3.1. Читання вхідних даних.

3.2. Для кожного рядка:

i) пропуск початкових пробілів, порожніх рядків та коментарів та виділення підрядка, що містить значення температури.

ii) перетворення підрядка зі значенням температури у формат з плаваючою комою (double val).

iii) перетворення отриманого значення у фіксований формат – конвертація значення val у 32-бітне ціле число зі знаком (int32_t fixed), використовуючи фіксований формат Q(30.2) (шляхом множення на 100.0 та округлення).

Крок 4: Взаємодія з апаратним прискорювачем.

4.1. Запис: Записати значення fixed у регістр входу апаратного фільтра (REG_IN). Ця дія запускає обробку на FPGA.

4.2. Синхронізація (очікування готовності): Увійти у цикл опитування (polling), постійно зчитуючи біт готовності (біт 0) з регістру статусу фільтра (REG_STATUS). Очікувати, доки біт готовності не дорівнюватиме 1.

4.3. Зчитування: Зчитати 32-бітний результат фільтрації (int32_t out_fixed) із регістру виходу (REG_OUT).

Крок 5: Деконверсія та запис виходу.

5.1. Перетворення у плаваючу кому: Перетворити `out_fixed` назад у формат із плаваючою комою (`double out_val`) шляхом ділення на 100.0.

5.2. Запис послідовного номера вибірки (n), сирого значення (val) та відфільтрованого значення (`out_val`) у вихідний файл CSV.

Крок 6: Очищення ресурсів.

6.1. Звільнення відображеної пам'яті (`munmap`).

6.2. Закриття дескриптору пристрою `/dev/mem` та файлових потоків.

Проектування та реалізація конвеєрної архітектури трирівневого фільтраційного модуля на FPGA. Апаратний модуль `filter` реалізує триступеневий послідовний конвеєр для цифрової фільтрації, що використовує протокол *Avalon-MM Slave* для обміну даними з хост-процесором. Фільтр працює з 32-бітними цілими числами зі знаком (фіксований формат обчислень).

Розміри вікон та коефіцієнт α були підібрані у ході проведення серії експериментів, які детально описані авторами в роботі [6]. Вони краще зберігають загальний тренд та відображають плавні зміни середовища за короткий час. Менші значення параметрів призводили до недостатнього згладжування, тоді як більші — до надмірної інерційності та втрати оперативності реакції.

Архітектура та інтерфейс

Інтерфейс: *Avalon-MM Slave* (3-бітний адресний простір).

Регістри:

`REG_STATUS` – реєстр стану; біт 0 – *ready* (результат готовий); біт 1 – *busy* (обробка триває).

`REG_IN` – реєстр для запису вхідного семпла (запускає конвеєр).

`REG_OUT` – реєстр для зчитування відфільтрованого семпла.

Конвеєр: Обробка здійснюється за кінцевим автоматом (*FSM*) із чотирма станами (`S_IDLE`, `S_MED`, `S_MA`, `S_EXP`), де кожен етап фільтрації займає один тактовий цикл, забезпечуючи конвеєрну обробку з латентністю три такти.

Етапи конвеєрної фільтрації

Конвеєр обробляє дані послідовно через три незалежні каскади, реалізовані у режимі *Register-Register*.

Етап 1: Медіанний фільтр (*Stage 1: Median Filter*)

Вхідний семпл записується у кільцевий буфер (`med_buf`) розміром 5.

Використовується комбінаційний блок сортування (на основі бульбашкового сортування) що на кожному такті зчитує вміст буфера та обчислює медіану (середній елемент).

Результат медіани (`median_comb`) фіксується у реєстрі `stage1_out` на наступному такті.

Етап 2: Фільтр ковзного середнього (*Stage 2: Moving Average, MA*)

Вхідний семпл записується у кільцевий буфер (`ma_buf`) розміром 4.

Реалізований як ефективний рекурсивний фільтр, який уникає перерахунку всієї суми.

На кожному такті оновлюється загальна сума (S) у 48-бітному акумуляторі за формулою:

$$S_{new} = S_{old} + x[n] - x_{old}, \quad (1)$$

де $x[n]$ – це `stage1_out`, а x_{old} – найстаріший елемент з буфера `ma_buf`.

Ділення на 4 (усереднення) виконується швидкою бітовою операцією логічного зсуву вправо на 2 біти.

Результат фіксується у реєстрі `stage2_out`.

Етап 3: Експоненційне згладжування (*Stage 3: Exponential Smoothing*)

Параметри: Коефіцієнт згладжування $\alpha = 0.4$.

Реалізує фільтр першого порядку за формулою:

$$y[n] = \frac{1}{\alpha_{den}} * (\alpha_{num} * x[n] + (1 - \alpha_{num}) * y[n - 1]), \quad (2)$$

де α_{num} – фіксовані чисельник коефіцієнта згладжування, α_{den} – фіксований знаменник коефіцієнта згладжування, $x[n]$ – це результати роботи другого фільтра, а $y[n - 1]$ – попередній результат загальної фільтрації.

Множення виконується з розширеною 48-бітною точністю для запобігання переповненню проміжних результатів.

Кінцевий результат після ділення на 10 фіксується як `stage3_out` та `data_out_reg`, після чого встановлюється прапор `ready_out` і кінцевий автомат переходить у стан `S_IDLE`.

Керування кінцевим автоматом (*FSM*):

1. початковий стан `S_IDLE` переходить у `S_MED`, після надходження нового значення даних в регістр, `busy = 1`, `ready_out = 0`;

2. стан `S_MED` переходить у `S_MA` після обчислення медіани;

3. `S_MA` переходить в `S_EXP` після обчислення ковзкого середнього;

4. `S_EXP` повертається в початковий стан, `S_IDLE`, після обчислення експоненціального згладжування.

Методологія експериментального дослідження. Для перевірки гіпотези дослідження було проведено серію експериментів, спрямованих на оцінку ефективності трирівневого фільтруючого конвеєра, реалізованого на FPGA платформи DE1-SoC. Основна мета експериментів полягала у визначенні здатності апаратної системи обробляти сенсорні потоки в режимі реального часу, а також у порівнянні ефективності апаратної фільтрації з програмною реалізацією на ARM Cortex-A9 за показниками рівня шумів, стабільності сигналів і затримки обробки.

Взаємодія з процесором організована через інтерфейс Avalon-MM, що забезпечує можливість подачі вхідних даних та зчитування результатів через регістри. Архітектура складається з трьох послідовних етапів: п'ятиточкового медіанного фільтра, фільтра ковзного середнього з вікном 4 та експоненційного згладжування з коефіцієнтом $\alpha = 0.4$. Така комбінація дозволяє поетапно зменшувати різні типи шумів: імпульсні перешкоди, високочастотні флуктуації та нестабільність тренду.

Функціональні блоки об'єднані в конвеєр із внутрішніми регістрами між стадіями, які забезпечують фіксовану затримку та передавання сигналів `valid` для синхронізації обробки. Передача вхідного сигналу здійснюється шляхом запису в реєстр Avalon-MM, після чого дані проходять усі три рівні фільтрації без зупинок. Вихідне значення доступне в окремому регістрі, а стан конвеєра контролюється через статусні сигнали `ready_out` та `busy`, що дозволяє відстежувати готовність результату у реальному часі.

Обчислення виконуються у фіксованому форматі обчислень для забезпечення апаратної ефективності. Конвеєрна організація дає змогу досягти високої пропускну здатності: після подачі першого значення вихід стабілізується через п'ять тактів (за шириною першого вікна фільтрації), а надалі кожен такт формує новий оброблений семпл. У результаті внутрішня затримка між стадіями становить лише одну тактову подію, що забезпечує стабільну швидкість обробки 1 семпл за такт, на порядок випереджаючи послідовні реалізації на процесорі.

Перед безпосередньою апаратною реалізацією на платі DE1-SoC було виконано моделювання роботи багаторівневого фільтруючого конвеєра у середовищі ModelSim. Це дозволило перевірити коректність функціонування апаратних модулів (медіанного фільтра, фільтру ковзного середнього та методу експоненційного згладжування) на рівні часових сигналів та оцінити їхню здатність поетапно зменшувати різні типи шумів у поточкових даних. Результати симуляції представлені у вигляді часових діаграм на рис. 1.

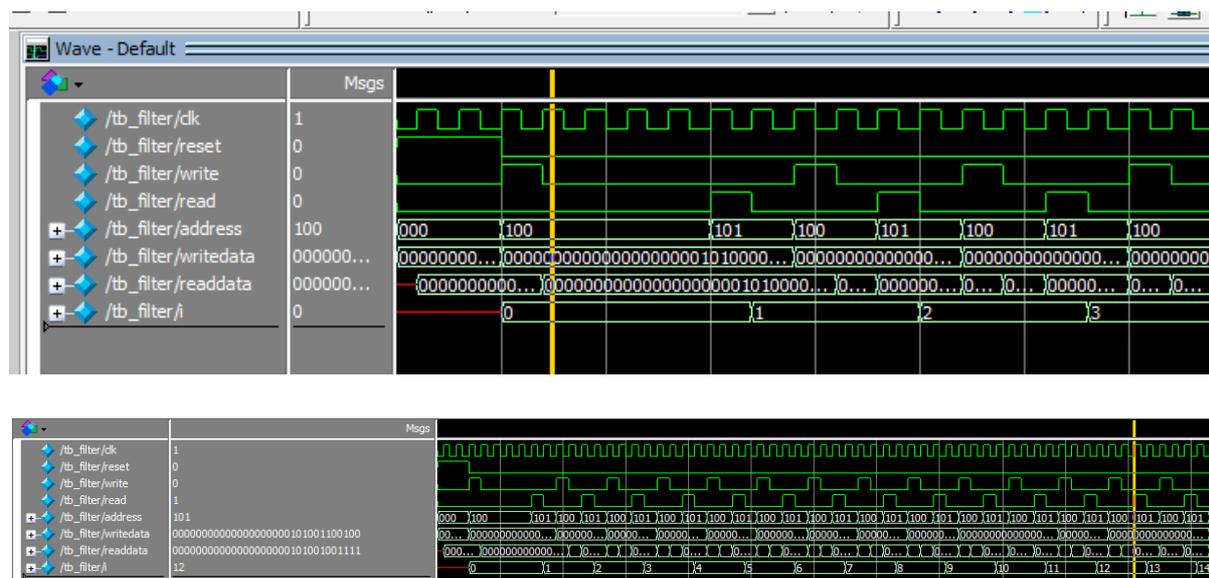


Рис. 1. Часові діаграми симуляції роботи в середовищі ModelSim

Під час верифікації апаратних модулів у середовищі ModelSim перевірялися часові діаграми valid-сигналів та передавання даних між блоками, що підтвердило відсутність втрати семплів і правильну послідовність обробки.

Розроблена модель була синтезована та протестована на апаратній платформі DE1-SoC. Тестування здійснювалося на реальних температурних даних, які дозволили перевірити роботу алгоритмів згладжування на зашумлених даних. Початкові температурні вимірювання були зібрані протягом тривалого часу з реального датчика, що характеризується підвищеним рівнем шуму. Для демонстрації переваг високошвидкісної обробки даних на FPGA (1 семпл за такт) дані вимірювань були перетворені у синтетичний потік, який подавався на FPGA-платформу, без урахування фізичної інерційності температурного процесу. Такий підхід дозволив абстрагуватися від особливостей конкретного фізичного сигналу та зосередитися на аналізі продуктивності апаратної реалізації алгоритму. У цьому контексті температурний сигнал використано як репрезентативний приклад реальних шумових даних.

Отримані експериментальні результати підтверджують ефективність побудованого фільтра. Початковий сигнал містив значні короткочасні стрибки, характерні для фізичних сенсорів, що чітко спостерігалось на ділянках приблизно 10–20 та 40–50 семплів. Після проходження медіанного фільтра імпульсні шуми було усунуто без спотворення трендів. Ковзне середнє додатково приглушило залишкові випадкові коливання, сформувавши плавні переходи. Фінальний блок експоненційного згладжування показав стабільну зміну значень з мінімальним рівнем шумів, що дозволяє зробити сигнал придатним для точних систем керування.

Порівняння з послідовною реалізацією на процесорі показало, що пропускна здатність збільшилась завдяки безперервній роботі конвеєра: система не очікує завершення попередньої обробки для прийому нового семплу. Загальний ефект – значне зменшення шумів та формування плавного, достовірного тренду вихідного сигналу, підтверджене як симуляцією, так і тестами на реальних даних.

На рис. 2 представлені результати обробки температурних даних за допомогою трирівневого фільтра.

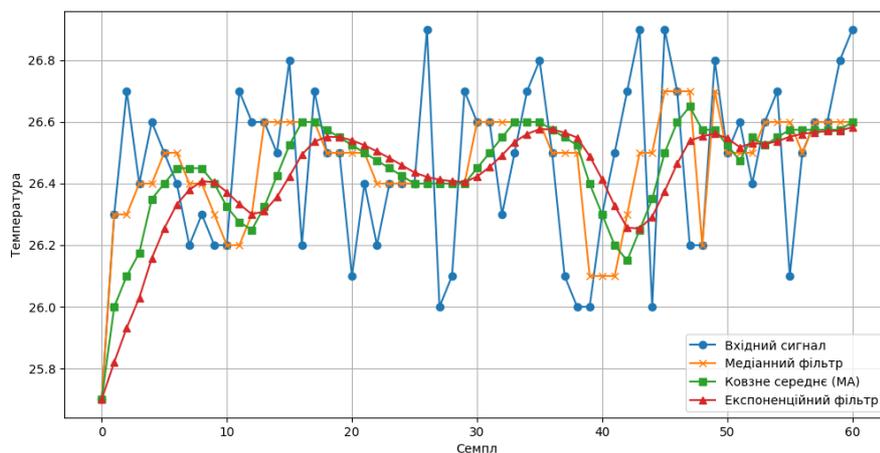


Рис 2. Графіки показників температури під час різних етапів фільтрації

Синя лінія відповідає вхідному сигналу, що складається з реальних вимірювань температури, поданих у модуль фільтрації по одному семплу. На графіку видно, що вхідний сигнал містить численні стрибки (наприклад, на 10–20 та 40–50 семплах), які не відображають реальних змін температури. Помаранчева лінія демонструє результати роботи медіанного фільтра, який усуває раптові імпульси, зберігаючи загальний тренд сигналу. Зелена лінія відображає результат застосування фільтра ковзного середнього, що згладжує залишкові коливання і надає сигналу більш плавний характер. Нарешті, червона лінія показує вихід трирівневого фільтра після експоненційного згладжування, підкреслюючи плавність тренду та забезпечуючи можливість прогнозування подальших значень сигналу.

Аналіз результатів дослідження. Реалізація багаторівневої системи фільтрації сенсорних даних платформі FPGA-SoC, що поєднує вбудоване процесорне ядро архітектури ARM Cortex-A9 та програмовну логіку FPGA, дозволяє ефективно розподілити обчислювальні функції між апаратною та програмною частинами системи. FPGA виконує обчислювально інтенсивні операції фільтрації у вигляді паралельної конвеєрної обробки, тоді як вбудоване програмне забезпечення на ARM-процесорі під керуванням ОС Linux забезпечує керування процесом обробки, ініціалізацію апаратних модулів, а також збір і подальшу передачу оброблених даних до системи моніторингу.

Обмін даними між процесором ARM і логікою FPGA здійснюється через внутрішню шину Avalon, що функціонує всередині SoC та не потребує використання зовнішніх інтерфейсів передавання даних (SPI, Ethernet, USB). Це дозволяє мінімізувати накладні витрати на обмін інформацією та забезпечити підвищену пропускну здатність і низьку затримку локального передавання потокових даних.

Реалізація механізмів обміну на рівні модуля ядра ОС Linux додатково зменшує затримки, пов'язані з переходами між просторами користувача та ядра, і надає прямий доступ до регістрів з відображенням у пам'ять та системи обробки апаратних переривань. У сукупності це забезпечує більш детермінований та високопродуктивний обмін даними порівняно з звичайними реалізаціями, що дозволяє розглядати запропоновану архітектуру як орієнтовану на обробку потокових даних у режимі, близькому до реального часу.

Таким чином, поєднання внутрішньої шини Avalon, вбудованого ARM Cortex-A9 та апаратної обробки на FPGA створює умови для реалізації високопродуктивного механізму локальної фільтрації сенсорних потоків на edge-рівні, що забезпечує підвищення продуктивності, зменшення затримок та збереження гнучкості системи.

Попереднє масштабування даних у фіксованому форматі обчислень дозволило знизити витрати апаратних ресурсів та забезпечити стабільність обробки без необхідності використання чисел з плаваючою комою. Експериментальні результати підтвердили, що конвеєрна організація модуля багаторівневої фільтрації забезпечує отримання одного об-

робленого значення на кожному такті після заповнення конвеєра, що значно зменшує загальну затримку системи. Для трьох послідовних етапів фільтрації внутрішня латентність становить лише три такти, а пропускна здатність досягає 1 семпл за такт, що є суттєвою перевагою порівняно з програмною реалізацією на ARM Cortex-A9.

Для програмної реалізації без апаратного прискорення на процесорі ARM Cortex-A9 використана мова програмування C безпосередньо на платформі FPGA-SoC, з використанням компілятора GCC (GNU Compiler Collection) з третім рівнем оптимізації, що забезпечує підвищену швидкість виконання програм. Порівняння програмної реалізації без апаратного прискорення та апаратної реалізації показало, що використання FPGA дозволяє скоротити час обробки на 26%. Отриманий результат є особливо важливим для систем, де частота надходження даних може зростати, а затримка сигналу має залишатися мінімальною. Крім того, експериментально підтверджено покращення якості відфільтрованого сигналу: медіанний фільтр усуває імпульсні сплески, фільтр ковзного середнього згладжує високочастотні коливання, а експоненційне згладжування забезпечує стабільний тренд без втрати реальної динаміки сигналу.

Результати верифікації в середовищі ModelSim підтвердили коректність часових залежностей та безперервність потоків даних на всіх етапах конвеєра. Тестування на апаратному прототипі DE1-SoC засвідчило надійну роботу системи в реальних умовах і продемонструвало її потенціал для застосування в системах автоматизації, моніторингу та інших IoT-рішеннях, де критично важлива швидка та достовірна попередня обробка сенсорних сигналів на edge-рівні.

Висновки.

1. У роботі розроблено апаратний трирівневий фільтраційний модуль для реалізації системи обробки сенсорних даних у режимі реального часу, що відрізняється від відомих рішень поєднанням медіанного фільтра, ковзного середнього та експоненційного згладжування в єдиному конвеєрі. Така архітектура забезпечує паралельну та прискорену обробку поточкових даних, що підтверджено зменшенням затримки на 26% порівняно з програмною реалізацією на процесорі ARM.

2. Розроблений програмно-апаратний механізм взаємодії процесора ARM та логіки FPGA на SoC через інтерфейс Avalon-MM із використанням фіксованого формату обчислень, що забезпечує низьколатентний та високопродуктивний обмін даними, мінімальні витрати логічних ресурсів FPGA і можливість масштабування системи для обробки мультисканальних потоків без значного збільшення апаратної складності.

3. Експериментальні дослідження підтвердили покращення якості відфільтрованого сигналу: усунення імпульсних шумів, згладжування високочастотних коливань та стабілізацію тренду. Результати демонструють підвищення надійності та точності локальної обробки сенсорних потоків, що робить створений модуль придатним для застосування в розподілених системах моніторингу, промислових IoT-рішеннях та інших реальних задачах edge-обробки даних.

Отримані результати підтверджують наукову новизну роботи: розроблена методика поєднання трирівневої апаратної фільтрації з високошвидкісним інтерфейсом обміну даними на Avalon-MM дозволяє реалізувати детерміновану, масштабовану та ефективну обробку сенсорних потоків на edge-рівні, що суттєво перевищує продуктивність традиційних програмних рішень на вбудованих процесорах.

Заява про використання генеративного ШІ та технологій на основі ШІ в процесі написання тексту статті

Під час написання цього матеріалу автори використовували модель генеративного штучного інтелекту ChatGPT для пошуку літератури яка відповідає тематиці дослідження та оформлення списку використаних джерел. Після використання ChatGPT автори переглянули та відредагували зміст за потреби і взяв на себе повну відповідальність за зміст публікації.

Список використаних джерел

1. Kulanov, V., & Perepelitsyn, A. (2024). Analysis of application of FPGA technologies in IoT. *Aerospace Technic and Technology*, (5), 85–94. <https://doi.org/10.32620/aktt.2024.5.09>.
2. Hryshchuk, O., & Zagorodnyuk, S. (2025). Managing energy consumption in FPGA-based edge computing systems with soft-core CPUs. *Journal of Edge Computing*, 4(1), 57–72. <https://doi.org/10.55056/jec.717>.
3. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., et al. (2023). Edge-computing-driven Internet of Things: A survey. *ACM Computing Surveys*, 55(8), Article 174, 1–41. <https://doi.org/10.1145/3555308>.
4. Magyari, A., & Chen, Y. (2022). Review of state-of-the-art FPGA applications in IoT networks. *Sensors*, 22, Article 7496. <https://doi.org/10.3390/s22197496>.
5. Satpathy, S., Trivedi, M. C., Malhotra, S., Chaudhary, S., & Sharma, V. (2024). FPGA-enhanced IoT methods for disease pre-screening and prediction: An energy optimization approach. *Measurement: Sensors*, 33, Article 101213. <https://doi.org/10.1016/j.measen.2024.101213>.
6. Haidai, A., & Klymenko, I. (2025). Methodology of adaptive data processing in IoT monitoring systems with multilevel sensor data filtering and self-tuning. *Information, Computing and Intelligent Systems*, (7), 110–126. <https://doi.org/10.20535/2786-8729.7.2025.341409>.

References

1. Kulanov, V., & Perepelitsyn, A. (2024). Analysis of application of FPGA technologies in IoT. *Aerospace Technic and Technology*, (5), 85–94. <https://doi.org/10.32620/aktt.2024.5.09>.
2. Hryshchuk, O., & Zagorodnyuk, S. (2025). Managing energy consumption in FPGA-based edge computing systems with soft-core CPUs. *Journal of Edge Computing*, 4(1), 57–72. <https://doi.org/10.55056/jec.717>.
3. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., et al. (2023). Edge-computing-driven Internet of Things: A survey. *ACM Computing Surveys*, 55(8), Article 174, 1–41. <https://doi.org/10.1145/3555308>.
4. Magyari, A., & Chen, Y. (2022). Review of state-of-the-art FPGA applications in IoT networks. *Sensors*, 22, Article 7496. <https://doi.org/10.3390/s22197496>.
5. Satpathy, S., Trivedi, M. C., Malhotra, S., Chaudhary, S., & Sharma, V. (2024). FPGA-enhanced IoT methods for disease pre-screening and prediction: An energy optimization approach. *Measurement: Sensors*, 33, Article 101213. <https://doi.org/10.1016/j.measen.2024.101213>.
6. Haidai, A., & Klymenko, I. (2025). Methodology of adaptive data processing in IoT monitoring systems with multilevel sensor data filtering and self-tuning. *Information, Computing and Intelligent Systems*, (7), 110–126. <https://doi.org/10.20535/2786-8729.7.2025.341409>.

Дата першого надходження статті до видання: 01.12.2025
Дата прийняття статті до друку після рецензування: 19.12.2025

UDC 004.6:004.75:004.382

Anatolii Haidai¹, Iryna Klymenko²

¹PhD student

National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» (Kyiv, Ukraine)
E-mail: tolya.hei@gmail.com. ORCID: <http://orcid.org/0000-0001-9330-414X>

²Doctor of Technical Sciences, Professor, Professor at the Department of Computing Engineering
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» (Kyiv, Ukraine)
E-mail: klymenko.iryana@lil.kpi.ua. ORCID: <http://orcid.org/0000-0001-5345-8806>

HIGH-PERFORMANCE HARDWARE-SOFTWARE SYSTEM FOR EDGE PROCESSING OF SENSOR DATA STREAMS IN IoT SYSTEMS ON AN FPGA-SoC PLATFORM

The paper addresses the development and implementation of a high-performance hardware–software system for multi-level filtering of sensor measurements on an FPGA-SoC platform with an integrated ARM processor running the Linux operating system. The aim of the study is to ensure parallel processing of streaming sensor data at the edge level of IoT systems with minimal and deterministic latency, reduced network load, and improved scalability for multi-channel data streams.

To achieve this goal, a hardware filtering module was implemented as a three-stage pipelined architecture combining a median filter, a moving average filter, and exponential smoothing. A hardware–software interaction mechanism between the

embedded ARM processor and FPGA logic was implemented using the Avalon-MM interface, which provides high-performance exchange and processing of streaming sensor data in real time using fixed-point representation.

Experimental results show that the hardware implementation of multilevel sensor stream filtering on FPGA produces one processed sample per clock cycle after pipeline filling, with an internal latency of three clock cycles and a throughput of one sample per cycle. A comparison with a software implementation on an ARM Cortex-A9 processor demonstrates a 26% reduction in processing time. The experiments also confirm improved signal quality through suppression of impulse noise, smoothing of high-frequency fluctuations, and stabilization of the signal trend without loss of useful signal dynamics.

The developed sensor stream processing system, based on the combination of three-level hardware filtering on an FPGA-SoC platform and a high-speed Avalon-MM interface, provides deterministic, scalable, and efficient local edge-level data processing, significantly outperforming traditional software-based solutions on ARM processors. The proposed approach is suitable for distributed monitoring systems, industrial IoT applications, and other real-time scenarios where high performance and predictable timing characteristics are critical.

Keywords: *FPGA; data filtering; ARM; IoT.*

Fig.: 2. References: 6.