

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І  
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»  
УДК 004.31

«До захисту допущено»  
Завідувач кафедри СПСКС  
Тарасенко В.П.  
\_\_\_\_\_ (підпис) \_\_\_\_\_ (ініціали, прізвище)  
“ \_\_\_\_ ” \_\_\_\_\_ 2018 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності 123 Комп'ютерна інженерія  
Системне програмування

на тему: СПЕЦІАЛІЗОВАНІ ЗАСОБИ НА ПЛІС ДЛЯ ОБРОБКИ  
ГЕТЕРОГЕННИХ ДАНИХ

Виконав: студент II курсу, групи КВ-72мп  
(шифр групи)

Хомук Богдан Олександрович \_\_\_\_\_ (підпис)  
(прізвище, ім'я, по батькові)

Керівник: к.т.н., доцент Клятченко Я.М. \_\_\_\_\_ (підпис)  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент: \_\_\_\_\_ (підпис)  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_ (підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

Системне програмування

ЗАТВЕРДЖУЮ

Завідувач кафедри СПСКС

Тарасенко В.П.

\_\_\_\_\_

\_\_\_\_\_

(підпис)

(ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Хомуку Богдану Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Спеціалізовані засоби на ПЛІС для обробки гетерогенних даних»  
науковий керівник дисертації к.т.н., доцент Клятченко Я.М.  
затверджені наказом по університету від 30 жовтня 2018 р. №4030-с.
2. Термін подання студентом дисертації 7 грудня 2018 р.\_\_\_\_\_
3. Об'єкт дослідження: є бездротова сенсорна мережа збору даних.
4. Предмет дослідження: є метод і спосіб балансування бездротової сенсорної мережі збору даних на основі ПЛІС.
5. Перелік завдань, які потрібно розробити полягає в огляді і аналізі балансування бездротової сенсорної мережі збору даних на основі ПЛІС, розробці бездротової сенсорної мережі та забезпеченні енергозбереження бездротової сенсорної мережі.
6. Перелік ілюстративного матеріалу презентація (кількість аркушів: 14).

7. Перелік публікацій: 1) XI Конференція молодих вчених «Прикладна математика та комп'ютеринг» ПМК-2018-2 «Підвищення енергоефективності сенсорних мереж за допомогою засобів на ПЛІС»; 2) IV Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами «Паралельна обробка даних на засобах ПЛІС», «Кросплатформене розподілення сенсорних мереж за допомогою засобів на ПЛІС».

8. Дата видачі завдання 5 вересня 2017 .

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Вивчення літератури за тематикою проекту	25.11.2017	
	Аналіз існуючих рішень	19.12.2017	
	Підготовка матеріалів першого розділу магістерської дисертації	09.02.2018	
	Підготовка матеріалів другого розділу магістерської дисертації	21.03.2018	
	Підготовка матеріалів третього розділу магістерської дисертації	22.04.2018	
	Підготовка матеріалів четвертого розділу магістерської дисертації	23.05.2018	
	Підготовка матеріалів п'ятого розділу магістерської дисертації	24.06.2018	
	Оформлення документації магістерської дисертації	25.07.2018	
	Попередній розгляд магістерської дисертації на кафедрі	26.11.2018	

Студент

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (прізвище, ініціали)

Науковий керівник дисертації

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (прізвище, ініціали)

## РЕФЕРАТ

**Актуальність теми.** В даний час в системах автоматизації, моніторингу за станом об'єктів все частіше успішно використовується технологія бездротових сенсорних мереж (БСМ).

Найбільшого поширення набули стаціонарні бездротові сенсорні мережі, для яких характерне їх фіксований стан в процесі роботи. Останнім часом в сенсорних мережах активно розвивається напрямок, пов'язаний з використанням для них автономних джерел живлення на базі альтернативних джерел енергії.

**Об'єктом дослідження** є бездротова сенсорна мережа збору даних.

**Предметом дослідження** є метод і спосіб балансування бездротової сенсорної мережі збору даних на основі ПЛІС.

**Мета роботи** полягає в огляді і аналізі балансування бездротової сенсорної мережі збору даних на основі ПЛІС, розробці бездротової сенсорної мережі та забезпеченні енергозбереження БСМ.

**Методи дослідження.** Для реалізації мети дослідження і вирішення поставлених завдань будуть використані наступні наукові методи і підходи: методи системного аналізу, моделювання, методи дискретного програмування.

### **Наукова новизна:**

- розроблено та досліджено спосіб балансування бездротової сенсорної мережі збору даних на основі ПЛІС, що дозволяє вирішувати практичні завдання, в тому числі і завдання великої розмірності;

- розроблено та досліджено метод енергетичного балансування бездротової сенсорної мережі, що зменшує енергоспоживання вузлів мережі і збільшує тривалість безперервної роботи такої мережі.

**Практична цінність** отриманих в роботі результатів полягає в тому, що запропонована реалізація бездротової сенсорної мережі може використовуватися для системи моніторингу радіочастотного спектра і радіоелектронних засобів і супровід електронних інформаційних ресурсів та інформаційних систем, що можуть тривалий час працювати без заміни джерела живлення.

**Апробація роботи.** Результати роботи пройшли апробацію або знаходяться на стадії публікації на конференціях:

– XI конференція молодих вчених «Прикладна математика та комп'ютинг» ПМК-2018-2;

– IV Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами;

Результати роботи опубліковані в 3 наукових працях, з яких 3 – тези доповідей.

**Структура та обсяг роботи.** Магістерська дисертація складається з вступу, п'ятих розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень.

У першому розділі розглянуті загальні відомості про бездротові сенсорні мережі, методи маршрутизації, агрегація даних, забезпечення надійної передачі даних БСМ, система моніторингу на базі БСМ та функціональне призначення БСМ.

У другому розділі надані загальні відомості про ПЛІС, обґрунтування вибору ПЛІС, та порівняння ПЛІС з іншими мікросхемами.

У третьому розділі надано огляд архітектури FPGA, логіка об'єднання структури, розширення логіки FPGA.

У четвертому розділі надана інформація про сімейство ПЛІС - Spartan-3AN, були розглянуті архітектурні особливості, конфігурації, маркування упаковки, вимоги до програмного забезпечення.

У п'ятому розділі надана інформація про оцінку БСМ, розглянута структура створеного прийомопердавача, який був оптимізований, для більш тривалої безвідмовної роботи, проведені експериментальні дослідження БСМ.

У висновках надана інформація про отримані результати і короткий опис всієї пройденої роботи, що була обговорена в попередніх розділах.

Робота виконана на 86 аркушах, містить 4 додатків та посилання на список використаних літературних джерел з 24 найменувань. У роботі наведено 29 рисунків та 8 таблиць.

**Ключові слова:** бездротові сенсорні мережі, сенсори, радіопередача, ПЛІС, FPGA, Spartan-3AN.

## РЕФЕРАТ

**Актуальность темы.** Сейчас в системах автоматизации мониторинга за состоянием объектов очень часто использует технологии беспроводных сенсорных сетей (БСС).

Самыми распространенными беспроводными сенсорными сетями, есть сети, для которых характерно их фиксированный состояние во время выполнения работы. В сенсорных сетях, сейчас, активно набивает развитие такое направление, что связано с использованием автономных источников питания для длительного процесса работы.

**Объектом исследования** есть беспроводная сенсорная сеть обработки и сбора данных.

**Предметом исследования** есть способ и метод балансировки беспроводной сенсорной сети обработки и сбора данных на основе ПЛИС.

**Цель работы** заключается в разработке беспроводной сенсорной сети, анализе и осмотре балансировки беспроводной сенсорной сети обработки и сбора данных на основе ПЛИС, и обеспечении энергосбережения БСС.

**Методы исследования.** Для того чтобы реализовать цели решения поставленных задач данного исследования будут использованы такие методы как: моделирования, методы дискретного программирования, методы системного анализа.

### **Научная новизна:**

- были исследованы и разработаны способы балансировки беспроводной сенсорной сети обработки и сбора данных на основе ПЛИС, что позволило решить практические задачи, а также задачи с большой размерностью;

- были исследованы и разработаны методы энергетической балансировки беспроводной сенсорной сети, что позволило значительно уменьшить затраты энергопотребление узлов сети и при этом продолжительность непрерывной работы сети увеличилось.

**Практическая ценность** полученных в работе результатов заключается в том, что предложенная реализация беспроводной сенсорной сети может использоваться для системы мониторинга, а также обработки и сбора данных информационных систем, которые могут длительное время работать без замены источника питания.

**Апробация работы.** Результаты работы прошли апробацию или находятся на стадии публикации на конференциях:

- XI конференция молодых ученых «Прикладная математика и компьютеринг» ПМК-2018-2;

- IV Международная научно-техническая Internet-конференция «Современные методы, информационное, программное и техническое обеспечение систем управления организационно-техническими и технологическими комплексами;

Результаты работы опубликованы в 3 научных работах, из которых 3 - тезисы докладов.

**Структура и объем работы.** Магистерская диссертация состоит из введения, пяти глав, заключения и приложений.

Во введении дана общая характеристика работы, выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований.

В первом разделе рассмотрены общие сведения о беспроводных сенсорных сетях, методы маршрутизации, агрегация данных, обеспечения надежной передачи данных БСС, система мониторинга на базе БСС и функциональное назначение БСС.

Во втором разделе предоставлены общие сведения о ПЛИС, обоснование выбора ПЛИС, и сравнения ПЛИС с другими микросхемами.



В третьем разделе дан обзор архитектуры FPGA, логика объединения структуры, расширение логики FPGA.

В четвертом разделе представлена информация о семействе ПЛИС - Spartan-3AN, были рассмотрены архитектурные особенности, конфигурации, маркировка упаковки, требования к программному обеспечению.

В пятом разделе предоставлена информация об оценке БСС, рассмотрена структура созданного приемопередавача, который был оптимизирован для более длительной безотказной работы, проведены экспериментальные исследования БСС.

В выводах предоставлена информация о полученных результатах и краткое описание всей пройденной работы, которая была обсуждена в предыдущих главах.

Работа выполнена на 86 листах, содержит 4 приложений и ссылки на список использованных литературных источников из 24 наименований. В работе приведены 29 рисунков и 8 таблиц.

**Ключевые слова:** беспроводные сенсорные сети, сенсоры, радиопередача, ПЛИС, FPGA, Spartan-3AN.

## ABSTRACT

**Relevance of the topic.** At present, the technology of wireless sensor networks (WSN) is increasingly being successfully used in automation systems and monitoring the state of objects.

The most widespread stationary wireless sensor networks, which are characterized by their fixed state in the process. Recently, a trend has been actively developing in sensor networks related to the use of autonomous power sources for them based on alternative energy sources.

**The object of the research** is a wireless sensor network data collection.

**The subject of research** is the method and method of balancing a wireless sensor network data acquisition based on FPGA.

**The purpose of the work** is to inspect and analyze the balancing of the FPGA-based wireless sensor network data collection, develop a wireless sensor network and ensure energy-saving base stations.

**Research methods.** To achieve the goal of the research and solve the set tasks, the following scientific methods and approaches will be used: methods of system analysis, modeling, and methods of discrete programming.

### **Scientific novelty:**

- a method of balancing a wireless FPGA-based sensor network has been developed and investigated, which allows solving practical tasks, including large-scale tasks;

- method of energy balancing a wireless sensor network has been developed and investigated, reduces the power consumption of network nodes and increases the duration of the continuous operation of such a network.

**The practical value** of the results obtained in the work lies in the fact that the proposed implementation of a wireless sensor network can be used for monitoring the radio frequency spectrum and radio electronic means and tracking electronic

information resources and information systems that can work for a long time without changing the power source.

**Approbation of work.** The results of the work have been tested or are being published at conferences:

- XI Conference of Young Scientists "Applied Mathematics and Computing" PMK-2018-2;

- IV International Scientific and Technical Internet-Conference "Modern Methods, Information, Software and Technical Support of Management Systems for Organizational, Technical and Technological Complexes;

The results of the work are published in 3 scientific papers, of which 3 are theses of the reports.

**Structure and scope of work.** The master thesis consists of introduction, five chapters, conclusion and applications.

In the introduction, a general description of the work is given, the current state of the problem is assessed, and the relevance of the research direction is substantiated.

The first section covers general information about wireless sensor networks, routing methods, data aggregation, ensuring reliable WSN data transmission, a monitoring system based on the WSN, and the functionality of the WSM.

The second section provides general information about the FPGA, the rationale for the selection of the FPGA, and the comparison of the FPGA with other chips.

The third section provides an overview of the architecture of FPGA, the logic of combining the structure, the extension of the logic of FPGA.

The fourth section provides information about the FPGA family - Spartan-3AN, architectural features, configurations, packaging marking, software requirements were considered.

The fifth section provides information on the evaluation of the WSN, reviewed the structure of the created receiver, which was optimized for longer uptime, conducted experimental studies of the WSN.

The appendices provide information on the results obtained and a brief description of all the work done, which was discussed in previous chapters.

The work was done on 86 pages, contains 4 annexes and references to the list of references used from 24 titles. The work contains 29 figures and 8 tables.

**Keywords:** wireless sensor networks, sensors, radio transmission, FPGA, FPGA, Spartan-3AN.

## ЗМІСТ

<b>ВСТУП</b> .....	20
<b>РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО СЕНСОРНІ МЕРЕЖІ</b> .....	22
<b>1.1 Бездротові сенсорні мережі</b> .....	22
<b>1.2 Методи маршрутизації в бездротових сенсорних мережах</b> ..	23
<b>1.2.1 Самоорганізована та надійна мережева технологія</b> ...	25
<b>1.2.2 Низька вартість IP-технологій з'єднання</b> .....	26
<b>1.2.3 Технологія автоматичного регулювання потоку</b> .....	27
<b>1.3 Маршрутизація в бездротових сенсорних мережах</b> .....	27
<b>1.4 Агрегація даних</b> .....	32
<b>1.5 Забезпечення надійної передачі даних в бездротових сенсорних мережах</b> .....	34
<b>1.6 Система моніторингу на базі бездротової сенсорної мережі</b> ..	35
<b>1.7 Функціональні призначення і сфери застосування</b> .....	36
<b>РОЗДІЛ 2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ПЛІС ТА ОБГРУНТУВАННЯ ЙОГО ВИБОРУ</b> .....	38
<b>2.1 Передумови до розвитку ПЛІС. Переваги ПЛІС</b> .....	38
<b>2.2 Порівняльний аналіз ПЛІС і НВІС</b> .....	41
<b>2.3 Класифікація пограмованих логічних інтегральних схем (ПЛІС)</b> .....	43
<b>2.3.1 Класифікація ПЛІС за ступенем інтеграції</b> .....	43
<b>2.4 Особливості проектування цифрових пристроїв на базі ПЛІС</b> .....	43
<b>РОЗДІЛ 3 ЗАГАЛЬНИЙ ОГЛЯД АРХІТЕКТУРИ FPGA</b> .....	46
<b>3.1 Логіка об'єднання структури</b> .....	46
<b>3.2 Повернення і взаємодія</b> .....	49
<b>3.3 Розширення логіки FPGA</b> .....	54
<b>3.4 Конфігурації FPGA</b> .....	58
<b>3.4.1 Статична оперативна пам'ять з довільним доступом (SRAM)</b> .....	59
<b>3.4.2 Флеш пам'ять</b> .....	60
<b>РОЗДІЛ 4 ОГЛЯД СІМЕЙСТВА SPARTAN-3AN FPGA</b> .....	61

4.1 Огляд архітектурних особливостей .....	61
4.2 Конфігурація .....	63
4.3 Флеш-пам'ять системи.....	64
4.3 Можливості вводу-виводу .....	66
4.4 Маркування упаковки .....	67
4.5 Характеристики перемикачів.....	68
4.6 Вимоги до програмного забезпечення.....	68
<b>РОЗДІЛ 5 ЕТАПИ СТВОРЕННЯ СЕНСОРНОЇ МЕРЕЖІ ДЛЯ МОНІТОРИНГУ ДАНИХ.....</b>	<b>70</b>
5.1 Теоретична оцінка характеристик бездротової мережі.....	74
5.2 Структура прийомопередавача .....	77
5.3 Структура енергоспоживання прийомопередавача.....	79
5.4 Функціонування приймача в режимі ретрансляції .....	80
5.5 Енергоспоживання приймача в режимі ретрансляції в відсутності енергозбереження.....	81
5.6 Енергоспоживання приймача в режимі ретрансляції з енергозбереженням .....	87
5.7 Енергоспоживання приймача в режимі передачі даних і в «сплячому» режимі .....	94
5.8 Експериментальне дослідження енергоспоживання приймач .....	95
5.9 Експериментальне дослідження бездротової сенсорної мережі .....	95
<b>ВИСНОВКИ .....</b>	<b>99</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....</b>	<b>100</b>

## ДОДАТКИ

Додаток 1. Презентація

Додаток 2. Лістинг розробленої програмного коду

Додаток 3. Публікації

Додаток 4. Довідка про впровадження

## **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ**

БВВ – Блоки вводу-виводу

БСМ – бездротова сенсорна мережа

ІМС – Інтегральна мікросхема

ІС – Інтегральна схема

КРСМ – кросплатформена розподілена мережа

ЛЕ – Логічні елементи

ЛО – локальний об'єкт

ЛОМ – локальний об'єкт мережі

ЛРО - локально-розподілений об'єкт

МК – мікроконтролер

НВІС – надвелика інтегральна схема

ОЗП – оперативний запам'ятовувальний пристрій

ПЗ – програмне забезпечення

ПЛІС – Програмовано логічна інтегральна схема

РСМ - розподілена сенсорна мережа

САПР – Система автоматизованого проектування

ТРО - Територіально розподілений об'єкт

ЦАП – цифро-аналоговий передавач

ASIC – Application-Specific Integrated Circuit

CPLD – Complex Programmable Logic Device

FPGA – Field Programmable Gate Array

HDL – Hardware Description Language

LUT – Look-Up Table

RAM – Random Access Memory

SRAM – static random access memory

SPLD – Simple Programmable Logic Device

WBAN – wireless body area network



## ВСТУП

З ростом продуктивності обчислювальних засобів зростає і обчислювальна складність алгоритмів, а також вимоги по швидкодії з боку користувача. Розробка цифрових пристроїв, що мають високу швидкодію і складність, неможлива без застосування спеціалізованих систем автоматизованого проектування (САПР), систем моделювання і макетування. Комплексні вимоги до готового виробу, такі як низька вартість, малий час розробки і модернізації, гранична швидкодія, змушують розробників використовувати універсальні засоби і методи проектування. Зазначеним якостям в даний час відповідають технології проектування пристроїв на основі мікропроцесорів, а також технологія проектування з використанням програмованих логічних інтегральних схем (ПЛІС).

Найважливішим завданням сучасних підприємств є надання послуг або виробництво продукції з максимальною ефективністю і мінімальною собівартістю. В рамках підприємства протікає безліч технологічних процесів, що вимагають безперервного контролю для забезпечення дотримання встановлених норм і промислових стандартів. З огляду на масштаби сучасних підприємств і високий ступінь складності процесів, актуальною стає завдання всеосяжного автоматизованого моніторингу з використанням сучасних засобів інформаційних технологій.

ПЛІС є порожній чіп. "На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС визначається не на фабриці при виготовленні, а задається за допомогою додаткового програмування (проектування) за допомогою спеціальних засобів: програматорів та програмного забезпечення" [1]. Ця технологія дозволяє розробити мікросхему з власною архітектурою.

ПЛІС - високоінтегровані гнучкі універсальні логічні пристрої з внутрішньосистемним перепрограмуванням. Збільшення попиту на пристрої з можливістю зміни їх внутрішньої структури в реальному часі з швидкою перебудовою виконуваних функцій розширило сфери застосування ПЛІС. Доцільність її застосування обумовлюється необхідністю або розробити

оригінальну апаратуру, або замінити звичайну інтегральну схему (ІС), що дозволяє зменшити розміри пристрою, знизити споживану потужність і збільшити надійність. Найбільш ефективним є використання ПЛІС в виробках, які потребують нестандартних схемотехнік.

Бездротова сенсорна мережа (БСМ) - це бездротова система, що представляє собою розподілену, самоорганізовану і стійку до відмов окремих елементів мережу мініатюрних обчислювальних пристроїв з автономним джерелом живлення. Вузли такої системи транслюють повідомлення один через одного, забезпечуючи значну площу покриття мережею при малій потужності передавача.

В магістерській дисертації розглянута багаторівнева гетерогенна мережа, керована вузлами вищого рівня. Для цієї архітектури треба створити формальну модель, обрати відповідне архітектурне рішення та програмну реалізацію. Роль, яку відіграє модель у процесі проектування динамічної сенсорної мережі, обумовлює підвищені вимоги до інструменту моделювання та відповідності моделі реальній мережі.

Для створених моделей необхідно провести аналіз коректності роботи, виявити додаткові обмеження, внесені у модель способом її програмної реалізації, проаналізувати коректність їх застосування та ступінь відповідності реальній мережі.

Необхідність забезпечення тривалої автономної роботи бездротових портативних пристроїв є проблемою, яка визначає життєздатність, область застосування і конкурентноздатність бездротових технологій і пристроїв на їх основі. Для вирішення питань енергоспоживання та енергозбереження [1-2] можуть використовуватися різні підходи: конструктивні рішення, що дозволяють використовувати малоспоживаючі компоненти, організація протоколів зв'язку, що знижують навантаження на пристрої, введення сплячих режимів і ін.

## РОЗДІЛ 1

### ЗАГАЛЬНІ ВІДОМОСТІ ПРО СЕНСОРНІ МЕРЕЖІ

Можна поділити сенсорні мережі по двох основних типах: гомогенні (є однорідними) і гетерогенні (є неоднорідними) сенсорні мережі.

Для однорідних бездротових сенсорних мереж характеристики будуть однаковими, якщо аналізувати енергію батареї і функціональні можливості. У сенсорних мережах можуть бути два, три або більше видів вузлів, при різних енергетичних і функціональних можливостях. Існують три поширених типів неоднорідностей по ресурсам для сенсорних вузлів: обчислювальні неоднорідності, неоднорідності по мережевим можливостям і неоднорідності енергії. Обчислювальна неоднорідність пов'язана з тим, що означає, що в гетерогенному вузлі є більш потужний процесор і більший розмір пам'яті, якщо порівнювати з типовим сенсорним вузлом. На базі того, що застосовують більш потужні обчислювальні ресурси вузли можуть дати комплексну обробку за даними і здійснення їх довгострокового зберігання. Неоднорідність в мережеских можливостях означає, що в гетерогенному вузлі існує більш висока пропускна здатність і використовуються більш потужні радіопередавачі, ніж в типових вузлах. Неоднорідність енергії означає, що в гетерогенному сенсорному вузлі є можливості або постійного електроживлення від мереж, або можливості по заміні батарей або акумуляторів протягом терміну служби сенсорних вузлів. Є переваги розміщення різнорідних вузлів в сенсорних мережах: Зменшується затримка при передачі даних, збільшується тривалість життєвого циклу мереж, підвищується надійність в передачі даних.

#### 1.1 Бездротові сенсорні мережі

Бездротові сенсорні мережі (Wireless Sensor Networks) є особливим класом мереж, що складаються з великої кількості розподілених в просторі мініатюрних пристроїв (вузлів) з автономним живленням, об'єднаних між собою в єдину мережу за допомогою технології бездротового з'єднання.

Впровадження в даний пристрій вимірювальних функцій або функцій взаємодії зі стороннім обладнанням робить можливим використання

бездротових сенсорних мереж для вирішення завдання розподіленого моніторингу і віддаленого управління різними об'єктами і фізичними процесами.

Бездротові сенсорні мережі мають деякі характерні властивості: однорангова топологія; самоорганізація; масштабованість; безпечна передача даних.

Бездротові сенсорні мережі, як правило, побудовані на базі тимчасової фізичної топології, що призводить до відсутності центрального пристрою, який контролює всі інші пристрої в мережі. В даному випадку кожен вузол є незалежним і виконує одночасно роль і «клієнта» (інших пристроїв), і «сервера» (для інших пристроїв) в рамках мережі. На додаток до цього, така фізична топологія забезпечує відмовостійкість мережі.

Під самоорганізацією мається на увазі здатність бездротової сенсорної мережі, будучи цілісною і живою системою, реагувати на зміни навколишнього середовища шляхом зміни своїх внутрішніх процесів, поведінки і потоків даних. До подібного роду процесів належать:

- маршрутизація трафіку між вузлами мережі (в тому числі, маршрутизація трафіку в мережі з мобільними вузлами, маршрутизація трафіку в мережах зі змінною доступністю вузлів);

- кластеризація.

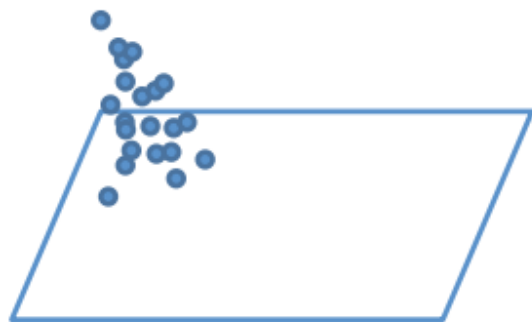
Відповідно до вимоги масштабованості бездротова сенсорна мережа повинна мати можливість легко підключити до себе нові вузли.

Забезпечення безпечної передачі даних в бездротових сенсорних мережах є однією з найважливіших задач при організації систем моніторингу і включає:

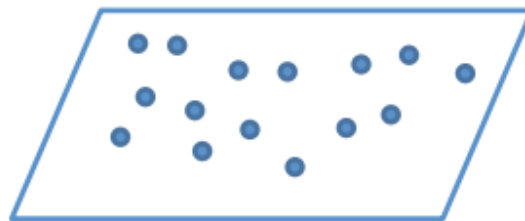
- шифрування трафіку між вузлами;
- аутентифікацію вузлів мережі і трафіку;
- стійкість мережі до атак на відмову в обслуговуванні.

## **1.2 Методи маршрутизації в бездротових сенсорних мережах.**

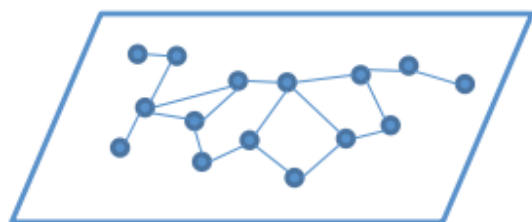
Як правило, БСМ складається з декількох вузлів сенсорної мережі та шлюзу для підключення до Інтернету. Загальний процес розгортання БСМ виглядає наступним чином (див. рис 1.1): по-перше, вузли мереж датчиків транслюють свій стан в оточенні та отримують статус від інших вузлів, щоб виявити один одного. По-друге, вузли мереж датчика організовані у зв'язану мережу відповідно до певної топології (лінійна, зірка, дерево, сітка тощо). Нарешті, відповідні шляхи обчислюються на побудовану мережу для передачі даних зондування. Потужність мережевих вузлів датчиків зазвичай забезпечується батареями, тому відстань передачі вузлів WSN коротка. Відстань передачі може бути від 800 до 1000 метрів у відкритому зовнішньому середовищі з лінією зору. Це буде різко знижуватися у випадку критого оточення навколишнього середовища до приблизно декількох метрів. Для того, щоб розширити охоплення мережі, мережа датчиків використовує режим багатоканальної передачі. Тобто вузли мережевого сенсора є як передавачем, так і приймачем. Перший сенсорний мережевий вузол, вихідний вузол, надсилає дані до сусіднього вузла для передачі даних шлюзу. Сусідній вузол пересилає дані до одного з сусідніх вузлів, які знаходяться на шляху до шлюзу. Пересилання повторюється, доки дані не надходять до шлюзу, місця призначення. Протоколи та деякі методи реалізації БСМ можуть бути адаптовані до зрілої архітектури та технологій бездротових та дротових комп'ютерних мереж. Проте особливостями БСМ є самоорганізація, самореалізація, обмежена енергія вузлів та нестабільні канали передачі [3].



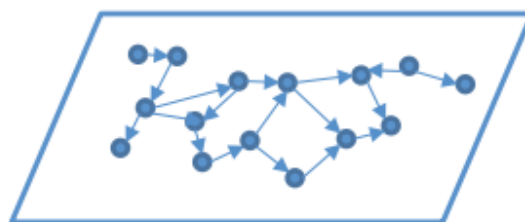
**Розміщення сенсорів**



**Пробудження і виявлення**



**Під'єднання до мережі**



**Маршрутизація та передача**

Рисунок 1.1 - Організація та передача БСМ

### **1.2.1 Самоорганізована та надійна мережева технологія**

Позиції вузлів БСМ є випадковими, а вузли можна переміщувати, захищати та втручатися. Топологія мережевих сіток має великі переваги в плавності та надійності в порівнянні з іншими топологіями мережі. Самоорганізований підхід управління мережевими вузлами може значно підвищити надійність мережі, що призведе до інтелектуальної технології мережевих мереж, як показано на рисунку 1.2. У спеціальній технології мережевих технологій сітка, вузол спершу контролює сусідні вузли та вимірює потужність сигналу, а потім вибирає відповідний сусідній вузол для синхронізації часу та надсилає запит на об'єднання. Потім сусідній вузол передає запит на шлюз. Шлюз отримує запит і призначає мережеві ресурси для вузла. На основі сіткової мережі, вузли мережевого датчика можуть бути призначені двома або більше шляхами передачі для підвищення надійності мережі. Синхронізований сіточний протокол (ССП) [4] мережі пилу може підтримувати самоорганізуючу мережу та підтримувати мережу, що складається з сто вузлів [5].

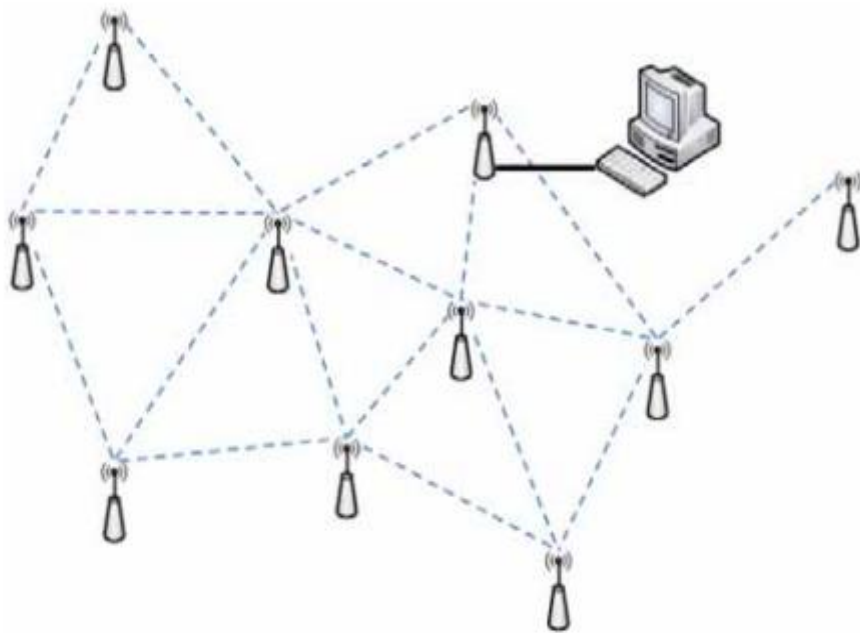


Рисунок 1.2 - Самоорганізована мережа

### 1.2.2 Низька вартість IP-технологій з'єднання

У дизайні ранніх сенсорних мереж зазвичай використовуються внутрішні адреси для керування вузлами мереж датчиків. Довжина адреси була відносно короткою і придатною для реалізації в вузлах малої потужності вбудованих датчиків. Проте, метод внутрішнього керування адресами несумісний з IP-методом Інтернету, що збільшує складність взаємодії між вузлами сенсорної мережі та традиційними вузлами IP-мережі. Тому існує потреба у вирішенні проблеми з'єднання БСМ та IP-мережі. Традиційні адреси IPv4 поступово вичерпані, а нова технологія IPv6 має величезний адресний ресурс, який підходить для широкого кола розгортання сенсорної мережі. В результаті 6LoWPAN малопотужних бездротових технологій на основі IPv6 виникла [6]. 6LoWPAN взагалі реалізував спрощений протокол IPv6 над протоколом IEEE 802.15.4. Зсув заголовка та перезавантаження фрагментації пакетів здійснюється шляхом додавання шарів адаптації між шаром IP і каналом зв'язку, що є надійним способом досягнення адаптивного протоколу між мережею IPv6 та мережею датчиків, як показано на малюнку 1.2. Продукти сенсорної мережі компанії Sensinode на базі NanoStack [7] та компанії TI на базі CC-6LoWPAN [8] використовують технологію 6LoPAN для

забезпечення масштабованості, безперервної та надійної взаємодії між сенсорною мережею та IP-мережею.

### **1.2.3 Технологія автоматичного регулювання потоку**

Однією з відмінностей між БСМ і традиційними провідними мережами є нестабільність бездротового зв'язку. У БСМ зв'язку між вузлами чутливі до перешкод і закупорювання, що призводить до невдалої передачі сигналу.

Традиційна мережа - це стабільна дротова мережа, дані якої будуть втрачені лише через затори. Принцип управління потоком полягає в тому, що відправник даних налаштовує трафік відправки відповідно до ситуації втрати передачі даних. Коли відбувається втрата даних, відправник зменшує швидкість передачі. І коли дані не втрачені, відправник збільшує швидкість передачі. Такі механізми управління потоком більше не підходять для БСМ [8], оскільки втрата даних в сенсорних мережах в основному викликана заторів, перешкодами та оклюзіями. Чисте зменшення швидкості передачі не може вирішити проблему, але лише знижує продуктивність мережі. Для вирішення проблеми деградації продуктивності мережі в умовах нестабільної передачі пропонується адаптивний потік керування. Адаптивний поточний контроль перевіряє причину втрати пакетів та регулює потоки передачі. Тим часом, згідно з якістю посилення та кількістю помилок передачі, найкраща швидкість передачі даних для передачі даних між вузлами визначається як пріоритет, щоб забезпечити хорошу стабільність мережі при розгляді відстані та пропускної здатності передачі.

### **1.3 Маршрутизація в бездротових сенсорних мережах.**

Маршрутизація пакетів є однією з найбільш актуальних завдань в області БСМ, і ефективність її рішення істотно впливає на енергоспоживання, надійність, пропускну здатність і інші показники якості обслуговування мережі.

Бездротові сенсорні мережі мають ряд ключових відмінностей від інших типів бездротових мереж передачі інформації, таких як локальні бездротові



мережі та мобільні епізодичні мережі. Перелічимо основні особливості БСМ і вимоги, які до них пред'являються:

- великі масштаби мережі - кількість вузлів в мережі може досягати десятків тисяч;
- обмежені ресурси вузлів - ємність автономного джерела живлення, обчислювальна потужність і пам'ять мікропроцесора, пропускна здатність каналів зв'язку дуже обмежені;
- розміщення вузлів - розташування вузлів в просторі може бути випадковим або детермінованим, їх розподіл по площі (обсягу) покриття мережі може бути як рівномірним, так і нерівномірним;
- складна топологія - в загальному випадку мережа має багатокоміркову топологію, всі або більшість вузлів нерухомі;
- види трафіку - в залежності від розв'язуваної прикладної задачі потрібна підтримка типів трафіку «багато-до-одного», «один-до-багатьох» і «багато-до-багатьох»;
- модель генерації повідомлень - вузли можуть ініціювати передачу пакетів за часом (періодично), за подією або за запитом від зовнішнього споживача інформації, а також можливі різні комбінації перерахованих варіантів;
- різномірність вузлів і з'єднань - вузли можуть мати різні енергетичні ресурси, обсяги пам'яті, а бездротові канали відрізняються швидкістю передачі даних, надійністю, дальністю зв'язку;
- самоорганізація і відмовостійкість - вузли повинні самостійно налаштовуватися на етапі розгортання системи, а також в процесі роботи адаптуватися до умов навколишнього простору та поточного режиму експлуатації;
- масштабованість - кількість службового мережевого трафіку і необхідний обсяг пам'яті вузлів повинні мінімально або зовсім не залежати від загального розміру мережі;

- час життя мережі - потрібно забезпечити тривалий термін експлуатації мережі при автономних джерелах живлення вузлів.

На основі описаних характеристик БСМ можна навести кілька причин того, що застосування в БСМ традиційних алгоритмів маршрутизації недоцільно або неможливо зовсім.

По-перше, через велику кількість вузлів в мережі неможливо організувати глобальну систему адресації, оскільки накладні витрати на підтримку адрес в актуальному стані будуть занадто великі. Отже, традиційні техніки маршрутизації на основі IP-адрес незастосовні в БСМ Крім того, великі масштаби і епізодичний характер розгортання БСМ роблять неприпустимими на практиці будь-які дії по настройці мережевої взаємодії вузлів.

По-друге, багато програм БСМ (наприклад, розподілені системи моніторингу) засновані на типі трафіку «багато-до-одного»: джерелами даних можуть бути всі вузли мережі, а споживачем є одна базова станція (в деяких випадках кілька), тому всі потоки трафіку спрямовані від кінцевих вузлів до, базової станції, при цьому для економії ресурсів мережі може виконуватися агрегація даних у міру їх передачі до споживача. Тип трафіку «один-до-багатьох» може використовуватися, наприклад, для передачі команд управління від базової станції до кінцевих вузлів або при оновленні програмно-апаратних засобів вузлів. Найбільш загальним варіантом є тип трафіку «багато-до-багатьох», при якому джерелами і споживачами інформації можуть бути будь-які вузли мережі. На такий тип трафіку спираються, зокрема, багато системи «внутрішньомережевого» зберігання і обробки даних. Отже, механізми маршрутизації для БСМ повинні забезпечувати ефективну передачу пакетів при різних типах трафіку.

По-третє, при розробці методів маршрутизації для БСМ необхідно враховувати істотні обмеження ресурсів вузлів (енергія, пам'ять і обчислювальна потужність вузлів, пропускна здатність каналів зв'язку) і

використовувати їх максимально ефективно, задовольняючи при цьому специфічним вимогам, що пред'являються до такого роду мереж.

По-четверте, в більшості додатків БСМ після розгортання положення вузлів в просторі не змінюється, за винятком, можливо, декількох мобільних вузлів, в той час як в традиційних бездротових мережах вузли можуть часто і вільно переміщатися. Проте, топологія сенсорної мережі є динамічною через відмови вузлів з різних причин (фізичне пошкодження, відключення для технічного обслуговування, виснаження джерела енергії) і коливань якості з'єднань між ними (зміна характеристик трас проходження сигналів, включення і відключення джерел перешкод), тому алгоритм маршрутизації повинен володіти стійкістю до цих змін і адаптуватися до умов експлуатації при мінімальних витратах на передачу службового трафіку.

По-п'яте, БСМ є вузькоспеціалізованими системами, тому механізм маршрутизації повинен забезпечувати гнучке налаштування параметрів якості обслуговування під особливості конкретного прикладного завдання для досягнення максимальної ефективності. Як правило, першорядним вимогою є мінімальне енергоспоживання вузлів, проте в деяких завданнях важливі такі характеристики як латентність, рівномірність розподілу трафіку.

З огляду на сказане, перерахуємо основні завдання, які необхідно вирішити для створення ефективного методу маршрутизації в БСМ, і характерні для БСМ властивості, які слід при цьому брати до уваги:

Розміщення вузлів. Спосіб розгортання мережі залежить від прикладної задачі. У деяких з них вузли вручну поміщаються в заздалегідь відомі місця, при цьому можлива наявність фіксованої стаціонарної інфраструктури мережі. В інших додатках розміщення вузлів по території виконується випадковим чином (наприклад, викиданням з літального апарату), тому отримана в результаті розподіл вузлів може бути нерівномірним. Отже, оскільки для зв'язку між вузлами використовуються малопотужні низько споживають радіочастотні прийом передавачі, необхідна «естафетна» маршрутизація пакетів для покриття всієї площі розгортання мережі, при цьому в загальному

випадку не можна розраховувати на можливість організації стаціонарної інфраструктури мережі.

Енергоспоживання. Сенсорні вузли працюють від автономних джерел живлення (наприклад, батареї або акумулятори) з кінцевою ємністю, при цьому від БСМ в цілому потрібно тривалий термін експлуатації без заміни джерел живлення вузлів. Можливо також отримання електроенергії від навколишнього середовища, але генератори, що перетворюють, наприклад, сонячну або механічну (зокрема, вібрацію) енергію в електричну, що не виробляють достатньої потужності для того, щоб не враховувати питання мінімізації енергоспоживання, а також дорогі і далеко не завжди доступні до застосування. Як правило, серед елементів вузла приймач має найбільшу потужність споживання, тому головний спосіб зменшити середнє енергоспоживання вузла полягає в мінімізації активності в радіоканалі (передача і прийом даних, прослуховування каналу). З огляду на, що кожен вузол є не тільки джерелом або одержувачем інформації, а й у разі необхідності проміжним ретранслятором пакетів, оптимізація обсягів і напрямів потоків трафіку є важливим завданням рівня маршрутизації.

Модель генерації повідомлень. У додатках БСМ можуть зустрічатися різні варіанти моделі генерації повідомлень: за часом (періодично), за подією, за запитом або комбінації перерахованого. Періодична передача повідомлень використовується в системах збору даних і моніторингу, тобто вузли періодично опитують підключені до них зовнішні датчики і передають оброблені дані базової станції. Згідно з моделями генерації повідомлень за подією і за запитом дані передаються тільки в якості реакції на будь-яку подію (наприклад, сигнал з датчика перевищив заданий поріг) або у відповідь на запит базової станції. Отже, здатність методу маршрутизації підтримувати різні моделі генерації повідомлень буде істотно впливати на параметри мережі.

Різноманітність вузлів і з'єднань. У більшості робіт по БСМ передбачається, що мережа складається з однакових вузлів, але в багатьох

прикладних задачах може знадобитися використання вузлів з різними апаратними можливостями (обчислювальна потужність, ємність джерела енергії) в залежності від їх функціонального призначення. Крім того, бездротові з'єднання між вузлами можуть відрізнятися пропускнуою спроможністю, дальністю, надійністю, тому в процесі маршрутизації необхідно враховувати різноманітність складових елементів мережі.

**Відмовостійкість.** Передбачається, що умови експлуатації БСМ можуть бути жорсткими, тому часто будуть мати місце вихід з ладу вузлів і порушення з'єднань між ними. Отже, для забезпечення високої надійності всієї системи в цілому метод маршрутизації повинен автоматично формувати нові маршрути в обхід відмовили вузлів, витрачаючи на переконфігурацію якомога менші ресурси.

**Масштабованість.** БСМ можуть включати до декількох десятків тисяч вузлів, тому кількість службового трафіку і обсяг пам'яті. Займаючись службовою інформацією для виявлення і підтримки маршрутів в актуальному стані, повинні, мінімально або зовсім не залежати від загальної кількості вузлів в мережі.

**Якість обслуговування.** У більшості завдань першорядним вимогою є мінімізація енергоспоживання для збільшення часу життя мережі, але можливі й інші критерії якості обслуговування: затримка передачі повідомлення (латентність), пропускна здатність, рівномірність розподілу трафіку і т. д.

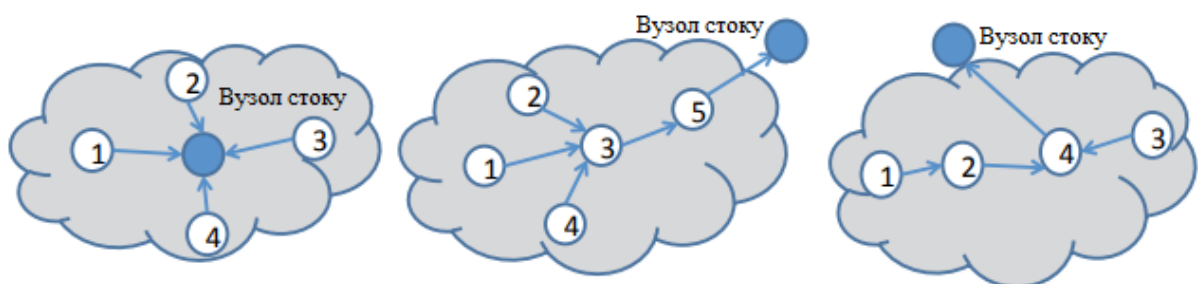
#### **1.4 Агрегація даних**

У середовищах мережевого сенсора з обмеженим енергоспоживанням це непридатне в багатьох аспектах енергії акумулятора, можливостях обробки, ємності та пропускнуої здатності зв'язку, для кожного вузла для передачі даних у вузол раковини. Це пояснюється тим, що в сенсорних мережах з високим рівнем охоплення інформація, що повідомляється сусідніми вузлами, має деяку ступінь надмірності, що призведе до передачі даних окремо в кожному вузлі при одночасному споживанні пропускнуої здатності та енергії всієї мережі датчиків, що скорочує час життя мережі.

Щоб уникнути вищезазначених проблем, введені методи агрегації даних. Агрегація даних - це процес інтеграції декількох копій інформації в одну копію, що є ефективним і здатним задовольняти потреби користувачів середніх вузлів датчиків.

Впровадження агрегації даних сприяє як економії енергії, так і отримання точної інформації. Енергія, споживана при передачі даних набагато більша, ніж при обробці даних в мережах сенсорів. Тому за допомогою локальної обчислювальної та вузлової здатності вузла виконуються операції агрегування даних для видалення великої кількості надлишкової інформації, щоб мінімізувати обсяг передачі та заощаджувати енергію. У складному мережевому середовищі важко забезпечити точність інформації, отриманої лише шляхом збору декількох зразків даних з розподілених вузлів датчиків. В результаті моніторингу даних того самого об'єкту потрібна спільна робота декількох датчиків, що ефективно підвищує точність та надійність отриманої інформації.

Продуктивність протоколу агрегації даних тісно пов'язана з топологією мережі. Після цього можна проаналізувати деякі агрегаційні дані протоколів відповідно до топологій мережі зірок, дерева та мережі, як показано на рисунку 1.3 [9].



а) Зоряна топологія БСМ б) Дерево топологія БСМ в) Ланцюг топологія БСМ

Рисунок 1.3 - Три види топологій БСМ: зірка, дерево, ланцюжок

Технологія агрегації даних може заощадити енергію та покращити точність інформації, а також приносити жертви в інших сферах. З одного боку, в процесі передачі даних, пошук агрегованих вузлів, операцій агрегації даних

та очікування прибуття інших даних, ймовірно, збільшиться у середній затримці мережі. З іншого боку, порівняно з традиційними мережами, сенсорні мережі мають більшу швидкість передачі даних. Агрегація даних може суттєво зменшити надмірність даних, але випадково втрачати більше інформації, що зменшує надійність мережі датчиків [9].

### **1.5 Забезпечення надійної передачі даних в бездротових сенсорних мережах.**

При вирішенні завдань, пов'язаних із забезпеченням безпечної передачі даних в бездротових сенсорних мережах, слід враховувати наступні фактори:

1) обмежені обчислювальні ресурси вузла мережі (обсяг оперативної пам'яті мікроконтролера становить від 1 до 4 КБ, тактова частота - 20 МГц), а в деяких випадках і обмежені ресурси самої мережі (середня швидкість передачі між вузлами може становити до 100 байт / с);

2) відсутність центрального (головного) вузла - відсутність єдиної точки прийняття рішень - з огляду на те, що в основі бездротових сенсорних мереж лежить однорангова фізична топологія;

3) невеликий розмір пакета даних, що передається по бездротовому каналу між вузлами мережі, який, як правило, становить 10-100 байт.

Таким чином, традиційні криптографічні алгоритми не можуть бути використані для забезпечення безпеки даних в бездротовій сенсорній мережі з огляду на наведених вище причин. В даний час є актуальним вирішення наступних завдань:

1) розробка алгоритмів шифрування, орієнтованих на застосування в пристроях з обмеженими обчислювальними ресурсами;

2) розробка алгоритмів шифрування, які можуть працювати з різними довжинами блоків, враховуючи при цьому енергоспоживання пристрою і ймовірності колізій;

3) розробка алгоритмів обміну ключовою інформацією.

В результаті проведених авторами досліджень був розроблений блоковий симетричний алгоритм шифрування даних для застосування в

вузлах в бездротовій сенсорній мережі. Схема шифрування алгоритму заснована на використанні мережі Фейстеля і дискретної хаотичного відображення. Подібна комбінація структурних елементів дозволяє балансувати між розміром шифруючого блоку, криптографічної стійкістю вихідної послідовності і тривалістю процедури шифрування.

Розроблений алгоритм реалізований на платформі 8-бітних мікроконтролерів Atmel і вбудований в базову функціональність бездротового приймача типу ППС-40А. Робочі характеристики зазначеної реалізації алгоритму: використання пам'яті для зберігання програм - 762 байт, використання оперативної пам'яті - 241 байт, максимальна швидкість обробки даних - 88,2 Кбіт / с (при тактовій частоті мікроконтролера 20 МГц).

### 1.6 Система моніторингу на базі бездротової сенсорної мережі.

Бездротова сенсорна мережа, в свою чергу, є частиною системи моніторингу - апаратно-програмного комплексу, в який крім бездротової мережі також входить сервер і робоче місце користувача (рис. 1.4).



Рисунок 1.4 - Структурна схема моніторингу

Бездротова сенсорна мережа складається з деякого числа приймачів, здійснює збір інформації від підключеного обладнання або датчиків на певній території і передає цю інформацію за допомогою комплексного каналу зв'язку в програмне забезпечення, встановлене на сервері. Сервер з встановленим програмним забезпеченням здійснює прийом, обробку та зберігання



інформації, що надходить від підключеного обладнання або датчиків, а також відправку на них, при необхідності, по команді користувача, що управляють. Робоче місце користувача являє собою персональний комп'ютер (або будь-який інший мобільний пристрій, наприклад, планшет), з якого здійснюється доступ до програмного забезпечення для аналізу інформації і, при необхідності, управління роботою підключеного обладнання. Програмне забезпечення здійснює зберігання і обробку надходить з приймачів інформації, і надає інтерактивний інтерфейс взаємодії користувача (споживача інформації або адміністратора системи) з системою. При цьому виконуються наступні функції:

- облік підключеного (активного) обладнання або датчиків;
- надання звітів;
- настройка контрольованих порогових рівнів параметрів, перевищення яких спричинить автоматичне повідомлення зацікавлених осіб за допомогою електронної пошти або СМС-повідомлень;
- експорт інформації в необхідний формат для подальшої обробки в інших системах (додатках);
- настройка параметрів роботи системи (періодичність збору даних, режими роботи і т.д.).

Доступ до програмного забезпечення для користувачів здійснюється з робочого місця за допомогою веб-браузера (наприклад, Internet Explorer). При такому використанні відпадає необхідність установки додаткового програмного забезпечення на комп'ютери користувачів, що значно прискорює процес первісного розгортання системи, а також значно спрощує впровадження нових функцій і виправлення помилок.

## **1.7 Функціональні призначення і сфери застосування**

Можна виділити наступні можливі функціональні призначення апаратно програмного комплексу:

- управління промисловим і технологічним обладнанням (пристрої, установки, апарати, машини);

- знімання інформації і параметрів з пристроїв вимірювання і контролю (лічильники електроенергії, тепла, води);

- вимірювання параметрів навколишнього середовища (температура, вологість, освітленість, вібрації, зовнішні фактори);

- контроль території, що охороняється.

До перспективних сфер застосування подібного роду систем слід віднести:

- сільське господарство (тепличне господарство, овочесховища);

- промисловість і виробництво (виробничі підприємства, склади);

- житлово-комунальне господарство;

- контроль території, що охороняється;

- лікувально-профілактичні установи;

- інші сфери, де необхідний оперативний моніторинг і управління обладнанням.

## **РОЗДІЛ 2**

### **ЗАГАЛЬНІ ВІДОМОСТІ ПРО ПЛІС ТА ОБГРУНТУВАННЯ ЙОГО ВИБОРУ**

"Програмовані логічні інтегральні схеми - ПЛІС є одними з найперспективніших елементів цифрової схемотехніки. ПЛІС являє собою кристал, на якому розташована велика кількість простих логічних елементів. Спочатку ці елементи не з'єднані між собою. З'єднання елементів (перетворення розрізаних елементів в електричну схему) здійснюється за допомогою електронних ключів, розташованих в цьому ж кристалі. Електронні ключі управляються спеціальною пам'яттю, в осередки якої заноситься код конфігурації цифрової схеми. Таким чином, записавши в пам'ять ПЛІС певні коди, можна зібрати цифровий пристрій будь-якого ступеня складності (це залежить від кількості елементів на кристалі і параметрів ПЛІС). На відміну від мікропроцесорів, в ПЛІС можна організувати алгоритми цифрової обробки на апаратному рівні. При цьому швидкодія цифрової обробки різко зростає" [10].

#### **2.1 Передумови до розвитку ПЛІС. Переваги ПЛІС**

Загальною тенденцією розвитку елементної бази цифрової схемотехніки, починаючи з появи перших інтегральних мікросхем і до теперішнього часу, є безперервне підвищення числа логічних елементів (ЛЕ), що розміщуються на кристалі, з одночасним зниженням питомої вартості одного елемента. Збільшення числа ЛЕ безперервно відкриває можливості створення все більш складних цифрових пристроїв, що розміщуються на одному кристалі. До основних (далеко не повних) позитивних результатів цієї тенденції можна віднести:

- постійне розширення функціональних можливостей і поліпшення споживчих властивостей кінцевих виробів;
- зменшення габаритів і споживаної потужності;
- підвищення надійності [11].

Відображенням цієї тенденції в складі елементної бази цифрової схемотехніки з'явився перехід від інтегральних мікросхем (ІМС) малої і середньої ступені інтеграції до великих (ВІС) і надвеликих (НВІС) інтегральних мікросхем. Одним з найбільш революційних результатів розвитку мікроелектроніки є можливість створення перших мікропроцесорів, що дало потужний поштовх до впровадження цифрових технологій обробки інформації в усіх сферах людської діяльності.

Однак далеко не всі практичні завдання цифрової схемотехніки можуть бути вирішені тільки з використанням одних мікропроцесорів. Це обумовлено органічно властивою всім мікропроцесорам особливістю, пов'язаною з тим, що рішення будь-якої задачі мікропроцесором завжди складається з послідовності кроків кінцевої тривалості, в той час як для вирішення багатьох завдань (в тому числі пов'язаних і з забезпеченням роботи самих мікропроцесорів) потрібні пристрої з мінімальною затримкою виконання логічних функцій. Існує три основних способи задовольнити цю потребу:

- використання наборів стандартної дискретної цифрової логіки загального застосування;
- використання замовних НВІС;
- використання програмованих логічних інтегральних схем (ПЛІС).

Набори дискретної цифрової логіки різних серій досить тривалий час були основною елементної бази для розробки цифрових пристроїв. До складу таких наборів входить велика кількість окремих мікросхем, призначених як для виконання базових логічних функцій (І, АБО, НЕ), так і для виконання функцій типових цифрових пристроїв, таких як тригери, регістри, лічильники, мультиплексори, дешифратори і т.д., що дає можливість використовувати їх для розробки більш складних функціонально закінчених цифрових пристроїв[12].

Основний недолік дискретної логіки полягає в тому, що для розробки кінцевих виробів зазвичай потрібна велика кількість мікросхем. Наслідком цього є велика кількість зовнішніх з'єднань, складність конструкції і великі

габарити друкованих плат, велика довжина сполучних провідників, складність побудови пристроїв з високою тактовою частотою, низька надійність. Для зменшення числа мікросхем при проектуванні мікропроцесорних систем був розроблений ряд периферійних ВІС, що представляють собою спеціалізовані цифрові пристрої, призначені для виконання деяких типових функцій в складі мікропроцесорних систем, такі як контролери динамічних операційних запам'ятовуючих пристроїв (ОЗП), контролери переривань, контролери прямого доступу в пам'ять, контролери шин і т.д. Однак навіть застосування периферійних ВІС не дозволяє повністю подолати основні недоліки дискретної цифрової логіки.

Найбільш кардинально проблема габаритів, швидкодії, спрощення конструкцій друкованих плат і забезпечення надійності вирішується шляхом розробки і виготовлення замовних НВІС. На жаль, цей шлях економічно виправданий тільки при великосерійному виробництві однотипних кінцевих виробів. Крім того, при використанні замовних НВІС можлива модифікація виробів вимагає істотних додаткових матеріальних і тимчасових витрат.

У той же час на практиці досить часто виникають потреби в розробці оригінальних цифрових пристроїв і виробів, не розрахованих на багатосерійне виробництво, для яких розробка замовних НВІС не прийнятна або з економічних причин, або за термінами виконання. Протягом тривалого часу єдиним шляхом для вирішення таких завдань було використання інтегральних мікросхем дискретної логіки і периферійних ВІС, адже можливості перших поколінь простих ПЛІС були вельми обмежені, а ціна складних ПЛІС дуже високою, крім того, були певні складнощі і з проектуванням цифрових пристроїв на ПЛІС.

В останні роки відбувся різкий прорив як в технології виготовлення ПЛІС, так і в розробці інструментальних засобів, призначених для проектування цифрових пристроїв на ПЛІС і випуску готових виробів. Технологічний прорив характеризується різким збільшенням числа еквівалентних логічних вентилів, що розміщуються на одному кристалі,

підвищенням робочої частоти з різким одночасним зниження як питомої, так і абсолютної вартості. Програмована логіка по темпам розвитку випереджають інші напрямки цифрової електроніки (універсальних мікропроцесорів, сигнальних процесорів, мікроконтролерів і замовних БІС) [12].

## **2.2 Порівняльний аналіз ПЛІС і НВІС**

Техніко-економічні показники сучасних ПЛІС досягли такого рівня, який забезпечує при випуску партій виробів до декількох сотень нижчу вартість кінцевих виробів, ніж застосування замовних НВІС. Очевидно, що найближчим часом цей поріг буде підвищуватися, забезпечуючи економічну доцільність застосування ПЛІС і в великосерійному виробництві. Крім суто економічних передумов, цьому сприяє ряд додаткових переваг сучасних ПЛІС, які полягають в тому, що, зберігаючи всі переваги однокристального рішення, властивого рекомендованим НВІС, пристрої на основі ПЛІС можуть дуже швидко і з малими витратами піддаватися модернізації. Завдяки наявності в багатьох типах ПЛІС вбудованих систем програмування і конфігурації, що дозволяють перепрограмувати їх прямо на місці без використання зовнішніх програматорів, пристрої на ПЛІС можуть модернізуватися навіть перебуваючи в постійній експлуатації у замовника. З цією метою провідні виробники програмованої логіки включають в засоби проектування пристроїв на ПЛІС підтримку оновлення версії ПЛІС через інтернет. Крім того, терміни проектування і випуску готової продукції на ПЛІС незрівнянно коротші, ніж розробка і виробництво замовлених НВІС, що в умовах динамічно мінливого ринку іноді може мати вирішальне значення.

Важливою обставиною є те, що для випуску готової продукції не потрібно того складного і дорогого технологічного обладнання, яке потрібно для виробництва замовних НВІС. Остання обставина відкриває середнім і навіть дрібним фірмам шлях на ринок виробів сучасної електронної техніки, який був раніше доступний тільки гігантам електронної промисловості (див. Таблицю 2.1) [13].

Таблиця 2.1 - Відносні рейтинги ефективності застосування НВІС і ПЛІС

Показники	Замовні НВІС	ПЛІС
Швидкодія	вуже висока	висока
Щільність упаковки	дуже висока	середня
Вартість при одиночному виробництві	дуже висока	середня
Вартість при серійному виробництві	Низька	висока
Час розробки	Великий	низький
Час виробництва	Великий	середній
Можливість модернізації	«жорстка» ектура	в неробочому режимі
Ризик виробника	дуже високий	високий
Ступінь автоматизації процесу проектування	середній	високий

Як ми можемо спостерігати, що НВІС має перевагу в вартості але ця перевага замовлених НВІС проявляється тільки при дуже великому обсязі виробництва ідентичних мікросхем.

Чималу роль в розширенні сфер застосування ПЛІС, скорочення часу і зниження трудовитрат на проектування зіграли і значні успіхи в створенні інструментальних засобів для розробки і випуску кінцевих виробів на ПЛІС, основу яких складають спеціальні пакети програм, що забезпечують весь виробничий цикл зі створення цифрових пристроїв на ПЛІС, від розробки схем до випуску готових виробів.

## **2.3 Класифікація пограмованих логічних інтегральних схем (ПЛІС)**

Різноманітність існуючих типів ПЛІС не дозволяє вибрати єдиний критерій для їх всеосяжної класифікації. Тому доцільно виділити набір класифікаційних критеріїв, що забезпечують можливість систематизації інформації про характеристики та особливості ПЛІС. До основних критеріїв класифікації ПЛІС слід віднести:

- ступінь інтеграції;
- класифікація по архітектурі;
- організація внутрішньої структури і структури матриці з'єднань;
- наявність внутрішньої RAM-пам'яті.

### **2.3.1 Класифікація ПЛІС за ступенем інтеграції**

Ступінь інтеграції ПЛІС характеризується логічною ємністю. Логічна ємність вимірюється числом логічних вентилів і визначає можливість ПЛІС забезпечити реалізацію цифрового пристрою заданої складності. Як правило, в якості базового логічного вентиля приймають елемент 2І-НЕ. Відповідно до обраного критерію ПЛІС поділяються на чотири підгрупи:

- Низького ступеня інтеграції - до 1500 логічних вентилів;
- Середнього ступеня інтеграції - від 1500 до 15000;
- Високої інтеграції - від 15000 до 150000;
- Надвисокої інтеграції - більше 150000 [14].

## **2.4 Особливості проектування цифрових пристроїв на базі ПЛІС**

Процес проектування та виготовлення цифрових пристроїв на основі стандартних інтегральних схем комбінаційного і послідовного типів малої і середньої ступені інтеграції, є вузькоспеціалізованим. Тобто в процесі створення пристроїв цифрової техніки, фахівці виконують певні індивідуальні функції. Відповідно процес уже складається із певних стадій, таких як підготовка виробництва, проектування, відпрацювання технологій та інше. Процес створення цифрових пристроїв на основі спеціалізованих інтегральних



схем високого ступеня інтеграції, характеризуються такою ж складністю. Тому використання ПЛІС дозволяє істотно скоротити обсяг цих стадій, фактично звести їх тільки до етапів проектування за допомогою електронної обчислювальної машини.

Істотною перевагою ПЛІС є їх універсальність і можливість швидкого програмування під виконання функцій практично будь-якого цифрового пристрою. ПЛІС являє собою напівфабрикат, на основі якого розробник, що володіє персональним комп'ютером, нескладними і відносно недорогими апаратними засобами програмування і спеціальним програмним забезпеченням, що називається системою автоматизованого проектування (САПР), має можливість проектування цифрового пристрою в рекордно короткі терміни. При цьому весь цикл проектування і програмування доступний всього одній людині - проектувальнику цифрових пристроїв на базі ПЛІС.

Всі сучасні САПР ПЛІС працюють під управлінням операційних систем сімейства Windows і використовують всі переваги її графічного інтерфейсу. Фірми-виробники інтегральних схем ПЛІС здійснюють зазвичай підтримку своєї продукції шляхом розробки і поширення таких САПР. Проектування ПЛІС в системі САПР передбачає виконання таких етапів:

- розробка структурної формули проектованого цифрового пристрою, її мінімізація і адаптація під тип і можливості використовуваної ПЛІС;
- створення нового файлу проекту, призначення сімейства ПЛІС для проекту;
- введення структурної формули або схеми проектованого пристрою за допомогою спеціальних програм - редакторів;
- компіляція введеної структурної формули або схеми, тобто отримання бітової послідовності завантажується в ПЛІС програми;
- часовий аналіз затримок в окремих елементах і перевірка працездатності всієї структури цифрового пристрою шляхом моделювання за допомогою спеціальних програм - симуляторів;

- програмування мікросхеми ПЛІС за допомогою спеціальних апаратних засобів – програматорів [15].

Проектувальнику в системі САПР доступні зазвичай великі програмні бібліотеки, що задають функції типових інтегральних схем транзитивно-транзисторної логіки і комплементарної структури металу-оксиду-напівпровідника. Це дозволяє синтезувати схему цифрового пристрою не тільки в певному базисі, тобто з елементарних функцій І, АБО, НЕ, але і використовуючи готовий програмний аналог існуючих реальних мікросхем. Широкі можливості при проектуванні представляє глобальна мережа Internet, через яку здійснюється поширення спеціально розроблених бібліотек компонентів, які не є в стандартних версіях САПР. Для завдання схеми проєктованого на базі ПЛІС цифрового пристрою широко використовуються мови опису апаратних засобів, в загальному випадку звані HDL (Hardware Description Language). За допомогою операторів такої мови можна задати типи комбінаційних або послідовних пристроїв, сформулювати вхідні впливи на них і зв'язки між ними, визначити константи і змінні в проєкті і т.д.

Серед лідерів у виробництві ПЛІС і САПР для них на сьогоднішній день можна виділити такі фірми, як Xilinx, Altera, Achronix. На їх частку припадає до 80% від загального виробництва ПЛІС, швидкодія яких на сьогоднішній день досягла близько сотень МГц, ємність програмованих матриць І - кількох мільйонів елементів і знизилась їх вартість. Все це, плюс найкоротші терміни проектування, висока надійність, яка пояснюється високим ступенем інтеграції окремих елементів в єдиному кристалі напівпровідника і, як наслідок, мінімальна кількість зовнішніх між'єднань, робить ПЛІС кращою елементної базою в порівнянні зі стандартними логічними мікросхемами комбінаційного і послідовних типів.

## РОЗДІЛ 3

### ЗАГАЛЬНИЙ ОГЛЯД АРХІТЕКТУРИ FPGA

#### 3.1 Логіка об'єднання структури

Взявши за приклад типовий настільний комп'ютер, то усередині є пристрої зберігання та зв'язку (жорсткі диски та мережеві карти), пам'ять і, звичайно ж, центральний процесор або процесор, де відбувається більша частина обчислень. FPGA відіграє подібну роль у телекомунікаційній конфігурації обчислювальної платформи. У дуже загальних термінах, у FPGA існує лише два типи ресурсів: логіка та взаємозв'язок. Логіка полягає в тому, що ми робимо такі речі, як арифметика,  $1 + 1 = 2$  і логічні функції,  $\text{if (ready) } x = 1 \text{ else } x = 0$ . З'єднання - це те, як ми отримуємо дані (наприклад, результати попередніх обчислень) від одного вузла обчислення до іншого. Найбільшу увагу потрібно приділити логіці.

З цифрової логіки та архітектури комп'ютерної архітектури відомо, що будь-яке обчислення може бути представлено як логічне рівняння (і в деяких випадках як логічне рівняння, де вхідні дані залежать від минулих результатів, FPGA також можуть зберігати стан). У свою чергу, будь-яке булеве рівняння можна виразити, як таблицю істинності. З цих смиренних початків можна побудувати складні структури, які можуть виконувати арифметику, такі як змінні і множники, а також структури, що приймають рішення, які можуть оцінювати умовні висловлювання. Поєднуючи їх, можна описати складні алгоритми, просто використовуючи таблиці істинності.

З цього основного спостереження цифрової логіки впливає таблиця істинності як обчислювальне серце FPGA. Більш конкретно, один елемент пристрою, який легко реалізує таблицю істинності, - це таблиця пошуку або LUT. З точки зору реалізації схеми, LUT може бути сформований з N:1 (N-to-one) мультиплексора та N-розрядної пам'яті. LUT просто перераховує таблицю істинності. Тому, використовуючи LUT, FPGA забезпечує загальну реалізацію довільної цифрової логіки. На малюнку 3.1 показана типова таблиця пошуку

N-входів, які знаходяться в сьгоднішніх FPGA. Фактично, майже всі комерційні FPGA розташовані на LUT, як їх основний будівельний блок.

LUT може обчислювати будь-яку функцію N входів, просто програмуючи таблицю пошуку таблицею істинності функції, що реалізовується. Як показано на рисунку 3.1, реалізовуючи функцію XOR 3-входа з 3-вхідним LUT (часто називається 3-LUT), призначається значення для пам'яті таблиці пошуку, така, що шаблон виділених бітів вибирає правильну відповідь рядка. Таким чином, кожен «рядок» дає результат 0, за винятком чотирьох випадків, коли XOR з трьох вибраних ліній дає 1.

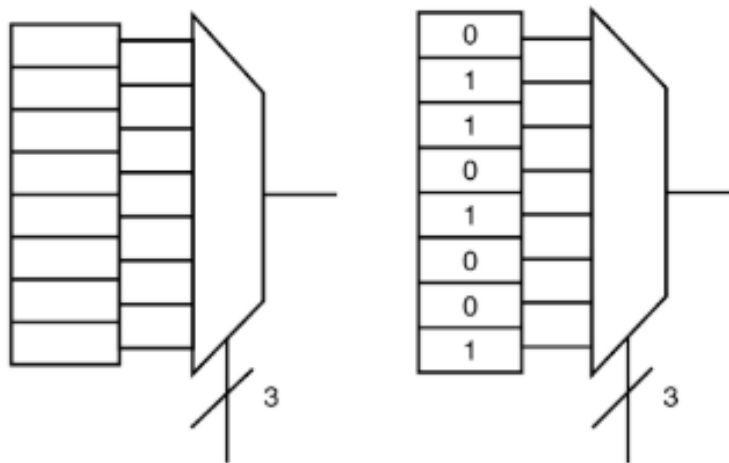


Рисунок 3.1 - Схема 3-LUT (а) та відповідний символ 3-LUT та таблиця істинності (б) для логічного XOR

Звичайно, більш складні функції та функції з більшою кількістю входів можуть бути реалізовані шляхом об'єднання декількох таблиць довідників. Наприклад, можна організувати єдиний 3-LUT у дисковості 8x1, але значення пошуку перепрограмоване, 8x1 оперативної пам'яті. Але основний блок, таблиця пошуку залишається незмінною.

Незважаючи на те, що LUT більш-менш обрано як найменший обчислювальний вузол у комерційно доступних FPGA, величина таблиць пошуку в кожному логічному блоці була широко досліджена [19]. З одного боку, більші таблиці пошуку дозволяють виконати більш складну логіку за логічним блоком, тим самим зменшуючи затримку проводки між блоками,

оскільки буде потрібно менше блоків. З іншого боку, менші таблиці пошуку можуть вимагати, щоб дизайн використовував більшу кількість логічних блоків, тим самим збільшуючи затримку електропроводки між блоками, зменшуючи час затримки за одиницю блоків.

Поточні емпіричні дослідження показали, що структура 4-LUT робить найкращий компроміс між зоною та затримкою для широкого кола тестових схем. Звичайно, оскільки обчислення FPGA еволюціонує на більш широкі арені і з цього приводу, Xilinx випустила FPGA на основі SRAM на Virtex-5 з архітектурою 6-LUT.

Питання про кількість LUT на один логічний блок також було обстежено [20], з емпіричними даними, що дозволяє групувати більше одного 4-LUT в єдиний логічний блок, що може покращити область і затриматись. Багато сучасних комерційних FPGA включають в себе в кожному логічному блоці кілька 4-LUT, щоб скористатися цим спостереженням.

Вивчення, як розміру LUT, так і кількості LUT на блок починають вирішувати питання про обчислювальні грануляції у ПЛІС. На одному кінці спектра досить проста структура невеликої довідкової таблиці (наприклад, 2-LUT) являє собою дрібнозернисту обчислювальну здатність. На іншому кінці, грубозернисті, можна передбачити більші обчислювальні блоки, такі як повні 8-бітні арифметичні логічні одиниці, більш характерні для процесорів. Як і у разі пошуку розміру таблиці, тонкозернисті блоки можуть бути більш продуманими при маніпуляціях та арифметиці бітового рівня, але вимагають комбінування для реалізації більших частин логіки. Контраст, що з більш грубозернистими блоками, які можуть бути більш оптимальними для обчислень, орієнтованих на дані, які працюють зі стандартними розмірами "слів" (8/16/32 біт), але є марними при здійсненні дуже простих логічних операцій. Практика в галузі промисловості полягає в тому, щоб досягти рівноваги гранулярності, використовуючи досить дрібнозернисті архітектури 4-LUT та збільшуючи їх з більш грубозернистими гетерогенними елементами, такими як мультиплікатори.

Обравши логічний блок, цього не достатньо, щоб реалізувати всі функції, які ми хочемо, у FPGA. За допомогою лише LUT, для FPGA не існує способу зберігання будь-якого стану сприйняття, тому забороняється здійснювати будь-яку форму послідовної логіки. Для усунення цієї ситуації потрібно додати простий однобітний елемент зберігання в базовий логічний блок у вигляді D-тригера.

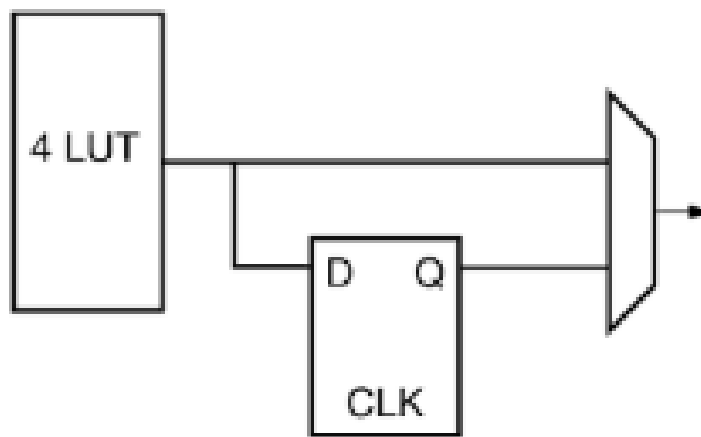


Рисунок 3.2 - Простий логічний блок для пошуку таблиці

Тепер логічний блок виглядає так, як на рисунку 3.2. Вихідний мультиплексор вибирає результат як за допомогою функції, що створюється таблицею пошуку, так і збереженим бітком у D-тригері.

### 3.2 Повернення і взаємодія

За допомогою LUT та D тригера можна визначити те, що зазвичай називається логічним блоком або функціональним блоком FPGA.

Поточні FPGA реалізують те, що часто називають архітектурою островних стилів. Як показано на рисунку 3.3, ця конструкція має логічні блоки, обклеєні в двовимірному масиві та взаємопов'язані певним чином. Логічні блоки утворюють острови та "плавають" у морському вузлу.

За допомогою цієї архітектури масивів обчислення виконуються просторово в структурі FPGA. Великі обчислення розбиті на шматки розміром 4-LUT і відображені в масиві фізичних логічних блоків. З'єднання

налаштоване на маршрутизацію сигналів між логічними блоками відповідно. За наявності достатніх логічних блоків можна зробити на FPGA будь-які розрахунки.

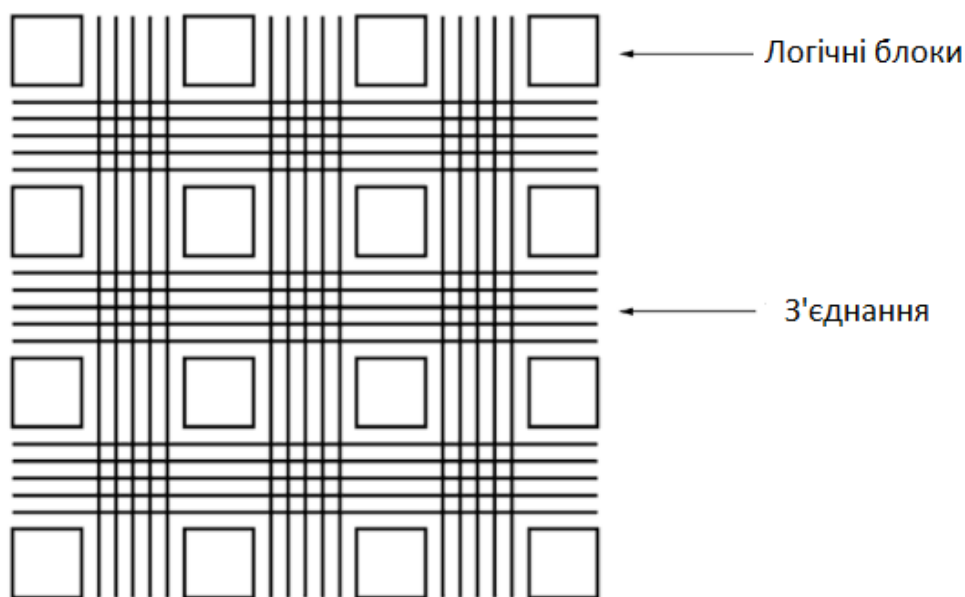


Рисунок 3.3 - Архітектура FPGA

На рисунку 1.3 показана структурна схема не є репрезентативною для будь-яких структур, що використовуються у реальних FPGA, але є більш мультиплікаційним заповнювачем.

1) Найближче сусідство Найближчі сусідні зв'язки так само просто, як це звучить. Дивлячись на 2x2 масив логічних блоків на рисунку 1.4, можна побачити, що єдиними потребами в цьому сусідстві є введення та виведення з'єднань у кожному напрямку: північ, південь, схід і захід. Це дозволяє кожному логічному блоку спілкуватися безпосередньо з кожним з його найближчих сусідів. Рисунок 3.4 - це приклад однієї з найпростіших архітектур маршрутизації. Хоча це може здатися майже виродженням, він використовувався в деяких комерційних FPGA. Звичайно, хоча це і є простим рішенням, ця структура страждає від серйозних проблем із затримкою та з'єднаннями. Якщо, замість масиву 2x2 - масив 1024x1024. з'єднання з найближчим сусідом лише затримує лінійну затримку з відстанню, оскільки сигнал має пройти через багато блоків(і багатьох вимикачів) для досягнення кінцевого призначення.

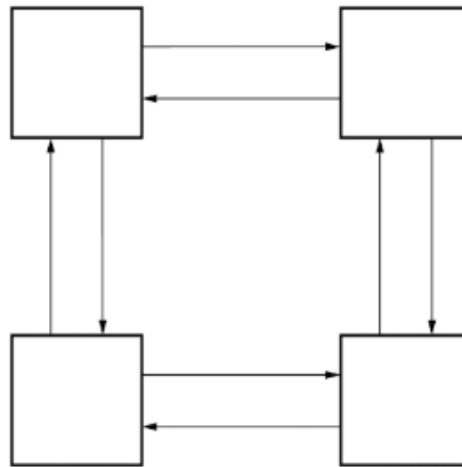


Рисунок 3.4 – Найближчі зв'язки між сусідами

З точки зору підключення, не маючи можливості обійти логічні блоки в структурі маршрутизації, потрібні всі маршрути, які потребують більше одного скачка перетинаючи логічний блок. За наявності лише однієї двонаправленої пари в кожному напрямку, це обмежує кількість сигналів логічного блоку, які можуть перетинатися. Сигнали, що проходять, не повинні перекривати сигнали, які активно споживаються та виробляються. Через ці обмеження структура найближчих сусідів рідко використовується винятково, але вона майже завжди доступна в поточних FPGA, часто доповнюється деякими прикладами.

2) Сегментовані. Відходячи від архітектури чистої логіки блоку, найбільш сучасні архітектури FPGA виглядають не так, як на рисунку 3.3 подібніше, як на рисунку 3.5. На рисунку 3.5 ми вводимо блок підключення та перемикач. Тут структура маршрутизації більш загальна та сітчаста. Логічний блок отримує доступ до сусідніх ресурсів зв'язку через блок з'єднання, який з'єднує вхідні й вихідні термінали логічного блоку з маршрутизацією ресурсів через програмовані комутатори або мультиплектори. Блок з'єднання (деталізований на рисунку 3.6) дозволяє вводити вхідні дані логічного блоку та виводи на будь-які горизонтальні та вертикальні доріжки, що збільшує гнучкість маршрутизації.



Блок перемикача з'являється, коли горизонтальна та вертикальна маршрутизація відстежується, як показано на рисунку 3.7. У загальному сенсі це просто матриця програмованих комутаторів, що дозволяє сигналу на треку підключитися до іншої доріжки. Залежно від конструкції блоку перемикача, це з'єднання може бути, наприклад, повертати кут у будь-якому напрямку або продовжувати прямо. Конструкція блоків перемикання являє собою цілу область досліджень сама по собі і має багато різноманітних конструкцій, які демонструють різний ступінь взаємодії та ефективності [21-23].

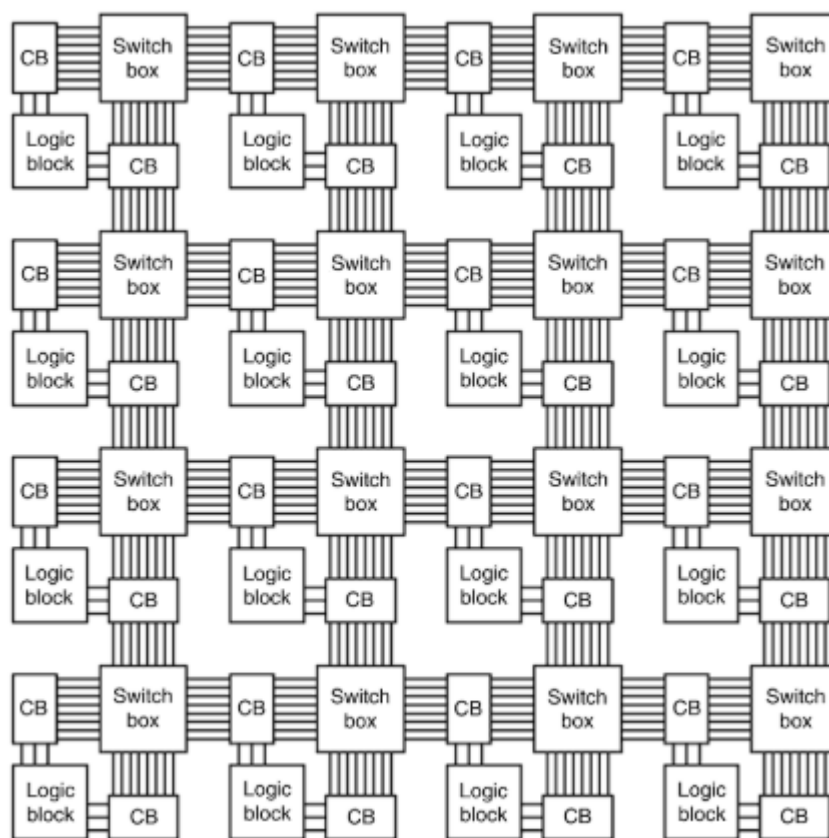


Рисунок 3.5 - Архітектура FPGA з підключеними блоками та перемикачами для підтримки більш складних структур маршрутизації

Маршрутизація найближчих сусідів може бути виконана, хоча і через пару блоків з'єднання та блок перемикання. Однак для сигнали, які потребують подорожей на великі відстані, окремі сегменти можуть бути разом переключені в блок перемикача, щоб разом з'єднати віддалені логічні блоки. Це спосіб емуляції довгих шляхів сигналу, які можуть охоплювати довільні

відстані. Результатом є довгий дрот, який фактично містить коротші "сегменти".

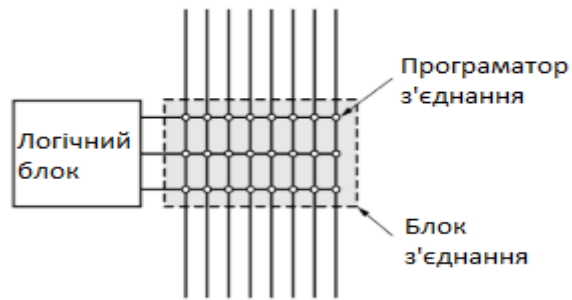


Рисунок 3.6 Деталі блоку з'єднання

Тільки ця архітектура взаємо з'єднання не покращує радикальних характеристик затримки сусідньої структури. Проте введення блоків з'єднання та перемикачів відокремлює інтерконнект від логіки, що дозволяє здійснювати маршрутизацію на великі відстані без споживання ресурсів логічного блоку.

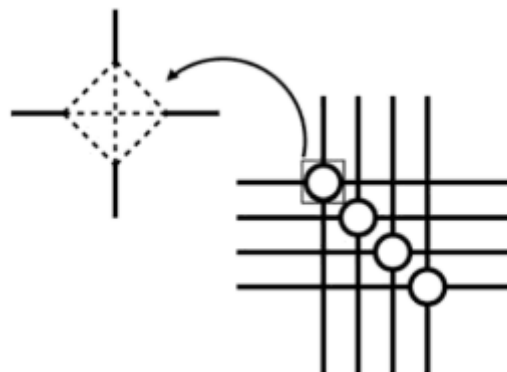


Рисунок 3.7 - ієрархічна маршрутизація

Щоб покращити структуру, потрібно впровадити довжину дроту. Наприклад, розгляньмо провід, який охоплює один логічний блок як  $L_1$ . У деяких сегментованих архітектурах маршрутизації можуть існувати довші У цьому блоці локальна, маршрутизація найближчих сусідів - це все, що є доступним. У свою чергу утворюється  $2 \times 2$  скупчення цих кластерів, що охоплює 16 логічних блоків. На цьому рівні ієрархії, довші дроти на межі менших,  $2 \times 2$  кластерів, з'єднують кожен кластер із чотирьох логічних блоків з іншими кластерами у групі вищого рівня. Це повторюється на більш високих рівнях ієрархії, з більшими кластерами та довгими провідниками.

Забезпечуючи меншу кількість ресурсів на вищих рівнях ієрархії, ця архітектура взаємозв'язків залишається площею ефективною, зберігаючи деякі довжини дроту для мінімізації затримки сигналів, які потребують перетину великих відстаней.

Як і в сегментованій архітектурі, точками з'єднання, які з'єднують один рівень ієрархії маршрутизації з іншою, можуть бути в будь-якій точці структури міжз'єднання. Нові точки в існуючих блоках перемикачів можуть бути створені або повністю незалежні перемикачів сайтів в іншому місці в з'єднуванні можуть бути створені спеціально для переміщення між рівнями ієрархії. Дроти, що дозволяють сигналам більшої відстані подорожувати більш ефективно.

### **3.3 Розширення логіки FPGA**

За допомогою логічного блоку, подібного до зображеного на рисунку 3.2, викладену в двовимірному масиві з підтримуючою структурою з'єднання, можна реалізувати будь-яку комбінаційну та послідовну логіку. Єдиним обмеженням є область з точки зору кількості доступних логічних блоків.

Сучасні архітектури інтерфейсу FPGA дозріли, щоб включати набагато більше, ніж прості найближчі сусідні з'єднання для підвищення продуктивності для загальної програми. Аналогічно, основні логічні елементи були збільшені, щоб підвищити продуктивність для звичайних операцій, таких як арифметичні функції та зберігання даних.

1) Швидкий ланцюг. Одним з основних дій, які FPGA, ймовірно, буде виконувати, - це доповнення. З основного блоку логіки очевидно, що можна реалізувати повноцінну структуру з двома логічними блоками, принаймні 3-LUT. Один логічний блок конфігурується для обчислення суми, а один налаштовано для обчислення переносу. Каскадні N пар логічних блоків разом дадуть простий N-бітний повний компенсатор

Критичний шлях цього типу додавання відбувається не з розрахунку суми бітів, а швидше від розриву переносу сигнал від бітів нижчого порядку до бітів вищого порядку (див. рис. 3.8). Цей шлях починається з первинних

входів низького порядку, проходить через логічний блок, виходить на межу підключення, до сусіднього логічного блоку тощо. Затримка накопичується на кожній точці перемикавання вздовж шляху.

Один розумний спосіб збільшити швидкість полягає в тому, щоб скоротити ланцюг переносу між додатковими логічними блоками. Це можна досягти шляхом надання виділеного, мінімально зміненого шляху з виходу логічного блоку, який обчислює сигнал перенесення на сусідню пару логічних блоків вищого порядку. Цей ланцюг перенесення не буде потрібно здійснювати на загальній мережі з'єднання. Додаючи мінімальну кількість накладних витрат (проводів), різко прискорюючи операцію додавання.

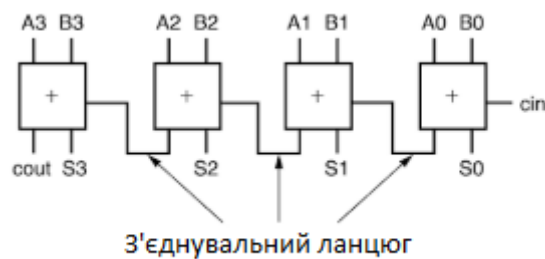


Рисунок 3.8 - Простий 4-бітний повний комплект

Ця функція змушує деякі обмеження на просторову компоновку багатобічного додавання. Щоб зберегти область перемикавання, спеціальний ланцюг переносу не може бути двонаправленою, що також обмежує вертикальну орієнтацію фізичного макета і диктує порядок бітів відносно один одного. Швидкий переносний ланцюг Xilinx XC4000E показаний на рисунку 3.9. Двонаправлені швидкі провідники проводяться по стовпах, а горизонтальні лінії - односпрямованими. Це дозволяє розміщувати великі суматорні структури в зигзагоподібному шаблоні в масиві і все-таки використовувати спеціальний з'єднувальний ланцюг зв'язку.

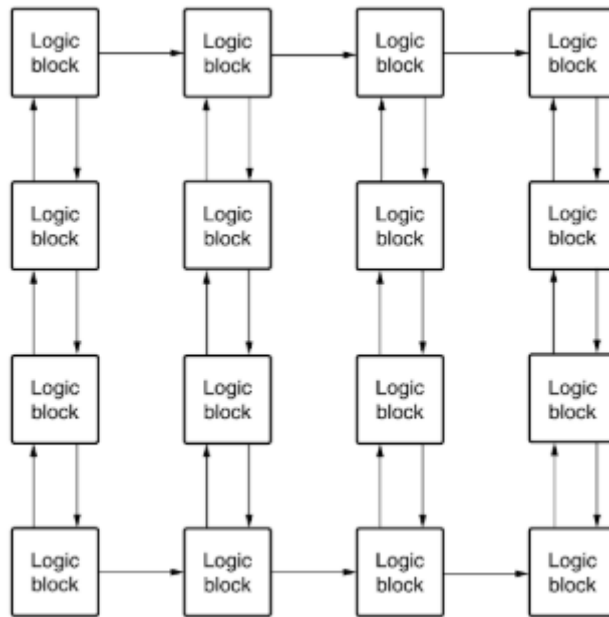


Рисунок 3.9 Xilinx XC4000E ланцюг швидкої передачі

Логіка швидкої передачі носіїв в даний час є звичайною у комерційних FPGA, при цьому фізичні обмеження дизайну в цій точці повністю абстраговані інструментами, наданими виробниками. Змішувачі цієї оптимізації спираються на здатність набору інструментів виявляти додатки в описі схеми дизайнера, а потім використовувати спеціальну логіку. За допомогою сьогоденніх інструментів така оптимізація є майже прозорою для кінцевого користувача.

2) Мультиплікатори. Якщо додавання в алгоритмах є звичайним явищем, то множення, звичайно, не рідкість. Кілька реалізацій доступні, якщо використовувати загальні логічні блоки, щоб побудувати множники. З регіонально-ефективного ітеративного зсуву-накопичувального методу до району-споживаного мультиплікатора масиву, можна використовувати логічні блоки для обчислення додатків або зберігання проміжних значень. Хоча, безумовно, можна реалізувати множення, можна це зробити лише за умови великої відстрочки затримки або великого відстані логічного блоку залежно від нашої реалізації. По суті, логічні блоки не дуже ефективні при виконанні множення.

Замість неефективного використання простого LUT для здійснення множення, можна маршрутизувати значення, які необхідно помножити на фактичні мультиплікатори, реалізовані в кремнії. FPGA обмінюється швидкістю та потужністю для конфігурування у порівнянні з їхніми аналогами ASIC (прикладна інтегральна схема).

Швидкі ланцюжки переносу, мультиплікатори накладають важливі конструктивні міркування та фізичні обмеження, але додаємо ще один варіант для обчислення нашої палітри операцій. Тепер справа тільки з гарним дизайном і хорошими інструментами для ефективного дизайну. Подібно швидким ланцюгам переносу, мультиплікатори є звичайними у сучасних FPGA.

3) ОЗУ. Ще одна область, яка сканує деяку настройку за межами загального FPGA-файлу, знаходиться в області зберігання даних на чіпі. Хоча логічні блоки можуть індивідуально надавати кілька бітів зберігання за допомогою структури таблиці пошуку та, в сукупності, для багатьох бітів, вони далеко не ефективно використовують ресурси FPGA. Подібно ланцюгу швидкого переносу та "жорсткому" мультиплікатору, архітектури FPGA надають своїм користувачам щедрі обсяги оперативної пам'яті, доступ до якої можна отримати з загальної тканини FPGA.

Статичні RAM є надзвичайно малими, і при фізичному розподіленні через FPGA це може бути дуже корисним для багатьох алгоритмів. Угруповавши багато статичних комірок пам'яті в банки пам'яті, дизайнери можуть реалізовувати великі ПЗУ для надзвичайно швидкого пошуку табличних обчислень і операцій з постійним коефіцієнтом, а також великих ОЗ для буферизації, чергування та використання базових скреттів - все з достоїнством простої стратегія синхронізації та швидкість, отримана шляхом уникнення комунікації з зовнішнім інтерфейсом до зовнішньої пам'яті. Сьогоднішні FPGA забезпечують в будь-якому місці від кілобітів до мегабіт виділеної оперативної пам'яті.

4) Блоки процесора. Взаємозв'язок усіх цих блоків, більшість комерційних FPGA тепер пропонують цілі процесори, що виділяються, у FPGA, іноді навіть більше, ніж один. У загальному сенсі, ПЛІС є надзвичайно ефективними при реалізації сировинних обчислювальних трубопроводів, експлуатації нестандартних бітових ширин, а також забезпечують дані та функціональний паралелізм. Включення виділених процесорів визнає той факт, що потоки алгоритмів, які є дуже процедурними і містять високий ступінь розгалуження, не можуть легко прискорити використання FPGA.

Цілі блоки процесора тепер можна знайти у високопродуктивних пристроях FPGA. Ці пристрої мають масштаб 300 MHz PowerPC-пристроїв, завершені без блоків з плаваючою комою. Вони здатні керувати цілою вбудованою операційною системою, а деякі навіть можуть перепрограмувати тканину FPGA навколо них. Ядро процесора не так легко експлуатуються як ланцюжки переносу, множителі та мікросхеми оперативної пам'яті, але вони представляють собою чіткий зсув у напрямку створення FPGA більш "орієнтованої" на платформу.

З традиційним центральним процесором на борту (і, можливо, до чотирьох), один FPGA може служити практично як ціла "система-на-чипі" - святий грааль системних інтеграторів та виробників вбудованих пристроїв. Завдяки стандартним мовам програмування та інструментальним чіпам, доступним для розробників, цілковитий проект може бути реалізований за допомогою одночипного рішення, що значно зменшує витрати та час на ринок.

### **3.4 Конфігурації FPGA**

Одним з визначальних особливостей FPGA є його здатність діяти як "порожнє обладнання" для кінцевого користувача. Забезпечуючи більшу продуктивність, ніж чисте програмне забезпечення для універсальних процесорів, і більшу гнучкість, ніж рішення з підтримкою ASIC з фіксованою функцією, FPGA залежить від конфігурування пристрою.

Кожен налаштовуваний елемент у FPGA вимагає 1 біт пам'яті для підтримки користувальницької конфігурації. Для простих LUT FPGA ці

програмовані місцеположення зазвичай містять вміст логічного блоку та зв'язність тканини маршрутизації. Конфігурація ПЛІС здійснюється шляхом програмування бітів пам'яті, підключених до цих програмованих розташувань відповідно до визначень користувачів. Для таблиць пошуку це означає, що заповнювати його є 1s та 0s. Для тканини маршрутизації програмування дозволяє та вимикає перемикачі уздовж провідних шляхів.

Конфігурацію можна вважати плоским бінарним файлом, вміст якого має карту, біт для біта, до програмованих бітів у FPGA. Цей бітовий потік генерується інструментами, специфічними для постачальників, після завершення розробки апаратного забезпечення. Хоча його точний фонт, як правило, не загально відомо, чим більше FPGA, тим більший бітовий потік стає. Звичайно, існує багато відомих способів зберігання одного біта двійкової інформації.

#### **3.4.1 Статична оперативна пам'ять з довільним доступом (SRAM)**

Як зазначено в попередніх розділах, найбільш широко використовуваним способом зберігання конфігураційної інформації в комерційно доступних FPGA є нестабільна статична оперативна пам'ять, або SRAM. Цей метод був популярним, оскільки забезпечує швидку та нескінченну реконфігурацію за відомими технологіями.

Недоліки SRAM надходять у формі енергоспоживання та волатильності даних. У порівнянні з іншими технологіями, описаними в цьому розділі, клітина SRAM є великою (6-12 транзисторів) і розсіює значну статичну потужність через струм витоку. Інший суттєвий недолік полягає в тому, що SRAM не підтримує вміст без живлення, а це означає, що при потужності FPGA не налаштовується та має бути запрограмований з використанням логіки та сховища позашляхової мережі. Це може бути зроблено за допомогою енергонезалежного запам'ятовуючого пристрою для зберігання конфігурації та мікроконтролера для виконання процедури програмування. Хоча це може здатися тривіальним завданням, воно збільшує кількість компонентів і



складність дизайну і перешкоджає FPGA на базі SRAM бути справді односпецифічним рішенням.

### **3.4.2 Флеш пам'ять**

Хоча менш популярна, ніж SRAM, кілька сімейств пристроїв використовують флеш-пам'ять для зберігання інформації про конфігурацію. Флеш-пам'ять відрізняється від SRAM, тому що вона не є енергонезалежною і може бути записана лише в обмежене число разів.

Незалежність флеш-пам'яті означає, що дані, записані на нього, залишаються при видаленні живлення. На відміну від FPGA на базі SRAM, FPGA залишається налаштованим за допомогою користувальницької логіки, навіть за допомогою циклів живлення, і не вимагає додаткового сховища або апаратного забезпечення для програмування при завантаженні. По суті, FPGA на базі Flash може бути готовий негайно.

Камера флеш-пам'яті також може бути зроблена за меншу кількість транзисторів порівняно з клітиною SRAM. Ця конструкція може призвести до зниження статичного споживання електроенергії, оскільки менше транзисторів, що сприяють потоку витоку.

Недоліки використання флеш-пам'яті для зберігання інформації про конфігурацію FPGA впливають із методів, необхідних для написання на нього. Як згадувалося, флеш-пам'ять має обмежений термін служби циклу запису, і часто має меншу швидкість запису, ніж SRAM. Кількість циклів запису залежить від технології, але, як правило, це сотні тисяч до мільйонів. Крім того, більшість методів запису Flash вимагають більшої напруги в порівнянні з звичайними схемами; вони вимагають додаткової мікросхемної схеми або структур, таких як зарядні насоси on-chip, щоб мати можливість виконувати Flash-запис.

## РОЗДІЛ 4

### ОГЛЯД СІМЕЙСТВА SPARTAN-3AN FPGA

Сімейство Spartan-3AN FPGA поєднує в собі найкращі атрибути провідного, недорогого FPGA з енергонезалежною технологією в широкому діапазоні щільностей. Сім'я поєднує в собі всі особливості сімейства FPGA Spartan-3A плюс провідна технологія вбудованої флеш-пам'яті для конфігурації та енергонезалежного зберігання даних. Сімейство Spartan-3AN FPGA відмінно підходить для обмежених у космосі додатків, таких як блейд-сервери, медичні пристрої, автомобільні розважальні програми, телематика, GPS та інші невеликі споживчі продукти. Поєднання технологій FPGA та Flash дозволяє мінімізувати кількість мікросхем, друкованих плат та загальну кількість, а також підвищити надійність системи. Інтерфейс внутрішньої конфігурації FPGA Spartan-3AN повністю автономний, що підвищує безпеку дизайну. Сім'я підтримує всю зовнішню конфігурацію. FPGA Spartan-3AN - перший у світі невластивий FPGA з MultiBoot, який підтримує два або більше файли конфігурації на одному пристрої, що дозволяє використовувати альтернативні конфігурації для поліпшення полів, режимів тестування або декількох системних конфігурацій.

#### 4.1 Огляд архітектурних особливостей

Архітектура FPGA Spartan-3AN сумісна з FPGA Spartan-3A. Архітектура складається з п'яти фундаментальних програмованих функціональних елементів:

- Налаштовані логічні блоки (CLB) містять гнучкими таблицями пошуку (LUT), які реалізують елементи логіки плюс, що використовуються як тригерів або фіксаторів.

- Блоки вводу / виводу (IOB) керують потоком даних між штифтами вводу / виводу та внутрішньою логікою пристрою. IOB підтримує двонаправлену передачу даних плюс 3-штатну операцію. Вони підтримують різні стандарти сигналу, включаючи кілька високопродуктивних диференціальних стандартів. Реєстри Double Data-Rate (DDR) включені.

- Блок пам'яті забезпечує зберігання даних у формі двокоротких блоків 18-Кбіт.
- Блоки мультиплікатора приймають два 18-бітних двійкових числа в якості входів і обчислюють продукт.
- Блоки цифрового диспетчера годинників (DCM) забезпечують самокалібрувальні, повністю цифрові рішення для розподілу, затримки, множення, розподілу та синхронізації фазових сигналів.

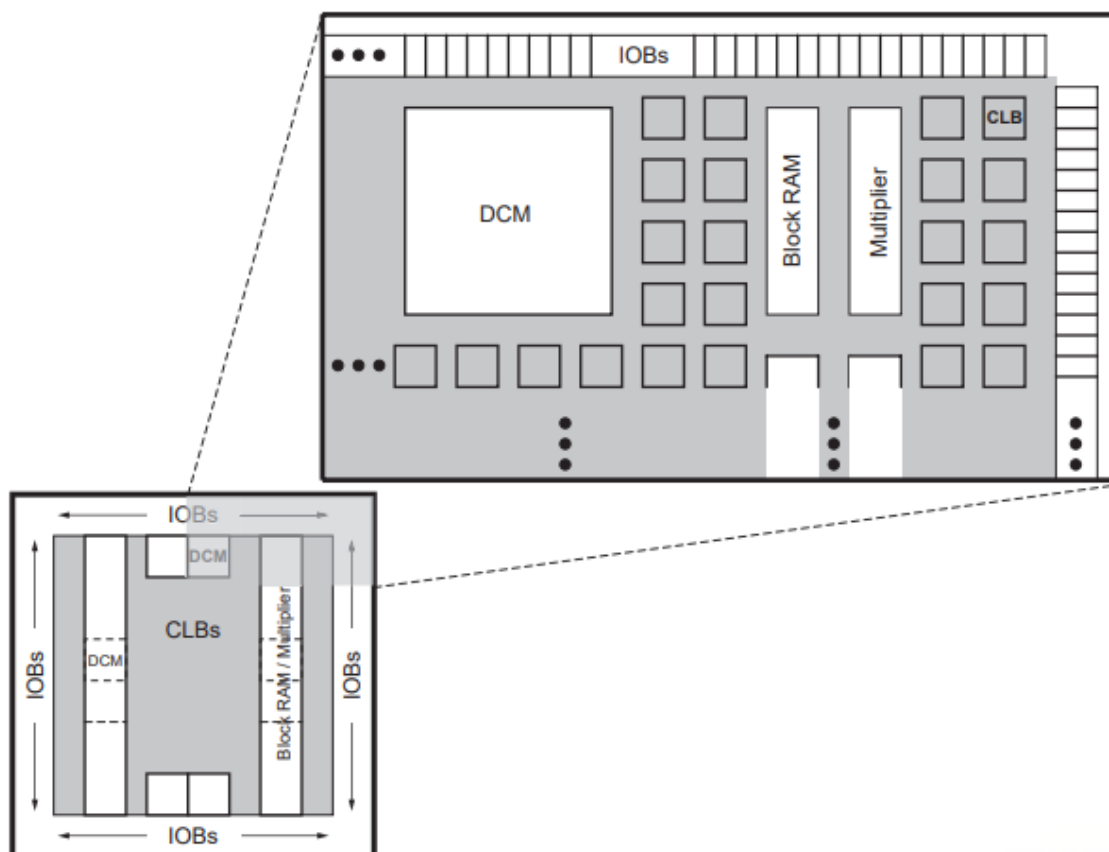


Рисунок 4.1 – Архітектура FPGA Spartan-3AN

Ці елементи організовані так, як показано на малюнку 4.1. Подвійне кільце шахових IOB оточує регулярний масив CLB. Кожен пристрій має два стовпці блокової пам'яті, крім XC3S50AN, який має один стовпець. Кожна оперативна колона складається з декількох 18-Кбітових ОЗУ. Кожний блок пам'яті асоціюється з виділеним мультиплікатором. DCM розташовані в центрі з двома зверху та двома у нижній частині пристрою. XC3S50AN має DCM лише у верхній частині, тоді як XC3S700AN та XC3S1400AN додають два DCM у середині двох стовпців блоку ОЗУ та множників. Spartan-3AN FPGA

має багату мережу слідів, що з'єднують усі п'ять функціональних елементів, передаючи сигнали між ними. Кожен функціональний елемент має пов'язану матрицю перемикача, що дозволяє кілька підключень до маршрутизації.

## 4.2 Конфігурація

Конфігурація Spartan-3AN FPGA запрограмовані шляхом завантаження даних конфігурації у надійні, перепрограміруємі, статичні фіксатори CMOS-конфігурації (CCL), які спільно контролюють всі функціональні елементи та ресурси маршрутизації. Дані конфігурації ПЛІС зберігаються в мікросхемі в енергонезалежній флеш-пам'яті або за межами PROM або деякого іншого енергонезалежного середовища, як на платі, так і поза ним. Після накладання енергії дані конфігурації записуються в FPGA, використовуючи будь-який з семи різних режимів:

- 1) налаштування з внутрішньої спам-пам'яті SPI (рис. 5.2)
  - 1.1 повністю самодостатній
  - 1.2 зменшений простір дошки
  - 1.2 легкий у використанні інтерфейс конфігурації
- 2) майстер серійний від платформи Xilinx Flash PROM
- 3) серійний периферійний інтерфейс (SPI) від зовнішнього стандартного SPI серійного спалаху
- 4) стандартний периферійний інтерфейс байтів (BPI) з стандартної версії x8 або x8 / x16 паралельно NOR Flash
- 5) словник серійний, типово завантажений з процесор
- 6) Slave Parallel, типово завантажений з процесора
- 7) граничний сканування (JTAG), який зазвичай завантажується з процесора або системного тестера.

Функція MultiBoot зберігає кілька файлів конфігурації в мікросхемі, що забезпечує тривалі терміни оновлення. MultiBoot також підтримує декілька системних рішень з єдиною платою, щоб мінімізувати запаси та спростити додавання нових функцій, навіть у полі. Гнучкість підтримується для виконання додаткових конфігурацій MultiBoot за допомогою зовнішньої

конфігурації. Протокол автентифікації пристрою Spartan-3AN запобігає клонування. Клонування дизайну, несанкціоноване перебудовування та повна інженерна технологія зводять вимоги безпеки пристрою до вищих та вищих рівнів. Аутентифікація переміщує захист від захисту бітових потоків до наступного рівня захисту на рівні дизайну, що захищає як дизайн, так і вбудований мікрокод. Алгоритм автентифікації повністю визначено користувачем, реалізований за допомогою логіки FPGA. Кожен продукт, генерація або дизайн може мати інший алгоритм та функціональність для підвищення безпеки.

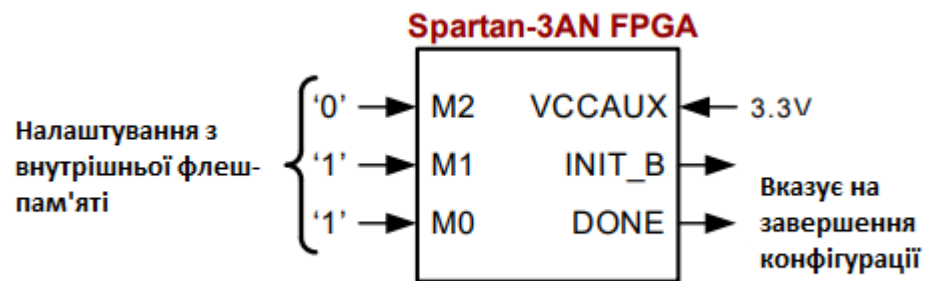


Рисунок 4.2 - Інтерфейс налаштування FPGA Spartan-3AN з внутрішньої спам-пам'яті SPI

### 4.3 Флеш-пам'ять системи

Кожна Spartan-3AN FPGA містить велику інтегровану SPI-серійну флеш-пам'ять, показану в Таблиці 4.1, яка використовується, перш за все, для зберігання біт-потоків конфігурації FPGA. Проте масив Flash-пам'яті достатньо великий, щоб зберігати принаймні дві MultiBoot FPGA-конфігурації бітових потоків або енергонезалежних даних, що вимагаються додатком FPGA, наприклад, процесорні програми MicroBlaze з кодуванням.

Таблиця 4.1 - Пристрій у флеш-пам'яті системи

Номер частини	Загальна флеш-пам'ять (біт)	FPGA Bitstream (біти)	Додаткова флеш-пам'ять (біт)
XC3S50AN	1,330,053	537,893	789,719
XC3S200AN	5,320,212	1,471,237	3,847,282
XC3S400AN	5,320,212	2,320,468	2,997,815
XC3S700AN	10640424	3361147	7,278,923
XC3S1400AN	21,280,849	5,849,014	15,430,694

Після конфігурації дизайн FPGA має повний доступ до вбудованої флеш-пам'яті через внутрішній інтерфейс SPI; контрольна логіка реалізується за допомогою логіки FPGA. Крім того, програма FPGA сама може зберігати енергонезалежні дані або надавати живі, оновлені Flash-системи.

Спа-картка Spartan-3AN у флеш-пам'яті системи підтримує передові послідовні функції Flash.

- 1) невеликий розмір сторінки спрощує енергонезалежне зберігання даних
- 2) випадково доступний, адресний байт
- 3) передавання серійних даних до 66 МГц
- 4) буферів сторінок SRAM
  - 4.1 перегляд даних Flash у програмуванні іншої сторінки Flash
  - 4.2 функції запису на базі EEPROM як
  - 4.3 два буфери в більшості пристроїв, один в XC3S50AN
- 5) видалення сторінок, блоків та секторів
- 6) секторові функції захисту даних та безпеки
  - 6.1 захист сектора: Захист сектора (змінюється)
  - 6.2 закриття сектора: дані сектора не змінюються (постійні)
- 7) 128-байтовий реєстр безпеки
  - 7.1 відокремлено від унікального ідентифікатора ДНК пристрою FPGA

7.2 64 -байт-фабричний запрограмований ідентифікатор, унікальний для вбудованої флеш-пам'яті

7.3 64-байтовий одноразовий програмований, користувальницький програмований поле

8) 100 000 циклів програмування / стирання

9) 20-річне утримання даних

10) комплексна підтримка програмного забезпечення

10.1 програмування прототипу в системі JTAG за допомогою кабелю USB Xilinx платформи та IMPACT ISE 9.1.02i або пізнішого програмного забезпечення

10.2 підтримка програмування продукту за допомогою програмістів BPM Microsystems з відповідним програмним адаптером

10.3 приклади дизайну, що демонструють програмування в системі за допомогою програми FPGA Spartan-3AN

#### **4.3 Можливості вводу-виводу**

Інтерфейс Spartan-3AN FPGA SelectIO підтримує безліч популярних одностадійних та диференційованих стандартів. Деякі користувальницькі введення / виводу є однонаправленими, лише вхідними штифтами.

FPGA Spartan-3AN підтримує наступні стандартні одиночні стандарти:

- 1) 3.3V низьковольтний TTL (LVTTTL)
- 2) КМОП низької напруги (LVCMOS) на 3.3V, 2.5V, 1.8V, 1.5V або 1.2V
- 3) 3.3V PCI на частоті 33 МГц або 66 МГц
- 4) HSTL I, II і III при 1.5 В і 1.8 В, звичайно використовуються в пам'яті додатків
- 5) SSTL I та II на частотах 1.8V, 2.5V і 3.3V, які зазвичай використовуються для роботи з пам'яттю.

FPGA Spartan-3AN підтримує такі диференційні стандарти:

- 1) LVDS, mini-LVDS, RSDS та PPDS на вході-виводі 2,5 В або 3,3 В
- 2) Шина LVDS I / O при 2.5 В • TMDS I / O при 3.3 Вт

#### 4.4 Маркування упаковки

На рисунку 4.3 наведено приклад верхнього розміру для FPGA Spartan-3AN у чотирьохдискових упаковках. На рисунку 4.4 показана верхня марка для FPGA Spartan-3AN у пакунках BGA. Маркування пакунків BGA практично ідентично тим, що використовуються для квадратної упаковки, за винятком того, що маркування обертається щодо індикатора кульки A1.

Комбінації частин "5C" та "4I" зі ступенем швидкості / температури можуть бути подвійні, позначені як "5C / 4I". Пристрої із подвійним знаком можуть бути використані як на пристроях -5C або -4I. Пристрої з однією позначкою гарантуються лише за вказаною шкалою швидкості та температурою.

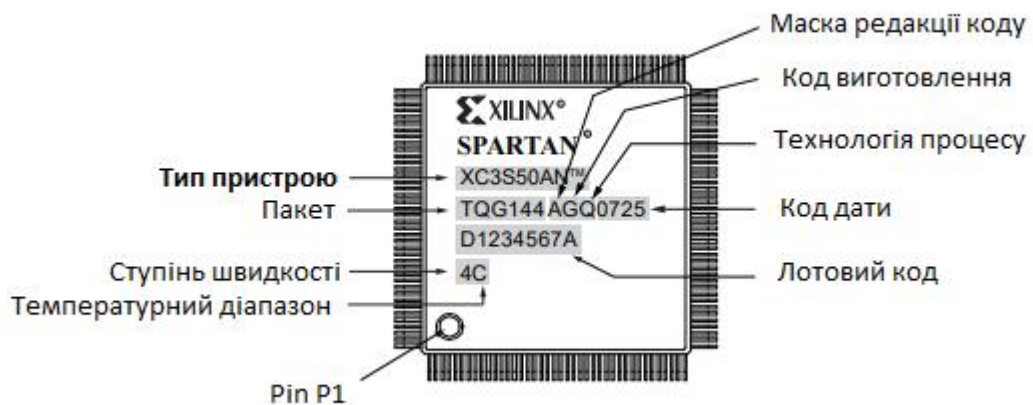


Рисунок 4.3 - Приклад маркування пакету QFP Spartan-3AN

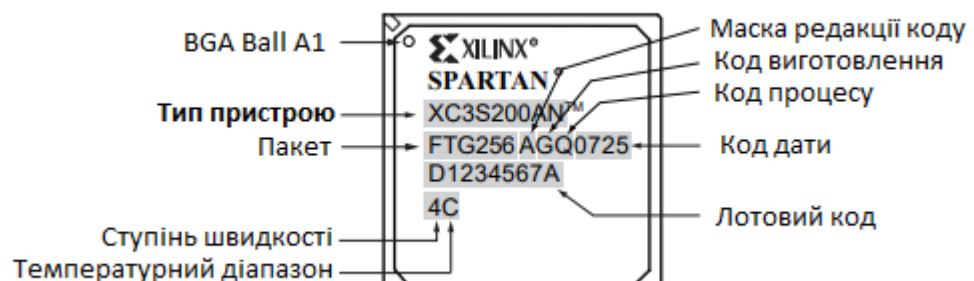


Рисунок 4.4 - Приклад маркування пакета Spartan-3AN BGA



## **4.5 Характеристики перемикачів**

Всі FPGA Spartan-3AN постачаються в двох ступенях швидкості: -4 та вища продуктивність -5. Характеристики перемикачів в цьому документі позначаються як попередній, попередній, попередній або виробничий. Кожна категорія визначається наступним чином:

Попередній перегляд: Ці специфікації базуються лише на оцінках і не повинні використовуватися для аналізу часу.

Аванс: Ці специфікації базуються лише на моделюванні та зазвичай доступні найближчим часом після встановлення специфікацій FPGA. Незважаючи на те, що рейтинги швидкості з цією назвою вважаються відносно стабільними та консервативними, певні недоотримання звітів все ще можуть відбутися.

Попередній характер: ці специфікації базуються на повній характеристиці раннього кремнію. Прилади та класи швидкості з цим призначенням мають на меті краще вказати очікувану продуктивність виробництва кремнію. Імовірність недотримання попередніх затримок суттєво зменшується в порівнянні з даними про підвищення.

Виробництво: Ці технічні характеристики затверджені, коли достатньо силового кремнію певного члена сімейства пристрою було охарактеризовано для забезпечення повної взаємозв'язку між файлами швидкості та пристроями через багато виробничих лотів. Існує не зафіксована затримка, і клієнти отримують офіційне повідомлення про будь-які подальші зміни. Як правило, найнижча швидкість означає перехід до виробництва перед швидшими ступенями швидкості.

## **4.6 Вимоги до програмного забезпечення**

Системи якісного виробництва повинні використовувати конструкції FPGA, зібрані за допомогою файлу швидкості, позначеного як статус виробництва. Проекти FPGA з використанням менш зрілої позначки швидкості файлу повинні використовуватися лише під час прототипування системи або попередньої підготовки. FPGA-проекти зі швидкісними файлами,

позначеними як Preview, Advance або Preliminary, не повинні використовуватися в системі якості виробництва.

Кожного разу, коли позначення швидкості файлу змінюється, коли пристрій досягає стану Production, перезапускається останнє програмне забезпечення Xilinx ISE® у проекті FPGA, щоб забезпечити, щоб дизайн FPGA містив останню інформацію про час та оновлення програмного забезпечення.

У деяких випадках ступінь швидкості випускається до виробництва в інший час, ніж коли файл швидкості випускається за допомогою етикетки виробництва. Будь-які розбіжності у маркуванні виправляються в наступних версіях файлів швидкості, для пристроїв, які можна вважати належним чином.

Всі обмеження параметрів є репрезентативними щодо найгірших умов напруги живлення та температурних переходів. Якщо не зазначено інше, опубліковані значення параметрів застосовуються до всіх пристроїв Spartan-3AN. Характеристики змінного та постійного струму визначаються з використанням однакових цифр як для комерційних, так і для промислових оцінок.

Параметри синхронізації та їх представницькі значення підбираються для включення, оскільки вони важливі як загальні вимоги до проектування, або вони вказують на основні характеристики продуктивності пристрою. Файли швидкості Spartan-3AN (v1.39), частина програмного забезпечення Xilinx Development, є оригінальним джерелом для багатьох, але не всіх значень. Для більш повних, точніших та найгірших даних використовуйте значення, що повідомляються статичним аналізатором синхронізації Xilinx (TRACE у програмному забезпеченні розробки Xilinx) та анотовані до симулятора netlist.

## РОЗДІЛ 5

### ЕТАПИ СТВОРЕННЯ СЕНСОРНОЇ МЕРЕЖІ ДЛЯ МОНІТОРИНГУ ДАНИХ

Одними з найважливіших завдань сучасного виробництва, де є потреба у цілодобовий моніторинг і автоматизоване управління технологічними процесами, є завдання контролю і управління, наприклад, контроль якості електричної енергії. В даний час для вирішення цих завдань використовуються різні стандартні і специфічні апаратні засоби і програмне забезпечення, створене як за допомогою універсальних засобів розробки, так і за допомогою специфічних засобів, таких як SCADA системи, які дозволяють розробляти алгоритми управління і забезпечувати візуальний інтерфейс системи. У будь-якому випадку створення програмного забезпечення (ПЗ) для моніторингу та управління вимагає високої кваліфікації програміста розробника. Однак, в разі, коли об'єкт спостереження є територіально розподіленим, з'являється додатковий рівень складності при реалізації подібної системи. Використання кросплатформеної розподіленої сенсорної мережі (КРСМ) може бути універсальним рішенням.

Під локальним об'єктом (ЛО) слід розуміти відокремлений об'єкт, розміри якого дозволяють застосовувати на ньому локальні обчислювальні мережі (ЛОМ) і шини передачі даних. Протяжність таких об'єктів - від одиниць до декількох сотень метрів. Прикладами ЛО є окремі адміністративні, промислові, громадські будівлі, обмежені території. Локально-розподілений об'єкт (ЛРО) - сукупність ЛО в межах виділеної території. прикладами даного класу є: комплекс будівель, будівель і відкритих майданчиків великого підприємства, що знаходяться в межах єдиної виділеної території, єдиний комплекс будівель. Максимальна протяжність таких об'єктів може становити від декількох сотень метрів до кількох кілометрів.

Територіально розподілений об'єкт (ТРО) - сукупність територіально відокремлених ЛО і ЛРО, віддалених один від одного і об'єднаних каналами зв'язку, з центральним об'єктом. До даного класу відносяться: регіональна

мережа об'єктів великого підприємства, об'єкта промисловості та багато ін. Максимальна взаємна віддаленість локальних об'єктів в складі РО від сотень метрів до кількох тисяч кілометрів.

Характер ТРО, а також можливість присутності кількох ЛО в складі, визначає використання для реалізації територіально розподіленої сенсорної мережі як ЛОМ, шин передачі даних RS-485, RS232, Bluetooth, ZigBee, так і бездротових мереж Wi-Fi, широкосмугових каналів Ethernet.

Кросплатформена територіально розподілена сенсорна мережа призначена для цілодобової автоматичної реєстрації даних, а також управління технологічними процесами в ручному режимі, а при використанні додаткових програмних засобів - і в автоматичному.

З огляду на територіального розподілення мережі, неминуче виникає питання гетерогенності вузлів мережі. Розподілені системи повинні функціонувати в середовищі, що представляє собою сукупність вузлів, виконаних в різній апаратної архітектурі, що працюють під управлінням різних операційних систем, пов'язаних з допомогою різних мережевих технологій, компоненти яких реалізовані з використанням різних мов програмування. Без цієї умови неможливе застосування КРСМ в різних, в тому числі і агресивних середовищах.

У цій ситуації очевидним є рішення використовувати проміжне програмне забезпечення. Шар проміжного програмного забезпечення забезпечує взаємодію між компонентами операційної системи і прикладних програмних забезпеченням вузла мережі. Використання подібної архітектури визначає ряд переваг: відсутність залежності прикладного програмного забезпечення від транспортних протоколів, можливість використовувати різні платформи при реалізації вузлів мережі, можливість використання різних мов програмування.

Сенсорна мережу подібних масштабів, що охоплює найрізноманітніші платформи і пристрої, містить інтелектуальні автономні модулі, з можливістю внутрішньої адаптації, і самоосвітою. Такими можливостями володіють

модулі на базі програмованої логічної інтегральної схеми (ПЛІС). В ході проектування кросплатформеної територіально розподіленої сенсорної мережі оптимальним рішенням є використання систем автоматизованого проектування (САПР) з можливістю моделювання і тестування пристроїв на базі ПЛІС.

Одним з таких програмних комплексів є зв'язка САПР Altium Designer 6.9. [25] і Xilinx ISE 9.2 [24].

САПР Altium Designer 6.9 являє собою систему наскрізного проектування і дозволяє паралельно розробляти проекти друкованої плати і ПЛІС. До безперечних достоїнств САПР Altium Designer 6.9 відноситься реалізована технологія Live Design [25], що дозволяє тестувати пристрої в реальному часі як на налагоджувальних комплектах так і на готових виробках, використовувати віртуальні інструменти.

Необхідність обробки і глибокого аналізу різних показників територіально розподілених об'єктів, в контексті завдань контролю якості електричної енергії диктує використання розподілених сенсорних мереж (PCM), єдиної бази даних для зберігання і обробки інформації.

Дослідження об'єктів, віднесених до класу або розглядаються як розподілені, можливо за допомогою адекватних інструментів. До таких інструментів можна безумовно віднести розподілені сенсорні мережі [16].

До етапу фізичної реалізації мережі доцільно провести тестування навантаження на базу даних, відпрацювати взаємодію компонентів сенсорної мережі, замінивши потік даних від сенсорів, тестовим набором даних, при цьому слід ввести програмні агенти, адаптовані роботу сенсорної мережі.

Побудова таких аналогів сенсорних мереж для проведення експерименту можливо з використанням технології віртуалізації додатків і віртуалізації операційних систем. Віртуалізація додатків дозволяє запускати різний набір програмних клієнтів на єдиній апаратній платформі, але в ізольованому просторі для кожного клієнта, в той час як технологія віртуалізації операційних систем має на увазі повну віртуалізацію всього

апаратного та програмного оточення операційної системи. До другого типу віртуалізації також слід віднести віртуальні середовища, що дозволяють емулювати роботу мобільних і вбудованих пристроїв [17, 18].

Повноцінний експеримент є основою досліджень у галузі науки. Однак, в деяких випадках, можливий перехід до часткової або повної віртуалізації експерименту з переходом на віртуальні аналоги систем.

Розподілена сенсорна мережа існує в трьох основних шарах взаємодії: мережевий, програмний, апаратний.

Очевидно, що абстрагуючись від конкретного обладнання, переходячи на рівень функціонального опису призначення вузла, а також його взаємодії з іншими віртуальними вузлами мережі можна говорити про достатній рівень віртуалізації апаратних аналогів вузлів мережі.

Такі функціональні аналоги фактично є програмними агентами, запущеними на реальних і віртуальних машинах, в залежності від рівня абстрагування і необхідності використання реального обладнання. Застосування віртуалізації додатків дає можливість повністю використовувати будь-які ресурси фізичного комп'ютера, а технологія віртуалізації операційних систем дозволяє підключати віртуальні порти передачі даних (USB, COM, LPT) до фізичних портів комп'ютера, таким чином, зберігається можливість підключення специфічного обладнання у віртуальне середовище. Віртуальні середовища, що дозволяють емулювати роботу мобільних і вбудованих пристроїв, володіють всіма комунікаційними можливостями, що і фізичні пристрої подібного класу. При цьому реалізація комунікаційних можливостей віртуальних коштів можлива завдяки використанню обладнання фізичного комп'ютера [19].

Сучасні методи і засоби розробки дозволяють згодом зробити перенесення розподіленої сенсорної мережі з віртуального простору з заміною програмних агентів фізичними сенсорами.

Таким чином здійснюється можливість використання різних типів віртуалізації для емуляції роботи кросплатформеної розподіленої сенсорної

мережі з використанням різних апаратних платформ, як промислових, так і широко поширених користувальницьких РСМ.

При розробці програмного забезпечення кросплатформеної сенсорної мережі (КРСМ), призначеної для вимірювання показників якості електричної енергії в електромережах територіально розподіленого об'єкта, а також при дослідженні параметрів транспортних інфраструктур, неминуче постає проблема гетерогенності оточення. Різні середовища, в яких функціонує КРСМ, різні платформи, за допомогою яких реалізуються вузли мережі, визначають використання різних мов, середовищ програмування. Витрати на розробку ПЗ великої територіально розподіленої мережі можуть стати істотними. Очевидно, що необхідна консолідація зусиль з розробки програмного забезпечення для подібних проектів.

Характер територіально розподілених об'єктів, а також можливість присутності кількох локальних об'єктів в їх складі, визначає використання для реалізації територіально розподіленої сенсорної мережі як ЛОМ, шин передачі даних RS-485, RS-232, Bluetooth, ZigBee, так і широкосмугових каналів Ethernet, бездротових мереж Wi-Fi.

Гетерогенне оточення КРСМ змушує розробника шукати баланс між енергоспоживанням пристроїв і їх комунікаційними можливостями. Використання інтегрованих OEM-модулів є оптимальним варіантом для налагодження і тестування функціональних і комунікаційних можливостей вузла КРСМ [14].

Таким чином, системи реалізовані з використанням єдиної технології, володіють можливостями роботи в гетерогенних мультиплатформових середовищах КРСМ, що дозволяє використовувати їх для вирішення завдань контролю стану і управління територіально розподіленими об'єктами.

## **5.1 Теоретична оцінка характеристик бездротової мережі**

Розробка і прийняття стандарту IEEE 802.15.6 для фізичного рівня персональних побутових сенсорних мереж були покликані вирішити наступні проблеми, властиві вузькосмуговим сигналам:

1) збільшити пропускну здатність WBAN (wireless body area network) за рахунок збільшення швидкості передачі в режимі «точка-точка» з 250 кбіт / с до 1-10 Мбіт / с;

2) підвищити надійність зв'язку в порівнянні з вузькосмуговими мережами:

3) спростити структуру мережі і алгоритми її роботи;

4) зменшити енергоспоживання компонентів мережі.

Пропускна здатність мережі. Фізична швидкість прямохатичного прийомопередвача, становить від 3 до 6 Мбіт/с. З аналізу пропускну здатності бездротових сенсорних мереж відомо, що гранично можлива пропускна здатність мережі в цілому в 5-6 разів нижче швидкості передачі між її вузлами в режимі «точка-точка». Таким чином, пропускна здатність мереж з прямохатичного прийомопередвача може досягати величин - 1 Мбіт/с, з, в той час як в мережах, що використовують вузькосмугові сигнали (на основі технології ZigBee), ця величина становить не більше 40 кбіт/с. Недостатня пропускна здатність може бути причиною нестійкої роботи мережі, як це спостерігалось, наприклад, в [13].

Підвищення надійності зв'язку. У радіосистемах малого радіусу дії, до яких відносяться приймачі розглянутих бездротових сенсорних мереж, зазвичай в якості допустимого рівня помилки приймається ймовірність помилки на біт не більше  $10^{-3}$ . У багатьох випадках така ймовірність помилки є прийнятною. В інших випадках, коли потрібно більш низький рівень, можна використовувати досить прості коди, що виправляють помилки, щоб зменшити рівень помилки на 1-3 порядки. Разом з тим застосування завадостійкого кодування підвищує енергоспоживання приймач, а також додатково знижує швидкість передачі в 1.5-2 рази, що не завжди допустимо.



В експериментах з прийомопередавачем в режимі передачі «точка-точка» було встановлено, що ймовірність помилок на біт  $\sim 10^{-6} - 10^{-7}$  є типовими для них в умовах використання апаратури в офісах і виробничих приміщеннях навіть без спеціального завадостійкого кодування. При використанні ретрансляцій ймовірність помилок зростає, але навіть при 4-5 ретрансляцій їх рівень, як правило, не перевищує  $10^{-5}$ . Тому можна очікувати, що при пакетній передачі даних при довжині пакетів менше 1 кбіт буде втрачатися не більше 1% пакетів.

Спрощення структури мережі і алгоритмів її роботи. У разі, якщо потоки інформації в мережі значно менше її пропускної здатності, можливе використання режимів з незалежним збором і передачею даних від кожного сенсорного пристрою. Незважаючи на відсутність координації між вузлами, зіткнення між пакетами, що передаються по мережі, малоімовірні. Такий режим роботи різко спрощує алгоритми роботи мережі в порівнянні з випадком синхронної роботи вузлів мережі. І за інших рівних умов, збільшує її надійність. У розглянутих мережах такий сценарій роботи може бути реалізований в значній кількості додатків.

Зменшення енергоспоживання. Аналіз енергоспоживання прямохаотичних прийомопередавачів показав, що при однаковій середній швидкості передачі енергоспоживання пристроїв в 3-5 разів менше, ніж у вузькосмугових систем на основі технології ZigBee.

Найбільш напружений режим роботи буде у ретрансляторів, оскільки через них проходить потік інформації відразу від декількох сенсорних вузлів. Але ретранслятори розташовуються в основному в фіксованих точках і можуть використовувати мережеву напругу.

Підсумовуючи вищесказане, можна очікувати, що на основі прямохаотичної технології передачі даних можуть бути реалізовані системи моніторингу, що відповідають вимогам стандарту IEEE 802.15.6.

Для обґрунтування практичної реалізованості подібних систем і їх очікуваних характеристик була створена експериментальна апаратура, що дозволяє реалізувати фрагменти мережі і проведена серія експериментів з нею.

## **5.2 Структура прийомопередавача**

У стандарті IEEE 802.15.6 в якості носіїв інформації пропонується використовувати імпульсні сигнали трьох типів: імпульси з лінійною частотною модуляцією (chirp pulses), хаотичні імпульси (chaotic pulses) і короткі імпульси (short pulses share).

Хаотичні імпульси як носій інформації для бездротових засобів зв'язку були запропоновані в ІРЕ ім. В.А. Котельникова РАН в 2000 р [16, 19, 20]. У 2007 р пропозицію ІРЕ ім. В.А. Котельникова РАН і компанії Sansung вони були включені в якості оптимального рішення в стандарт бездротового персонального зв'язку. Тоді хаотичні радіоімпульси були вперше визнані міжнародним науково-технічним суспільства в якості ефективного носія інформації для бездротових систем зв'язку.

До теперішнього часу розроблено та присутні на ринку кілька типів прийомопередавачів, що використовують хаотичні радіоімпульси. Це дозволяє зіставити вимоги стандарту IEEE 802.15.6 до прийомопередавачів з реальними пристроями.

Запропонований прийомопередавач призначений для передачі даних в режимі «точка - точка», а також для використання в бездротових сенсорних мережах. Він може використовуватися в режимі кінцевого пристрою, ретранслятора і спільно з комп'ютером в режимі базової станції. Пристрій має спеціальний роз'єм для підключення сенсорної плати і дозволяє отримувати з неї дані при передачі аналогового сигналу або через цифрові інтерфейси.

Структура пристрою представлена на рисунку 5.1. Її можна розділити на радіочастотну і цифрову частини.

До радіочастотної частини відносяться приймач, генератор з модулятором і НВЧ перемикач. Передавач складається з модулятора і генератора. Модулятор являє собою електронний ключ, який подає живлення

на генератор і забезпечує необхідну напругу на його вході. До складу приймача входять малoshумний підсилювач і логарифмічний детектор. Необхідність в НВЧ перемикачі пов'язана з тим, що прийом і передача здійснюються в одному діапазоні і за допомогою однієї приймально-передавальної антени. У той момент, коли здійснюється передача, ключ знаходиться в такому положенні, при якому забезпечується комутація виходу генератора зі входом антени, причому в канал приймача НВЧ сигнал не потрапляє. Під час прийому - навпаки. Таким чином, НВЧ перемикач здійснює розв'язку між приймачем і генератором.



Рисунок 5.1 - Структурна схема прийомопередавача

Функцію управління ключем виконує програмована логічна інтегральна схема (ПЛІС). Крім того, вона здійснює попередню обробку імпульсів, отриманих від приймача, формування імпульсів потрібної довжини і шпаруватості для передачі. Оскільки ПЛІС працює виключно з цифровими сигналами, то для перетворення аналогових імпульсів в цифрові використовується компаратор. Гранична напруга компаратора задається цифро-аналоговим перетворювачем (ЦАП), керованим мікро контролером (МК).

МК координує роботу всього приймача. Крім управління ЦАП (тобто завдання порогового напруги компаратора), МК відповідно до завантаженою програмою активізує режими прийому, передачі інформації, режим

очікування; містить інформацію, необхідну для роботи пристрою в мережі; включає в себе інтерфейси зв'язку з зовнішніми пристроями.

Робота приймача в мережі визначається такими характеристиками як довжина пакета даних, довжина фрейму даних, число пакетів в секунду.

Надалі, як приклад, за відсутності іншої домовленості, будемо використовувати набір робочих параметрів наведений в таблиці 5.1.

Таблиця 5.1 – Набір робочих параметрів

Параметр	Позначення	Значення
Швидкість передачі даних(корисна)	$R_{data}$	31 кбіт/с
Фізична швидкість передачі	$R_{phys}$	5 Мбіт/с
Довжина пакету даних	$L_{data\_pack}$	698 біт
Довжина фрейму даних	$L_{data\_in\_pack}$	620 біт
Тривалість імпульсу	$T_{pulse}$	77 нс

### 5.3 Структура енергоспоживання прийомопередавача

Блок-схема енергоспоживання приймача показана на рисунку 5.2 і включає в себе мікроконтролер (МК), ПЛІС, передавач, приймач, кварцовий генератор, компаратор, ЦАП, стабілізатор і джерело живлення (батареї). В процесі функціонування приймач постійно споживає електроенергію. Величина споживання є функцією часу, яка залежить від характеристик енергоспоживання окремих елементів приймача, включаючи обрані режими та умови їх роботи, і від складу елементів приймача, активних на різних часових інтервалах.

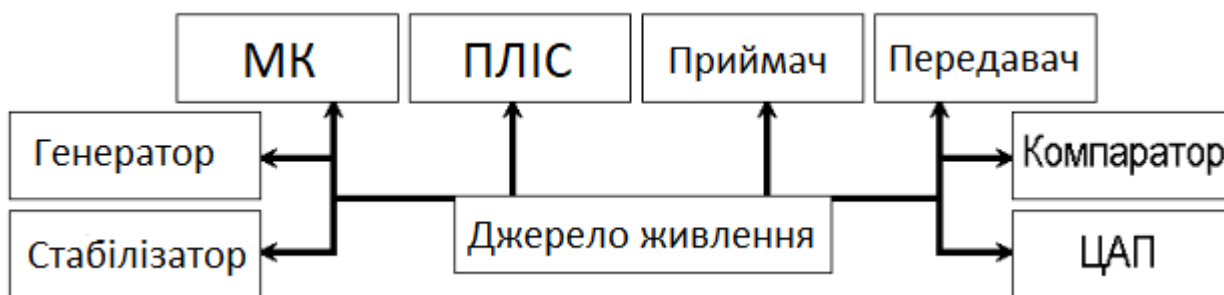


Рисунок 5.2 – Блок-схема енергоспоживання прийомопередавача

Наприклад, різні типи мікроконтролерів мають різні рівні споживання потужності крім цього, потужність, яка споживається мікро контролером, залежить від його завантаження і тактової частоти, на якій він працює в прийомопередавачі.

Потужність, яку споживають передавач і приймач, також залежить від обсягів інформації, що передається в одиниць часу інформації.

#### 5.4 Функціонування приймача в режимі ретрансляції

Основні робочі режими функціонування вузлів комунікаційної мережі - це режим передачі потоку даних по бездротовому каналу, режим ретрансляції сигналу і всілякі сплячі режими, якщо ніякі потоки даних в даний момент не передаються. Крім того, час від часу передається службова інформація - керуючі команди, але вважаємо, що ці процеси займають значно менше часу в порівнянні з основними режимами.

Для того щоб проаналізувати енергоспоживання пристрою, почнемо з ретрансляції пакетів, оскільки це найбільш складний режим, при якому задіяні всі блоки пристрою. Розглянемо цикл ретрансляції одного пакета даних.

У циклі ретрансляції пакета даних можна виділити чотири основні етапи: прийом пакета, обробка пакета, передача пакета, режим «очікування». Тут під режимом «очікування» мається на увазі пауза між двома ретрансляції послідовно йдуть пакетів.



Рисунок 5.3 – Етапи циклу ретрансляції пакету

### 5.5 Енергоспоживання приймача в режимі ретрансляції в відсутності енергозбереження

У якості відправної точки розглянемо енергоспоживання приймача, в якому не передбачені режими енергозбереження. У прийомопередавачі існує тільки три можливих стани: прийом, обробка пакета, передача. У паузах між пакетами приймач слухає ефір, тобто знаходиться в режимі прийому.

На рис. 5.4 показаний сценарій функціонування окремих блоків в циклі ретрансляції пакета.

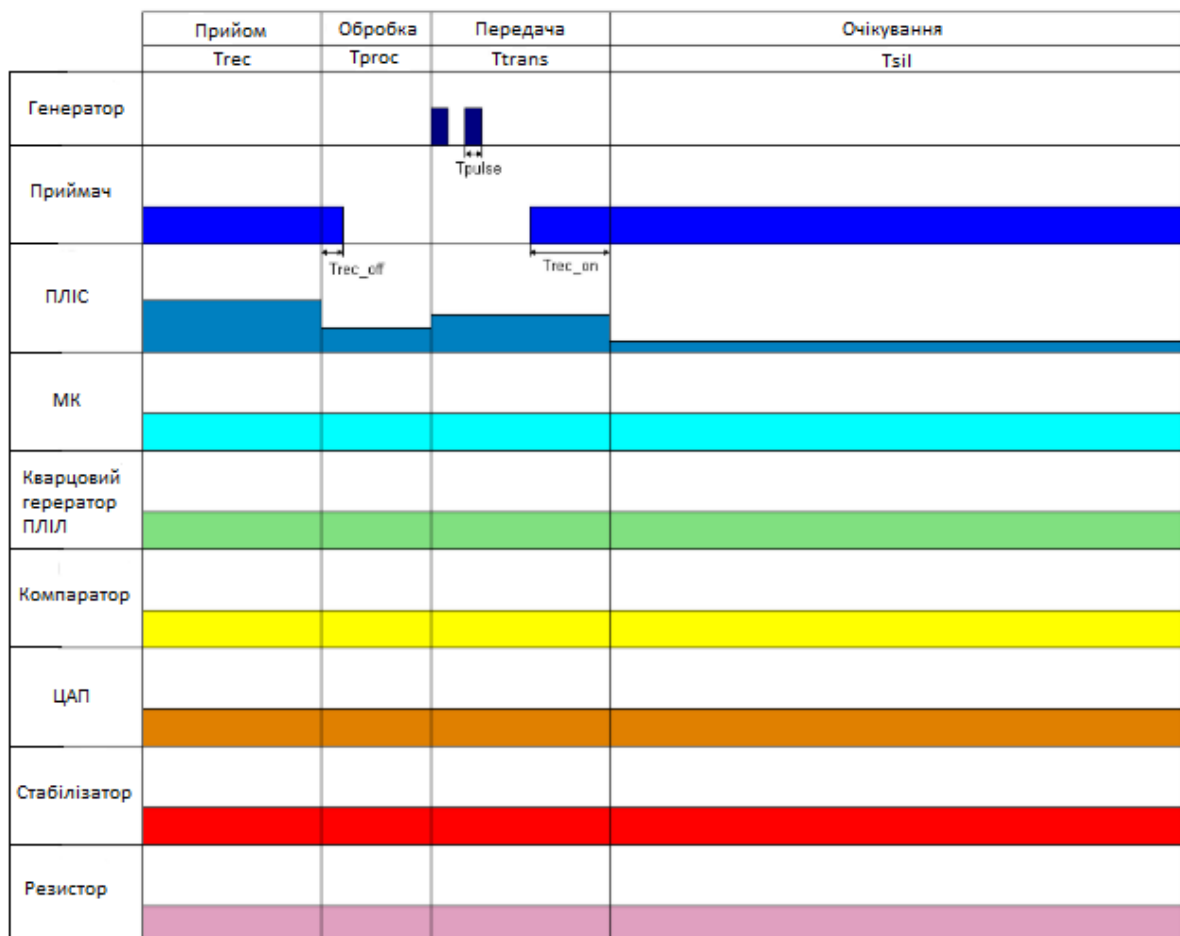


Рисунок 5.4 – Схема функціонування блоків приймача в циклі ретрансляції пакета даних. Варіан 1.

На кожному етапі ретрансляції пакету блоки приймача працюють по-різному в залежності від виконуваних ними на даному етапі функцій і заданого сценарію їх роботи. Сценарій роботи блоків приймача закладений в програму, виконувану мікро контролером, який управляє змінами режимів окремих блоків приймача.

Цикл ретрансляції пакету:

1) прийом пакета

На цій стадії відбувається прийом пакета протягом часу відповідного тривалості 1 пакету  $T_{rec} = T_{1bit} * T_{Ldata\_pack}$

- Генератор відключений
- Приймач включений, споживає струм  $I_{rec\_rec}$
- ПЛІС включений, споживає струм  $I_{PLIC\_rec}$
- Мікроконтроллер включений, споживає струм  $I_{MC\_rec}$
- Генератор ПЛІС включений, споживає струм  $I_{quartz\_rec}$
- Компаратор включений, споживає струм  $I_{compar\_rec}$
- ЦАП включений, споживає струм  $I_{DAC\_rec}$
- Стабілізатор включений, споживає струм  $I_{stab\_rec}$

2) обробка пакета

На цій стадії відбувається обробка прийнятого пакета протягом часу  $T_{proc}$ .

- Генератор відключений
- Приймач вимикається. Час виключення приймача  $T_{rec\_off}$ , при цьому він споживає струм  $I_{rec\_proc}$
- ПЛІС включений, споживає струм  $I_{PLIC\_proc}$
- Мікроконтроллер включений, споживає струм  $I_{MC\_proc}$
- Кварцовий генератор ПЛІС включений, споживає струм  $I_{quartz\_proc}$
- Компаратор включений, споживає струм  $I_{compar\_proc}$
- ЦАП включений, споживає струм  $I_{DAC\_proc}$

- Стабілізатор включений, споживає струм  $I_{stab\_proc}$

### 3) передача.

На цій стадії відбувається передача пакета протягом часу відповідного тривалості 1 пакету  $T_{trans} = T_{1bit} * T_{Ldata\_pack}$

- Якщо передається біт «1», то генератор включається на час відповідне тривалості одного інформаційного імпульсу  $T_{pluse}$  і вимикається на час відповідне захисному інтервалу. Якщо передається біт «0» генератор залишається у вимкненому стані. Процедура повторюється відповідно до кількості переданих біт.

- Приймач вимкнений протягом часу  $T_{trans} - T_{rec\_on}$ , в кінці етапу передачі пакета приймач включається протягом часу  $T_{rec\_on}$ , при цьому споживає струм  $I_{rec\_rec}$

- ПЛІС включений, споживає струм  $I_{PLIC\_trans}$

- Микроконтроллер включений, споживає струм  $I_{MC\_trans}$

- Кварцовий генератор ПЛІС включений, споживає струм  $I_{quartz\_trans}$

- Компаратор включений, споживає струм  $I_{compar\_trans}$

- ЦАП включений, споживає струм  $I_{DAC\_trans}$

- Стабілізатор включений, споживає струм  $I_{stab\_trans}$

### 4) очікування

У режимі без енергозбереження в паузах між пакетами пристрій прослуховує ефір, тобто знаходиться в режимі прийому протягом часу  $T_{sil} = L_{data\_in\_pack} / R_{data} - (T_{rec} + T_{proc} + T_{trans})$ .

Розглянемо розроблений прийомопередавач. Це пристрій, що працює в діапазоні частот 4-6 ГГц, Фізична швидкість передачі даних 5 Мбіт / с, яку випромінює потужність на цій швидкості -1 дБм, Дальність передачі 20-25 м. Це одна з перших версій приймачів, що використовуються в сенсорних мережах. У цьому приймачі не передбачені режими енергозбереження.



В даному пристрої генератор включається тільки для передачі даних, при цьому в безперервному режимі він споживає струм 58 мА. Для передачі окремих бітів генератор модулюється по живленню, тобто вмикається і вимикається. При цьому час включення-виключення генератора малий (~ 1-2 нс). За рахунок такої модуляції середнє споживання генератора зменшується пропорційно скважності проходження імпульсів. На етапах прийому і обробки пакета і в паузі між пакетами генератор повністю відключається і його споживання дорівнює 0 мА.

Приймач відключається на час обробки і передачі пакета (споживаний струм 0 мА). Весь інший час приймач знаходиться в робочому режимі чекаючи приходу пакету даних, споживаючи при цьому струм 39 мА.

ПЛІС включений постійно і споживає «холостому режимі» 2 мА, однак в залежності від дій, які він виробляє на різні етапи циклу прийому-передачі пакета його споживання зростає. Перемикання між режимами відбувається швидко.

Таблиця 5.2 - Середні значення споживання енергії

	Прийом	Обробка	Передача	Очікування
Генератор	0 мА	0 мА	68 мА (в неперервному режимі)	0 мА
Приймач	41 мА	0 мА час вимикання 155нс	0 мА час вмикання 15мкс, ток включення 39мА	39 мА
ПЛІС	9 мА	9 мА	9 мА	2 мА
МК	8 мА	8 мА	8 мА	8 мА
Компаратор	7.2 мА	7.2 мА	7.2 мА	7.2 мА
ЦАП	0.21 мА	0.21 мА	0.21 мА	0.21 мА
Стабілізатор	0.45 мА	0.45 мА	0.45 мА	0.45 мА

Решта блоки пристрою, такі як контролер, кварцовий осцилятор, компаратор, ЦАП і стабілізатор працюють постійно.

У таблиці 2 наведені середні значення споживання енергії кожним з блоків пристрою на різних етапах.

Час обробки пакета вважається рівним 1 мкс.

У таблиці 5.3 наведені середні значення споживання струму пристроєм під час прийому пакета, передачі пакета, в паузі між пакетами і загальне споживання пристрою з розрахунку, що передаються пакети довжиною 90 байт зі швидкістю 31 кбіт / с. Передбачається, що пристрій працює в цьому режимі 100% часу.

Таблиця 5.3 - Середні значення споживання струму

Споживання прийомопередавача	Середнє, мА
На етапі прийому	0.42
На етапі передачі	0.28
При ретрансляції (прийом + обробка + передача)	0.69
На етапі очікування (в паузах між пакетами)	71.6
Загальна	72.31

Середнє споживання енергії при такій швидкості передачі дорівнює приблизно 73 мА або 242 мВт. Якщо пристрій буде безперервно працювати в режимі ретрансляції на такій швидкості, то воно зможе працювати від батареї ємністю 1.2 Ач (2.4 Ач) протягом 18 (34) годин.

Отримані характеристики дозволяють забезпечити функціонування приймачів протягом декількох років, при використанні їх в додатках, в яких інформація від сенсорних блоків передається рідко, наприклад раз на місяць і обсяг переданих даних невеликий. Однак якщо мова йде про додатки, в яких потік даних від датчиків досить інтенсивний, для забезпечення довготривалої роботи мережі без заміни джерел живлення енергоспоживання приймачів має бути значно зменшено.

Наведені нижче малюнки ілюструють внесок в загальне споживання пристрою окремих блоків приймача на різних етапах циклу ретрансляції.

На рис. 5.5 представлена загальна картина споживання. Кольором показано споживання окремих блоків. Споживання блоків згруповано за етапами циклу ретрансляції.

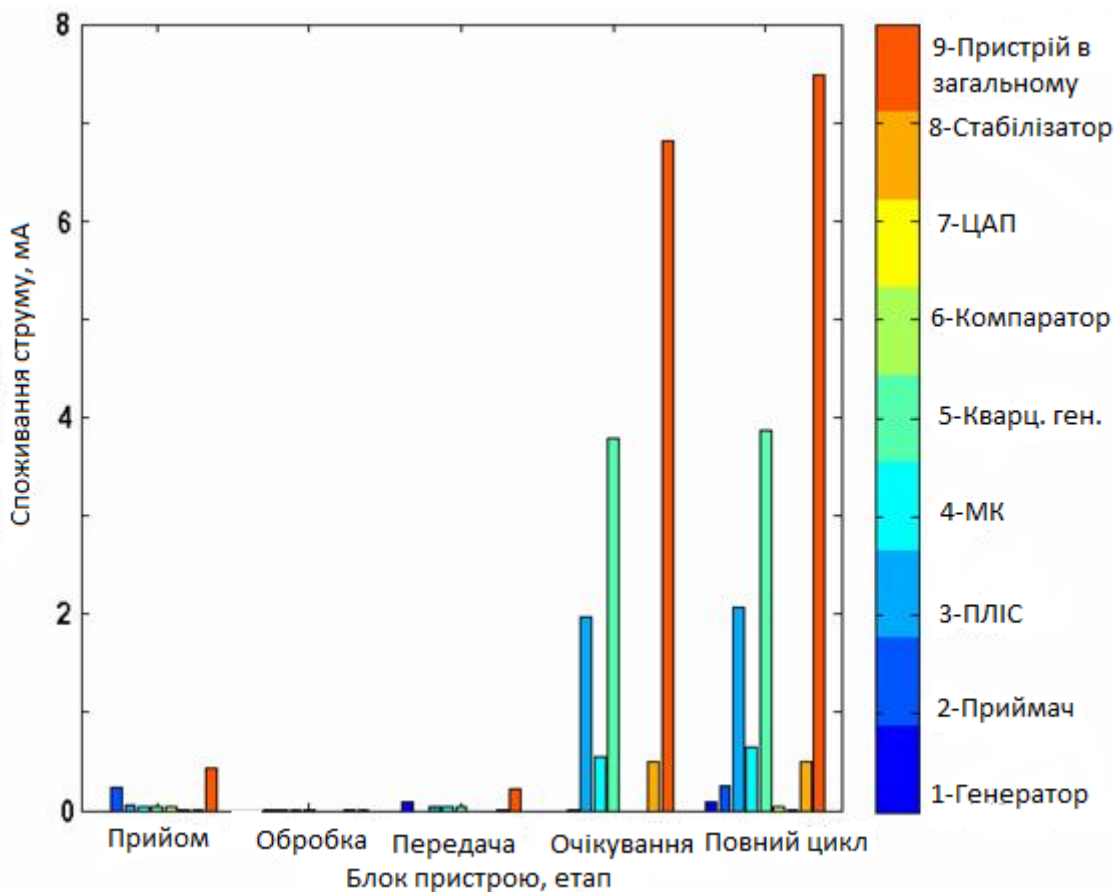


Рисунок 5.5 - Споживання блоків пристрою на різних етапах циклу ретрансляції пакетів.

Як показує аналіз малюнку, основне споживання пристрою припадає на етап очікування, при цьому блоком, який споживає найбільший струм, є приймач. Це пояснюється тим, що приймач сам по собі споживає в активному режимі досить великий струм, в порівнянні з іншими блоками, працює практично постійно, відключаючи лише на короткий проміжок часу. Інші блоки пристрою (за винятком генератора) так само працюють постійно, вносячи свій внесок в загальне споживання.

Для того щоб знизити енергоспоживання приймача необхідно оптимізувати функціонування його блоків. Для цього будуть використані одночасно два підходу: вибір блоків пристрою з більш низьким споживанням і організація режимів функціонування блоків пристрою.

### **5.6 Енергоспоживання приймача в режимі ретрансляції з енергозбереженням**

На першому етапі ми спробували знизити енергоспоживання приймача за рахунок більш раціональної організації роботи окремих блоків. Якщо у вихідній схемі велика частина блоків працює безперервно протягом усього циклу ретрансляції пакету, то в новому варіанті пропонується там, де можливо, відключати не задіяні на тому чи іншому етапі блоки або переводити їх в режими зі зниженим енергоспоживанням («сплячі режими») і повертати їх в активний режим безпосередньо до того моменту, коли вони повинні бути задіяні для виконання будь-яких функцій.

Наприклад, якщо в системі зв'язку передбачена синхронізація між пристроями, і приймач має інформацію про очікуваний час приходу наступного пакета, який він повинен прийняти, то приймач може не прослуховувати ефір постійно, а включатися тільки до певного моменту часу. Компаратор і ЦАП задіяні тільки під час прийому пакета, після того, як пакет прийнятий, їх можна відключити і включити знову до моменту прийому наступного пакета. Мікроконтролер активно працює під час прийому, обробки і передачі пакета, а на етапі очікування він може бути переведений в «сплячий» режим. Кварцовий генератор ПЛІС може бути відключений на час очікування наступного пакета. Однак необхідно повернути мікроконтролер і кварцовий генератор ПЛІС в активний режим до моменту приходу наступного пакета.

Тимчасова діаграма функціонування блоків приймача за таким варіантом приведена на рис. 5.6.

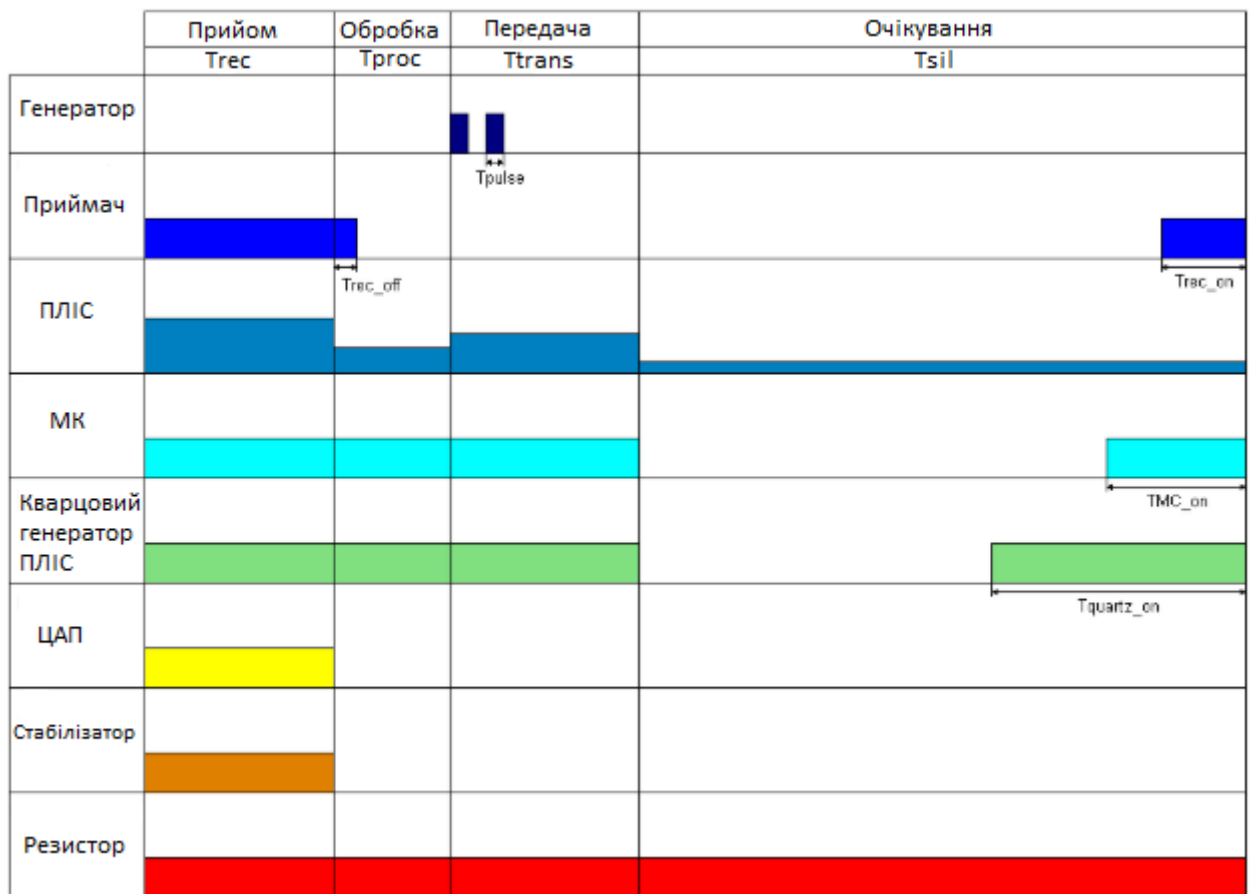


Рисунок 5.6 - Діаграма функціонування блоків приймача в циклі ретрансляції пакета даних. Варіант 2

Генератор включається тільки для передачі даних, при цьому в безперервному режимі он, як і раніше, споживає струм 58 мА. Для передачі окремих бітів генератор модулюється по харчуванню, тобто вмикається і вимикається. При цьому час включення-виключення генератора мало (~ 1-2 нс). Зменшення енергоспоживання за рахунок скважності проходження радіоімпульсу в таблиці не наведено, вони враховуються далі при розрахунку енергоспоживання пристрою і його блоків. На етапах прийому і обробки пакета і в паузі між пакетами генератор повністю відключається і його споживання дорівнює 0 мА.

Приймач працює тільки під час прийому пакета, споживаючи при цьому струм 39 мА. На час обробки, передачі пакета і в паузі між пакетами приймач відключений (споживаний струм 0 мА). Однак під час відключення приймача (155 нс) на етапі обробки пакета і під час включення приймача (15 мкс) для

прийому наступного пакета в кінці етапу мовчання приймач споживає струм 39 мА.

ПЛІС включений постійно і споживає «холостому режимі» 2 мА, однак в залежності від дій, які він виробляє на різні етапи циклу прийому-передачі пакета його споживання зростає. Так, при прийомі пакету ПЛІС споживає додатково 7.4 мА, при обробці пакета 2 мА, при передачі пакета 4.8 мА. Перемикання між режимами відбувається швидко і не враховується в подальших розрахунках.

Мікроконтролер під час прийому, обробки і передачі пакета знаходиться в активному режимі і споживає 8 мА. На час паузи між пакетами мікроконтролер переводиться в сплячий режим зі споживанням 9 мкА, при цьому перехід в сплячий режим відбувається швидко, а вихід їх сплячого режиму для прийому наступного пакета займає 1.3 мс при споживанні струму 8 мА.

Кварцовий генератор, що задає тактову частоту ПЛІС працює на частоті 46 МГц і в робочому режимі споживає 7.2 мА. В цьому режимі кварцовий генератор знаходиться під час ретрансляції пакета. На час паузи між пакетами кварцовий генератор відключається. Відключення відбувається швидко, в вимкненому стані кварцовий генератор струм не споживає. Для прийому наступного пакета кварцовий генератор знову почне працювати, що займає 10 мс при споживанні струму 7.2 мА. Компаратор і ЦАП працюють тільки на етапі прийому пакета, споживаючи при цьому 7.2 мА і 0.2 мА відповідно. На інших етапах компаратор і ЦАП відключені. Вмикання та вимикання цих блоків відбувається швидко. Споживання стабілізатора залежить від загального споживання інших блоків приймача. При високому загальному споживанні пристрою під час ретрансляції пакету стабілізатор споживає близько 1 мА, при падінні споживання пристрою на етапі мовчання споживання стабілізатора знижується до 0.5 мА.

Результати розрахунків енергоспоживання приймача наведені в таблиці 5.4.

Таблиця 5.4 – Енергозатрати прийомопередавача

Споживання прийомопередавача	Середнє, мА
На етапі прийому	0.42
На етапі передачі	0.22
При ретрансляції (прийом + обробка + передача)	0.64
На етапі очікування (в паузах між пакетами)	6.61
Загальна	7.26

З порівняння результатів, наведених в таблицях 5.3 і 5.4 видно, що введення енергозберігаючого сценарію функціонування блоків приймача дозволяє знизити енергоспоживання майже в 10 разів. Причому як впливає з аналізу малюнку 5.7 основна економія досягається за рахунок зниження витрат на етапі очікування.

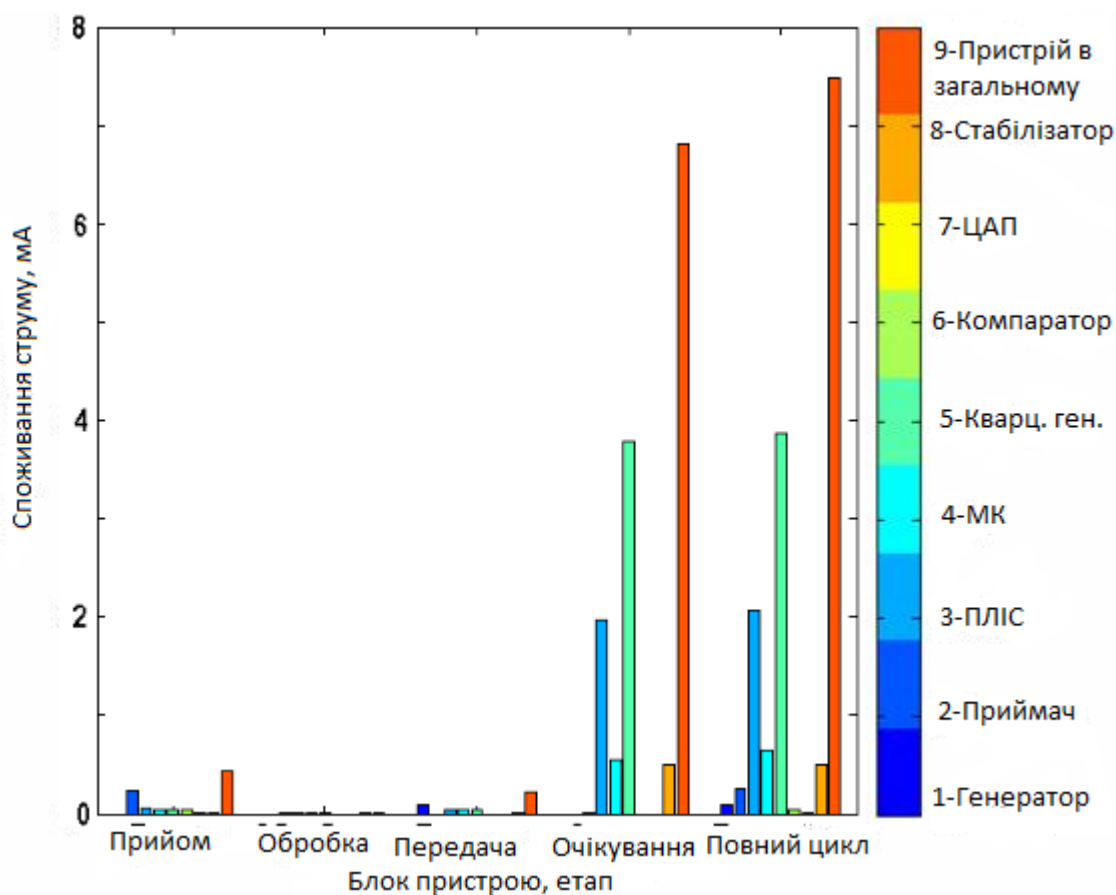


Рисунок 5.7 - Споживання блоків пристрою на різних етапах циклу ретрансляції пакетів.

Аналіз отриманих результатів показує, що основний внесок в загальне енергоспоживання в циклі ретрансляції пакету вносять ПЛІС і кварцовий генератор ПЛІС в паузі між пакетами. Це означає, що для подальшого зменшення енергоспоживання пристрою в першу чергу необхідно знизити споживання цих блоків на етапі очікування. Зменшення енергоспоживання ПЛІС можна домогтися, якщо використовувати іншу модель ПЛІС - MAX V [24]. Для цієї моделі характерне більш низьке базове споживання - 27 мкА, замість 2 мА для моделі MAX II [20].

Крім того, можна використовувати одинарний компаратор, замість подвійного використовуваного в базовій моделі, що дозволить знизити споживання компаратора вдвічі.

Після проведеної модифікації найбільш споживають блоком пристрою залишається кварцовий генератор ПЛІС. Для того щоб вирішити цю проблему пропонується подавати тактову частоту на ПЛІС ні з кварцового генератора, а з виходу мікроконтролера з наступним множенням частоти в ПЛІС. Це дозволить позбутися від кварцового генератора ПЛІС.

На рис. 5.8(а) і 5.8(б) показана схема подачі тактової частоти на ПЛІС в початковому варіанті (з кварцового генератора ПЛІС) і в новому варіанті (з мікроконтролера) відповідно.

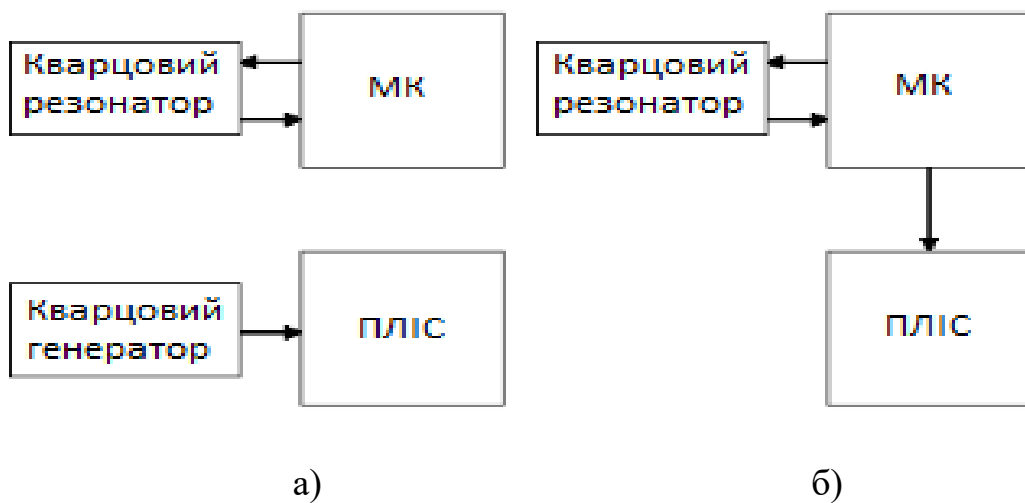


Рисунок 5.8 - Варіанти схеми подачі тактового сигналу на мікроконтролер і ПЛІС.



Сценарій функціонування блоків приймача при відсутності кварцового генератора ПЛІС представлений на рис. 5.9.

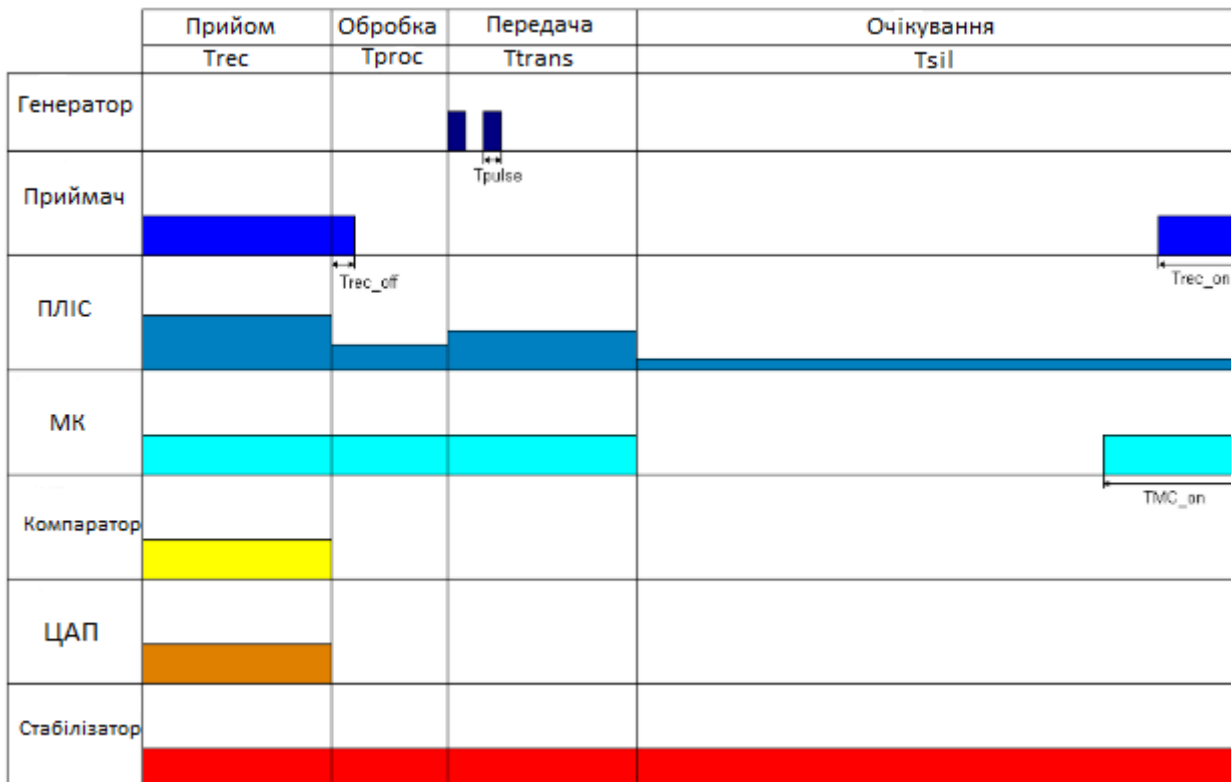


Рисунок 5.9 - Діаграма функціонування блоків приймача в циклі ретрансляції пакета даних. Варіант 3.

Споживання блоків пристрою на різних етапах циклу ретрансляції пакетів наведено на рис 5.10.

Як показує аналіз результатів в описаному випадку найбільше споживання припадати на мікроконтролер.

У пристрої тактова частота мікроконтролера формується за допомогою зовнішнього пасивного кварцового осцилятора. При перекладі мікроконтролера в сплячий режим кварцовий осцилятор відключається. При подальшому перекладі мікроконтролера в активний режим спочатку запускається кварцовий осцилятор. Мікроконтролер може працювати тільки коли кварцовий осцилятор вийде на стабільний режим. Цим фактором визначається повільне час переходу мікроконтролера в робочий режим (близько 1.3 мс). Споживання мікроконтролера в процесі переходу із сплячого

режиму в робочий режим вносить основний вклад в загальне споживання мікроконтролера і пристрої в цілому. Отримуємо наступні результати для енергоспоживання пристрою (Таблиця 5.5).

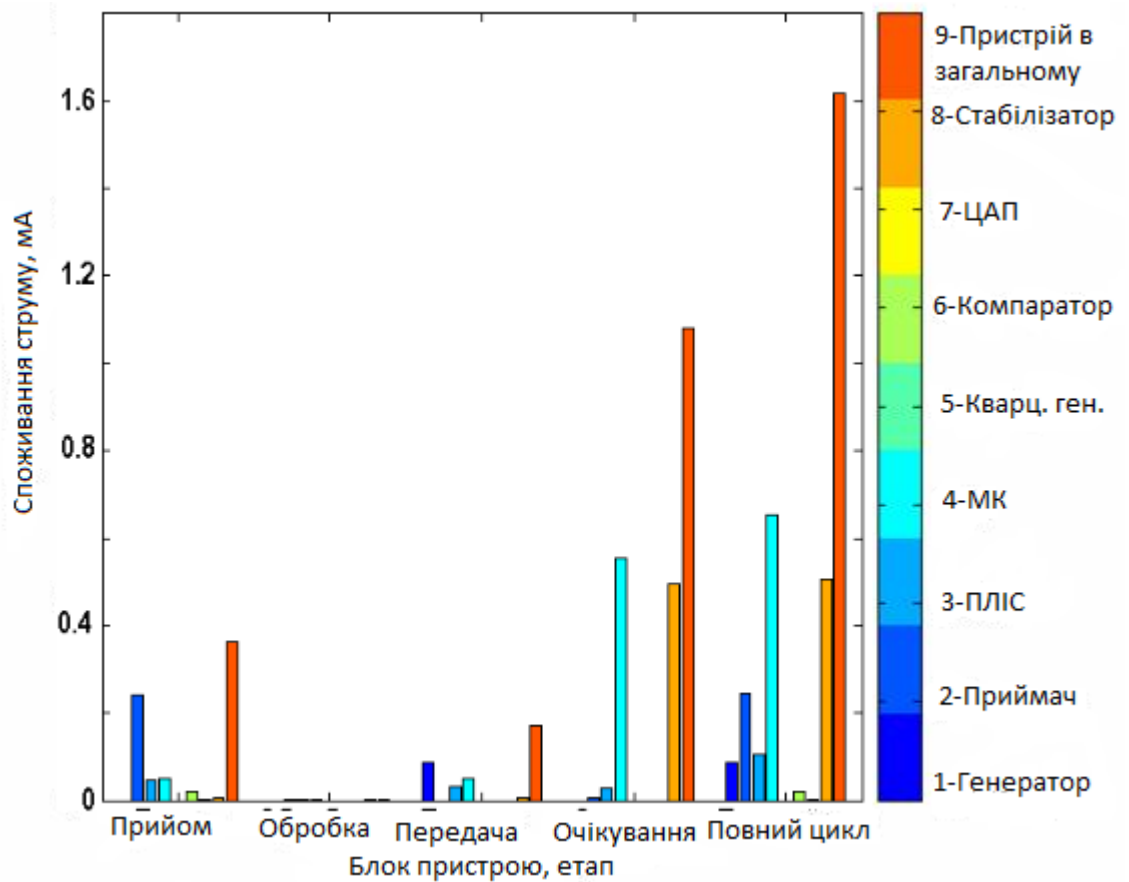


Рисунок 5.10 - Споживання блоків пристрою на різних етапах циклу ретрансляції пакетів.

Таблиця 5.5 – Енергозбереження прийомопередавача

Споживання прийомопередавача	Середнє, мА
На етапі прийому	0.34
На етапі передачі	0.16
При ретрансляції (прийом + обробка + передача)	0.52
На етапі очікування (в паузах між пакетами)	1.1
Загальна	1.57

В результаті проведеного зміни конструкції приймача можна знизити енергоспоживання в паузах між пакетами до 1 мА, а загальне споживання при ретрансляції пакетів на швидкості 31 кбіт / с до 1.5 мА.

Що означає, що при повному завантаженні на швидкості 31 кбіт / с пристрій зможе пропрацювати 800 годин (1600 годин) від батареї ємністю 1.2 Ач (2.3 Ач) або при 15% завантаженні близько 15 місяців.

Як показують розрахунки на швидкості 62 кбіт / с пристрій зможе пропрацювати 500 годин (980 годин) від батареї ємністю 1.1 Ач (2.2 Ач) або при 15% завантаженні близько 10 місяців.

## **5.7 Енергоспоживання приймача в режимі передачі даних і в «сплячому» режимі**

Наведені вище дані були отримані для найважливого режиму-режиму ретрансляції пакетів. А що ми маємо в разі передачі даних від сенсора? В цьому випадку пристрою немає необхідності прослуховувати ефір. Воно тільки час від часу шле пакети даних. Тому приймач, компаратор і ЦАП можна тримати вимкненими. Що призводить до споживання пристрою при передачі даних зі швидкістю 31 кбіт / с – 3.8 мВт.

Коли пристрій тривалий час не використовується для передачі або ретрансляції даних можна перевести його в сплячий режим. При роботі в глибокому сплячому режимі пристрій 1 раз в секунду прокидається і слухає ефір приблизно протягом 19 мкс. Якщо за цей час пристрій отримує з ефіру сигнал побудки, воно переходить в активну фазу, якщо немає - знову засинає.

У глибокому сплячому режимі всі блоки пристрою відключаються за винятком мікроконтролера, який знаходиться в режимі Power Down. В цьому режимі споживання струму мікроконтролером становить 25 мкА. Час виходу мікроконтролера з режиму Power Down - 4.2 мс. Крім того, ще 1 мс потрібно для встановлення частоти кварцового резонатора. Виходячи з цих даних маємо наступну оцінку для споживання пристрою в сплячому режимі.

При знаходженні в глибокому сплячому режимі протягом 100% часу пристрій споживає струм 77 мкА.

## 5.8 Експериментальне дослідження енергоспоживання приймач

На основі описаного підходу до енергозбереження був розроблений і виготовлений малозатратний приймач. В ході роботи над експериментальним макетом пристрою були внесені невеликі зміни в спочатку запропоновану схему функціонування пристрою, які проте не змінюють загального сенсу. У таблиці 6 наведено порівняння результатів розрахунку енергоспоживання приймача в режимі передачі даних і в режимі ретрансляції при швидкості передачі даних 62 кбіт / с і довжині пакетів даних 164 байт для математичної моделі системи і експерименту.

Таблиця 5.6 – Енергозатрати, мА

	Модель	Експеримент
Передача даних	2.4	2.7
Ретрансляція	3.2	3.9

При напрузі живлення пристрою 3.2 В споживання потужності пристрою з передачі даних на швидкості 62 кбіт / с одно 9.1 мВт, в режимі ретрансляції даних на швидкості 62 кбіт / с – 12.9 мВт.

Проведено дослідження енергоспоживання прийомопередавача і проаналізовані шляхи зниження енергоспоживання для забезпечення більш тривалої автономної роботи.

В результаті проведених досліджень запропонована структура приймача і розроблений сценарій функціонування окремих блоків приймача, що дозволяють забезпечити енергоспоживання на рівні 2.7 мА (9.1 мВт) в режимі передачі і 3.9 мА (12.9 мВт) в режимі ретрансляції даних зі швидкістю 62 кбіт

## 5.9 Експериментальне дослідження бездротової сенсорної мережі

Для дослідження БСМ прийомопередавача був проведений ряд експериментів з типовими варіантами топології мережі, з можливістю

використання в банківських системах, для моніторингу. Основними задачами проведення експериментів були перевірка працездатності запропонованої сенсорної мережі і оцінка її ефективності.

1) Експеримент по передачі даних із декількох приміщень.

В експериментах перевірялася робота мережі при прийомі даних від сенсорних вузлів, розташованих в двох приміщеннях, базова станція перебувала поза прямої видимості сенсорним вузлів, здійснювали збір даних. Топологія мережі мала вигляд «дерева» з двома гілками (рис. 1.36).

При проведенні випробування використовувалися 2 сенсорних вузла (У1, У2), 3 ретранслятора (У3, У4, У5) і 1 базова станція (У6). Схема експерименту представлена на рис. 5.11. Дані від сенсорного вузла У1 передавалися на ретранслятор У3, після чого подавали на ретранслятор У5 і потім приймалися базовою станцією У6. Аналогічно передавалися дані від сенсорного вузла У2: дані через ретранслятор У4 потрапляли на ретранслятор У5, після чого їх приймала базова станція. Після обробки і візуального представлення, на комп'ютері з'являлися дані від сенсорних вузлів У1 н У2.

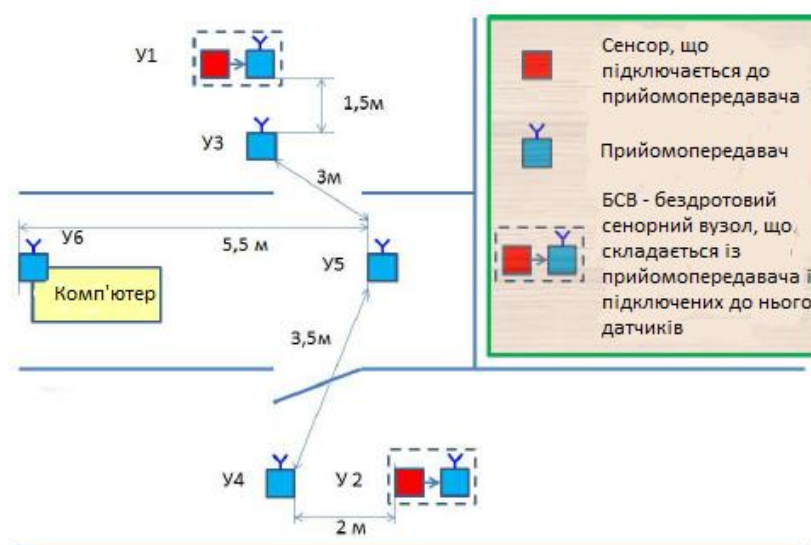


Рисунок 5.11 – Схема розміщення вузлів, при зборі та передачі даних

В експериментах з топологією типу «дерева», при незалежному знятті даних від сенсорних вузлів не спостерігалися колізій через зіткнення пакетів.

2) Експерименті з передачі даних від ансамблю сенсорних вузлів.

В експериментах вивчалась можливість роботи БСМ в разі одночасного збору інформації від декількох сенсорних вузлів, дані від яких надходили на базову станцію через ретранслятор. Топологія мережі була комбінацією топології «зірки» і «ланцюг»(рис. 1.3(б) і 1.3(в)).

При проведенні експериментів використовувалися 10 сенсорних вузлів (У1-У10), вузол ретранслятор (У11) і базова станція (У12). Всі сенсорні вузли мережі працювали в асинхронному режимі, посилаючи дані один раз в секунду.

Експерименти показали, що в такій топології мережі система стабільно працює в асинхронному режимі. При цьому теоретичні оцінки показують, що стійкий характер функціонування мережі зберігається при збільшенні числа сенсорних вузлів щонайменше в десять разів в порівнянні з числом вузлів, що використовувалися в експериментах.

### 3) Експеримент з передачі даних від рухомого сенсорного вузла.

Завданням експериментів являється перевірка роботи бездротової сенсорної мережі в разі передачі даних від сенсорного вузла, що переміщається уздовж ланцюжка ретрансляторів. Дані від вузла приймає ближній до нього ретранслятор. На відміну від попередніх випробувань, топологія мережі динамічно змінювалася залежно від положення сенсорного вузла (рис. 5.12).

При проведенні експериментів використовувався сенсорний вузол (У1), 3 прийомопередавачі–ретранслятори (В2, В3, В4) і базова станція (У5).

Дані від рухомого сенсорного вузла У1, розташованого на пересувний тумбі, надходили на той ретранслятор, в області якого знаходився в даний момент вузол У1, після чого передавалися на наступний в ланцюжку ретранслятор, що знаходиться ближче до базової станції У5 і т.д. Після серії ретрансляцій дані надходило на базову станцію У5.

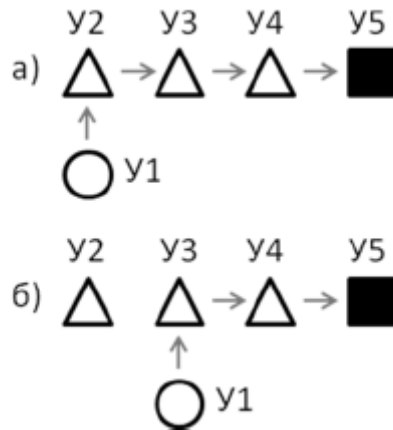


Рисунок 5.12 – Топологія мережі в експерименті з рухомих сенсорним вузлом

Положення вузлів мережі на початку експерименту, коли дані проходить через 3 ретранслятора (а) і пізніше, коли дані проходять через 2 ретранслятора (б).

У початковий момент часу тумба з сенсорним вузлом У1 розташовувалася поряд з ретранслятором У2. Дані від сенсора поступають на комп'ютер по маршруту, що проходить через ретранслятори У2, У3 н У4 (рис. 5.12а).

Потім вузол був зміщений в бік ретрансляторів У3 і У4 (рис 5.11б).

Під час проведення експериментів було встановлено, що дані від сенсорного вузла У1 безперервно надходять на базову станцію і відображаються на екрані комп'ютера. в тому числі при переміщенні мережі.

## ВИСНОВКИ

У даній магістерській дисертації розглянуті загальні відомості про сенсорні мережі та їхнє використання. Один із найоптимальніших варіантів для використання сенсорних мереж є моніторинг і контроль процесів роботи. Слід зазначити, що використання бездротових сенсорних мереж вимагає низьких енергозатрат, щоб вони могли працювати довгий період без зміни джерела живлення.

В дисертації розглянуті особливості структури ПЛІС та її ефективного використання для побудови бездротової сенсорної мережі. Під час виконання магістерської дисертації була створена і проаналізована сенсорна мережа.

Так, як ключовим показником БСМ, як і раніше залишається час їх автономної роботи від батареї, важливим завданням є вибір ефективного протоколу передачі даних, який дозволить оптимізувати споживання енергії мережі. Тому була проведена порівняльна характеристика за такими параметрами як енергоспоживання. Був проведений огляд і аналіз структурної схеми розробленого прийомопередавача, з використанням ПЛІС, для зменшення енергозатрат сенсорних мереж. Проведено дослідження енергоспоживання прийомопередавача і проаналізовані шляхи зниження енергоспоживання для забезпечення більш тривалої автономної роботи. Отримані експериментальні результати показують ефективність запропонованого методу для зниження енергозатрат при прийомі і передачі даних.

Подальший напрямок досліджень доцільно направити на підвищення пропускної здатності і поліпшення енергоефективності мережі в залежності від ймовірності бітової помилки, при використанні коригувальних кодів.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Altium Limited. Altium Designer 6.9 [Електронний ресурс]. – Режим доступу: [www.altium.com](http://www.altium.com) - Дата доступу: вересень 2018.
2. Xilinx. Xilinx [Електронний ресурс]. – Режим доступу: [www.xilinx.com](http://www.xilinx.com) - Дата доступу: вересень 2018.
3. Sensor Networks. Georgia Institute of Technology [Електронний ресурс]. – Режим доступу: [www.craigulmer.com/portfolio/unlocked/000919\\_sensorsimii/wireless\\_sensor\\_networks.ppt](http://www.craigulmer.com/portfolio/unlocked/000919_sensorsimii/wireless_sensor_networks.ppt) - Дата доступу: вересень 2018.
4. Time synchronized mesh protocol. International Symposium, Distributed Sensor Networks [Електронний ресурс]. – Режим доступу: <http://robotics.eecs.berkeley.edu/~pister/publications/2008/TSMR%20DSN08.pdf> - Дата доступу: вересень 2018.
5. Shenyang. Courtesy of Shenyang Institute of Automation 2014 [Текст]. – China, 2008
6. The wireless embedded Internet. [Електронний ресурс]. – Режим доступу: <http://elektro.upi.edu/pustaka.elektro/Wireless%20Sensor%20Network/6LoWPAN.pdf> - Дата доступу: вересень 2018.
7. Sensinode. Available. [Електронний ресурс]. – Режим доступу: [www.sensinode.com/EN/products/software.html](http://www.sensinode.com/EN/products/software.html) - Дата доступу: вересень 2018.
8. Evaluation kit. Texas Instruments. [Електронний ресурс]. – Режим доступу: [www.ti.com/tool/CC610WPAN-DK-868](http://www.ti.com/tool/CC610WPAN-DK-868) - Дата доступу: вересень 2018.
- 9 Алгоритмы маршрутизации. [Електронний ресурс]. – Режим доступу: <http://citforum.ru/nets/ito/2.shtml> - Дата доступу: вересень 2018.
- 10 Зотов В.Ю. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPACK ISE.[Текст]. –Москва, 2003. – стор. 624.
- 11 Проектирование компьютерных систем. типы данных. методы проектирования [Електронний ресурс]. – 2016. – Режим доступу: [https://university-sandbox.raccoongang.com/courses/course-v1:UDHTU+UDHTU\\_CS+2016\\_12/about](https://university-sandbox.raccoongang.com/courses/course-v1:UDHTU+UDHTU_CS+2016_12/about). - Дата доступу: вересень 2018.

12 Применение ПЛИС в цифровой схемотехнике. [Электронный ресурс]. - 2004 – Режим доступа: <http://vkoopilke.narod.ru/pls.html> - Дата доступа: вересень 2018.

13 Д.О. Москалец. Надвеликі інтегральні схеми з програмованою архітектурою в системах спеціального призначення [Текст]. – стор.3.

14. [Системы решение систем уравнений и неравенств](http://www.cleverstudents.ru/systems/solving_systems_Gauss_method.html) [Электронный ресурс]. – Режим доступа: [http://www.cleverstudents.ru/systems/solving\\_systems\\_Gauss\\_method.html](http://www.cleverstudents.ru/systems/solving_systems_Gauss_method.html) - Дата доступа: травень 2018.

15 Altium Innovation Station [Электронный ресурс]. – Режим доступа: [www.altium.com/AltiumInnovationStation](http://www.altium.com/AltiumInnovationStation) - Дата доступа: вересень 2018.

16 Володин К. И. Щербань А. Б. Распределенные интеллектуальные информационные системы. [Текст] - 2005. – стор. 67-89.

17 Володин К. И. Ладиков С. В. Система сбора информации для распределенной сенсорной сети [Текст] - 2006. – стор.44-66.

18 Володин К. И. Даянов М.Х. Сенсорные сети. [Текст] - 2005 - стор.75-91.

19 Волков Е.А. Числительный метод [Текст]. Москва, 1987 – стор 148.

20 The design of an SRAM-based field-programmable gate array—Part 1: Architecture. IEEE Transactions on VLSI Systems, The design of an SRAM-based field-programmable gate array—Part 1: Architecture. IEEE Transactions on VLSI Systems. [Текст] - 2005 - стор.89-123.

21 Н. Fan, J. Liu, Y. L. Wu, C. C. Cheung. On optimum switch box designs for 2-D FPGAs. Proceedings of the 38th ACM/SIGDA Design Automation Conference (DAC). [Текст] - 2007 - стор.323.

22 Xilinx, Inc. Xilinx XC4000E and XC4000X Series Field-Programmable Gate Arrays, Product Specification. [Текст] - 2003 - стор.453.

23 Altem Corp. Altera Strati Device Handbook. [Текст] - 2009 - стор.73.

24 Xilinx, Inc. Xilinx Virtex-II Pro m and Virtex-II Pro P Platform FPGA Handbook. [Текст] - 2005 - стор.235.