

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-54-20>

УДК 621.396

Озерчук Ігор Михайлович, провідний науковий співробітник

<https://orcid.org/0000-0001-7011-0772>

Український науково-дослідний інститут спеціальної техніки та судових експертиз Служби безпеки України, м. Київ, Україна

## АРХІТЕКТУРА ПЛАТФОРМИ ПРОГРАМНО-ВИЗНАЧЕНОГО РАДІО НА ОСНОВІ ПРОЦЕСОРА ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

**Озерчук І.М. Архітектура платформи програмно-визначеного радіо на основі процесора загального призначення.** У межах статті проведено розробку архітектури платформи програмно-визначеного радіо на основі процесора загального призначення. Наголошується, що прогрес у цифровій обробці радіосигналів, широке проникнення у всі сфери діяльності бездротових технологій зумовили появу нових видів радіосистем званих програмно-визначеним радіо. Підкреслюється, що основною тенденцією виступає локалізація аналогових елементів радіосистем (фільтрів, підсилювачів, генераторів несучої частоти, цифро-аналогових та аналого-цифрових перетворювачів) на входах/виходах радіосистем, широке застосування цифрових методів обробки радіосигналів, децентралізація цифрової обробки. Зазначається, що у класичному поданні програмно-визначена радіосистема (Software Defined Radio, SDR) являє собою центральний протокол, сигнально-інформаційний потік, процесор (або протокольний процесор), оснащений приймальним та передавальним блоками. Наведено архітектуру платформи SDR на основі GPP та підкреслено, що програмно-визначена радіосистема може складатися з кількох ПЛІС та обслуговувати кілька незалежних радіоканалів. Велика кількість комунікаційних процесорів забезпечує одночасну обробку кількох потоків даних. Самі собою комунікаційні процесори може бути кількох типів, кожен із яких оптимізовано до роботи з певним типом сигналів. Також, зазначається, що окремі типи процесорів можуть бути виділені для аналізу сигналів, збирання статистики або фільтрації пакетів. Можливості перепрограмування (повного чи часткового) дозволяють змінювати кількість та склад комунікаційних процесорів залежно від поточних умов роботи. Використання високошвидкісних сучасних послідовних приймачів, а також великої кількості паралельних каналів дозволяє поширити структуру між'єднань за межі одного кристала та з малими витратами поєднувати кілька ПЛІС у систему.

**Ключові слова:** канал, процесор, розробка, архітектура, платформа, програмно-визначене радіо, приймач.

**Ozerchuk I. A Software-Defined Radio Platform Architecture Based on a General-Purpose Processor.** Within the framework of the article, the architecture of the software-defined radio platform based on a general-purpose processor was developed. It is emphasized that progress in digital processing of radio signals, wide penetration of wireless technologies into all spheres of activity led to the emergence of new types of radio systems called software-defined radio. It is emphasized that the main trend is localization of analog elements of radio systems (filters, amplifiers, carrier frequency generators, digital-to-analog and analog-to-digital converters) at the inputs/outputs of radio systems, widespread use of digital methods of radio signal processing, decentralization of digital processing. It is noted that in the classic view, a software-defined radio system (Software Defined Radio, SDR) is a central protocol, a signal-information flow, a processor (or protocol processor), equipped with receiving and transmitting units. The architecture of the GPP-based SDR platform is presented and it is emphasized that a software-defined radio system can consist of several FPGAs and serve several independent radio channels. A large number of communication processors provides simultaneous processing of several data streams. Communication processors themselves can be of several types, each of which is optimized to work with a certain type of signals. Also, it is noted that certain types of processors can be dedicated to signal analysis, statistics collection or packet filtering. Possibilities of reprogramming (full or partial) allow changing the number and composition of communication processors depending on the current operating conditions. The use of high-speed modern serial receivers, as well as a large number of parallel channels, makes it possible to extend the structure of interconnections beyond the limits of one crystal and to combine several FPGAs into a system with low costs.

**Key words:** channel, processor, development, architecture, platform, software-defined radio, receiver.

**Вступ та постановка проблеми.** За останнє десятиліття кількість бездротових пристроїв, які використовуються різними бездротовими програмними доменами, такими як бездротові сенсорні мережі (WSN) [1], Інтернет речей (IoT) [2], базові станції стільникового зв'язку [3] тощо, надзвичайно зросла. Кілька бездротових технологій стандартизовано, щоб забезпечити взаємозв'язок між різними бездротовими пристроями, включаючи RFID, LoRa, 3GPP 3G/4G/5G тощо. Бездротовий пристрій може містити один або кілька бездротових приймачів-передавачів, що підтримують різні бездротові технології. Кожен трансивер виконує всі фізичні (PHY) і частину операцій рівня керування доступом до медіа (MAC) через інтегровані аналогові та цифрові схемні блоки. Дійсно, більшість аналогових операцій PHY-рівня реалізуються на виділеному та інтегрованому аналоговому обладнанні, такому як підсилювачі, радіочастотні (РЧ) синтезатори, фільтри тощо. Ця традиційна реалізація значно обмежує прямо чи опосередковано можливість програмування та гнучкість трансиверів для модернізації або роботи з кількома дротами.

Більше того, процесори бездротових трансиверів здебільшого запатентовані, що не дозволяє програмістам і дослідникам отримати доступ до перепрограмування коду. Альтернативним

рішенням, яке дозволяє програмістам і дослідникам легко керувати апаратним забезпеченням і програмувати програмне забезпечення бездротових приймачів-передавачів, є використання реалізації на основі процесора загального призначення (GPP) на основі програмно-визначеного радіо (SDR), яке є реконфігурованим. Крім того, оскільки рівні PHY і MAC виконуються в програмному забезпеченні хостом GPP і завдяки можливості реконфігурації та перепрограмуванню радіоприймача, платформу SDR можна використовувати для впровадження багатьох бездротових технологій.

**Аналіз останніх досліджень і публікацій.** Науковий підхід до висвітлення питань особливостей реалізації процесора загального призначення є різномірним та масштабним. У сучасній науковій площині з'являються роботи присвячені дослідженням платформ програмно-визначеного радіо, їх структурній складовій та технікам реалізації.

С.М. Семендй [4] розглянув можливість використання непрофесійного SDR радіообладнання та програмно-апаратних платформ з відкритим вихідним кодом для розробки та верифікації алгоритмів протидії РЕБ та підвищення автономності БпЛА на полі бою.

І.С. Пятін та Ю.М. Бойко [5] провели оцінку спотворень сигналу у HDL-реалізації системи зв'язку на базі середовища розробки Xilinx Zynq-7000. Алгоритм оцінки та корекції зміщення частоти розроблений за критерієм максимальної правдоподібності. Підсистема відновлення синхронізації символів виконана на основі кола фазового автопідлаштування частоти (ФАПЧ).

Із зарубіжних авторів варто відмітити роботи таких науковців як: Ксонне Метью, Хваліте Джерома, Бол Девіда [6], Бріджес К. [7], Утрілла Раміро, Родрігес-Суррунеро, Роберто, Мартін Хосе, Розас Альба, Араужо Альваро [8], Шмідт-Кноррек До., Пакалет Р., Мінвеген А., Дайдерсен У., Кемпф Т., Кноп Р., Ашейд Р. [9], Урс Х., Редді Р. Венката Шива, Гудодагі Равеендра, Судхаршан К., Аравінд Би. [10], Беккер Джон, Старобінський Давид [11], Албяті Мохаммед, Дано Ерік, Раджамані Раві, Томпсон Емі [12], Манко Анджело, Кастрільо Вітторіо [13], Рамос Маріо, Камачо Брінес Рауль, Кадавід Стерв'ятник Паола, Урда Роберт, Рестрепо Хуліо [14], Ань Сюян, Юй Сюей, Сун Вейлун, Хан Ле, Ян Тінгін, Лі Чжаодун, Су Чжібао [15], Перотоні Марсело, Скаллоп Марк, Бартолеті Рафаель [16] та інших.

Однак незважаючи на масштабність наукових досліджень питання реалізації платформи програмно-визначеного радіо на основі процесора загального призначення не викликає сумнівів.

**Постановка завдання.** Метою роботи є розробка архітектури платформи програмно-визначеного радіо на основі процесора загального призначення.

**Викладення основного матеріалу дослідження.** Конструкції реалізації звичайних бездротових трансиверів, як правило, не вистачає можливості перепрограмування, гнучкості та масштабованості. Таким чином, оновлення програмного забезпечення, зміна логіки спеціального апаратного забезпечення або повторне використання трансивера для реалізації стандарту бездротового зв'язку, відмінного від того, для якого розроблено трансивер, є обмеженими або відсутніми. Крім того, звичайні бездротові трансивери здебільшого запатентовані, що не дозволяє розробникам і дослідникам отримати доступ до перепрограмування набору інструкцій. Альтернативним рішенням для пом'якшення цих обмежень є використання платформ SDR. На додаток до функції програмування, платформа SDR також служить мультитехнологічним шлюзом, використовуючи кілька бездротових технологій за допомогою загального набору радіоприймачів [10]. Це також дозволяє повторно використовувати програмне забезпечення на кількох радіопристроях і завантажувати програмне забезпечення онлайн для впровадження нових стандартів і виправлення помилок.

Платформа SDR – це клас радіоприймачів, які керують аналоговою частиною RF/IF за допомогою плати аналогового пристрою з відкритим вихідним кодом, і реалізують всю цифрову частину за допомогою програмованого головного процесора. Програмований головний процесор може бути GPP, DSP ASIP або FPGA. У рамках даного дослідження розкрито особливості платформам SDR на основі GPP через її легку програмованість за допомогою мови високого рівня та її гнучкості для реконфігурації та обробки складних алгоритмів. Загальна архітектура платформи SDR на основі GPP проілюстрована на рис. 1. Вона в основному розділена на три частини: пристрій SDR, інтерфейс зв'язку і хост GPP. Кожен компонент платформи має свої параметри зв'язку, які сприяють загальній продуктивності платформи SDR.

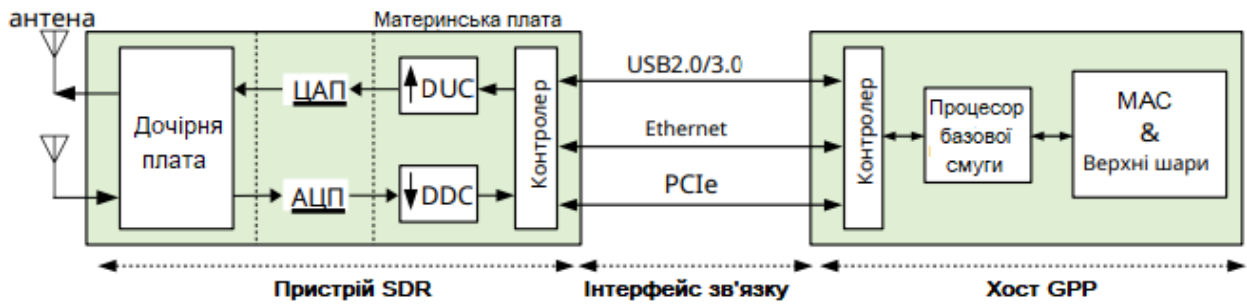


Рис. 1 – Загальна архітектура платформи SDR на основі GPP

Пристрій SDR – це невеликий портативний пристрій, який здатний передавати та приймати сигнали на різних частотах. Зазвичай він складається з керованих програмним забезпеченням аналогових РЧ/ПЧ і цифрових ПЧ інтерфейсів. Перша називається дочірньою платою, а друга – материнською.

Дочірня плата, по суті, відповідає за виконання аналогових функцій обробки РЧ/ПЧ, таких як фільтрація, підсилення, перетворення сигналів з РЧ в ПЧ і навпаки тощо. В основному діапазон робочих частот, ширина аналогової смуги частот і можливості каналу визначають сферу дочірньої плати для реалізації широкого спектру бездротових технологій з використанням пристроїв SDR. Дочірня плата з'єднує антену з материнською платою. Дійсно, більшість дочірніх плат інтегрують кілька ланцюгів вводу/ виводу для підключення кількох окремих антен, що забезпечує можливість одночасної передачі та прийому. Крім того, дочірні плати інтегрують аналогові входи/виходи для підключення АЦП і ЦАП материнської плати. На ринку дочірні плати є або окремими компонентами, або об'єднані з материнською платою, утворюючи єдину плату.

Материнська плата в основному відповідає за оцифрування та перетворення частоти дискретизації. Для виконання цих операцій материнські плати інтегрують АЦП/ЦАП і процесор DSP, які можна реалізувати за допомогою ASIC, DSP ASIP або FPGA. Він також об'єднує один або кілька комунікаційних інтерфейсів для з'єднання з хостом GPP. В основному він характеризується максимальною частотою дискретизації АЦП і ЦАП, роздільною здатністю АЦП і ЦАП, конструкцією процесора DSP і підтримуваним інтерфейсом зв'язку.

Материнська плата в пристроях SDR поєднує дочірню плату з хостом GPP. Він обмінюється вибіркою основної смуги з хостом GPP і аналоговими сигналами з дочірніми платами. Насправді отриманий аналоговий сигнал ПЧ від дочірньої плати спочатку оцифровується АЦП для отримання цифрових зразків ПЧ, а потім зменшується через DDC для отримання цифрових зразків основної смуги частот.

Нарешті, зразки базової смуги передаються на хост GPP через інтегрований інтерфейс зв'язку. У зворотному напрямку отримані зразки базової смуги спочатку підвищуються через DUC, щоб отримати цифрові зразки ПЧ, а потім ЦАП перетворюють на аналоговий сигнал ПЧ. Нарешті аналоговий сигнал ПЧ передається на дочірню плату [9]. На ринку материнські плати являють собою або окремі плати, що містять слоти для підключення дочірніх плат, або плату з інтеграцією дочірніх плат.

Дані передаються з хосту GPP на пристрій SDR і навпаки за допомогою дротових інтерфейсів зв'язку. Вони базуються на широко використовуваних комунікаційних інтерфейсах для передачі даних, таких як універсальна послідовна шина (USB 2.0, USB 3.0 тощо), Ethernet (стандартний, швидкий, гігабітний тощо) та взаємозв'язку периферійних компонентів (PCIe 1.x, PCIe 2.x тощо). Вони складаються з контролерів, таких як контролер мережевого інтерфейсу (NIC), встановлених як на хості GPP, так і на материнській платі SDR, для реалізації стандарту інтерфейсу зв'язку. Технологія комунікаційного інтерфейсу, реалізована контролером, має специфічні характеристики, визначені стандартами, такими як максимальна підтримувана швидкість, максимальний розмір корисного навантаження, максимальна довжина кабелю тощо. Щоб забезпечити більшу гнучкість швидкості передачі даних, деякі пристрої SDR включають кілька комунікаційних інтерфейсів. Контролери як пристрою SDR, так і хосту GPP мають реалізовувати однаковий стандарт інтерфейсу, але не обов'язково однаковою версією. Дійсно, різні версії одного стандарту можуть створювати з'єднання між двома сторонами, але вони повинні бути синхронізовані для ефективного обміну даними. У таких випадках стандарт зв'язку з найнижчою

швидкістю буде узгоджений шляхом автоматичного узгодження (як для Ethernet) або зворотної сумісності (наприклад, між USB 2.0 і USB 3.0) або вручну.

Хост GPP – це програмований пристрій, який може виконувати обчислювальні завдання на основі інструкцій, наданих програмним забезпеченням, що використовує мови програмування високого або нижчого рівня. Таким чином, хост GPP поєднує апаратне та програмне забезпечення та відповідає за їх взаємодію.

Компоненти вузла GPP здебільшого зібрані на одній платі. Ця плата містить внутрішні компоненти SOC, такі як GPP, внутрішня пам'ять, співпроцесори (GPU, DSP ASIP тощо) і можливі контролери для інтерфейсів зв'язку та зовнішні компоненти SOC, такі як зовнішня пам'ять, слоти розширення тощо.

GPP, який може бути мікропроцесором або мікроконтролером, відповідає за виконання цифрових операцій РНУ, MAC і верхнього рівня. На відміну від звичайних трансиверів, він має перевагу високого рівня програмування без модифікації апаратного забезпечення. Незважаючи на те, що він пропонує високу гнучкість для користувача, програмованість високого рівня зазвичай призводить до погіршення продуктивності процесора для задоволення вимог інтенсивних обчислювальних завдань обробки сигналів [11]. Дійсно, продуктивність (швидкість обробки) GPP значною мірою визначається тактовою частотою, де нижча тактова частота означає повільний процесор і менше споживання енергії.

GPP може мати одноядерний або багатоядерний процесор. Однак більшість GPP на даний момент базуються на багатоядерних (двоядерних, чотирьохядерних тощо) процесорах на одному фізичному центральному процесорі (CPU). Кожне ядро в багатоядерній однопроцесорній системі представляє один процесор або виконавчий блок, здатний виконувати процеси одночасно з іншими процесорами. Це збільшує кількість запитів, які потрібно обробити за один такт. Окрім тактової частоти та кількості ядер, на швидкість обробки істотно впливає архітектура системної шини (ширина шини, її тактова частота, кількість даних, які вона може передати за такт тощо). Розмір і рівень кешу ЦП також впливає на швидкість його обробки. Інші параметри, які можуть впливати на загальну швидкість обробки GPP, це кількість потоків, розмір пам'яті, кількість ALU, підтримка гіперпоточної обробки, розмір блоків SIMD (Single Instruction Multiple Data) тощо. Блоки SIMD дозволяють процесору виконувати одночасно одну і ту саму операцію для кількох блоків даних. Останні GPP підтримують архітектуру SIMD для підвищення продуктивності. Щоб досягти більшої обчислювальної продуктивності, GPP зазвичай доповнюються співпроцесорами, такими як GPU, FPGA та DSP ASIP.

Програмна частина хосту GPP керує роботою процесора, вхідним/ вихідним трафіком контролерів зв'язку та SDR-пристроєм. Зазвичай він розділений на три рівні: набір операцій, простір ядро та простір користувача. Набір інструкцій/операцій визначається як група операцій, які може виконувати процесор. Таким чином, код операції (об'єктний код), згенерований компілятором або асемблером може містити лише інструкції з цього набору. Набір операцій є одним із двох типів проектів архітектури набору операцій/інструкцій/команд (ISA): комп'ютери зі скороченим набором інструкцій (RISC) або комп'ютери зі складним набором інструкцій (CISC). ISA GPP може базуватися на CISC або RISC. Щоб використати переваги обох наборів інструкцій, сучасні GPP більшою мірою базуються на гібридному ISA (використовують зовнішні інструкції CISC, але внутрішні методи RISC). Крім того, використання RISC архітектури також може бути покращено шляхом додавання розширень дуже довгих машинних команд (VLIW), техніки, яка пропонує паралелізм рівня інструкцій.

Середнім рівнем архітектури програмної системи є ядро. Це серце операційної системи (ОС), що з'єднує простір користувача з апаратним процесором. Для взаємодії з апаратним забезпеченням ядро включає драйвери апаратного забезпечення, такі як драйвер процесора, драйвер жорсткого диска, драйвер мережевого контролера тощо. Для взаємодії з простором користувача ядро включає інтерфейс прикладних програм (API), який дозволяє програмам у просторі користувача отримувати доступ до системних ресурсів (наприклад, файлові системи, час GPP, віртуальна пам'ять тощо) та служби (наприклад, планування, обмін, обробка запитів на переривання (IRQ), перемикання контексту тощо). Саме ці служби впливають на продуктивність простору ядра з точки зору затримки та накладних витрат. Щоб зменшити затримку, необхідні додаткові функції, такі як обробник IRQ, планування процесу, зменшення кількості перемикань контексту тощо. З іншого боку, накладні витрати на ядро – це час, пов'язаний з керуванням такими ресурсами, як час GPP, пам'ять, диск тощо. Збільшення накладних витрат часто призводить до зменшення зайнятості

часу GPP і, як наслідок, пропускну здатності GPP. Оскільки для зменшення затримки ядра необхідні додаткові функції, накладні витрати ядра збільшаться. Очевидно, що існує компроміс між затримкою ядра та пропускну здатністю GPP, і баланс слід ретельно опрацьовувати відповідно до потреб користувача.

На вершині архітектури програмної системи знаходиться простір користувача, який складається з частини пам'яті, у якій виконуються програми користувача. Таким чином, програми користувача є функціями РНУ і МАС бездротових технологій. Програми користувача здебільшого написані з використанням мов програмування високого рівня, таких як С, С++, Java, Python, Matlab тощо. Також можна генерувати код програм користувача за допомогою текстових/ графічних мов програмування потоку даних, таких як програмування G, Python, С++, тощо. Ці мови програмування зазвичай входять до наборів інструментів розробки програмного забезпечення, таких як GNU Radio, LabVIEW, Matlab тощо. Набори інструментів надають бібліотеки для функцій DSP, бібліотеки для середовища виконання та компіляції, графічні засоби для створення графів потоку сигналів і генерування вихідного коду потокового графа тощо.

Компілятор користувацьких додатків є найважливішим елементом, який допомагає процесору досягти високої продуктивності та максимального часу виконання. Він відповідає за генерацію коду інструкцій за допомогою ISA цільового процесора. Якщо підтримується велика різноманітність цільових процесорів, компілятор називається загальним. Загальні компілятори також реалізують оптимізацію для покращення продуктивності GPP шляхом збільшення рівня паралелізму за допомогою трьох механізмів: паралелізм на рівні інструкцій (ILP), який дозволяє виконувати кілька інструкцій одночасно, паралелізм на рівні потоку (TLP), який дозволяє одночасному або псевдоодностанному запуску кількох потоків на одному чи кількох ядрах, і паралелізм на рівні даних (DLP), який дозволяє виконувати декілька елементів даних одночасно. Це передбачає оптимальне генерування вихідного коду за розміром і часом виконання відповідно до цільового процесора.

Щоб реалізувати бездротову технологію на платформах SDR або використовувати існуючі реалізації, необхідно, щоб продуктивність обраної платформи SDR (пристрій SDR, інтерфейс зв'язку та хост GPP) відповідала принаймні вимогам цільової бездротової технології. Ці вимоги в основному наведені з точки зору діапазону робочих частот, пропускну здатності, швидкості символів, бітрейту, затримки тощо.

Смуга частот платформи SDR – це робочий діапазон частот, який покриває пристрій SDR. Визначається на дочірніх платах за сигналами гетеродина (LO), які генеруються синтезатором частоти, таким як синтезатор фазового автопідстроювання частоти (PLL). Велика смуга частот потребує широкого діапазону частот гетеродина, а отже, і широкосмугових синтезаторів частот. Щоб охопити діапазон частот, підтримуваних дочірніми платами, пристрої SDR повинні використовувати відповідний тип антени.

Будь-який аналоговий або цифровий сигнал має смугу пропускання, визначену як зайнятий діапазон частот, що переносить більшу частину його енергії. Цей діапазон змінюється на кожному етапі сигнального ланцюга. Отже, він може бути виражений по-різному відповідно до етапу обробки сигналу. Дійсно, на РЧ передньому етапі це виражається як аналогова смуга пропускання або ширина РЧ каналу. На етапі АЦП/ЦАП це виражається як частота дискретизації ЦАП/АЦП. Коли сигнал обробляється на цифровому передньому кінці (DFE), його пропускну здатність виражається як частота дискретизації DFE. У каналі зв'язку між DFE і GPP смуга пропускання обмежена швидкістю інтерфейсу зв'язку. На хості GPP пропускну здатність виражається як швидкість передачі символу.

Смуга пропускання, виміряна в Гц, визначається радіочастотним інтерфейсом (дочірньою платою) пристрою SDR. Він налаштований головним чином аналоговим фільтром низьких частот (LPF) для зміни частоти від 0 Гц до заданої частоти зрізу фільтру.

На материнській платі пристроїв SDR вбудований АЦП/ЦАП може підтримувати одну або кілька частот дискретизації. Найвище значення частоти дискретизації визначає найбільшу аналогову смугу пропускання. Використання вибіркової частоти дискретизації є кращим, ніж фіксованої частоти, щоб адаптувати реальну аналогову смугу пропускання до необхідної смуги пропускання, яка встановлена програмними вимогами до частоти та виражається частотою дискретизації DFE. Коли найближча частота дискретизації більша, ніж частота дискретизації DFE, необхідне коригування за допомогою процесу інтерполяції та децимації.

**Висновки.** У межах даної роботи розроблено та детально описано архітектуру платформи програмно-визначеного радіо на основі процесора загального призначення. Оскільки частота дискретизації DFE має бути близькою до бітрейту хосту GPP, її можна використовувати для визначення необхідної пропускної здатності каналу, визначеною потребами програми користувача на хості GPP. Оскільки необхідна пропускна здатність каналу включена в реальну аналогову пропускну здатність, частота дискретизації DFE має бути меншою, ніж реальна аналогова пропускна здатність. Крім того, частоту дискретизації DFE слід збільшити або зменшити, щоб відповідати тактовій частоті ЦАП і АЦП. Деякі пристрої SDR вимагають строго цілочисельної інтерполяції та коефіцієнтів проріджування, і дуже бажано, щоб ці коефіцієнти були парними, і набагато краще, якщо коефіцієнти будуть мати дві потужності. Таким чином, вказівка відповідної частоти дискретизації DFE є ще однією вимогою, яку має враховувати користувач.

#### Список бібліографічного опису

1. Kumar Avinash. Wireless Sensor Network (WSN). 2020. DOI: 10.13140/RG.2.2.26851.27681.
2. Зубок В.Ю., Давидюк А.В. Архітектура інтернету речей та моделі обробки даних / Міжнародна науково-технічна конференція «Інтелектуальні технології лінгвістичного аналізу»: Тези доповідей. К.: НАУ, 2022. 56 с.
3. Микитенко С. Принципи формування архітектури мереж стільникового зв'язку п'ятого покоління. Computer-integrated technologies: education, science, production. 2022. P. 111-116. DOI: 10.36910/6775-2524-0560-2022-47-17.
4. Семендяй С. Використання технології когнітивного радіо для підвищення ефективності безпроводових систем передачі даних в умовах активного застосування засобів радіоелектронної боротьби. Кібербезпека: освіта, науки, техніка. Національний університет «Чернігівська політехніка», Чернігів, 2023. № 4(20). С. 220-229. DOI: 10.28925/2663-4023.2023.20.220229
5. Пятін І.С. Синхронізація систем зв'язку на основі SDR /Вісник Хмельницького національного університету. 2022. №5. С. 1-9.
6. Xhonneux Mathieu, Louveaux Jerome, Bol David. A sub-mW Cortex-M4 Microcontroller Design for IoT Software-Defined Radios. IEEE Open Journal of Circuits and Systems. 2023. P. 1-1. DOI: 10.1109/OJCS.2023.3270752.

#### References

7. Bridges C. Software Defined Radio (SDR) architecture to support multi-satellite communications. 2015. DOI: 10.1109/AERO.2015.7119186.
8. Utrilla Ramiro, Rodriguez-Zurrunero Roberto, Martin Jose, Rozas Alba, Araujo Alvaro. MIGOU: A Low-Power Experimental Platform with Programmable Logic Resources and Software-Defined Radio Capabilities. Sensors. 2019. № 19. P. 49-83. DOI: 10.3390/s19224983.
9. Schmidt-Knorreck C., Pacalet R., Minwegen A., Deidersen U., Kempf T., Knopp R., Ascheid G. Flexible front-end processing for software defined radio applications using application specific instruction-set processors. Conference on Design and Architectures for Signal and Image Processing, DASIP. 2012. P.1-8.
10. Urs H., Reddy R Venkata Siva, Gudodagi Raveendra, Sudharshan K., Aravind B. A Novel Algorithm for Reconfigurable Architecture for Software-Defined Radio Receiver on Baseband Processor for Demodulation. 2023. DOI: 10.1007/978-3-031-13577-4\_11.
11. Becker Johannes, Starobinski David. Snout: A Middleware Platform for Software-Defined Radios. IEEE Transactions on Network and Service Management. 2022. P. 1-1. DOI: 10.1109/TNSM.2022.3215626.
12. Albayati Mohammed, Dano Eric, Rajamani Ravi, Thompson Amy. A Model-Based Engineering Approach for Evaluating Software-Defined Radio Architecture. Systems. 2023. №11. 480 p. DOI: 10.3390/systems11090480.
13. Manco Angelo, Castrillo Vittorio. An FPGA Scalable Software-Defined Radio Platform for UAS Communications Research. Journal of Communication. 2021. № 2. P. 42-51. DOI: 10.12720/jcm.16.2.42-51.
14. Ramos Mario, Camacho Briñez Raul, Buitrago Cadavid Paola, Urda Robert, Restrepo Julio. Software Defined Radio, a perspective from education. Frontiers in Education. 2024. №8. DOI: 10.3389/educ.2023.1228610.
15. An Xuyang, Yu Xuwei, Song Weilong, Han Le, Yang Tingting, Li Zhaodong, Su Zhibao. A Software-Defined Distributed Architecture for Controlling Unmanned Swarm Systems. Electronics. 2023. №12. P. 37-39. DOI: 10.3390/electronics12183739.
16. Perotoni Marcelo, Vieira Marcos, Bartolleti Rafael. Experiments with a Holographic Radar Based on Software Defined Radio. IEEE Latin America Transactions. 2023. №21. DOI: 10.1109/TLA.2023.10305239.