

**ІВАНО-ФРАНКІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ НАФТИ І ГАЗУ**

На правах рукопису

**Яцків Наталія Георгіївна**

**УДК 004.627+004.415**

**МЕТОДИ ТА ЗАСОБИ СТИСНЕННЯ ДАНИХ В РОЗПОДІЛЕНИХ  
КОМП'ЮТЕРНИХ СИСТЕМАХ НА ОСНОВІ КОДІВ ПОЛЯ ГАЛУА**

**Спеціальність 05.13.13 – обчислювальні машини, системи та мережі**

**дисертація на здобуття наукового ступеня кандидата  
технічних наук**

**Науковий керівник**

**Николайчук Ярослав Миколайович**

**доктор технічних наук, професор**

**Івано-Франківськ – 2003**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
РОЗДІЛ 1	
ДОСЛІДЖЕННЯ ДЖЕРЕЛ ІНФОРМАЦІЇ ТА ОЦІНКА НАДЛИШКОВОСТІ КОДУВАННЯ ЇХ СТАНІВ В КОМП'ЮТЕРНИХ СИСТЕМАХ	14
1.1. Класифікація джерел інформації. Приклади різних об'єктів, які є джерелами інформації в розподілених комп'ютерних системах	14
1.2. Системні характеристики джерел інформації	18
1.2.1. Стани: технологічні, інформаційні, семантичні	22
1.2.2. Число і ансамбль контрольованих параметрів у заданому стані	24
1.2.3. Число рівнів кожного параметра і крок дискретизації	26
1.2.4. Ентропія станів джерела інформації	28
1.3. Класифікація та перспективи розвитку методів стиснення даних	30
1.4. Аналіз ефективності теоретико - числових базисів	35
ВИСНОВКИ	43
РОЗДІЛ 2	
МЕТОДОЛОГІЯ ТА ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ МЕТОДІВ ЗМЕНШЕННЯ НАДЛИШКОВОСТІ ЦИФРОВИХ ДАНИХ У КОМП'ЮТЕРНИХ СИСТЕМАХ	44
2.1. Логіко – статистичні та інтегрально – імпульсні моделі інформаційного стану джерел інформації	44
2.2. Методика оцінки коефіцієнта стиснення даних	51
2.3. Теоретичне обґрунтування потенціальних можливостей стиснення даних у базисі Крестенсона	58

2.4. Теоретичні основи та потенціальні можливості кодування цифрових даних у базисі Галуа на основі вертикальної інформаційної технології	64
2.5. Дослідження базисних функцій теоретико-числового перетворення Галуа	72
ВИСНОВКИ	79
РОЗДІЛ 3	
РОЗРОБКА ТА ДОСЛІДЖЕННЯ МЕТОДІВ СТИСНЕННЯ ДАНИХ У БАЗИСІ ГАЛУА	
3.1. Метод стиснення даних, представлених гармонічними сигналами	80
3.2. Одновимірні джерела інформації	91
3.3. Багатоканальні джерела інформації	95
3.4. Багатовимірні джерела інформації	98
3.5. Дослідження методу інтегрально-імпульсного кодування у базисі Галуа	104
ВИСНОВКИ	109
РОЗДІЛ 4	
РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ЗМЕНШЕННЯ НАДЛИШКОВОСТІ ІНФОРМАЦІЙНИХ ПОТОКІВ	
4.1. Інтегрально-імпульсний перетворювач з розширеними функціональними параметрами для систем обліку енергоносіїв	110
4.2. Спецпроцесор обробки даних на основі перетворення Крестенсона-Галуа	118
4.3. Процесор стиснення даних на основі базисних функцій Галуа	122
4.4. Структура спецпроцесора формування потоків даних логіко-статистичних інформаційних моделей	128

4.5. Алгоритм та програма стиснення даних в контролерах енергетичних систем	130
4.5.1. Функціональна схема програмного забезпечення	132
4.5.2. Програмна реалізація методу стиснення даних енергетичних об'єктів	134
ВИСНОВКИ	137
ОСНОВНІ ВИСНОВКИ	138
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	140
ДОДАТКИ	149

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

РКС	– розподілена комп'ютерна система
НКМ	– низова комп'ютерна мережа
ОК	– об'єкт керування
ДІ	– джерело інформації
ТЧП	– теоретико-числове перетворення
ЛСІМ	– логіко-статистичні інформаційні моделі
НГА	– нафтогазова автоматика
ЦОС	– цифрова обробка сигналів
СЗК	– система залишкових класів
ІКМ	– імпульсно-кодова модуляція
ІК	– інтегрально – імпульсне кодування
ВІТ	– вертикально-імпульсна технологія
ІІІ	– інтегрально-імпульсний перетворювач
ПНЧ	– перетворювач напруга – частота
ПЛІС	– програмована логічна інтегральна схема

## ВСТУП

Швидкий розвиток науки і техніки вимагає широкого використання пристроїв обробки, передавання і запам'ятовування постійно зростаючих об'ємів даних.

Розвиток інформаційних мереж і систем, зокрема автоматизованих систем різних рівнів, систем передавання звукових повідомлень і телевізійних зображень, комунікаційних систем, систем передавання вимірювальної, біомедичної або радіолокаційної інформації, систем контролю, телекерування і телесигналізації, обчислювальних мереж, систем автоматизованого проектування, збору і обробки даних, інформаційно-довідкових систем привів до значного збільшення інформаційних потоків між територіально розподіленими джерелами і споживачами повідомлень.

Це стимулювало дослідження задач підвищення ефективності систем передавання даних з обмеженою пропускнуою здатністю. Такі дослідження дозволили виявити що велика частина повідомлень, які передаються по каналах зв'язку є малоінформативними, тобто несуть надлишкові або несуттєві відомості. В результаті виникла задача стиснення об'єму даних, які передаються або запам'ятовуються, шляхом усунення природної надлишковості дискретних та неперервних повідомлень.

Для підвищення ефективності використання комунікаційних та інформаційно-обчислювальних ресурсів вказаних систем використовують різні методи і засоби.

Комплекс задач, вирішення яких забезпечить створення високоефективних систем зі стисненням даних включає, в першу чергу, синтез і аналіз методів кодування станів джерел інформації і розробки апаратно-програмних засобів для їх реалізації на базі сучасної цифрової техніки.

Особливо гостро проблема стиснення даних постала в даний час, в зв'язку з необхідністю обробляти і передавати великі масиви даних або

реєструвати і зберігати їх в запам'ятовуючих пристроях.

Завдяки використанню методів стиснення даних суттєво знижується собівартість формування, передавання, цифрової обробки та зберігання даних в інформаційних системах та комп'ютерних мережах.

При застосуванні досліджуваних методів в розподілених комп'ютерних системах реального часу ефективно вирішується також задача підвищення їх надійності та адаптації до режимів роботи в умовах відхилення технологічних процесів від норми, виявлення передаварійних ситуацій та робота в умовах частих переривань при ліквідації аварійних станів джерел інформації. Перспективною науково-технічною задачею в цій галузі є можливість ідентифікації станів джерел інформації без декодування стиснених даних.

**Актуальність теми.** Швидкий розвиток інформаційних систем і мереж, до яких належать розподілені комп'ютерні системи (РКС), обчислювальні мережі, системи збору та обробки даних, зумовлює істотне збільшення інформаційних потоків між територіально розподіленими джерелами та користувачами повідомлень, а також породжує задачу зберігання постійно зростаючих обсягів інформації.

Для підвищення ефективності використання комунікаційних та інформаційно-обчислювальних ресурсів вказаних систем застосовують методи зменшення надлишковості даних. Це дає змогу значно розвантажити канали зв'язку та системи обробки і зберігання даних.

Пошуку нових шляхів стиснення даних на основі ефективного кодування та теоретико-числових перетворень (ТЧП) присвячено багато наукових робіт. Зокрема, необхідно відзначити значний внесок відомих вчених: Харкевича О. О., Вошні О. Г., Макса Ж., Хартлі Р., Рабінера Л., Рейдера У., Голда Б., Акушського І. Я., Николайчука Я. М., Ольховського Ю. Б.

В той же час застосування найпоширеніших базисів для реалізації методів зменшення надлишковості даних, в тому числі базисів Радемахера,

Крейга, Уолша, перетворень Адамара та інших нерекурентних ТЧП, на сьогодні вже майже вичерпало свої можливості щодо їх вдосконалення та розширення сфери застосування.

Практика розробки теоретичних основ вертикальної інформаційної технології на основі рекурентних ТЧП у базисі Галуа відкриває нові можливості створення ефективних методів зменшення надлишковості цифрових даних і розвитку сфери їх застосування для ряду інформаційних потоків, що характеризуються специфічними властивостями. Наприклад, зі стрибкоподібними змінами амплітуди спектральних, кореляційних та ентропійних характеристик, що, як відомо, призводить до різкого зниження ефективності кодування існуючими методами, а в окремих випадках до збільшення обсягів даних стосовно початкових. Крім того, відомі методи стиснення даних практично не захищені від збоїв і помилок, особливо вставок і стирань у біт-орієнтованих потоках, що ускладнює і знижує ефективність їх застосування в системах реального часу, котрі передбачають керування складними технологічними об'єктами.

Тому розробка, дослідження і розвиток ефективних методів стиснення цифрових даних, котрі краще адаптовані до системних характеристик специфічних квазістаціонарних джерел інформації (ДІ) та забезпечують меншу чутливість стиснених даних до збоїв і помилок у процесі їх передавання, зберігання і цифрової обробки, а також створення відповідних апаратних та програмних засобів стиснення даних, є актуальною науково-технічною задачею.

***Зв'язок роботи з науковими програмами, планами, темами.***

Дисертаційна робота виконана в рамках плану наукових досліджень, що проводяться кафедрою комп'ютерних технологій і систем управління (КТ і СУ) Івано-Франківського Національного технічного університету нафти і газу (ІФНТУНГ) та Карпатським державним центром інформаційних засобів і технологій НАН України (КД ЦІЗІТ) за держбюджетними темами ІФНТУНГ:



– Д-60 НДР “Розробка теоретичних основ вертикальної інформаційної технології та вдосконалення методів автоматизованого контролю розподілених об’єктів нафтогазовидобувних підприємств”, № 01940001278;

– НДР “Розробка методологічних основ (теорії, моделей, алгоритмів, процедур і технічних засобів) діагностування і автоматизованого управління об’єктами нафтогазового комплексу України” (ТЗ ИГАУ-573/99).

Кафедрою спеціалізованих комп’ютерних систем (СКС) Тернопільської академії народного господарства (ТАНГ): НДР “Розробка теорії, методології побудови та технічних засобів спеціалізованих комп’ютерних систем”, № 0102U00576А.

Науково-технічною програмою КД ЦЗІТ та інституту мікропроцесорних систем керування об’єктами електроенергетики (ІМСКОЕ): “Створення концепції інформаційно-керуючих систем контролю та обліку енергоносіїв, систем керування технологічними процесами електроенергетичних об’єктів та захисту їх обладнання”, Львів, 2000 р.

Усі вищеназвані роботи виконувались за безпосередньою участю автора.

**Мета і задачі дослідження.** Метою роботи є вирішення актуальної науково-технічної задачі розвитку методів стиснення даних на основі розробки теоретичних положень, методів, технічних і програмних засобів зменшення надлишковості цифрових даних у базисі Галуа.

Для досягнення мети необхідно вирішити наступні задачі:

– проаналізувати існуючі методи та інформаційні технології стиснення даних у комп’ютерних системах на основі представлення потоків повідомлень в різних базисах: унітарному, Крестенсона, Хаара, Крейга, Радемахера та Галуа;

– визначити взаємозв’язок між системними параметрами та станами джерел інформації, що характеризують ентропію;

– провести теоретичне та експериментальне дослідження, вдосконалити існуючі та розробити нові методи стиснення даних на основі кодів поля

Галуа;

– дослідити можливість використання нових принципів стиснення даних для зменшення надлишковості в розподілених комп'ютерних та телекомунікаційних системах;

– здійснити апробацію запропонованих методів в діючих розподілених комп'ютерних системах (РКС);

– розробити і впровадити технічні засоби та інформаційні технології для реалізації розроблених методів стиснення даних в промисловості.

*Об'єкт досліджень* – низові розподілені комп'ютерні мережі автоматизованих систем зі стаціонарними та квазістаціонарними джерелами інформації.

*Предмет дослідження* – методи та засоби стиснення даних в базисі Галуа, які забезпечують ефективне зменшення надлишковості цифрових даних в РКС.

*Методи дослідження.* Для розробки методів і засобів стиснення даних у комп'ютерних системах використовувались методи теорії інформації, статистичного аналізу, теорії кодування у різних базисах і прикладної теорії цифрових автоматів.

Дослідження інформаційних технологій кодування даних у базисі Галуа, проведені з метою визначення оптимальних умов їх застосування, ґрунтуються на методах ентропійних досліджень.

Під час проведення експериментальних досліджень та обробки їх результатів використовувались методи теорії ймовірності, а також методи цифрового моделювання і програмної емуляції роботи цифрових процесорів обробки даних. Розробка технічних засобів здійснювалась з використанням методів схемо - та системотехніки.

***Наукова новизна одержаних результатів:***

– вдосконалені логіко-статистичні інформаційні моделі шляхом заміни унітарного коду послідовностями Галуа, що дало змогу кодувати мітки часу без введення надлишковості;

– вперше запропоновано базисні функції Галуа нульового та першого порядку, на основі яких розроблено нові методи стиснення цифрових даних, які забезпечують зменшення обсягів даних у 3-8 разів при кодуванні процесів, для яких класичні методи практично непрацездатні, що створює умови їх ефективного використання у стандартних системах передавання, архівації і зберігання даних;

– вперше запропоновано метод стиснення даних, представлених гармонічними сигналами, що базується на реєстрації  $1/8$  періоду сигналу, та кодуванні повторів бітами Галуа, коефіцієнт стиснення дорівнює 7,6 раза;

– розроблено модифікований метод адаптивного кодування даних в базисі Галуа, що забезпечує стиснення даних у 1,6-2 рази при 50% активних відліків і не вводить надлишковості на відміну від існуючих при 90% активних відліків;

– розроблено новий метод кодування багатовимірних джерел інформації на основі перетворення Крестенсона–Галуа, який дає змогу перейти від багатовимірного до одновимірного представлення даних, забезпечує зменшення службових даних при передаванні інформації, криптографічне кодування і захист від помилок.

#### ***Практичне значення одержаних результатів:***

– розроблено та впроваджено технічні засоби і технології, які базуються на використанні запропонованих методів стиснення даних, що дало змогу підвищити ефективність роботи РКС;

– розроблено алгоритми та програми стиснення даних, що суттєво зменшують надлишковість і тим самим знижують собівартість формування, зберігання і передавання інформації у комп'ютерних мережах;

– розроблено та виготовлено дослідний взірець інтегрально-імпульсного перетворювача з розширеними функціональними можливостями, що дало можливість розширити сферу його застосування.

*Теоретичні і практичні результати дисертаційної роботи використані та впроваджені:*

– у дослідно-конструкторських роботах, що проводились КД ЦІЗІТ НАН України та ІМСКОЕ “Створення концепції інформаційно-керуючих систем контролю та обліку енергоносіїв, систем керування технологічними процесами електроенергетичних об’єктів та захисту їх обладнання”;

– на ВАТ “Івано-Франківськцемент” у комп’ютерній системі контролю та обліку витрати енергоносіїв “ALFIYA”;

– у навчальному процесі кафедри СКС Тернопільської академії народного господарства при проведенні лекційних, практичних і лабораторних занять з курсів “Компресія даних”, “Низові комп’ютерні мережі” та “Системи передавання даних”.

**Особистий внесок здобувача.** Основні положення і результати дисертаційної роботи отримані автором самостійно. У публікаціях, написаних у співавторстві, здобувачеві належить: [2] – архітектура комп’ютерної системи ідентифікації семантичних, технологічних та інформаційних станів; [3] – аналітичні залежності коефіцієнта стиснення даних багатоканальних комп’ютерних систем збору технологічних даних; [4] – обґрунтування і класифікація ДІ та визначення їхніх системних характеристик з точки зору квазістаціонарних властивостей, розрахунок ентропії квазістаціонарних ДІ для різних методів зменшення надлишковості даних; [6] – структура спецпроцесора обробки даних на основі перетворення Крестенсона – Галуа ; [7] – аналіз методів цифрового кодування даних на фізичному рівні комп’ютерних систем ; [8] – дослідження методу стиснення даних на основі кодів поля Галуа в багатоканальних системах; [9] – метод кодування багатовимірних ДІ; аналітичні залежності коефіцієнта стиснення при різних методах кодування багатовимірних ДІ в базисі Крестенсона та Галуа.

**Апробація результатів дисертації.** Основні результати дисертаційної роботи доповідались та обговорювались на: Міжнародній конференції “Сучасні проблеми радіоелектроніки, телекомунікацій та комп’ютерної інженерії” TCSET`2002 – Львів-Славськo, 2002; Міжнародній науково-

практичній конференції “Автоматизація виробничих процесів” МНПК АВТ–2002 – Хмельницький, 2002; Науково-технічній конференції “Вимірювальна та обчислювальна техніка в технологічних процесах” – Хмельницький, 2002; Міжнародній конференції “Досвід проектування і застосування САПР в мікроелектроніці” CADSM 2003 – Львів-Славсько, 2003, а також доповідалось та обговорювалось на наукових семінарах кафедр КТ і СУ ІФНТУНГ та СКС ТАНГ.

**Публікації.** Результати досліджень опубліковані у 9 друкованих працях автора, з них 7 статей у фахових журналах та збірниках наукових праць, 2 доповіді у збірниках науково-технічних конференціях.

**Структура дисертації.** Дисертаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг роботи становить 177 сторінок, з них 139 сторінок основного тексту, що включає 66 рисунків і 8 таблиць. Список використаних джерел – 83 найменування, додатки – на 29 сторінках.

## РОЗДІЛ 1

## ДОСЛІДЖЕННЯ ДЖЕРЕЛ ІНФОРМАЦІЇ ТА ОЦІНКА НАДЛИШКОВОСТІ КОДУВАННЯ ЇХ СТАНІВ У КОМП'ЮТЕРНИХ СИСТЕМАХ

### 1.1. Класифікація джерел інформації. Приклади різних об'єктів, які є джерелами інформації у розподілених комп'ютерних системах

Сучасні автоматизовані комп'ютерні системи низового рівня призначені для роботи в реальному часі і найчастіше представляються трирівневою зірково-магістральною архітектурою (рис. 1.1) [1, 2].

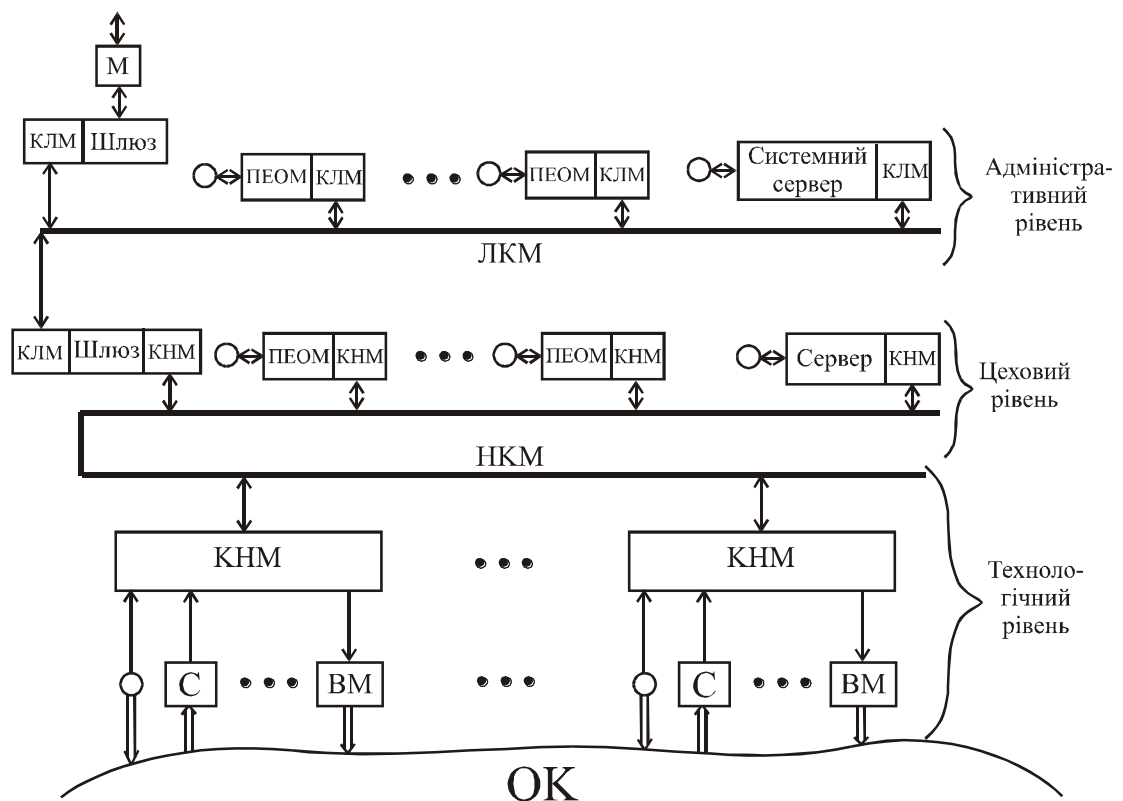


Рис. 1.1. Тривірнева зірково-магістральна архітектура автоматизованої системи: ОК – об'єкт керування; С – сенсор; ВМ – виконавчий механізм; КНМ – контролер низової мережі; М – модем; О – оператор; КЛМ – контролер локальної мережі; НКМ – низова комп'ютерна мережа; ЛКМ – локальна комп'ютерна мережа.

Особливістю функціонування такої архітектури є можливість колективного використання ресурсів різних сенсорів ОК. При цьому КНМ виконують три системні функції:

- а) управління технологічними процесами в режимі локальної автоматики;
- б) комунікаційне транспортування інформації від сенсорів в ПК верхніх рівнів і команд управління ВМ;
- в) цифрова обробка сигналів сенсорів, формування інформаційних моделей, архівація на рівні КНМ.

Як правило функціонування таких систем здійснюється під керівництвом стандартних операційних систем OS/2, Windows 9x, Windows NT, Unix, Linux.

В роботі [3] приведена класифікація ДІ, яка включає наступні їх типи:

- одноканальне однобітове (рис. 1.2 а);
- одноканальне багаторівневе (рис. 1.2 б);
- багатоканальне однобітове (рис. 1.2 в);
- багатоканальне багаторівневе (рис. 1.2 г).

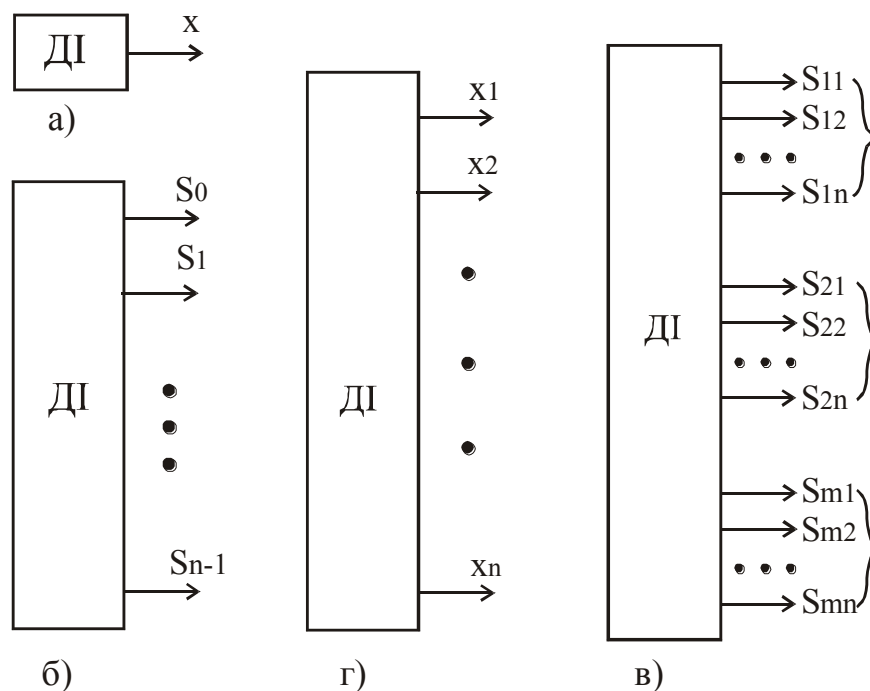


Рис. 1.2. Типи ДІ:  $x_n$  – булеві сигнали;  $S_{mn}$  – кодові сигнали.

Одержана інформація про ОК та про сигнали, що його описують, подається у формах:

- доступних безпосередньо вимірюванню входів, виходів, змінних стану;
- моделей ОК, ДІ або сигналів;
- оцінок сигналів та станів.

Властивості кожного з вищезгаданих класів ДІ можуть бути описані у вигляді сигнальної, статистичної та ентропійної моделей.

Інформація про ДІ надається у НКМ у цифровому вигляді. Тому у названих моделях доцільно дослідити системні властивості різних класів дискретних ДІ, відповідно властивості аналогових ДІ можуть бути описані в класі дискретних ДІ з певною заданою точністю.

Визначення інформаційних станів ОК ґрунтується на теорії і технології кодування дискретних ДІ. При цьому найбільш актуальним завданням є кодування квазістаціонарних ДІ, до яких відносять широкий клас промислових об'єктів.

Приклади різних об'єктів, які є джерелами інформації. Методологічні основи побудови низових комп'ютерних мереж розроблені в роботі [2], де визначена класифікація системних об'єктів низових мереж, згідно якої ДІ є наступні п'ять системних об'єктів (рис. 1.3):

- а) ЕОМ;
- б) користувачі;
- в) системи передавання даних;
- г) дані;
- д) об'єкти керування.

Кожний з вищеназваних об'єктів є функціональним об'єктом наступного типу [2]:

- а) джерелом інформації;
- б) середовищем перетворення даних;
- в) середовищем передавання даних;



г) приймачем інформації.

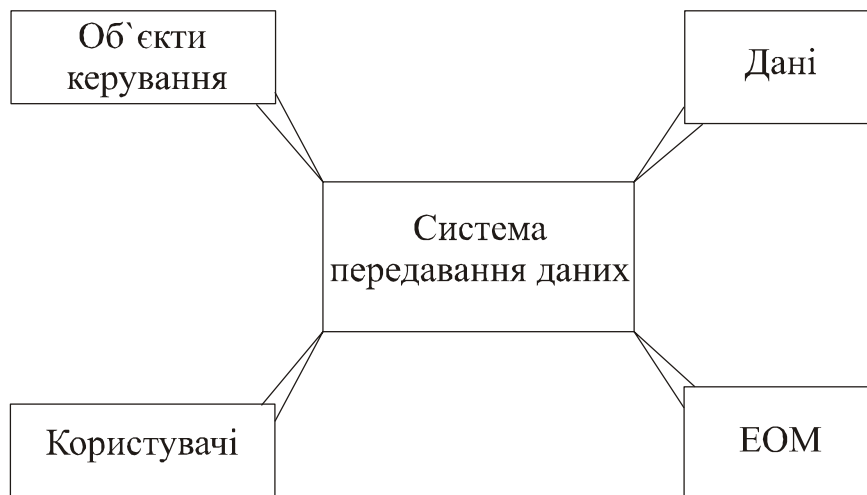


Рис. 1.3. Об'єкти НКМ.

Взаємодія системних і функціональних об'єктів представляє узагальнену архітектуру будь-якої комп'ютерної системи низового рівня (рис. 1.4) [2].

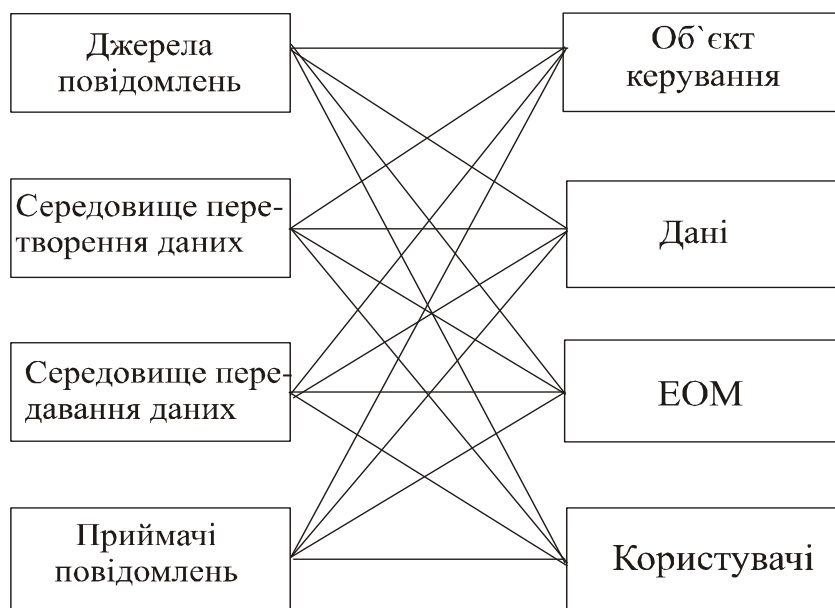


Рис. 1.4. Взаємодія функціональних і системних об'єктів НКМ.

Таким чином, задача визначення різних системних об'єктів, які можуть бути ДІ, зводиться до опису інформаційних характеристик кожного з названих системних об'єктів (додаток А).

## 1.2. Системні характеристики джерел інформації

Джерела інформації досліджуються:

- з метою ідентифікації їх станів;
- як об'єкти керування і регулювання;
- як джерела повідомлень для дистанційного передавання інформації;
- як джерела даних, що підлягають збору, передаванню, обробці і зберіганню в базі даних.

Існує два великих класи джерел інформації:

- джерела аналогових сигналів –  $x(t)$ ;
- джерела дискретних сигналів –  $x_i$ .

Джерела аналогових сигналів описуються наступними системними характеристиками (рис. 1.5):

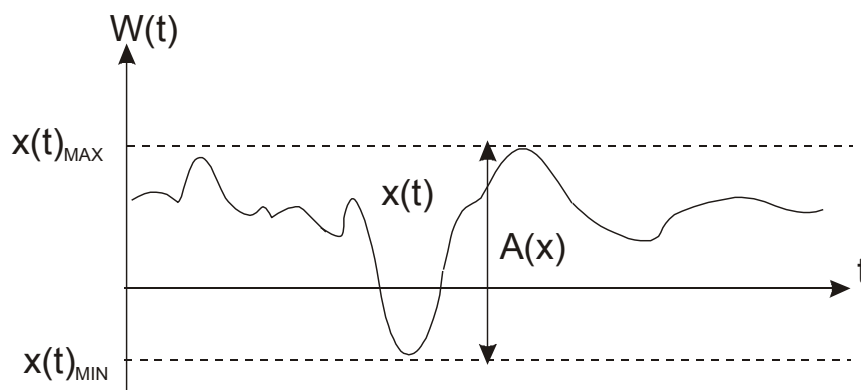


Рис. 1.5. Сигнал аналогового ДІ.

а) максимальним значенням амплітуди

$$A_x = x(t)_{\max};$$

б) смугою частот:

$$\Delta F = f_{\max} - f_{\min},$$

де  $f_{\max}$ ,  $f_{\min}$  – відповідно максимальна і мінімальна частота;

в) статистичним розподілом амплітуд  $P(X)$ ;

г) математичним очікуванням  $M_x$ ;

д) середньоквадратичним відхиленням  $\sigma_x$ ;

е) автокореляційною функцією  $R_{xx}(\tau)$ ;

є) спектром  $S(\omega)$ .

Дискретні джерела інформації описуються наступними характеристиками:

$$W(t) = F[A_x, M_x, D_x, \sigma_x, R_{xx}(j)],$$

де  $A_x$  – діапазон квантування,  $A_x = x(t)_{\max} - x(t)_{\min}$ ;

$M_x$  – математичне очікування;

$D_x$  – дисперсія;

$\sigma_x$  – середньоквадратичне відхилення;

$R_{xx}(j)$  – автокореляційна функція.

Залежно від характеру поведінки в часі ДІ поділяються на детерміновані, стаціонарні випадкові (рис. 1.6 а), квазістаціонарні (рис. 1.6 б) та нестаціонарні ДІ [4].

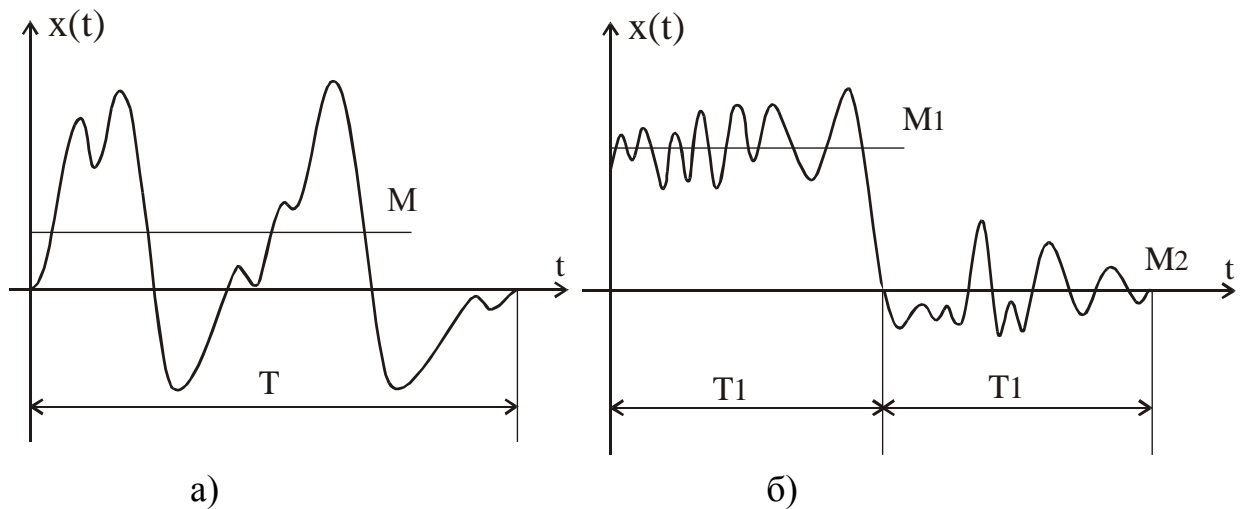


Рис. 1.6. Сигнальна модель ДІ: а) стаціонарного; б) квазістаціонарного: М – математичне очікування; Т – час спостереження.

При цьому умова стаціонарності описується наступними рівняннями:  $M_x = const$ ,  $D_x = const$ ,  $R_{xx}(\tau) = const$ ,  $S(\omega) = const$ , які визначаються згідно формул [2]:

$$M_x = \frac{1}{n} \cdot \sum_{i=1}^n x_i;$$

$$D_x = \frac{1}{n} \cdot \sum_{i=1}^n (x_i - M_x)^2 ;$$

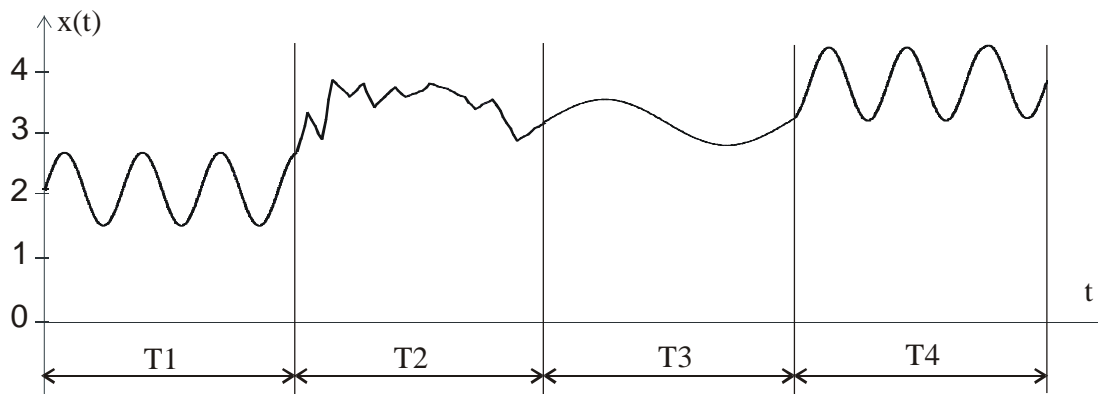
$$R_{xx}(j) = \frac{1}{n} \cdot \sum_{i=1}^n ((x_i - M_x) \cdot (x_{i+j} - M_x)) ;$$

$$S(\omega_i) = \frac{1}{r} \sum_{j=0}^m \left[ \frac{R_{xx}(j)}{D_x} \cdot \cos(\omega_{ij}) \right]^2 ,$$

де  $r$  – інтервал кореляції.

Якщо одна з цих характеристик стрибкоподібно змінюється в часі, то такі ДІ є квазістаціонарними. Якщо хоча б одна з характеристик змінюється на всьому інтервалі, то такі ДІ є нестаціонарними.

Сигнальна модель квазістаціонарного ОК може бути використана для порівняння ефективності різних методів зменшення надлишковості (рис. 1.7).



T1	T2	T3	T4
$M_1$	$M_2$	$M_2$	$M_3$
$D_1$	$D_1$	$D_2$	$D_1$
$R_{xx_1}(j)$	$R_{xx_2}(j)$	$R_{xx_3}(j)$	$R_{xx_1}(j)$
$S_1(\omega)$	$S_2(\omega)$	$S_3(\omega)$	$S_1(\omega)$

Рис. 1.7. Сигнальна модель квазістаціонарного процесу:

T1 – T4 – час спостереження.

Нестаціонарні ДІ описуються наступними рівняннями:  $M_x = \text{var}$  ,

$$D_x = \text{const}; R_{xx}(\tau) = \text{const}; S(\omega) = \text{const}.$$

Аналіз літературних джерел показав, що кожна з системних характеристик має наступні модифікації [2, 4]:

- а) математичне очікування – вибіркове, ковзне, вагове;
- б) дисперсія – стаціонарна, зважена;
- в) автокореляційна функція – полярна, знакова, коваріаційна, структурна, модульна, еквівалентності;
- г) спектри в базисах Уолша, Хаара, Крестенсона, Фур'є.

Для побудови інформаційних моделей ДІ необхідно провести розрахунок системних характеристик, на основі яких можливе зменшення надлишковості.

Досвід експлуатації автоматизованих комп'ютерних систем у промисловості показує, що об'єкти керування, представлені технологічними установками, системи обліку витрати енергоносіїв та інші належать до класу квазістаціонарних ДІ.

Водночас, на технологічному рівні дуже важливою є операція кодування станів ДІ, яка б передбачала зменшення надлишковості цифрових даних та оснащення контролерів низового рівня потужним математичним і програмним забезпеченням для автоматичного дослідження системних характеристик ДІ.

Ефективна реалізація методів стиснення даних зменшує вимоги до пропускної здатності каналів зв'язку та обсягів пам'яті ЕОМ на верхніх рівнях автоматизованих комп'ютерних систем.

Відомі методи зменшення надлишковості даних: адаптивні [5, 6, 7], методи приростів, дельта – модуляція [8] є недостатньо ефективними для кодування квазістаціонарних ДІ.

Тому задача розробки інформаційних технологій та алгоритмів кодування і зменшення надлишковості для квазістаціонарних та нестаціонарних об'єктів є особливо актуальною.

### 1.2.1. Стани: технологічні, інформаційні, семантичні

Аналіз світового рівня, досвід впровадження та тенденції розвитку нафтогазової автоматики підтверджують її важливу роль у вдосконаленні технології видобутку, транспорту та зберігання нафти і газу.

Окрім цього, степінь оснащення технологічних установок засобами автоматики та функціональні можливості влаштованого мікропроцесорного інтелекту на рівні сенсорів суттєво впливає на рівень техногенно–екологічної безпеки паливно–енергетичного комплексу. Важливим фактором забезпечення екологічної безпеки технологічних установок нафтогазового комплексу є необхідний високий рівень надійності нафтогазового обладнання, апаратури автоматики, а також її власна іскро– та вибухонебезпечність. Велике значення має адаптація характеристик нафтогазової автоматики (НГА) до конкретних технологічних процесів, які розглядаються як ДІ, що перебувають у різних класифікованих станах. Причому, кожний стан ДІ або ОК несе різну семантичну інформацію про їх екологічну безпеку.

Дослідження нафтогазових технологічних установок як ДІ показує, що вони належать до класу квазістаціонарних ОК. Причому, в кожний конкретний момент часу такі об'єкти класифікуються трьома характеристиками станів [4]:

- а) семантичні ( $S_i; i = 1, 2, \dots, s$ );
- б) технологічні ( $X_j; j = 1, 2, \dots, q$ );
- в) інформаційні ( $I_k; k = 1, 2, \dots, h$ ),

де  $s, q, h$  – відповідна кількість класифікованих станів ОК.

Серед наведених характеристик станів типових технологічних установок нафтогазового комплексу (табл. 1.1) семантичні – відображають і характеризують техногенно–екологічний стан, технологічні стани ОК визначаються класифікацією технологічних операцій, а інформаційні – кількістю та ансамблем контрольованих параметрів.

## Класи станів технологічних установок нафтогазового комплексу

Технологічна установка ОК	Стани ОК		
	семантичні	технологічні	інформаційні
1	2	3	4
Установка буріння	Буріння	Буріння	Буріння
	Викид	СПО (Спуск-підйом обладнання)	СПО
	Аварія	Промивка	Промивка
	Ліквідація	Компановка	Компановка
		Випробування	Простій
		Аварія	
		Ремонт	
		Простій	
Глибинно-насосна установка	Видобуток нафти	Пуск-зупинка	Пуск-зупинка
	Викид	Видобуток	Видобуток
	Аварія	Випробування	Пуск-зупинка
	Ліквідація	Ремонт	Видобуток
		Аварія	Випробування
		Простій	Простій
		Освоєння	
Фонтанна свердловина	Видобуток нафти	Видобуток	Видобуток
	Викид	Випробування	Випробування
	Аварія	Ремонт	Простій
	Ліквідація	Аварія	
		Ліквідація	

1	2	3	4
Товарно-сировинний стан резервуара	Зберігання	Будівництво	Наповнення-злив
	Перелив	Зберігання	Зберігання
	Зливання	Наповнення	Компаундування
	Розгерметизація	Компаундування	Простій
	Ліквідація	Злив	
		Аварія	
		Простій	

Аналогічні принципи ідентифікації станів та підтримки техногенно-екологічної безпеки ОК нафтогазового комплексу реалізуються для засобів діагностики технічного стану нафтогазового обладнання, а також комп'ютерних систем контролю та обліку витрати енергоносіїв.

### 1.2.2. Число і ансамбль контрольованих параметрів у заданому стані

Інформаційна система контролю та ідентифікації станів ОК повинна бути обладнана відповідними засобами автоматики і включати: сенсори ( $C_p$ ); логічні та евристичні аналізатори семантичних ( $S_i$ ), технологічних ( $X_j$ ) та інформаційних ( $I_k$ ) станів ОК; промислові контролери (ПрК), низову комп'ютерну мережу (НКМ) та базовий комп'ютер-сервер диспетчерського центру (рис. 1.8) [4].

Оператор технологічної установки (О) виконує функції евристичної ідентифікації семантичних та екологічних станів ОК, які інваріантні різним технологічним та інформаційним станам ОК, і не можуть бути однозначно визначені логічними аналізаторами  $S_i, X_j, I_k$  автономно.



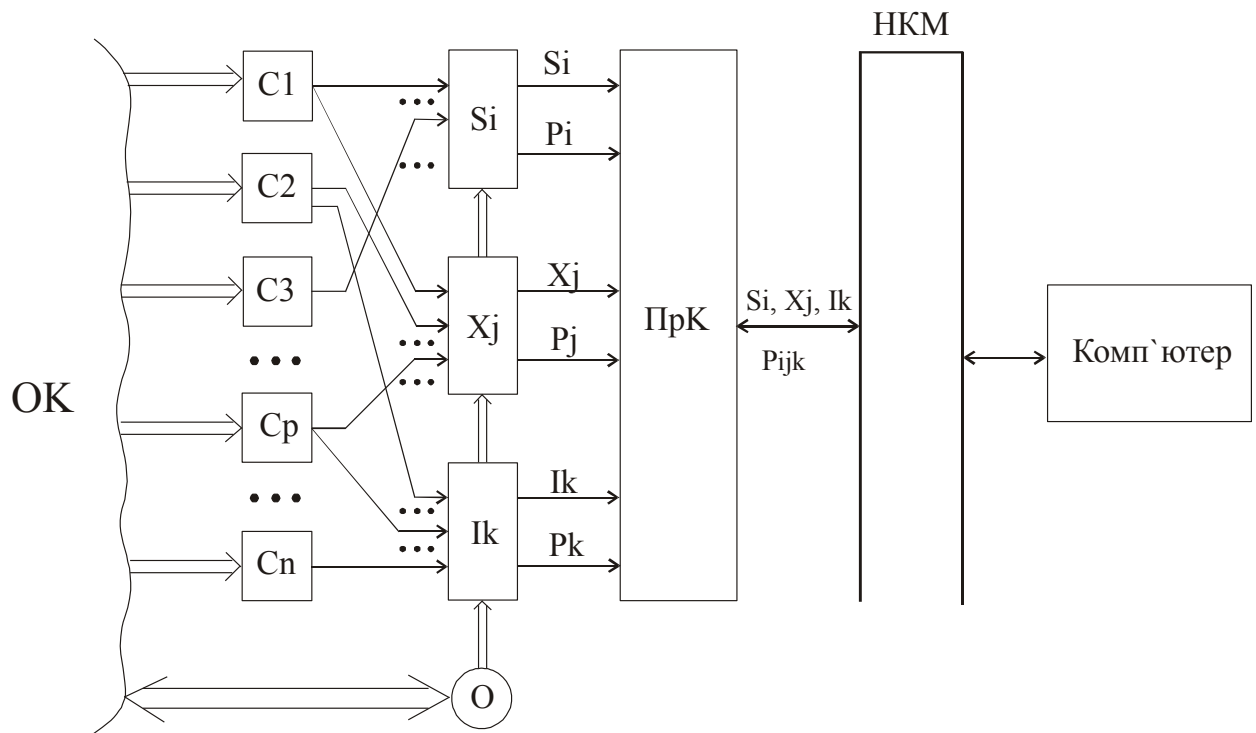


Рис. 1.8. Архітектура системи ідентифікації семантичних, технологічних та інформаційних станів ОК:

О – оператор нафтогазової технологічної установки;  $P_i, P_j, P_k$ , – формалізовані параметри семантичних, технологічних та інформаційних станів ОК;  $P_{ijk}$  – характеристичний параметр екологічного стану ОК.

Приведені у табл. 1.1 приклади класифікації семантичних характеристик станів технологічних установок нафтогазової промисловості є основою для побудови комп'ютерних систем екологічної безпеки.

Інформаційна система контролю та ідентифікації станів ОК дає змогу зменшити надлишковість даних на етапі первинної обробки. Обсяг даних, що формується усіма сенсорами на порядок перевищує обсяг вихідних даних промислового контролера, що через НКМ надходять до комп'ютера–сервера. Скорочення обсягів даних досягається за рахунок кодування стану ОК та формалізованого параметра, що описує даний стан. Якщо об'єкт контролю має 15 сенсорів, кожний з яких формує 10-ти розрядне значення і може перебувати в одному з восьми станів, то застосування запропонованої системи дозволяє скоротити обсяг даних зі 150 біт до 13 біт, а в окремих випадках до 3 біт.

При цьому важливу роль відіграє побудова в реальному масштабі часу логіко-статистичних інформаційних моделей [4, 9, 10].

### 1.2.3. Число рівнів кожного параметра і крок дискретизації

При передаванні, зберіганні та обробці інформації значні переваги має дискретна форма подання сигналів. Коли первинні сигнали інформаційних сигналів є неперервними відбувається їх перетворення на дискретні.

Кількість рівнів квантування залежить від обраної точності відновлення сигналу, чим вища точність, тим більше рівнів потрібно. Найбільш поширене рівномірне квантування, що зумовлено простотою технічної реалізації.

Більшість існуючих телеметричних систем побудовано за принципом передачі миттєвих значень контрольованого неперервного параметру, які розташовані на певній часовій відстані одне від одного, що називається кроком дискретизації. Очевидно, що із зменшенням кроку дискретизації  $\Delta t$ , відновлений сигнал більш точно буде відображати початкову функцію, але це призводить до збільшення часу зайнятості каналу зв'язку, підвищує вимоги до пропускну здатності каналу. Тому крок дискретизації слід вибирати таким, щоб з необхідною точністю отримати початкову функцію.

Вибір кроку дискретизації здійснюється за одним із наступних критеріїв [11, 12]:

– частотний критерій Котельнікова:

$$\Delta t = \frac{1}{2f_c},$$

де  $f_c$  – максимальна частота спектру сигналу;

– кореляційний критерій Желєзнова:

$$\Delta t \leq \tau_0, T_c \gg \tau_0,$$

де  $\tau_0$  – інтервал кореляції,  $T_c$  – тривалість сигналу;

– критерій допустимого відхилення:

а) при кусково-лінійній апроксимації

$$\Delta t = \sqrt{\frac{8\delta_{\max}}{\left(\frac{d^2 x(t)}{dt^2}\right)_{\max}}},$$

б) при ступінчатій апроксимації поліномом нульового порядку

$$\Delta t = \frac{|\delta|_{\max}}{\left(\frac{dx(t)}{dt}\right)_{\max}},$$

де  $\delta$  – похибка апроксимації,  $x(t)$  – значення функції.

Критерій максимального відхилення використовують для детермінованих процесів, коли відомий закон зміни вихідної функції  $x(t)$ .

При випадковому характері вихідної функції доцільно використовувати критерій середньоквадратичного відхилення.

Раціональний вибір кроку дискретизації та кількості рівнів квантування (рис. 1.9) дає можливість усунути першу причину виникнення надлишковості – завищення розмірності вектору даних, що надаються користувачу, коли для правильного уявлення про процес достатня менша точність подання даних. Це дозволяє скоротити обсяг даних на етапі їх формування.

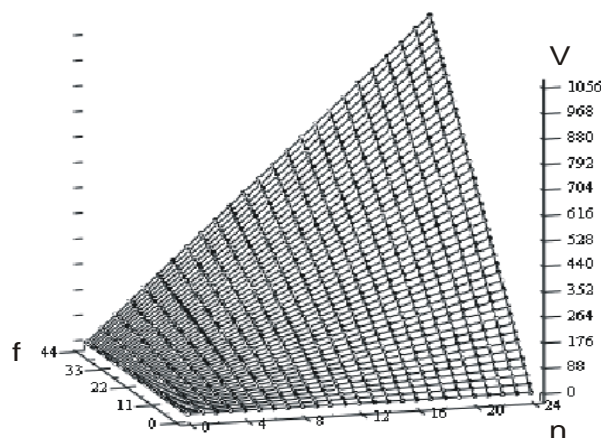


Рис. 1.9. Залежність об'єму даних від зміни частоти дискретизації та кроку квантування:  $V$  – об'єм даних,  $n$  – розрядність даних,  $f$  – частота дискретизації.

Для вибору кроку дискретизації нестационарного процесу потрібно визначити крок для стаціонарного процесу, який еквівалентний даному нестационарному, за поведінкою кореляційної функції на найбільш динамічній ділянці.

Врахування ефекту дискретизації сигналу призвело до застосування нового математичного апарату – апарату дискретної математики. Ефект квантування враховується простою апроксимацією дійсних чисел раціональними з заданою величиною похибки, при цьому не повністю враховувались характер і властивості цифрового подання сигналів при побудові моделей. Оскільки значення сигналу приймають цілочисельні рівні, можна використовувати абстрактний апарат алгебраїчних систем, зокрема, апарат кінцевих полів і кілець. В результаті досягається зменшення обсягу обчислень при реалізації цифрової обробки сигналу.

Значення сигналу  $x(t)$  можна розглядати не тільки як підмножину поля комплексних чисел, але як і підмножину інших алгебраїчних систем, що мають структуру кільця або поля, в тому числі і кінцевих (наприклад, кінцеве кільце залишків за модулем цілого числа, кінцеве поле Галуа). Слід зазначити, що в даному випадку вноситься мінімальна надлишковість [13–16].

#### **1.2.4. Ентропія станів джерела інформації**

Для підвищення ефективності системи передавання інформації необхідно збільшувати ентропію повідомлень. Ентропія повідомлень залежить від закону розподілу ймовірностей. Отже, для підвищення ефективності необхідно здійснити перерозподіл густини елементів повідомлення.

Відомо, що при наявності кореляційних зв'язків між елементами повідомлення їх ентропія зменшується. Тому, підвищення ефективності

інформаційних систем досягається за рахунок усунення або послаблення взаємозв'язків між елементами повідомлень, тобто за рахунок декореляції повідомлень.

На прикладі табл. 1.1 розглянемо ентропію різних станів ДІ.

Ентропія семантичного стану. Установка буріння може перебувати в одному з чотирьох станів: “буріння” –  $x_1$ ; “викид” –  $x_2$ ; “аварія” –  $x_3$  та “ліквідація аварії” –  $x_4$ . Останні два стани  $x_3$  і  $x_4$  є корельованими, тому для визначення ентропії стану “ліквідація аварії” потрібно враховувати умовну ймовірність –  $p(x_4/x_3)$ . Для перших трьох станів ентропія визначається за формулою Шенона [17]:

$$H_c(x) = \sum_{k_c=1}^n p(x_{k_c}) \cdot \log \frac{1}{p(x_{k_c})}.$$

Третій і четвертий стани (“аварія” і “ліквідація аварії”) утворюють простий ланцюг Маркова, ймовірність кожного повідомлення якого  $x_k$  цілком визначена, якщо відомо передане перед ним повідомлення  $x_i$ . Ентропія визначається за формулою [18]:

$$H(x) = \sum_{k=1}^n \sum_{i=1}^n p(x_i) \cdot p(x_k | x_i) \log \frac{1}{p(x_k | x_i)},$$

де  $p(x_k | x_i)$  – умовна ймовірність передачі  $x_k$ , якщо попереднім повідомленням було  $x_i$ ;

$p(x_i)$  – безумовна ймовірність передачі  $x_i$ .

Перші три стани є статистично незалежні.

Загальна ентропія семантичного стану згідно [19] дорівнює:

$$H_c(x) = \sum_{k_c=1}^3 p(x_{k_c}) \cdot \log \frac{1}{p(x_{k_c})} + p(x_3) \cdot p(x_4 | x_3) \cdot \log \frac{1}{\log p(x_4 | x_3)}.$$

Ентропія інформаційного стану. Усі п'ять перелічених у табл.1.1 станів: буріння, спуск-підйом обладнання, промивка компановка і простій є статистично незалежними, отже, з врахуванням їх ймовірностей  $p(x_k)$

значення ентропії інформаційного стану дорівнює

$$H_i(x) = \sum_{k=1}^5 p(x_k) \cdot \frac{1}{\log p(x_k)}.$$

Ентропія технологічного стану. Буріння, СПО, промивка, компановка, випробування та аварія є некорельованими між собою, аварія, ремонт і простій утворюють вищевказаний ланцюг Маркова, отже, ентропія технологічного стану дорівнює

$$H_i(x) = \sum_{k=1}^6 p(x_k) \cdot \frac{1}{\log p(x_k)} + \sum_{k=7}^8 \sum_{i=6}^8 p(x_i) \cdot p(x_k | x_i) \cdot \frac{1}{\log p(x_k | x_i)},$$

де  $p(x_i)$  – ймовірність  $(k-1)$  – технологічного стану.

Ентропія повідомлення при наявності кореляційних зв'язків між елементами, відповідно і обсяг інформації, що передається, зменшується, причому це зменшення буде тим інтенсивніше, чим сильніше кореляційні зв'язки і чим більша кількість елементів буде охоплена цими зв'язками [3, 19].

На основі проведеного аналізу показано, що максимальну ентропію має технологічний стан джерела інформації, семантичний – мінімальну.

### **1.3. Класифікація та перспективи розвитку методів стиснення даних**

Будь-яка класифікація має умовний характер і в значній мірі залежить від мети розробки та найважливіших ознак. Класифікація методів стиснення даних за найбільш загальними ознаками включає [6]:

#### **1. Причини виникнення надлишковості.**

1.1. Завищення розмірності вектору даних  $W(t)$ , що надаються

користувачу, викликане неврахуванням властивостей користувача, якому для правильного сприйняття станів об'єкта достатньо мати вектор меншої розмірності. Отже, перша причина виникнення надлишковості пов'язана з психофізичними властивостями користувача. Розробка методів зменшення надлишковості такого виду має евристичний характер.

1.2. Завищення числа компонент  $\{x_m(t)\}_{m=1, \overline{M}}$  процесу  $x(t)$ , що породжується об'єктом, призводить до підвищення розмірності повідомлення  $y(t)$  і оцінок  $\hat{y}(t)$  і  $\hat{x}(t)$ . Це пов'язано з неоптимальністю методів отримання відомостей щодо параметрів процесу.

Друга причина виникнення надлишковості полягає в тому, що об'єкт розглядається в просторі станів завищеної розмірності (задачі класифікацій, розпізнавання образів, ідентифікації об'єкта та інші).

1.3. Неповне використання характеристик процесу  $x(t)$ , що зумовлено використанням неоптимальних методів. При цьому розробка оптимальних методів стиснення даних, які повністю враховують відомості про процес  $x(t)$ , являє теоретичний інтерес для дослідження потенційно можливої ефективності стиснення.

1.4. Неможливість цілковитого знання поведінки реалізацій або досліджуваних характеристик процесу  $x(t)$  і проектування системи на основі усереднених або граничних значень характеристик процесу  $x(t)$ , тобто природна надлишковість.

1.5 Створення надлишковості з метою підвищення достовірності оцінок і отриманих відомостей  $W(t)$ . При цьому використовуються відомі методи резервування інформаційних каналів, завадостійкого кодування повідомлень і т.д. Ця надлишковість є цілком контрольованою і керованою з боку розробника системи.

Отже, всі методи стиснення даних за першою класифікаційною ознакою – видом надлишковості, що усувається – поділяються на класи методів, що реалізують стиснення даних з врахуванням позицій 1 – 4. Методи

зменшення розмірності  $W(t)$  і  $x(t)$  розглядаються разом із загальними задачами дослідження об'єкта: класифікації, ідентифікації та іншими.

Стиснення даних, як правило, пов'язане з усуненням третього і четвертого видів надлишковості.

2. Тип повідомлення  $y(t)$ .

За другою ідентифікаційною ознакою методи стиснення поділяються на призначені для обробки аналогових, аналого-дискретних і дискретних повідомлень.

3. Число координат повідомлення, що обробляються методом стиснення.

Ці методи стиснення даних поділяються на орієнтовані на обробку кожної координати повідомлення окремо (роздільна обробка) і декількох координат повідомлення одночасно (сумісна обробка). Сумісна обробка дає змогу при наявності детермінованого або стохастичного зв'язку координат повідомлення отримати переваги у стисненні порівняно з окремою обробкою, але складніше реалізується системами стиснення даних.

4. Предметна область, на яку орієнтований метод стиснення (визначений клас об'єктів).

Виділяють класи методів, які не пов'язані з конкретною предметною областю (універсальні), і ті, що використовують специфіку об'єктів (спеціальні). Прикладами останніх є методи стиснення звуку, зображення, тексту.

5. Число елементів (при цифровій обробці число кодових слів), одночасно оброблюваних з метою стиснення.

Перший клас складають методи, в яких кожний елемент (кодове слово) обробляється окремо. Другий клас складають групові (блочні) методи, в яких одночасно обробляється декілька вибірок (кодових слів). Блочні методи дозволяють підвищити ефективність стиснення даних джерел з пам'яттю.

6. Можливість відновлення початкової (вихідної) реалізації повідомлення  $y(t)$  з контрольованою похибкою.



Тут виділяють методи стиснення без втрат і методи стиснення з втратами. В першому випадку відновлення із заданою точністю можливе, а в другому – ні, оскільки при цьому  $z(t)$  (вихідне повідомлення) являє собою узагальнені параметри об'єкта.

Серед методів без втрат, слід окремо відзначити методи стиснення цифрових даних, які гарантують, при відсутності завад у каналі, точне відтворення вихідного повідомлення.

Автором розроблена класифікація методів стиснення даних за областю застосування: універсальні, методи стиснення звуку, зображення, технологічних даних (рис. 1.10).

Універсальні алгоритми лежать в основі архіваторів (pkzip, arj, lha) і програм динамічного стиснення дисків (Stacker, DoubleSpace) [20 – 25].

Основна ідея: друге і наступне входження деякого рядка символів у повідомлення замінюються посиланням на його першу появу в повідомленні. Застосовуються варіанти LZ77, LZSS, LZW (файли формату GIF, TIFF), BTLZ (рекомендація МККТТ V.42 bis). Використовуються для стиснення текстів і графіки [26 – 30].

Оцифрований звук погано піддається стисненню без втрат. Отже, для стиснення звуку у вигляді цифрового чи аналогового сигналу необхідно застосовувати алгоритми стиснення з втратами. Найбільше поширення одержали наступні методи [31 – 35]:

- лінійна імпульсно-кодова модуляція (ЛІКМ);
- логарифмічна (з компандуванням за  $A - \mu$  - законі);
- адаптивна ІКМ (АІКМ), диференціальна ІКМ (ДІКМ);
- адаптивна диференціальна ІКМ (АДІКМ);
- адаптивна дельта-модуляція (АДМ);
- кліпірування мовного сигналу;
- статистичне ущільнення;
- застосування вокодерів (пристроїв для стиснення, кодування і передавання мови) [36–39].

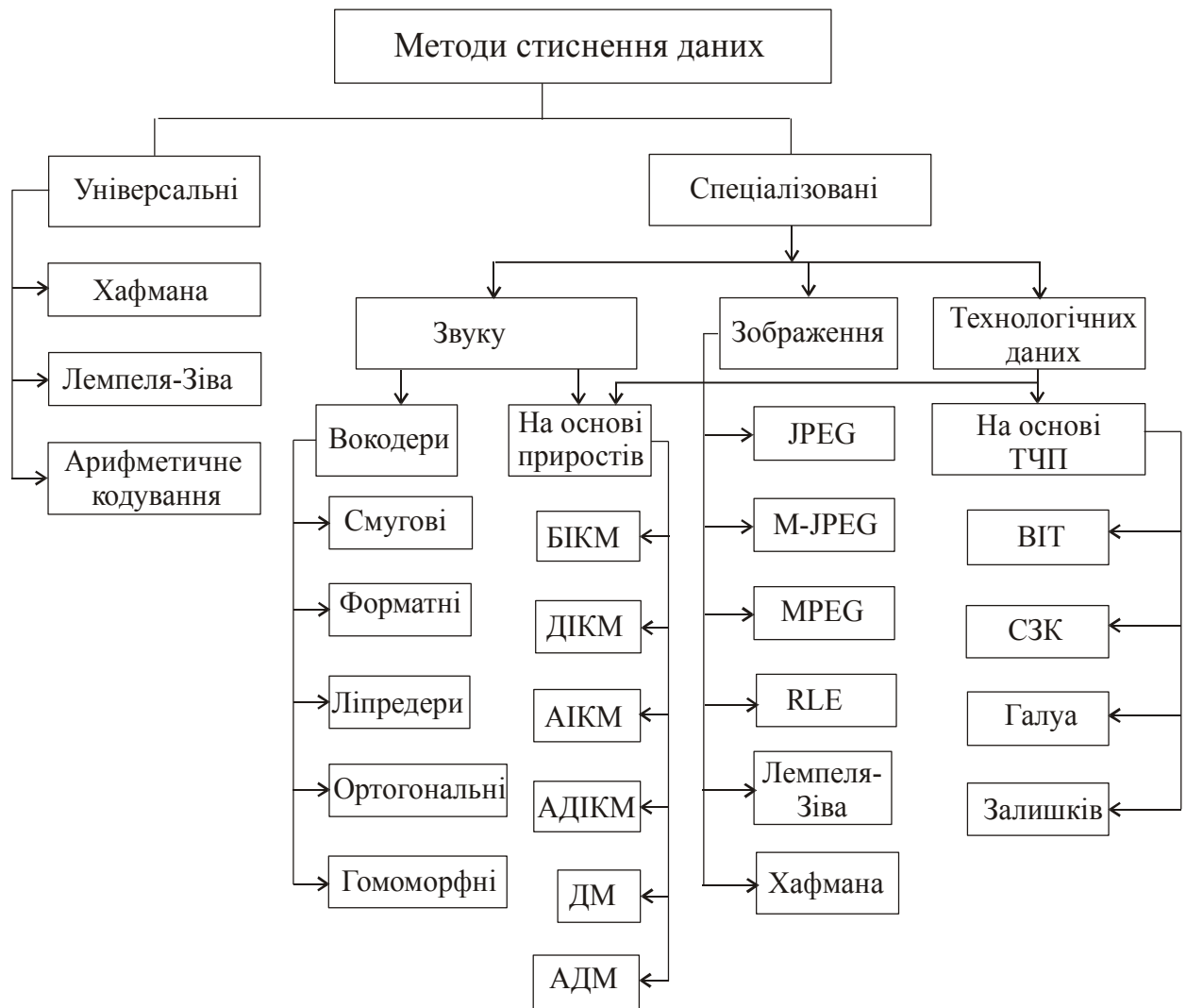


Рис. 1.10. Класифікація методів стиснення даних: JPEG – Joint Photographic Expert Group (алгоритм розроблений групою фахівців в галузі фотографії); M-JPEG – Motion-JPEG; MPEG – Moving Pictures Expert Group (алгоритм розроблений групою експертів у галузі цифрового відео); RLE – Run Length Encoding (кодування довжин повторів); LZ – Lempel – Ziv (Зіва – Лемпеля); ТЧП – теоретико-числове перетворення; ВІТ – вертикальна інформаційна технологія; СЗК – система залишкових класів.

Застосовуються також алгоритми стиснення без втрат:

- стиснення за рахунок пауз у сигналі (паузи видаляються з вихідного сигналу, а їхня тривалість запам'ятовується);
- при збереженні і передаванні музики замість оцифрованого звуку – використання інформації про те, які ноти, коли і на якому інструменті потрібно виконувати (інтерфейс MIDI).

Отже, при розробці методів зменшення надлишковості повідомлень необхідно враховувати наступні параметри:

- швидкість створення повідомлення;
- якість відновленого сигналу;
- складність алгоритмів кодування, декодування;
- комунікаційні затримки (для потокового кодування).

В даний час широко досліджуються методи стиснення звукових повідомлень, зображення, відео, про що свідчать чисельні публікації, але значно менше уваги приділяється розробці спеціалізованих методів стиснення технологічних даних, що є необхідною умовою для вдосконалення автоматизованих комп'ютерних систем керування технологічними об'єктами. Перспективним в цьому напрямі є розробка методів стиснення даних з використання дискретних базисів та теоретико-числових перетворень.

#### **1.4. Аналіз ефективності теоретико-числових базисів**

З розвитком обчислювальної техніки особливе значення має використання теоретико-числових базисів. Основною вимогою при цифровій обробці є завдання зведення до мінімуму часу машинного виконання операцій. Це особливо актуально в області керування, зв'язку, а також при цифровій обробці сигналів (ЦОС). Перетворення Фур'є має недолік – великі обсяги обчислень. Розробка швидкого перетворення Фур'є дала змогу суттєво прискорити обробку інформації, але залишилась необхідність у виконанні великої кількості операції множення, яка займає більшу частину часу при машинній обробці даних. Операції множення виконуються при розкладі функцій в ряд Фур'є і при виконанні інтегрального перетворення Фур'є.

Наприклад при розкладі функції  $x(t)$  в ряд Фур'є з використанням формули [40]:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cdot \cos k\omega_1 t + b_k \cdot \sin k\omega_1 t),$$

$$\text{де } a_k = \frac{2}{T} \cdot \int_{-T/2}^{T/2} x(t) \cdot \cos \cdot k\omega_1 t dt;$$

$$b_k = \frac{2}{T} \cdot \int_{-T/2}^{T/2} x(t) \cdot \sin \cdot k\omega_1 t dt.$$

При наближеній апроксимації функції  $x(t)$  тригонометричними рядами Фур'є, а також при інших видах обробки інформації операцію множення потрібно виконувати мільйони разів і це займає основну частину апаратних ресурсів.

Кусково-постійні базисні функції мають на обраному інтервалі часу постійні значення, отже, при розкладі функції  $x(t)$  в ряд замість вказаної базисної функції  $\cos k\omega_1 t$  введена величина, яка є для даного інтервалу розкладу постійною. Якщо при всіх  $k$  на заданому інтервалі значення базисної функції дорівнює 1 або  $-1$ , то у формулі обчислення коефіцієнтів  $a_k$  ряду замість виразу  $\cos k\omega_1 t$  будуть відповідно 1 або  $-1$ . Зникає необхідність в виконанні кожний раз операції множення. Якщо базисна функція на даному інтервалі часу дорівнює 0, то дорівнюють нулю на всьому інтервалі і коефіцієнти ряду, що визначаються. При наближеному обчисленні коефіцієнтів розкладу функції в ряд за базисними функціями, що приймають значення 1,  $-1$ , 0 необхідно виконувати простіші, ніж множення, операції – додавання і віднімання.

Від вибору базисних функцій залежать такі характеристики комп'ютерних систем, як коефіцієнт стиснення даних, похибка відновлення неперервних сигналів та складність реалізації.

В ЦОС на ЕОМ використовують базисні функції Уолша, Хаара, Крестенсона, Галуа [40 – 43].

Функції Уолша одержали широке використання в ЦОС завдяки тому, що вони приймають тільки одне із двох значень  $+1$  або  $-1$ .

Функції Уолша одержуємо із функцій Радемахера, заданих на інтервалі  $[0, 1]$  [40]:

$$R_m(x) = \begin{cases} +1, \text{ якщо } \frac{2 \cdot i}{2 \cdot m} \leq x \leq \frac{2 \cdot i + 1}{2 \cdot m} \\ -1, \text{ якщо } \frac{2 \cdot i + 1}{2 \cdot m} \leq x \leq \frac{2 \cdot i + 2}{2 \cdot m} \end{cases},$$

де  $m$  – порядок функції ( $m=1,2,\dots$ ),  $i$  – порядковий номер інтервалу задання функцій  $R_m(x)$ , ( $i=0,1,2,\dots,2^{m-1}$ ).

Функції Уолша можна отримати із функцій Радемахера за правилом утворення двійкових чисел, графіки цих функцій приведені на рис. 1.11.

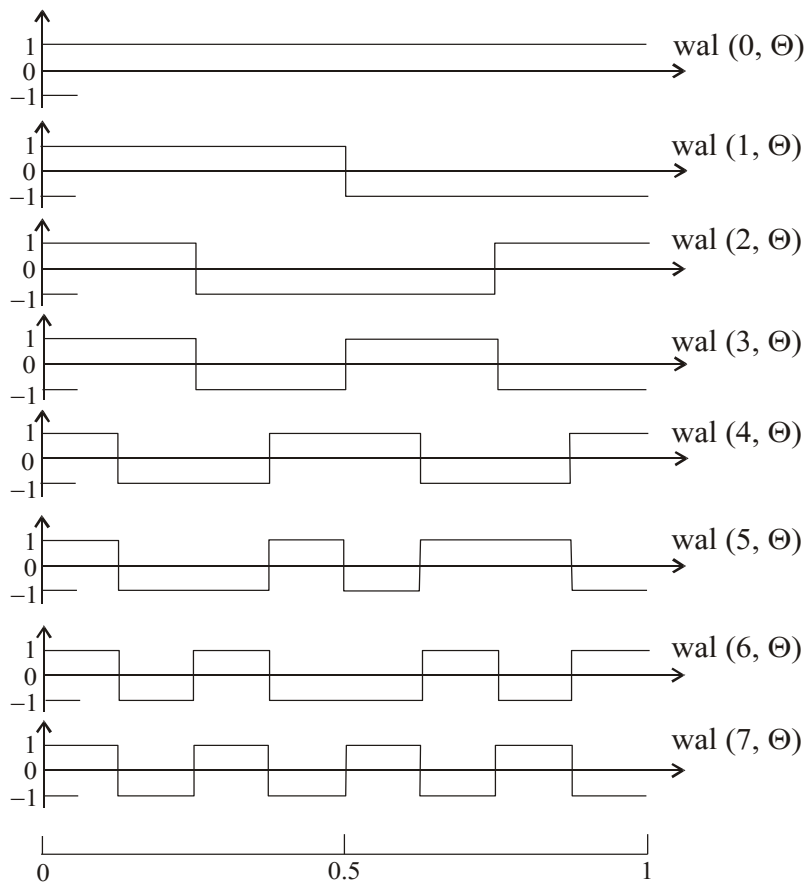


Рис. 1.11. Функції Уолша.

Розклад функції  $F(x)$  за функціями Уолша має наступний вигляд [40]:

$$F(x) \approx \sum_{m=0}^n a_m \cdot W_m(x),$$

коефіцієнти  $a_m$  обчислюються за формулою:

$$a_m = \int_0^1 F(x) \cdot W_m(x) dx. \quad (1.1)$$

Із формули (1.1) випливає, що для обчислення функцій Уолша необхідно інтегрувати  $F(x)$  на інтервалі задання  $W_m(x)$  та просумувати ці значення з врахуванням знаків  $W_m(x)$  на цих інтервалах.

Для формування функцій Уолша можна використовувати і те, що добуток будь-яких двох функцій Уолша дає деяку іншу функцію Уолша [41]. Якщо  $n_1$  – номер першої функції,  $n_2$  – номер другої, то номер функції одержаної після їх перемноження визначається як:  $n_1 \oplus n_2$ , тобто  $wal(n_1, \Theta) \times wal(n_2, \Theta) = wal(n_1 \oplus n_2, \Theta)$ .

Використання функцій Уолша має ряд переваг, зокрема, за функціями Уолша можна проводити розклад довільних сигналів в ряд Уолша-Фур'є, і вони приймають тільки два значення (+1 або -1) і тому зручні для обробки на ЕОМ.

При виконанні операції згортки сигналів та розробки цифрових фільтрів доцільно використовувати функції Хаара [43].

Функції Хаара визначаються при кожному значенні  $\Theta$ , ( $0 \leq \Theta < 1$ ) двома величинами, які позначимо  $l$  і  $n$ , де  $l$  – номер підрозділу в системі функцій Хаара;  $n$  – номер функції в відповідному підрозділі.

Для функцій Хаара використовують позначення  $har(l, n, \Theta)$ ,  $X_l^n(\Theta)$  або  $H_l^n(\Theta)$ .

Перші вісім функцій Хаара представлені на рис. 1.12, матриця дискретних значень функцій Хаара має вигляд [43]:

1	1	1	1	1	1	1	1
1	1	1	1	-1	-1	-1	-1
$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	0	0	0	0
0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$
2	-2	0	0	0	0	0	0
0	0	2	-2	0	0	0	0
0	0	0	0	2	-2	0	0
0	0	0	0	0	0	2	-2

Кожний рядок матриці відповідає певній функції Хаара (рис. 1.12.).

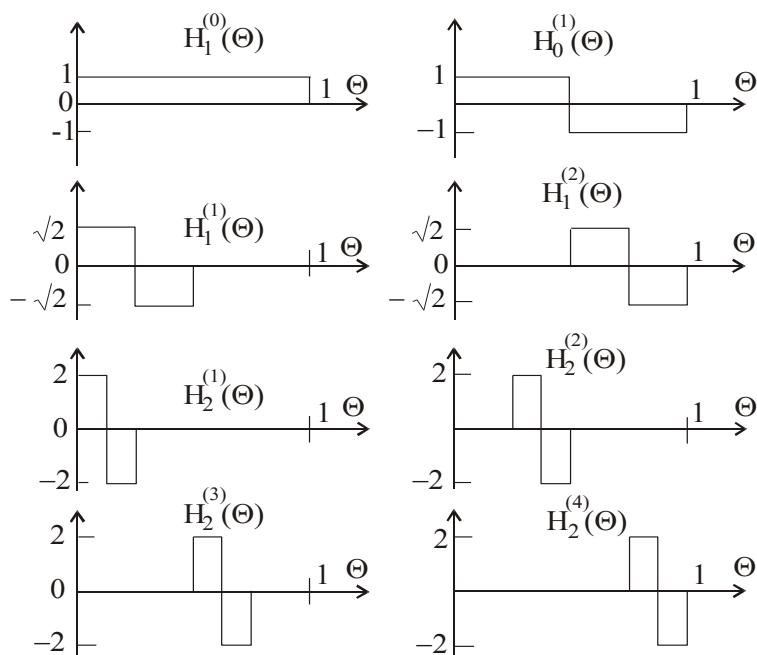


Рис. 1.12. Функції Хаара.

Для формування  $N$  функцій Хаара використовують формулу [43]:

$$H_l^{(n)}(\Theta) = \begin{cases} 2^{l/2} & \text{при } \frac{n-1}{2^l} \leq \Theta < \frac{n-1/2}{2^l} \\ -2^{l/2} & \text{при } \frac{n-1/2}{2^l} \leq \Theta < \frac{n}{2^l} \\ 0 & \text{при інших } \Theta \in [0, 1] \end{cases}$$

де  $0 \leq l < \log_2 N$  і  $1 \leq n \leq 2^l$ .

Розклад  $\varphi(x)$  в ряд за функціями Хаара проводять за наступним алгоритмом [43]:

обчислюють спектр функції  $\varphi(x)$ :

$$C_1^{(n)} = 2^{-m+l} \sum_{x=0}^{2^m-1} \varphi(x) H_l^{(n)}(x), \quad (1.1)$$

і представляють  $\varphi(x)$  в вигляді ряду

$$\varphi(x) = C_0^{(0)} H_0^{(0)}(x) + \sum_{l=0}^{m-1} \sum_{n=1}^{2^l} C_l^{(n)} H_l^{(n)}(x). \quad (1.2)$$

В цих формулах  $m$  має раніше вказане значення;  $l = 0, 1, \dots, m-1$ ;  $n = 1, 2, \dots, 2^l$ .

Виходячи з універсальної множини (рис. 1.13) можна сформувати усі відомі дискретні кодові бази.

0	0	1	0	1	0	0	1	1	1
1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	1	1	0	0	1
1	0	0	0	1	0	1	1	1	0
0	1	1	0	1	1	0	0	1	0
0	1	1	1	1	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0
0	0	0	1	1	0	1	0	0	1
0	1	0	0	1	0	1	1	0	0
1	0	0	1	0	0	1	1	0	1

Рис. 1.13. Універсальна множина.

Відбувається це наступним чином: вибираємо з універсальної множини стовпці, що містять необхідні для формування даного базису елементи, і впорядковуємо їх згідно правил створення конкретного базису.

Матриця базису Хаара має вигляд:

$$M_H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Об'єм матриці Хаара  $V_H = N^2$ ,

де  $N$  – кількість елементів в стовпці матриці.

Ефективність кодування в базисі Хаара:  $M_H = \frac{N}{V_{har}} = \frac{N}{N^2} = \frac{1}{N}$ .

Для унітарного базису об'єм кодової матриці і значення ефективності аналогічні попередньому випадку:  $V_u = N^2$ , і  $M_u = \frac{1}{N}$  відповідно.

$$M_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Об'єм матриці базису Радемахера

$$M_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

обчислюється згідно виразу:

$$V_R = N \cdot \log_2 N,$$

ефективність кодування з використанням даної матриці:

$$M_R = \frac{N}{N \cdot \log_2 N}.$$

Об'єм кодової матриці Грея  $V_{Gr} = N \cdot \log_2 N$ ,

ефективність  $M_{Gr} = \frac{1}{\log_2 N}$ .

Кодова матриця Грея:

$$M_{Gr} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Матриця базису Галуа:

$$M_{Gal} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Базис Галуа задається матрицею-стовпцем, об'єм якої дорівнює  $N$ .

Ефективність кодування:  $M_{Gal} = \frac{N}{N} = 1$ .

Оскільки об'єм кодової матриці базису Галуа мінімальний, відповідно дослідження та застосування даного базису для розробки методів стиснення даних є найбільш перспективним.

## ВИСНОВКИ

1. Проведений аналіз трирівневої зірково-магістральної архітектури з метою дослідження надлишковості даних на етапах формування, передавання та цифрової обробки. Визначено системні та функціональні об'єкти низових комп'ютерних мереж.

2. Аналіз промислових автоматизованих комп'ютерних систем показав, що об'єкти керування, які представлені технологічними установками, належать до класу квазістаціонарних ДІ і характеризуються трьома станами: семантичним, технологічним, інформаційним. Визначено стани основних технологічних установок нафтогазового комплексу, що дозволило розробити архітектуру комп'ютерної системи ідентифікації семантичних, технологічних та інформаційних станів ОК, використання якої дозволить зменшити надлишковість даних на етапі їх формування.

Розрахунок ентропії станів ДІ показав, що максимальну ентропію має інформаційний стан ДІ, а мінімальну – технологічний стан ДІ.

3. Проведено аналіз існуючих класифікацій методів стиснення та розроблена їх класифікація за предметною областю: універсальні методи стиснення, методи стиснення звуку, зображення, технологічних даних.

4. Проведено аналіз теоретико-числових кодових базисів, визначено об'єм кодових матриць та оцінено ефективність кодування. Проведені дослідження показали, що мінімальний об'єм кодової матриці має базис Галуа, отже, використання даного базису для розробки методів стиснення є найбільш ефективним та перспективним.

## РОЗДІЛ 2

### МЕТОДОЛОГІЯ ТА ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ МЕТОДІВ ЗМЕНШЕННЯ НАДЛИШКОВОСТІ ЦИФРОВИХ ДАНИХ У КОМП'ЮТЕРНИХ СИСТЕМАХ

#### 2.1. Логіко-статистичні та інтегрально – імпульсні моделі інформаційного стану джерел інформації

Дослідження систем зі стисненням даних може проводитись експериментальним та аналітичним методами. У першому випадку результати дослідження цілком визначаються умовами досліду, параметрами та структурою системи, типом визначеного процесу, які важко змінювати. У зв'язку з цим, більш перспективними є аналітичні методи, хоча вони і вимагають чіткої математичної постановки вихідної задачі: ймовірнісний опис повідомлень, вибір операторів його перетворення, вибір критерію якості.

Реальні фізичні процеси можна описувати на макрорівні, коли використовуються характеристики джерела за великий інтервал часу, або на мікрорівні, коли використовуються короткочасні властивості ДІ, а також на структурному рівні, коли окремі елементи процесу можна класифікувати на основі виділення специфічних ознак. Перші два види опису базуються на представленні процесу у вигляді експериментальної кривої (з неперервним або дискретним часом). Структурний опис дозволяє виявити дискретну природу об'єкту і представити його у вигляді ланцюжка символів із заданого алфавіту. При цьому на вихідний опис джерела впливає характер подальшої обробки повідомлення.

Макроопис доцільно використовувати для процедур стиснення даних фіксованої структури (неадаптивних), мікроопис – для адаптивних алгоритмів

зменшення надлишковості, структурне представлення – для методів, що базуються на синтаксичному підході до розпізнавання образів.

Таким чином, розробка моделі джерела інформації повинна відповідати вимозі універсальності для узгодження вказаних видів опису і адекватно відображати реальні нестационарні фізичні процеси.

Приведені в розділі 1 приклади класифікації семантичних станів технологічних установок нафтогазової промисловості є основою для побудови сенсорів та техногенно-екологічних комп'ютерних систем безпеки. Важливу роль відіграє побудова в реальному масштабі часу розроблених логіко-статистичних (ЛСІМ) та інших інформаційних моделей [4, 10, 44].

Особливо це стосується кодування семантичних станів ДІ, коли необхідно відслідковувати критичні ситуації технологічних об'єктів, до яких належать аварії, викиди нафти, коротке замикання.

Кодування нульовими та одиничними бітами унітарного базису вимагає додатково фіксувати час, оскільки ми не можемо відразу визначити момент часу, в який відбулась зміна.

Використання бітів Галуа дозволяє вдосконалити існуючу систему, позбавити її вказаного недоліку. Нульовим значенням ставляться у відповідність прямі біти Галуа, а одиничним – інвертовані. Це дозволяє визначити час, коли відбулася зміна. Для визначення моменту часу необхідно декодувати останніх  $n_g$  біт послідовності Галуа, що передують інвертованому біту  $\bar{G}$  (де  $n_g$  – розрядність кодону Галуа).

Перша ЛСІМ фіксує вихід за границі апертури “норми” амплітудних значень характеристичних параметрів  $P_{ijk}$  (рис. 2.1) згідно аналітичного виразу:

$$L_1 = \begin{cases} G, & P_{ijk} < E_1 \\ \bar{G}, & P_{ijk} \geq E_1 \end{cases},$$

де  $E_1$  – границя апертури амплітудних значень.

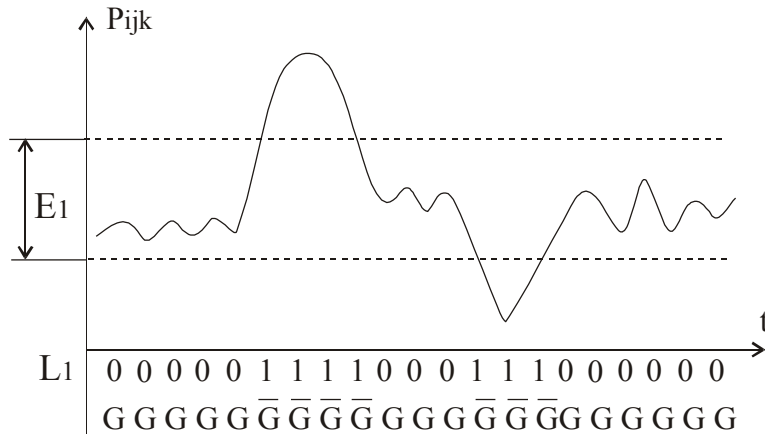


Рис. 2.1. Перша ЛСІМ.

Друга ЛСІМ фіксує зміну динаміки характеристичного параметру  $P_{ijk}$ , амплітуда якого не виходить за границі апертури  $E_1$  (рис. 2.2) згідно наступного аналітичного виразу:

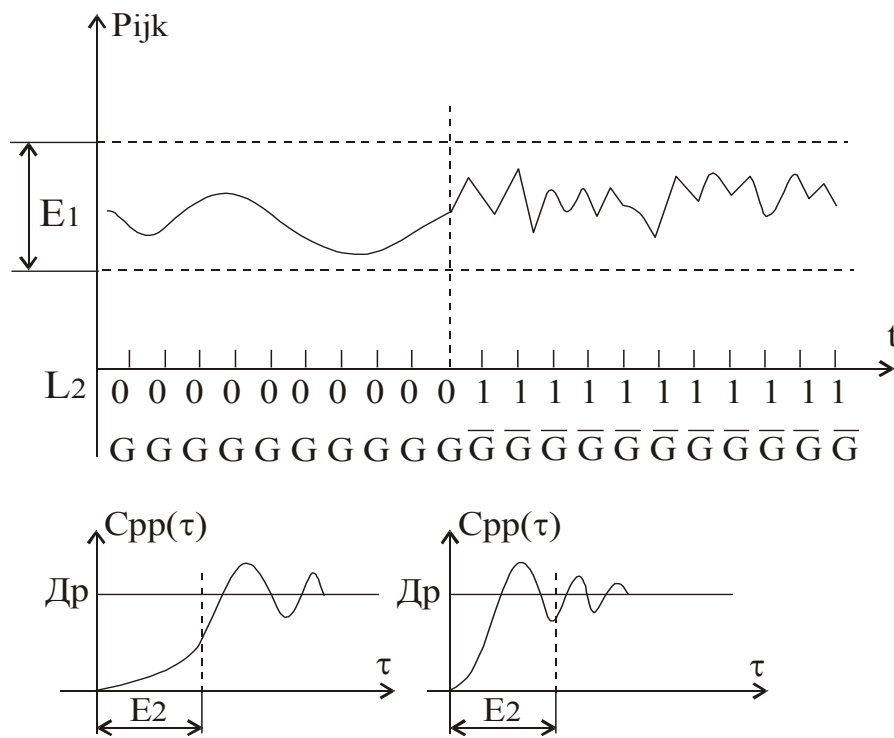


Рис. 2.2. Друга ЛСІМ.

$$L_2 = \begin{cases} G, & C_{pp}(\tau) < E_2 \\ \bar{G}, & C_{pp}(\tau) \geq E_2 \end{cases},$$

де  $C_{pp}(\tau)$  – автоструктурна кореляційна функція  $P_{ijk}$ ;  $\tau$  – часове зміщення;

$$C_{pp}(\tau) = \frac{1}{n} \sum_{i=1}^n \sum_{j=0}^{\tau} (P_{ijk} - (P_{ijk} + \tau))^2.$$

Третя ЛСІМ фіксує зміну знаку коефіцієнта взаємочореляції між двома характеристичними параметрами  $P_{ijk}$  і  $q_{ijk}$  з амплітудами, що не виходять за межі апертури (рис. 2.3), один з яких  $q_{ijk}$  вважається контрольним або еталонним згідно виразу:

$$L_3 = \begin{cases} G, & \rho_{pq} < 0 \\ \bar{G}, & \rho_{pq} \geq 0 \end{cases}.$$

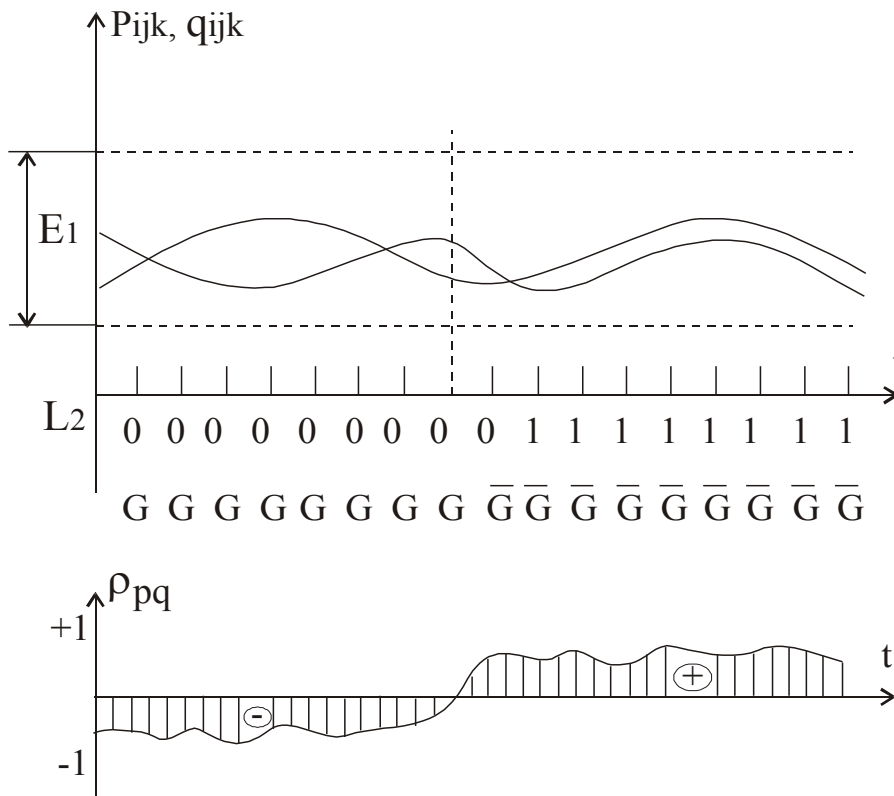


Рис. 2.3. Третя ЛСІМ.

Четверта ЛСІМ базується на спектральному аналізі контрольованого сигналу. Множина  $E1$  визначає спектральний склад сигналу, який відповідає нормальному режиму роботи ДІ (рис. 2.4):

$$L_4 = \begin{cases} G, & E1 = w, \\ \bar{G}, & E1 \neq w. \end{cases}$$

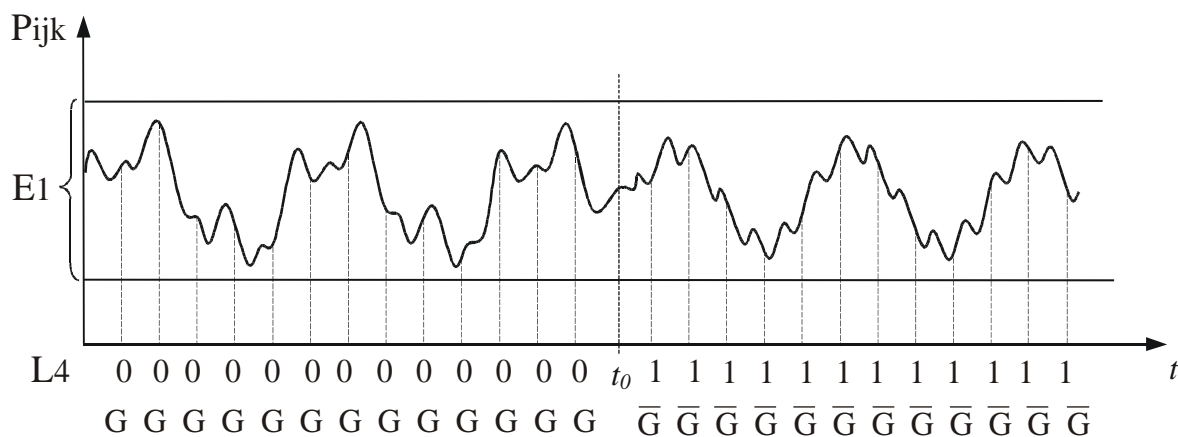


Рис. 2.4. Четверта ЛСІМ.

Запропонована в роботі [10] п'ята ЛСІМ базується на матриці коефіцієнтів взаємкореляції сигналів в каналах ДІ і оцінки глобальної дисперсії  $D_G$ . Вона дозволяє зафіксувати зменшення значень  $D_G$  нижче встановленої апертури  $E_5$ , що відповідає руйнуванню кореляційних зв'язків і деградації системи в цілому. На відміну від попередніх ЛСІМ дана модель виражається одною булевою змінною  $g$  (рис. 2.5), яка оцінює загальний стан багатоканального ДІ:

$$g = \begin{cases} G, & D_G > E_5 \\ \bar{G}, & D_G \leq E_5 \end{cases}.$$

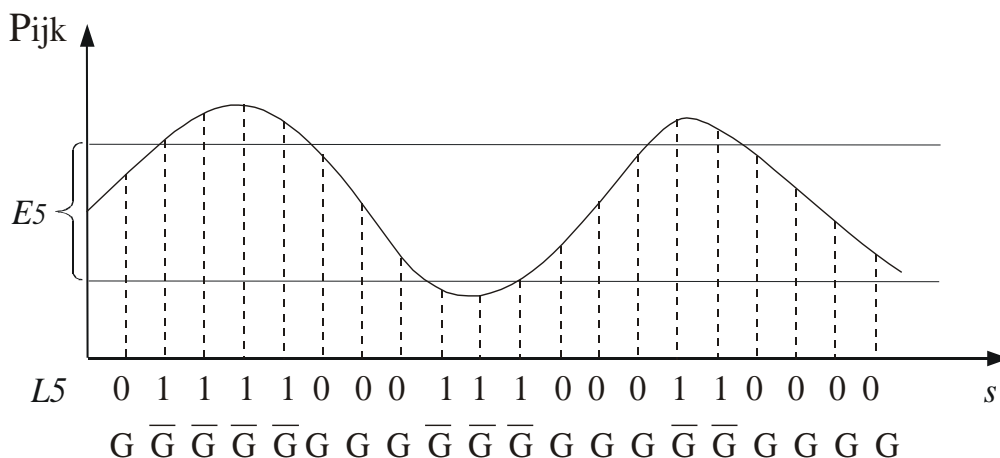


Рис. 2.5. П'ята ЛСІМ на основі глобальної дисперсії.

Шоста ЛСІМ, запропонована автором, базується на оцінці відхилення значення контрольованого параметру від еталонного. Задається  $\varepsilon_0$ ,



обчислюється величина відхилення сигналу, після цього здійснюється порівняння обчисленого і заданого значення.

У разі, якщо відхилення реального сигналу від передбачуваного не перевищує заданого значення, у канал зв'язку поступають прямі біти Галуа, при перевищенні допустимого відхилення біт Галуа інвертується. Інвертовані біти Галуа вказують на номер відліку (або ж час), коли відбулось дане відхилення. При нормалізації процесу у канал зв'язку знову поступають прямі біти Галуа.

Якщо відомо, що для ДІ на протязі тривалого часу відхилення сигналу не перевищує заданого значення, то доцільно використати модифікований метод, суть якого полягає в наступному: якщо відхилення сигналу є в межах норми, то в канал зв'язку нічого не передається, але біти Галуа постійно генеруються на передавальній стороні, останні  $n_g$  бітів записуються в буфер. У кожний момент слідування наступного біту Галуа вміст буферу оновлюється. У разі недопустимого відхиленні у канал зв'язку передається весь вміст буферу та інвертований біт Галуа.

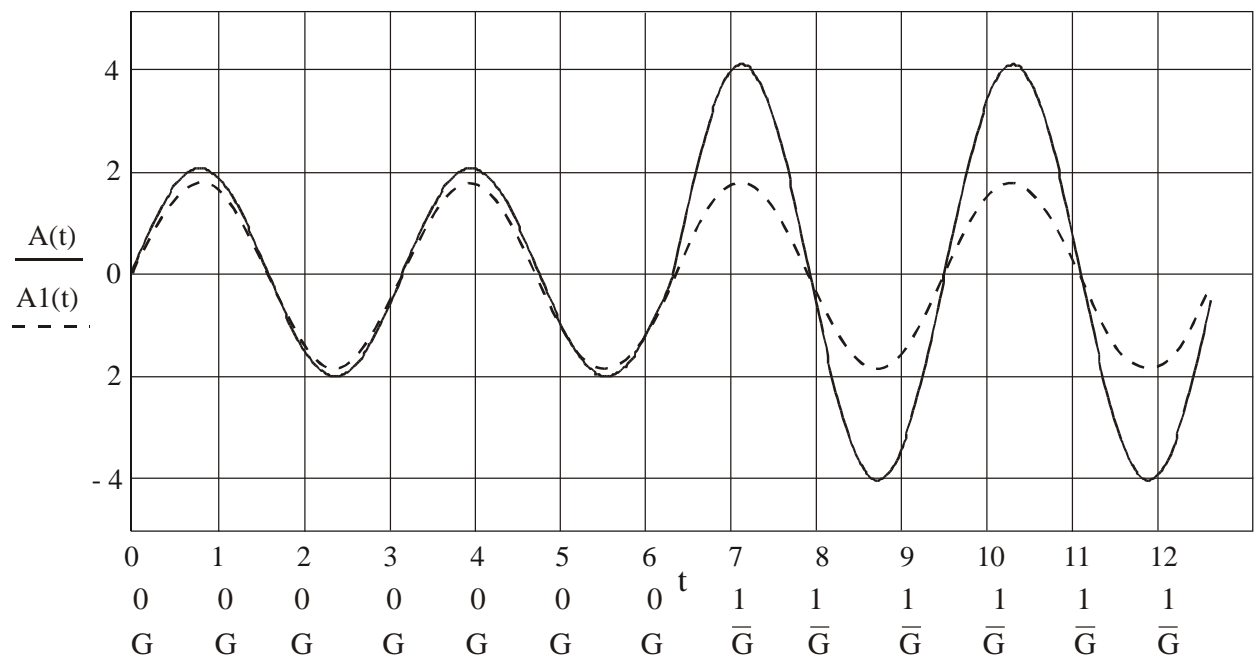


Рис. 2.6. Шоста ЛСІМ.

Значення шостої ЛСІМ визначаються за формулою:

$$L_6 = \begin{cases} G, & \varepsilon < \varepsilon_0 \\ \overline{G}, & \varepsilon \geq \varepsilon_0 \end{cases}.$$

В загальному випадку застосування ЛСІМ дозволяє ефективно зменшити обсяг даних. Якщо діапазон квантування контрольованого параметру  $A$ , то розрядність коду для представлення одного відліку визначається за формулою:

$$n = \hat{E}[\log_2 A].$$

Початковий об'єм даних при такому представленні:

$$I_0 = m \cdot \hat{E}[\log_2 A],$$

де  $m$  – кількість відліків.

Застосування ЛСІМ дозволяє зменшити об'єм даних до величини  $I = m$ , оскільки для кодування кожного відліку використовується один біт.

Отже, коефіцієнт стиснення дорівнює

$$k_c = \frac{\hat{E}[\log_2 A] \cdot m}{m} = \hat{E}[\log_2 A].$$

Аналогічні ЛСІМ можуть бути побудовані на основі відхилень статистичних, кореляційних, спектральних, ентропійних, кластерних, хемінгових та інших інформаційних моделей ОК.

Важливу інформацію про стан екологічної безпеки несуть інтегральні моделі семантичних та інформаційних станів ОК. При цьому досягається суттєве збільшення інформації про аномальні перевищення характеристичних параметрів  $P_{ijk}$  границь норми при великому числі сенсорів та низькій швидкості їх опитування  $\Delta t > \Delta t_0$ , де  $\Delta t_0$  - розрахунковий інтервал часової дискретизації.

## 2.2. Методика оцінки коефіцієнта стиснення даних

У 1989 році група дослідників запропонувала оцінювати коефіцієнт стиснення універсальних методів за допомогою набору файлів, що отримав назву Calgary Compression Corpus (CalgCC) [21]. Набір складається з 14 файлів, більшість з яких є тексти на англійській мові або мовах програмування (табл.2.1). Пізніше до цих 14 файлів додали ще 4 тексти на англійській мові. Але звичайно оцінку проводять на стандартному наборі з 14, а не з 18 файлів.

За останні 10 років CalgCC відіграв значну роль у розвитку методів стиснення даних без втрат.

Таблиця 2.1

Опис файлів, що складають стандартний CalgCC

Файл	Розмір, байт	Опис
1	2	3
Bib	111261	Бібліографічний список в форматі UNIX "refer" ASCII
Book1	768771	Художня книга: Т. Hardy. "Far from the madding crowd", неформатований текст ASCII. Містить велику кількість OCR-помилки (неправильно розпізнані символи)
Book2	610856	Технічна книга: "Principles of computer speech", формат UNIX "troff", ASCII
Geo	102400	Геофізичні дані, 32-бітні числа
News	377109	Набір повідомлень конференцій Usenet, формат ASCII
Obj1	21504	Об'єктний файл для EOM типу VAX
Obj2	246814	Об'єктний файл для ПК Apple Macintosh
Paper1	53161	Технічна стаття: Witten, Neal, Cleary. "Arithmetic coding for data compression", формат UNIX "troff", ASCII
Paper2	82199	Технічна стаття: Witten. "Computer (in) security" формат UNIX "troff", ASCII

1	2	3
Pic	513216	Факсимільна двоколірна картинка, 1728x2376 точок, являє собою дві сторінки технічної книги на французькій мові, відскановані з роздільною здатністю 200 точок на дюйм
Progc	39611	Програма на мові Сі, ASCII
Prog1	71646	Програма на мові Ліпс, ASCII
Progp	49379	Програма на мові Паскаль, ASCII
Trans	93695	Розшифрування термінальної сесії, формат редактора "EMACS", ASCII

Розмір стандартного CalgCC становить 3,141,622 байт. Повний CalgCC займає 3251493 байт

Серед конкурентів CalgCC відзначимо:

– Canterbury Compression Corpus (CantCC), що складається з двох частин-стандартного набору "Standard Set" (11 файлів загальною довжиною 2816188 байт) і набору великих файлів "Large Set" (4 файли, 16005619 байт), запропонований тією ж групою дослідників, що і CalgCC, в якості альтернативи морально застарілому CalgCC;

– набори файлів з Archive Comparison Test (ACT): 3 текстових файли, 3 виконуваних, 2 звукових і 8 повноколірних 24-бітових зображень, а також вищеописані CalgCC повний CantCC стандартний, і останній набір – це демо-версія гри Worms2 (159 файлів загальним розміром 17 Мб);

– файли Compression Comparison Test Вадима Юкіна (VYCCT, 8 файлів різних типів);

– набори файлів з тестів Art Of Lossless Data Compression (ARTest):

– 627 повноколірних зображень, 2066 Мб в 12 наборах;

– 1231 текстовий файл загальною довжиною 500 Мб в 6 наборах, в тому числі CantCC "Large Set" і 663 російських текстів;

– 5960 різноманітних файлів, 382 Мб в 10 наборах.

Серед стандартних наборів тестових зображень найбільш відомі чотири: JPEG Set, PNG Set, Waterloo Images та Kodak True Color Images.

Для спеціалізованих методів стиснення даних не існує аналогічних тестових файлів.

Аналіз літературних джерел показав наявність різних методик визначення коефіцієнта стиснення даних  $k_c$ . Зокрема в [11] коефіцієнт стиснення  $k_c$  визначається як відношення кількості відліків  $n_m$  на приймаючій стороні при рівномірній часовій дискретизації до кількості відліків  $n$  при зменшенні об'єму повідомлення за деякий проміжок часу при однаковій кількості рівнів квантування:

$$k_c = \frac{n_m}{n}.$$

В роботі [45], виходячи із  $\varepsilon$ -ентропії джерела неперервного сигналу, коефіцієнт стиснення є відношення двох величин, які представляють собою об'єми таблиць, одержаних при двох різних способах опису неперервного сигналу з похибкою  $\varepsilon_d$ . В якості таблиці з максимальним об'ємом  $H_{\max}^\varepsilon$  використовується таблиця одержана при ступінчатій апроксимації сигналу з постійним кроком. Тоді відносний коефіцієнт стиснення визначається як

$$k_{c.v} = \frac{H_{\max}^\varepsilon}{H^\varepsilon},$$

де  $H^\varepsilon$  – об'єм таблиці при вибраному методі опису сигналу.

Абсолютний коефіцієнт стиснення визначається із співвідношення:

$$k_{c.a} = \frac{H_{\max}^\varepsilon}{H_a^\varepsilon},$$

де  $H_a^\varepsilon$  – об'єм таблиці при повністю виключеній  $\varepsilon$ -надлишковості для заданого класу повідомлення.

В [46] стиснення представлено як збільшення ентропії повідомлення, що передається шляхом виключення надлишкових вибірок.

Коефіцієнт стиснення  $k_c$  визначається за формулою:

$$k_c = \frac{N_{\text{вх}}}{N_{\text{вих.сер}}},$$

де  $N_{\text{вх}}$  – кількість вхідних вибірок даних;  $N_{\text{вих.сер}}$  – середнє число значимих даних.

В [45] стиснення – це передача найбільш суттєвих змін в стані об'єкту. Вказується, що стиснення можливе в реальному часі і пакетному режимі. Головною метою системи стиснення є зменшення смуги частот каналу зв'язку для передавання даного повідомлення протягом визначеного часу або зменшення часу для передавання цього повідомлення в заданій смузі частот.

Дійсним коефіцієнтом стиснення рахується коефіцієнт

$$k_c = \frac{N_{\text{вх}}}{N_{\text{вих}}},$$

де  $N_{\text{вх}}$  – кількість двійкових розрядів на вході пристрою стиснення;  $N_{\text{вих}}$  – кількість двійкових розрядів на його виході.

В [45] стиснення даних визначається в неперервному узгодженні часового кроку квантування з характером зміни виміряного параметру (адаптивного квантування) і підвищення ефективності кодування виміряної, адресної і синхронізуючої інформації. Це дозволить при передаванні використовувати канал зв'язку з меншою пропускнуою здатністю або зменшити час передавання або використовувати для зберігання інформації запам'ятовуючі пристрої меншого об'єму.

Для оцінки ефективності використовується коефіцієнт:

$$k_e = \frac{L_1}{L_2},$$

де  $L_1$  – довжина коду за весь час передавання повідомлення;

$L_2$  – довжина коду стисненого повідомлення.

Найбільш повна методика оцінки коефіцієнта стиснення запропонована в [11], в якій для оцінки коефіцієнта стиснення пропонується

використовувати наступні показники:

а) середню частоту адаптивної дискретизації сигналу  $F_a$  при фіксованій похибці апроксимації;

б) абсолютний коефіцієнт стиснення:

$$k_{c.a} = \frac{F_p}{F_a},$$

де  $F_p$  – частота рівномірної дискретизації сигналу;

$F_a$  – середня частота адаптивної дискретизації сигналу;

в) відносний коефіцієнт стиснення:

$$k_{c.в} = \frac{F_{p(n=0)}}{F_{a(n=k)}},$$

де  $F_{p(n=0)}$  – частота рівномірної дискретизації при апроксимації сигналу поліномом нульового порядку;

$F_{a(n=k)}$  – середня частота адаптивної дискретизації сигналу при  $n = k$ .

Відносний коефіцієнт стиснення дозволяє оцінювати стиснення сигналу по відношенню до деякої базової моделі сигналу і порівнювати окремі пристрої стиснення між собою;

г) граничний коефіцієнт стиснення:

$$k_{c.гр} = \frac{F_{1д}}{F_a},$$

де  $F_{1д}$  – середня частота дискретизації в ідеальній системі стиснення;

$F_a$  – частота адаптивної дискретизації сигналу.

Оцінку коефіцієнта  $k_c$  проводять також шляхом порівняння об'єму повідомлення  $V_0$ , одержаного в результаті стиснення даних з об'ємом  $V$ , отриманих при деякому вихідному способі формування повідомлення:

$$V = F \cdot T \cdot H,$$

де  $F$  – ширина спектру сигналу;

$T$  – проміжок часу;

$$H = \log_2(P/P_m);$$

$P$  і  $P_m$  – відповідно середня потужність сигналу і шуму.

Коефіцієнт стиснення по смузі частот в каналі зв'язку при  $T = const$ ,  $H = const$  визначається:

$$k_{c.cm} = \frac{F}{F_0},$$

де  $F_0$  – ширина спектру сигналу при стисненні даних.

Зменшення надлишковості виміряних даних приводить до зменшення завадостійкості передавання даних, так як при спотворенні або стиранні із-за завади однієї або декількох координат повідомлення можуть виникнути великі похибки при відновленні неперервної функції  $x(t)$  по окремих дискретних значеннях.

В розглянутих методиках оцінки коефіцієнта стиснення не враховують ряд важливих показників, таких як: складність алгоритму кодування, а відповідно і вартість процесора кодування, складність алгоритму декодування, час декодування, чутливість до помилок.

Автором запропоновано для оцінки методів стиснення використовувати комплексний коефіцієнт ефективності (рис. 2.5), який можна визначати як добуток або суму факторів, що впливають на ефективність методу стиснення даних

$$K_{ef1} = \prod_{i=1}^n k_i,$$

$$K_{ef2} = \sum_{i=1}^n k_i,$$

де  $n$  – кількість факторів;

$k_1$  – вартість процесора кодування (складність алгоритму, число операцій, пропорційно машинний час);

$k_2$  – коефіцієнт стиснення;



$k_3$  – складність алгоритму декодування, час декодування;

$k_4$  – чутливість до помилок;

$k_5$  – коефіцієнт розмноження помилок.

Проведемо дослідження коефіцієнту ефективності методів стиснення даних в залежності від зміни значень окремих факторів для універсальних алгоритмів: арифметичного кодування та Хафмана (рис. 2.5).

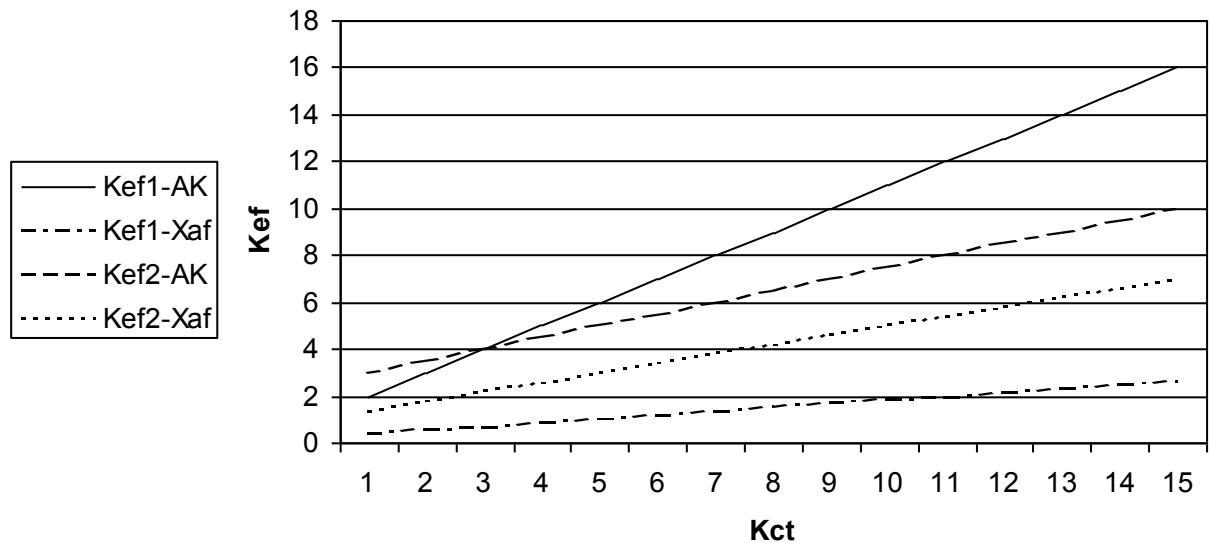


Рис. 2.5. Комплексний коефіцієнт ефективності:

Xaf – метод Хафмана;

AK – арифметичне кодування.

Приведені графічні залежності показують, що використання добутку для обчислення даного коефіцієнту в повній мірі враховують зміни кожного з вищевказаних факторів, а отже, його використання є більш доцільним. При використанні суми необхідно вводити додаткові вагові коефіцієнти, щоб зрівноважити вплив кожного з факторів.

В залежності від типу даних набір факторів може бути розширений введенням додаткових коефіцієнтів, для відео-даних це: якість зображення, швидкість надходження кадрів, редактованість, масштабування.

Комплексний коефіцієнт ефективності дозволяє визначити найбільш придатний алгоритм для вирішення поставленої задачі. Врахування основних

факторів, як складових значення коефіцієнту ефективності, відбувається з врахуванням особливостей кожної окремої задачі.

### 2.3. Теоретичне обґрунтування потенціальних можливостей стиснення даних у базисі Крестенсона

Теоретико-числовий базис Крестенсона породжує систему числення залишкових класів (СЗК).

Метод зменшення надлишковості технологічних сигналів на основі СЗК базується на основі теорії діофантових рівнянь і залишків [12, 13, 47, 48]:

$$x_i = a_i \cdot p + b_i,$$

де  $a_i$  – ранг;  $p$  – модуль;  $b_i$  – найменший невід’ємний залишок.

Діофантове рівняння:

$$x_i \equiv b_i \pmod{p},$$

або операція прямого кодування по залишках:

$$b_i = \text{res } x_i \pmod{p},$$

де  $\text{res}$  – символ операції отримання залишку.

Зворотня операція:

$$x_i = \overset{\vee}{E} \left[ \frac{x_{i-1} - b_i}{p} + 0,5 \right] \cdot p + b_i, \quad (2.1)$$

де  $\overset{\vee}{E}[\cdot]$  – цілочисельна функція з округленням до меншого цілого.

Умова однозначності кодування методом залишків виконується, якщо

$$\Delta x_{\max} \leq \frac{p-1}{2}, \quad (2.2)$$

для  $p = 5$ ,  $\Delta x_{i \max} \leq 2$ ,  $p = 7$ ,  $\Delta x_{i \max} \leq 3$ .

Інформація на рис. 2.6 закодована у вигляді залишків. При виконанні умови (2.2) процес  $x_i$  можна однозначно представити послідовністю залишків  $b_i$ .

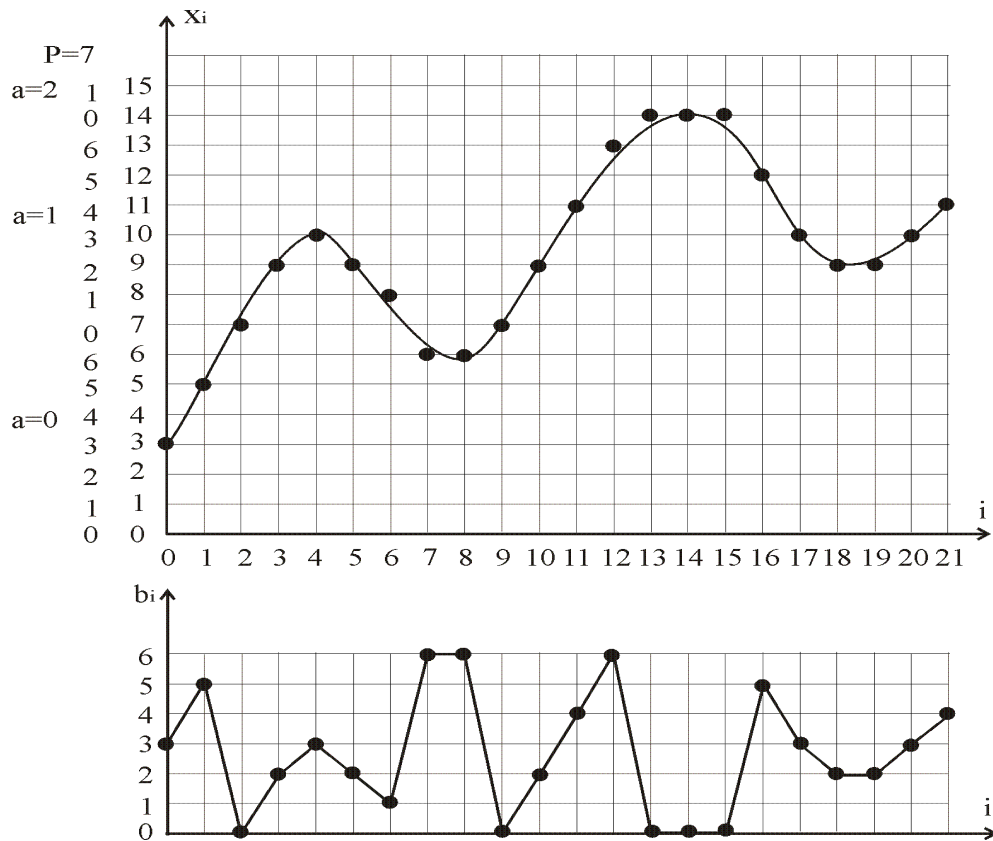


Рис. 2.6. Кодування аналогового сигналу методом залишків.

Ефект стиснення даних досягається за рахунок представлення інформаційних відліків відповідними залишками меншої розрядності.

Розрядність коду  $x_i$  визначається за формулою Хартлі:

$$n = \hat{E}[\log_2 A],$$

де  $n$  – розрядність двійкового коду для представлення величини  $x_i$ ;

$\hat{E}[\cdot]$  – цілочисельна функція з округленням до більшого цілого;

$A$  – діапазон квантування сигналу  $0 \leq x_i \leq A$ .

Розрядність коду залишків  $b_i$  визначаємо за формулою:

$$n_z = \hat{E}[\log_2 p].$$

Отже, формула для коефіцієнту стиснення буде мати наступний вигляд:

$$k = \frac{n}{\hat{E}[\log_2 p]}.$$

При кодуванні аналогового сигналу, представленого на рис 2.6, об'єм масиву становить:

$$V = n \cdot m = 4 \cdot 22 = 88 \text{ біт.}$$

Після кодування методом залишків об'єм масиву становить:

$$V = n_z \cdot m = 3 \cdot 22 = 66 \text{ біт.}$$

Декодування даних відбувається за формулою (2.1), при  $x_0 = 3$  визначаємо  $x_1$ :

$$x_1 = \check{E} \left[ \frac{x_0 - b_1}{p} + 0.5 \right] \cdot p + b_1 = \check{E} \left[ \frac{3 - 5}{7} + 0.5 \right] \cdot 7 + 5 = 5;$$

$$x_2 = \check{E} \left[ \frac{5 - 0}{7} + 0.5 \right] \cdot 7 + 0 = 7;$$

$$x_3 = \check{E} \left[ \frac{7 - 2}{7} + 0.5 \right] \cdot 7 + 2 = 9;$$

$$x_4 = \check{E} \left[ \frac{9 - 3}{7} + 0.5 \right] \cdot 7 + 3 = 10; \text{ і т.д.}$$

З приведених розрахунків значень сигналу  $x_1 \div x_4$  по відповідних залишках видно, що отримані значення відповідають значенням інформаційних відліків до виконання операції кодування.

Отже, для представленого на рис. 2.6 сигналу при використанні методу кодування та зменшення надлишковості даних на основі залишків коефіцієнт стиснення дорівнює  $k = 1,33$  рази.

Апаратна реалізація принципу перетворення аналогового сигналу  $x(t)$  в цифровий сигнал, представлений в системі залишкових класів, приведена на рис. 2.7 [49].

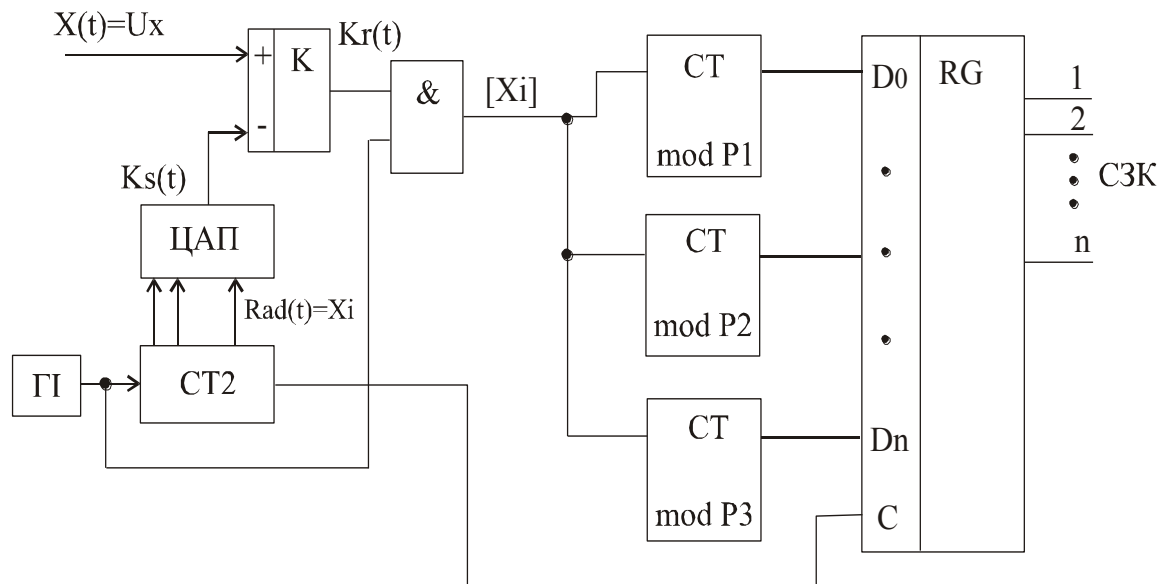


Рис. 2.7. Структурна схема кодера в базисі Крестенсона по модулях P1, P2, P3.

Робота представленої структурної схеми (рис.2.7) відбувається наступним чином: в початковий момент, який відповідає часу  $t = 0$ , лічильник CT2 і регістр RG знаходиться в нульовому стані. Вхідний аналоговий сигнал  $x(t)$  представляється у вигляді пропорційної напруги  $U_x$ , яка подається на вхід компаратора, на опорний вхід якого поданий вихідний сигнал цифро-аналогового перетворювача (ЦАП). При цьому відбувається порівняння значення ступінчатої функції  $Ks(t)$  зі значенням аналогової функції  $x(t)$ .

В результаті отримаємо  $x_i$ :

$$x_i = E \left[ \frac{x(t)}{\delta} \right],$$

Високочастотний сигнал, який модулює генератор ГІ, подається одночасно на вхід лічильника CT2 і на один з входів логічної схеми “Г”. При цьому в процесі формування двійкового коду на виходах лічильника CT2, що відповідає генеруванню системи функцій базису Радемахера  $Rad(t)$ , формується двійковий код  $x_i$ , який в ЦАП перетворюється у відповідну ступінчасту функцію  $Ks(t)$  базису Крестенсона  $x(t) \leq Ks(t)$  [49].



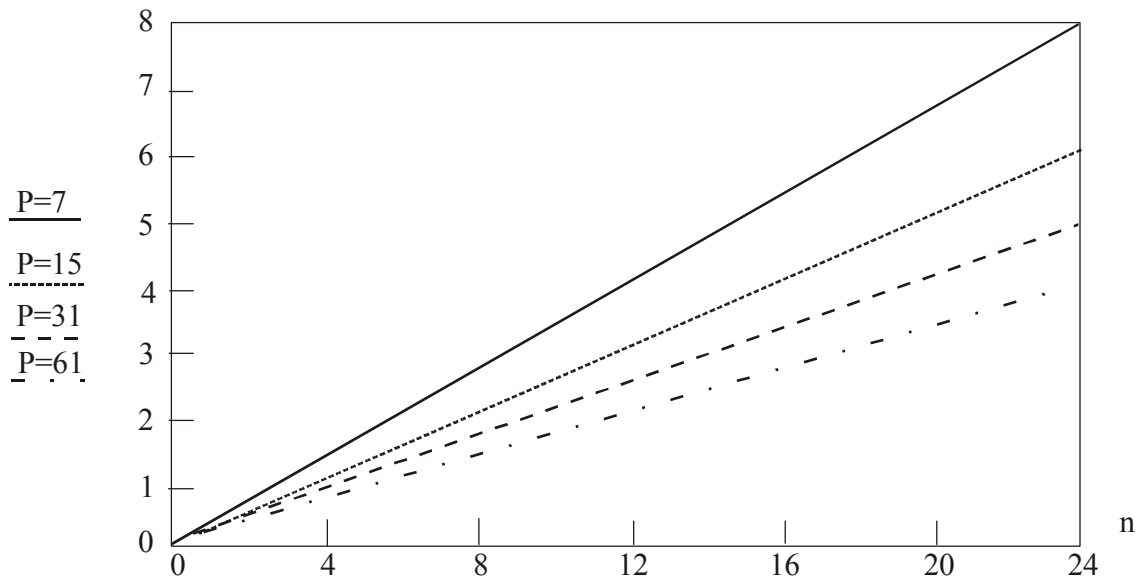


Рис. 2.9. Коефіцієнт стиснення даних в залежності від модуля  $P$ .

Метод залишків доцільно використовувати для кодування процесів з низькою динамікою (телефонна розмова, вимірювання температури та інші).

Представлення даних в системі залишкових класів дає змогу здійснювати паралельну обробку інформації без значного ускладнення обчислювальних засобів.

Особливістю СЗК залишається простота реалізації прямого та зворотного перетворень.

Використання СЗК спрощує побудову систем збору інформації, а також дозволяє вирішувати клас задач, що є невизначеними в позиційних системах числення.

## 2.4. Теоретичні основи та потенціальні можливості кодування цифрових даних у базисі Галуа на основі вертикальної інформаційної технології

Повна система залишків по модулю простого числа  $p$  утворює кінцеве поле порядку  $p$ , яке позначається через  $GF(p)$  і називається простим полем Галуа. Елементи поля  $GF(p) \in \{0, 1, 2, \dots, p-1\}$ , а операції “+”, “-”, “ $\cdot$ ”, “:” виконуються по модулю  $p$  [50 – 58].

Наприклад:  $GF(2)$  – двійкове поле,  $\{0, 1\}$ ,  $GF(3)$  – трійкове поле,  $\{0, 1, 2\}$ , в якому  $1 + 2 = 3 = 0 \pmod{3}$ ;  $2 \cdot 2 = 4 = 1 \pmod{3}$ ;  $1 - 2 = -1 = 2 \pmod{3}$ .

Кінцеві поля  $GF(p^r)$  порядку  $p^r$  утворюються з допомогою непривідних поліномів степені  $r$ .

При використанні примітивних непривідних поліномів  $\pi(x)$  просте поле  $GF(p)$  можна розширити до поля  $GF(p^r)$  за рахунок приєднання кореня  $a$  поліному  $\pi(x)$ , тобто з допомогою порівняння по двох модулях  $p$  і  $\pi(x)$ .

В табл. 2.2 представлені примітивні непривідні поліноми  $\pi(x)$  характеристики 2 з коефіцієнтами із простого поля  $GF(2)$ , а також поліноми характеристик 3, 5, 7 малих степенів з мінімальним числом ненульових коефіцієнтів. З математичної точки зору вибір полінома несуттєвий, так як всі кінцеві поля одного і того ж порядку ізоморфні, але вибір поліному суттєвий з точки зору апаратної реалізації. Детальні таблиці непривідних і примітивних поліномів представлені в [50].

Серед ефективних з точки зору апаратної реалізації двійкових многочленів необхідно виділити тричлени  $\pi(x) = x^r + x^k + 1$ .

Якщо елемент  $\alpha \in GF(2^r)$  представляє собою корінь непривідного двійкового тричлена степені  $r$ , то перші  $r$  степенів елемента  $\alpha$



представляють собою ефективний базис для запису поля  $GF(2^r)$ , так як множення на  $\alpha$  може бути виконано з допомогою  $r$  – розрядного регістра, в зворотній зв'язок якого входить один суматор з двома входами (рис. 2.10)

Таблиця 2.2.

Непривідні примітивні поліноми  $\pi(x) = x^r + f(x)$  степенів  $r$  і характеристик  $p$ .

$p$	$r$	$\pi(x); [x^r = f(x)]$	$p$	$r$	$\pi(x); [x^r = f(x)]$	
2	2	$x^2 + x + 1$	2	17	$x^{17} + x^3 + 1$	
	3	$x^3 + x + 1; x^3 + x^2 + 1$		18	$x^{18} + x^7 + 1$	
	4	$x^4 + x + 1$		19	$x^{19} + x^5 + x^2 + x + 1$	
	5	$x^5 + x^2 + 1$		20	$x^{20} + x^3 + 1$	
	6	$x^6 + x + 1$		21	$x^{21} + x^2 + 1$	
	7	$x^7 + x + 1; x^7 + x^3 + 1$		22	$x^{22} + x + 1$	
	8	$x^8 + x^4 + x^3 + x^2 + 1$		23	$x^{23} + x^5 + 1$	
	9	$x^9 + x^4 + 1$		24	$x^{24} + x^7 + x^2 + x + 1$	
	10	$x^{10} + x^3 + 1$		25	$x^{25} + x^3 + 1$	
	11	$x^{11} + x^2 + 1$		26	$x^{26} + x^6 + x^2 + x + 1$	
	12	$x^{12} + x^6 + x^4 + x + 1$		3	2	$x^2 + x + 2; [x^2 = 2 \cdot x + 1]$
	13	$x^{13} + x^4 + x^3 + x + 1$			3	$x^3 + 2 \cdot x + 1; [x^3 = x + 2]$
	14	$x^{14} + x^{10} + x^6 + x + 1$			4	$x^4 + x + 2; [x^4 = 2 \cdot x + 1]$
	15	$x^{15} + x + 1$		5	2	$x^2 + x + 3; [x^2 = 4 \cdot x + 2]$
	16	$x^{16} + x^{12} + x^3 + x + 1$		7	2	$x^2 + x + 3; [x^2 = 6 \cdot x + 4]$

Складність виконання обчислень в полі  $GF(p^r)$ , а відповідно конструкція і вартість обладнання, які здійснюють ці обчислення, суттєво залежать від вибору представлення поля.

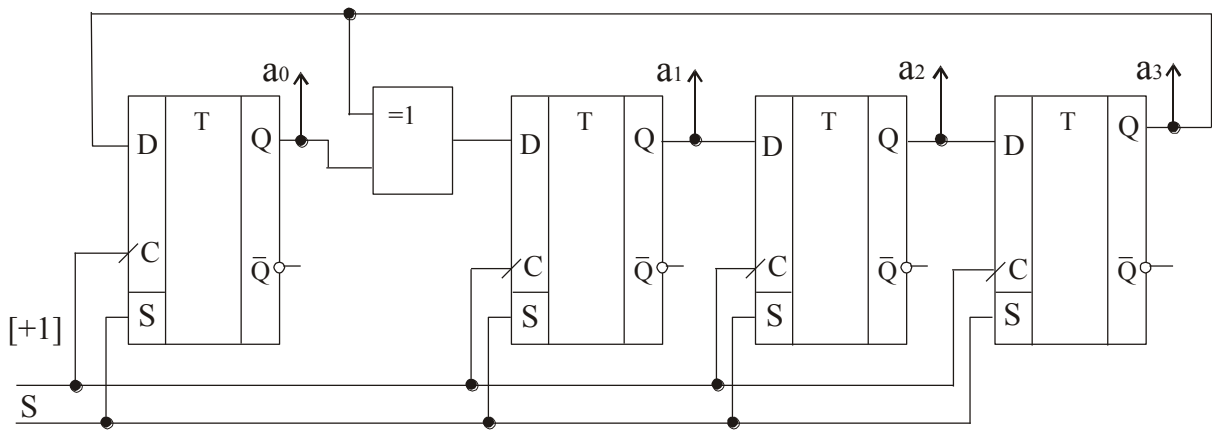


Рис. 2.10. Формувач елементів поля Галуа  $GF(2^4)$ .

Можливі різні представлення елементів розширеного кінцевого поля характеристики 2, при цьому правильний вибір варіанту представлення дозволяє спростити правила виконання операцій над елементами поля.

Найбільше поширення одержали наступні способи представлення елементів поля характеристики 2 [50]:

- а) в вигляді десяткових номерів (модифікованих логарифмів  $Log$ );
- б) степенів  $\alpha^i$  примітивного елемента  $\alpha$ ;
- в) логарифмів  $\log(\alpha^i)$  з основою  $\alpha$ ;
- г) двійкових векторів;
- д) поліномів;
- е) розкладання по нормальному базису та інші [56].

Умовно можна виділити два види представлення – в полярних та прямокутних координатах.

Векторне представлення елементів поля  $GF(2^4)$  можна отримати послідовно за допомогою схеми (рис. 2.10), використовуючи примітивний многочлен  $g(x) = x^4 + x + 1$ . Клас залишків  $\{x\} = \alpha$ , де  $\alpha$  – корінь многочлена  $x^4 + x + 1$ , є примітивним елементом поля  $GF(2^4)$ . Якщо в тригер молодшого розряду занести одиницю, а в інші тригери нулі, то одержимо представлення послідовних степенів елемента  $\alpha$  в формі приведеній в таблиці 2.3.

Представлення поля  $GF(2^4)$ 

Полярні координати			Прямокутні координати	
Десятковий номер N	Степінь $\alpha^i$	Логарифм $\log_{\alpha} \alpha^i$	Двійковий вектор	Поліном $\sum_{i=0}^3 \alpha^i \cdot x^i$
0	$\alpha^{-\infty}$	$-\infty$	0000	0
1	$\alpha^0$	0	0001	1
2	$\alpha^1$	1	0010	$x$
3	$\alpha^2$	2	0100	$x^2$
4	$\alpha^3$	3	1000	$x^3$
5	$\alpha^4$	4	0011	$x+1$
6	$\alpha^5$	5	0110	$x^2+x$
7	$\alpha^6$	6	1100	$x^3+x^2$
8	$\alpha^7$	7	1011	$x^3+x+1$
9	$\alpha^8$	8	0101	$x^2+1$
10	$\alpha^9$	9	1010	$x^3+x$
11	$\alpha^{10}$	10	0111	$x^2+x+1$
12	$\alpha^{11}$	11	1110	$x^3+x^2+x$
13	$\alpha^{12}$	12	1111	$x^3+x^2+x+1$
14	$\alpha^{13}$	13	1101	$x^3+x^2+1$
15	$\alpha^{14}$	14	1001	$x^3+1$
1	$\alpha^{15} = \alpha^0$	15=0	0001	1

Від векторного представлення (двійкових комбінацій) елементів поля можна перейти до їх представлення з допомогою поліномів, якщо

співставити двійкові розряди із степенями змінної  $x$ , які збільшуються справа наліво від 0 до  $r-1$ , так для  $r=4$ :

$$\begin{array}{cccc} x^3 & x^2 & x^1 & x^0 \\ 1 & 1 & 1 & 1 \leftrightarrow x^3 + x^2 + x + 1 \\ 1 & 0 & 1 & 1 \leftrightarrow x^3 + x + 1 \end{array}$$

Схему формування елементів поля  $GF(2^4)$  в зворотному порядку отримаємо, змінивши підключення суматора по mod 2 (рис. 2.11).

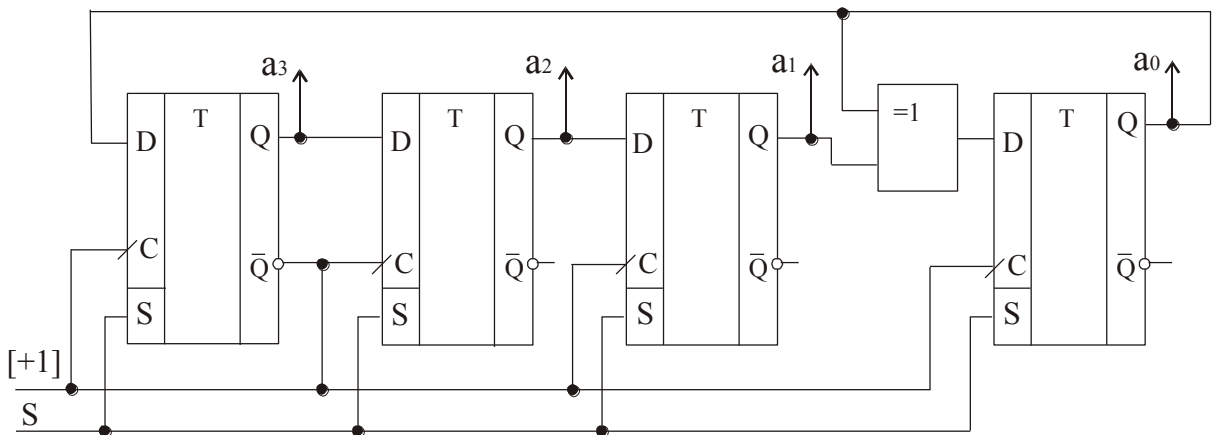


Рис. 2.11. Схема формування елементів поля  $GF(2^4)$  в зворотному порядку.

Ненульові елементи поля утворюють циклічну (мультиплікативну) групу порядку  $2^r - 1$ .

Кінцеві поля використовуються для побудови багатьох відомих кодів (циклічні, БЧХ, Ріда-Соломона, згорткові та інші) і їх декодування, а також в математиці: в теорії блок-схем, в кінцевих геометріях [55, 56].

При завадостійкому кодуванні найбільшу практичну цінність мають поля, порядок яких не перевищує  $2^{12}$ . Поля  $GF(2^r)$  при  $r > 12$  використовуються в системах зв'язку для побудови псевдовипадкових послідовностей.

Важливою властивістю кінцевих полів, що відрізняє їх від нескінчених полів, є наявність циклічної мультиплікативної групи порядку  $p^r - 1$ .

Кінцева мультиплікативна циклічна група визначається як множина елементів  $1, \alpha, \alpha^2, \dots, \alpha^{r-1}$  при умові  $\alpha^r = 1$ .

Рекурентні співвідношення або різницеві рівняння:

$$\sum_{j=0}^k h_j \cdot \alpha_{i+j} = 0 \quad (2.3)$$

або

$$\alpha_{i+k} = -\sum_{j=0}^{k-1} h_j \cdot \alpha_{i+j}, \quad (2.4)$$

де  $h_0 \neq 0$ ,  $h_k = 1$  і кожне  $h_j$  належить полю  $GF(q)$ .

Розв'язком цих рівнянь є послідовність  $\alpha_0, \alpha_1, \alpha_2 \dots$  елементів поля  $GF(q)$ .

Співвідношення (2.4) визначає правило обчислення  $\alpha_k$  по заданих значеннях величин  $\alpha_0, \alpha_1, \dots, \alpha_{k-1}$ . По відомих значеннях  $\alpha_0, \alpha_1, \dots, \alpha_k$  можна знайти  $\alpha_{k+1}$  і т.д.

Для обчислення суми (2.4) і, відповідно, для обчислення величин  $\alpha_k$  по значеннях  $k$  попередніх членів послідовності використовується лінійна послідовна перемикаюча схема (рис. 2.12). Вихідні величини  $\alpha_0, \alpha_1, \dots, \alpha_{k-1}$  заносимо в розряди пристрою, послідовні зсуви якого відповідають обчисленню послідовних символів, вихід після  $i$ -го зсуву дорівнює  $\alpha_i$ .

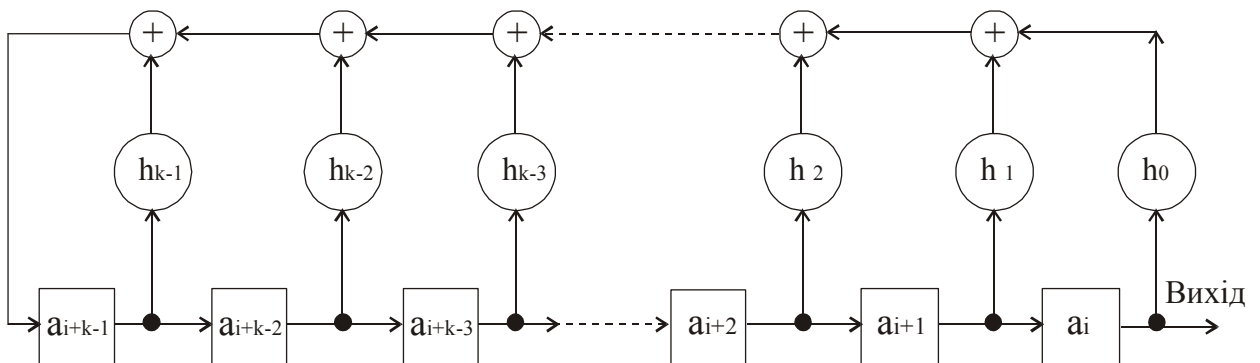
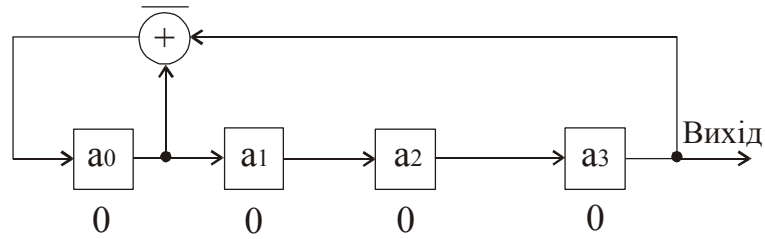


Рис. 2.12. Регістр зсуву із зворотнім зв'язком.

На рис. 2.13 показано реєстр зсуву, лінійні зворотні зв'язки якого описуються непривідним примітивним багаточленом  $h(x) = x^4 + x + 1$  над полем  $GF(2)$ .



Десяткове значення	Вихідний стан реєстру			
	1	2	3	4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	0	1	1	1
5	1	0	1	1
6	1	1	0	1
7	0	1	1	0
8	0	0	1	1
9	1	0	0	1
10	0	1	0	0
11	1	0	1	0
12	0	1	0	1
13	0	0	1	0
14	0	0	0	1
	0	0	0	0

Рис. 2.13. Схема формування та елементи поля  $GF(2^4)$ .

Вихід зчитується з правого кінця реєстра і дорівнює двійковій послідовності з періодом  $N = 2^4 - 1$ :

$$a = a_0, a_1 \cdots a_{13}, a_{14}.$$

При початковому стані 0001 регістра зсуву із зворотнім зв'язком вихідна послідовність має вигляд

$$a = a_0 a_1 \cdots = 100010011010111|1000 \cdots.$$

Недоліком представлених схем є необхідність завантаження початкового стану регістру.

Даний недолік можна усунути, якщо замість операції “виключаюче АБО” використати операцію “виключаюче АБО інвертоване” (рис. 2.13).

При початковому стані регістру 0 0 0 0, вихідна послідовність буде мати вигляд:

$$000011101100101|0000.$$

Приведеною послідовністю можна закодувати десяткові числа від 0 до 14. Згідно з рекурентним співвідношенням (2.4) по відомих значеннях  $a_0 a_1 a_2 a_3$  можна знайти  $a_4$  і т.д. Отже, послідовні зсуви регістру однозначно відповідають обчисленню наступних значень циклічної послідовності Галуа.

Для кодування значень технологічних параметрів на основі вертикальної інформаційної технології використовують поліноми порядку  $2^r$ , де  $r > 20$ . Щоб закодувати десяткові цифри від 0 до 1048575 кодом поля Галуа, необхідно вибрати непривідний примітивний поліном  $h(x) = x^{20} + x^3 + 1$  (табл. 2.3).

## 2.5. Дослідження базисних функцій теоретико-числового перетворення Галуа

Найповніше дискретизовані і квантовані функції досліджені в базисі Радемахера. Причому, розроблені теоретичні і методологічні основи генерації базових елементарних функцій:  $y_i = n \cdot x_i$ ;  $x_i^n$ ;  $\sqrt[n]{x_i}$ ;  $\log_n x_i$ ;  $e^{x_i}$ ;  $\cos x_i$ ;  $\sin x_i$  і т. д, а також відповідні структури спецпроцесорів для їх генерації та цифрової обробки [59].

Метод дельта-модуляції, розроблений академіком Харкевичем [60], знайшов широке застосування в техніці кодування та стиснення технологічних даних [35].

В основу принципу дельта-модуляції покладена процедура кодування на основі вибору такого кроку квантування сигналів по рівню  $\delta$  та кроку дискретизації  $\Delta t$ , при якому виконується умова:

$$\Delta_i = \begin{cases} 1, & x_i - x_{i-1} = +1 \\ 0, & x_i - x_{i-1} = -1 \\ 0, 1, \dots & x_i - x_{i-1} = 0 \end{cases}, \quad (2.5)$$

де  $x_i$  - поточне значення відліку стану джерела інформації.

Метод дельта-модуляції не характеризується повнотою базисних функцій, коли градієнт наростання значень  $x_i$  на інтервалі дискретності перевищує умову (2.5).

Неповнота системи базисних функцій методу дельта-модуляції обумовлює головний недолік названого методу, який полягає в тому, що при невиконанні умови (2.5) спостерігається відставання або часове запізнення фактичних станів джерела інформації при декодуванні стиснених даних.

В той же час в досліджуваному базисі Галуа така робота не проведена. Тому є актуальною задача розробки каталогу названих дискретизованих і квантованих функцій в базисі Галуа.



Наявність такого каталогу створить основу та допоможе теоретично дослідити потенційні можливості стиснення даних в базисі Галуа.

Розглянемо лінійну функцію  $y_i = n \cdot x_i$ .

Перші елементарні базисні лінійні функції Галуа нульового порядку  $y_i = G_i$  показані на рис. 2.14.

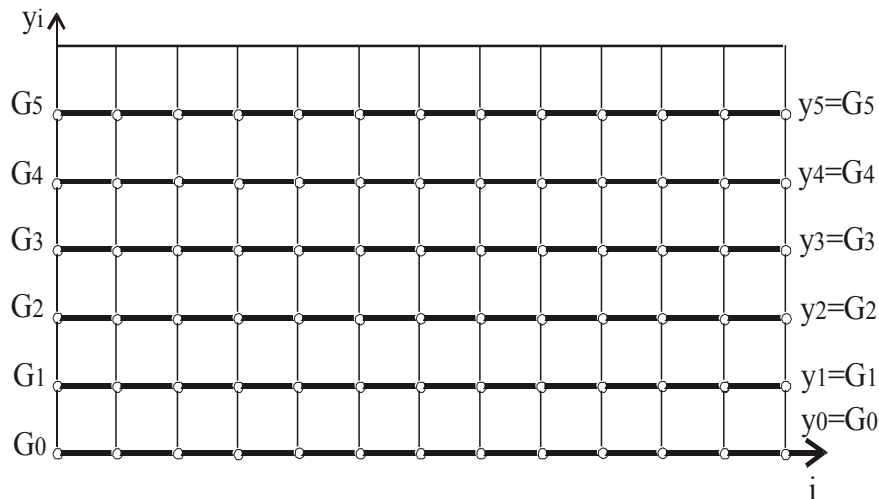


Рис. 2.14. Елементарні базисні лінійні функції Галуа.

З рис 2.14 видно, що існує інваріантність представлення різних функцій однаковими послідовностями бітів Галуа, тому даний метод приводить до неоднозначності кодування аналогічно представленню інтегралів з невизначеною величиною константи  $C$ .

Для того, щоб уникнути даного недоліку, кожна з функцій кодується послідовністю Галуа, яка генерується різним ключем: для  $n_g = 4$  (табл. 2.4).

Такий спосіб кодування інформаційних потоків в базисі Галуа забезпечує коефіцієнт стиснення даних

$$k_{c1} = \hat{E}[\log_2 A],$$

де  $A$  – діапазон квантування, при  $A = 1024$ ,  $k_{c1} = 10$  (рис. 2.15).

Застосування даного методу можливо при виконанні умови, що досліджуваний об'єкт перебуває у певному стані на протязі  $n_g + 1$  тактів, де  $n_g$  – розрядність кодону Галуа.

## Фазові базисні функції

Назва	Кодон	Послідовність Галуа
$G_0$	0000	0000101001101111
$G_1$	1000	1000010100110111
$G_2$	1100	1100001010011011
$G_3$	1110	1110000101001101
$G_4$	1111	1111000010100110
$G_5$	0111	0111100001010011
$G_6$	1011	1011110000101001
$G_7$	1101	1101111000010100
$G_8$	0110	0110111100001010
$G_9$	0011	0011011110000101
$G_{10}$	1001	1001101111000010
$G_{11}$	0100	0100110111100001
$G_{12}$	1010	1010011011110000
$G_{13}$	0101	0101001101111000
$G_{14}$	0010	0010100110111100
$G_{15}$	0001	0001010011011110

Без врахування вказаної умови коефіцієнт стиснення визначається за формулою:

$$k_{c2} = \frac{n \cdot m}{(n+1) \cdot f_a + (m - f_a)}$$

Приклад квазістаціонарного процесу, який однозначно можна представити біт-орієнтованою базисною функцією Галуа  $y_i = G_i$ , що описується лінією, показаний на рис. 2.16.

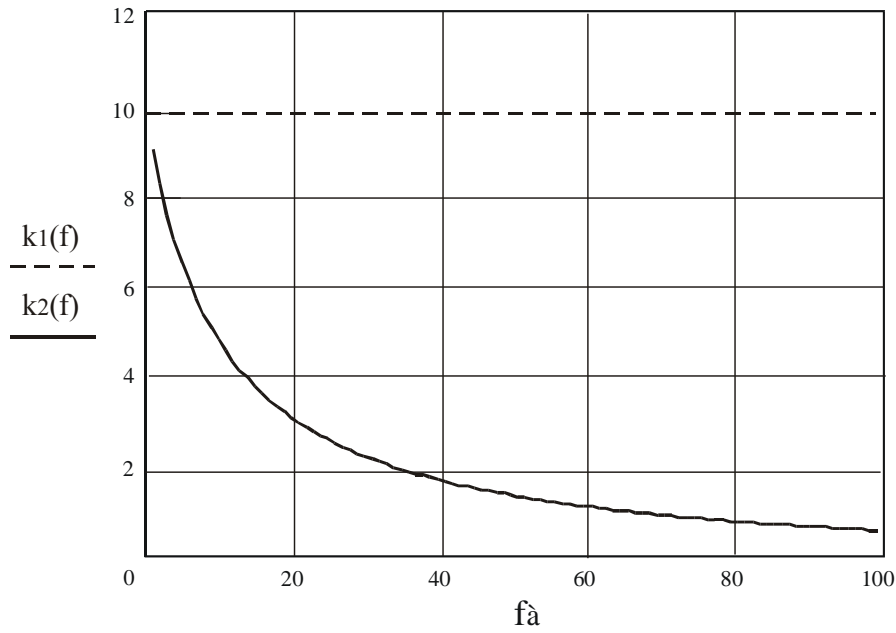


Рис. 2.15. Залежність коефіцієнта стиснення від кількості активних відліків.

Практичним застосуванням запропонованого методу є кодування станів об'єктів керування [4].

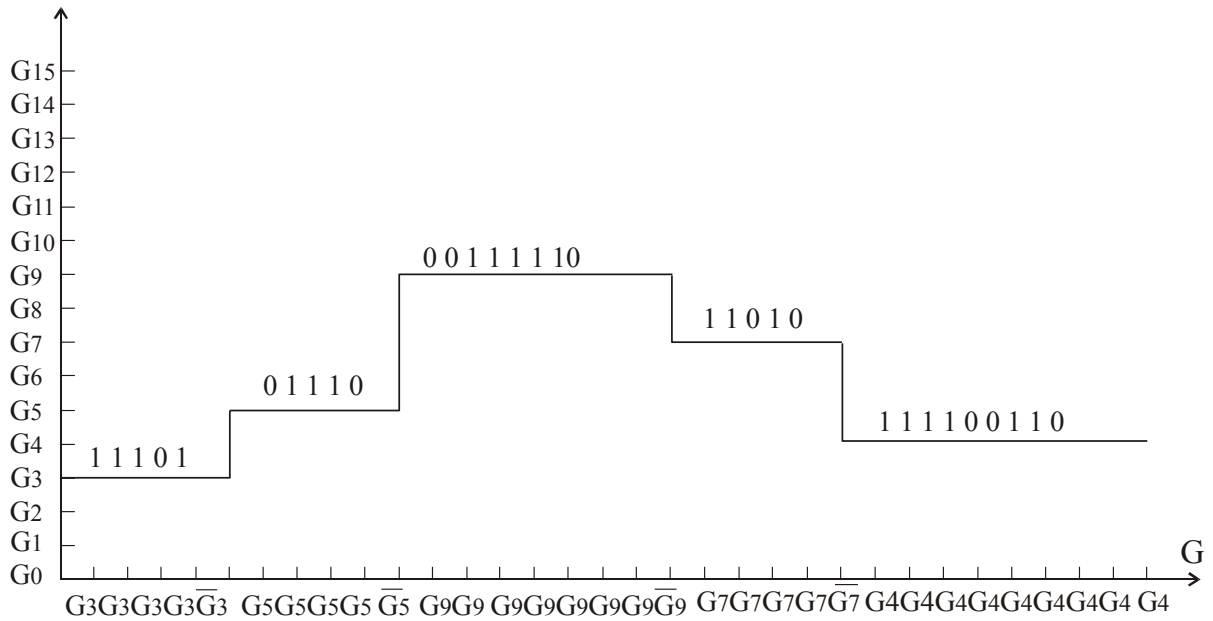


Рис. 2.16. Приклад кодування на основі базисних функцій Галуа нульового порядку.

Базисні функції Галуа нульового порядку доцільно використовувати для кодування станів об'єктів керування. Розглянемо приклади об'єктів, наведених в таблиці 1.1. Установка буріння має 4 семантичні, 8

технологічних та 5 інформаційних станів. Якщо у відповідність кожному стану поставити певний кодон Галуа, то отримаємо для кожного технологічного стану наступний код:

0001101 – Буріння;  
 0010110 – Аварія;  
 0110001 – Викид;  
 1100010 – Ліквідація.

Якщо об'єкт керування в даний момент часу перебуває у стані буріння, то послідовність бітів має вигляд 0001101. У разі його переходу в інший стан останній біт, на момент зміни стану вказаної вище послідовності, інвертується 0001100 і генерується послідовність на основі нового кодону, який вказує в якому стані знаходиться ОК в даний момент часу.

Цей метод дозволяє отримати істотне зменшення об'єму інформації. Якщо поставити у відповідність кожному стану ОК певний рівень (1–8), то кожний відлік потрібно кодувати 3 бітами, застосування запропонованого методу дозволяє кодувати 3 бітами тільки перший момент часу, а решту моментів часу представляти одним бітом. Тому на передаючій стороні генератор постійно формує послідовність Галуа, а передає кодон тільки в тому випадку, коли значення контрольованого параметру змінилося, тобто відліки стали активні.

Оскільки технологічні установки, як правило, перебувають у певному стані на протязі тривалого часу, що значно перевищує частоту опитування, то дане кодування є ефективним, причому, чим більша тривалість перебування ОК в кожному стані, тим вища ефективність кодування, тобто більший коефіцієнт стиснення.

У випадку, коли зміна значення функцій відбувається на протязі менше  $n$  тактів, необхідно вводити базисні функції Галуа першого порядку (рис. 2.17).

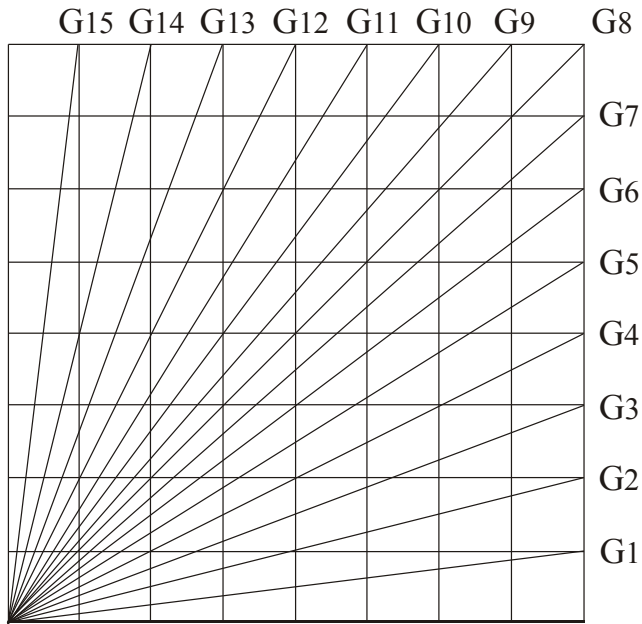


Рис.2.17. Базисні функції Галуа першого порядку.

При цьому біт-орієнтоване кодування дискретних функцій в базисі Галуа однозначно можливе для стрибкоподібних і лінійно-наростаючих функцій, що ілюструється прикладом (рис. 2.18).

Кодування даних за допомогою фазових базисних функцій Галуа є більш ефективне, ніж кодування за допомогою різних кодових ключів.

Метод кодування з використанням базисних функцій Галуа першого порядку доцільно використовувати для кодування інтегрованих значень параметрів об'єктів керування. Кут нахилу ліній визначає швидкість зростання інтегралу.

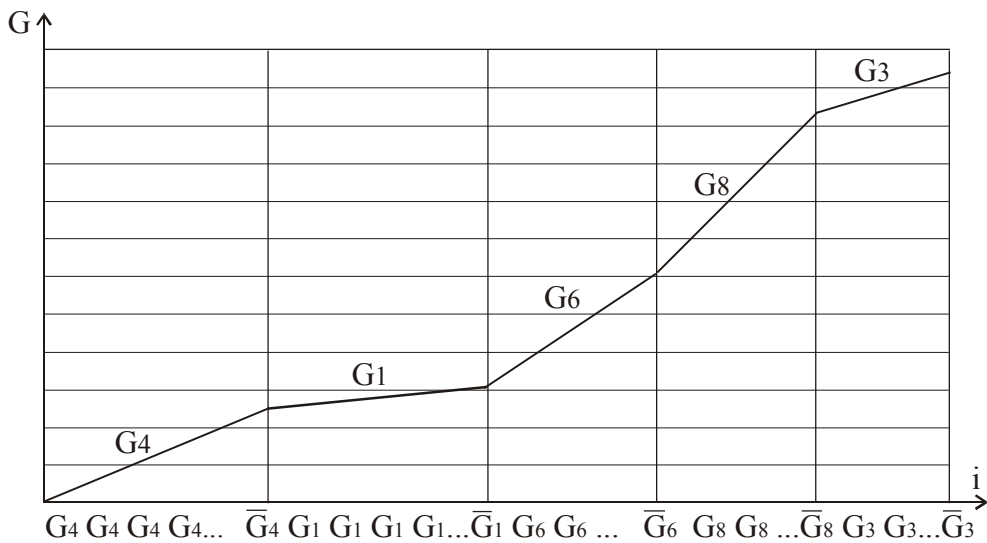


Рис. 2.18. Кодування на основі базисних функцій першого порядку.

Кожному куту нахилу відповідає кодон Галуа, що дозволяє однозначно декодувати значення інтегралу функції. Запропонований метод позбавлений недоліку, що полягає у постійному зростанні розрядності відліків, оскільки кут нахилу ліній знаходиться у визначеному діапазоні.

## ВИСНОВКИ

1. Вдосконалено існуючі ЛСІМ шляхом заміни унітарного коду послідовністю Галуа, що дозволило фіксувати час зміни параметрів сигналу без введення надлишковості. Розроблена нова ЛСІМ, яка базується на оцінці відхилення значення контрольованого параметру від передбачуваного і дозволяє ефективно відслідковувати аварійні ситуації на об'єктах електроенергетики.

2. Проведений аналіз та дослідження різних методик оцінки коефіцієнта стиснення показали, що найбільш ефективним є використання комплексного коефіцієнта ефективності, який обчислюється як добуток або сума окремих факторів.

3. Проведено дослідження методу стиснення технологічних даних в базисі Крестенсона. Коефіцієнт стиснення методу кодування на основі залишків дорівнює 1,5–3 і залежить від типу даних. Перевагою використання перетворень системи залишкових класів є незалежне утворення інформаційних розрядів, що дає можливість їх паралельної обробки. Розроблено структуру кодера в базисі Крестенсона.

4. Запропоновано базисні функції Галуа нульового та першого порядку, на основі яких розроблено та досліджено метод стиснення даних. Використання базисних функцій Галуа першого порядку для кодування інтегрального значення контрольованого параметру, на відміну від вертикальної інформаційної технології, забезпечує рівномірне слідування бітів Галуа при передаванні.

## РОЗДІЛ 3

### РОЗРОБКА ТА ДОСЛІДЖЕННЯ МЕТОДІВ СТИСНЕННЯ ДАНИХ У БАЗИСІ ГАЛУА

#### 3.1. Метод стиснення даних, представлених гармонічними сигналами

Особливо гостро задача ефективного використання ресурсів пам'яті постала при розробці цифрового реєстратора миттєвих значень струмів та напруг у аварійних режимах роботи електричної мережі.

Значення контрольованих параметрів електроенергетичних систем та мереж описуються гармонічними сигналами.

Нехай маємо один період синусоїди, яка описується рівнянням

$$x_1(t) = A_0 \cdot \sin(\omega \cdot t),$$

де  $A_0$  – амплітуда;  $\omega$  – частота;  $t$  – час.

Побудуємо на цьому ж графіку (рис. 3.1) косинусоїди  $x_2(t) = A_0 \cdot \cos(\omega \cdot t)$  і  $x_3(t) = A_0 \cdot \cos(\omega \cdot t + \pi)$  і синусоїду  $x_4(t) = A_0 \cdot \sin(\omega \cdot t + \pi)$ .

Таким чином, отримаємо систему рівнянь:

$$\begin{cases} x_1(t) = A_0 \cdot \sin(\omega \cdot t) \\ x_2(t) = A_0 \cdot \cos(\omega \cdot t) \\ x_3(t) = A_0 \cdot \cos(\omega \cdot t + \pi) \\ x_4(t) = A_0 \cdot \sin(\omega \cdot t + \pi) \end{cases}.$$

Представлені функції мають ряд особливих точок (рис. 3.1) залежно від параметру  $t$ :



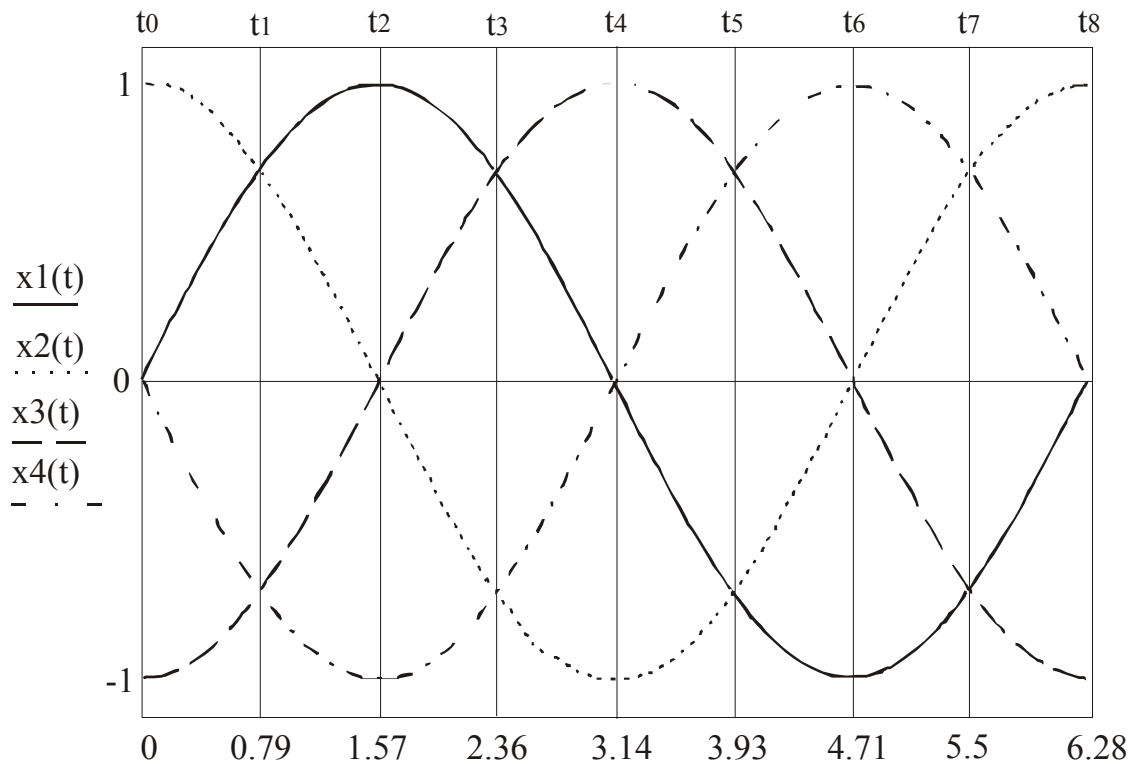


Рис. 3.1. Чотирифазний гармонічний сигнал.

$$\begin{aligned}
 t = 0, & \quad x_2(t) = A_0 \\
 t = 1, & \quad x_1(t) = x_2(t) \\
 t = 2, & \quad x_1(t) = A_0 \\
 t = 3, & \quad x_1(t) = x_3(t) \\
 t = 4, & \quad x_3(t) = A_0 \\
 t = 5, & \quad x_3(t) = x_4(t) \\
 t = 6, & \quad x_4(t) = A_0 \\
 t = 7, & \quad x_4(t) = x_2(t)
 \end{aligned} \tag{3.1}$$

Оскільки функція є періодичною, то в точці  $t = 8$  маємо повторення  $t = 0$ .

При подачі на вхід порогових систем (рис. 3.2) вхідного синусоїдального сигналу з довільною фазою і чотирифазним пороговим приймачем, описаного системою рівнянь (3.1), на виході отримаємо послідовність імпульсів (рис. 3.3).

Групування імпульсів щодо рівності похідної і сигналу, відносно точок максимуму амплітуди синусоїди дозволяє однозначно визначити фазу

вхідного сигналу при декодуванні (відновленні) інформації.

В результаті гармонічний синусоїдальний сигнал можна закодувати, використовуючи сумісне представлення в базисі Радемахера і Галуа, наступним чином:

$$\{A_0, t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}.$$

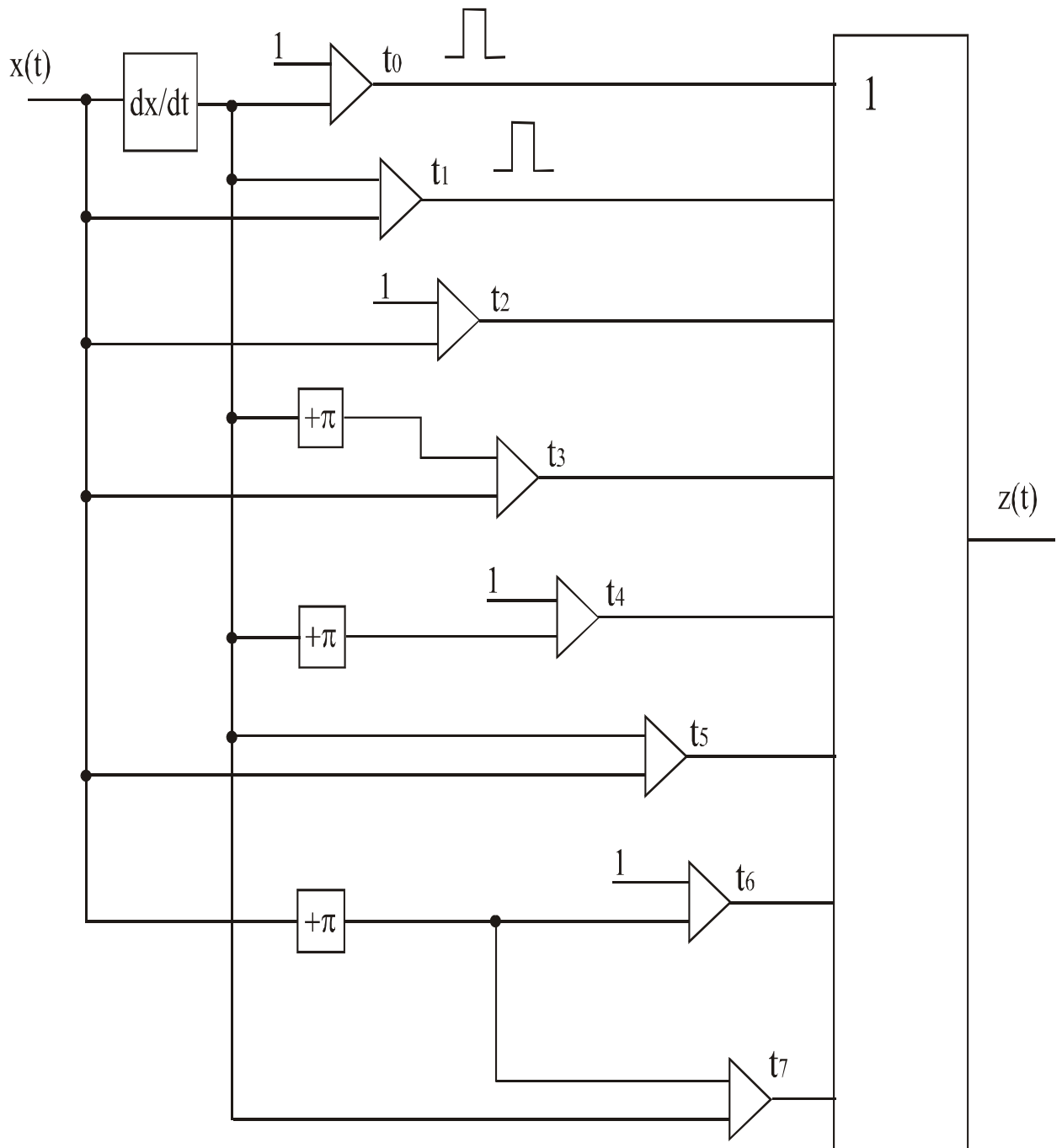


Рис. 3.2. Порогова схема:

$z(t)$  – реакція порогової схеми на вхідний синусоїдальний сигнал.

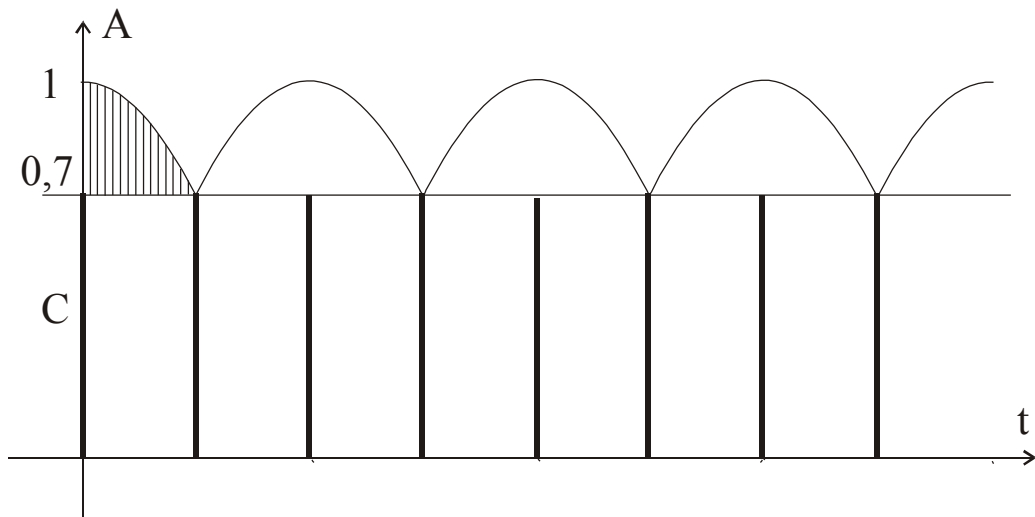


Рис. 3.3. Реакція порогової схеми на вхідний синусоїдальний сигнал.

В результаті, всі ці функції можуть бути зведені до одного фрагменту (рис. 3.4), за допомогою якого можна однозначно відновити сигнал.

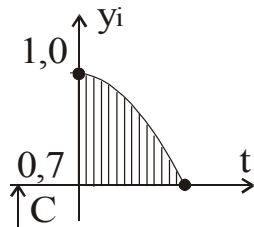


Рис. 3.4. Фрагмент сигналу:  $C=0,707$ .

Декодування гармонічного сигналу відбувається на основі алгоритму (рис. 3.5):

$$x(t) = \begin{cases} t_0 \leq t < t_1, & f_0(t) = \int_0^t (z_0(t) + C) dt; \\ t_1 \leq t < t_2, & f_1(t) = \overrightarrow{C + z_0(t)}; \\ t_2 \leq t < t_3, & f_2(t) = C + z_0(t); \\ t_3 \leq t < t_4, & f_3(t) = \frac{d(\overrightarrow{z_0(t) + C})}{dt}; \\ t_4 \leq t < t_5, & f_4(t) = \frac{d(z_0(t) + C)}{dt}; \\ t_5 \leq t < t_6, & f_5(t) = 0 - \overrightarrow{(z_0(t) + C)}; \\ t_6 \leq t < t_7, & f_6(t) = 0 - ((z_0(t) + C)); \\ t_7 \leq t < t_8, & f_7(t) = 0 - \left( \int_0^t \overrightarrow{(z_0(t) + C)} dt \right). \end{cases} \quad (3.2)$$

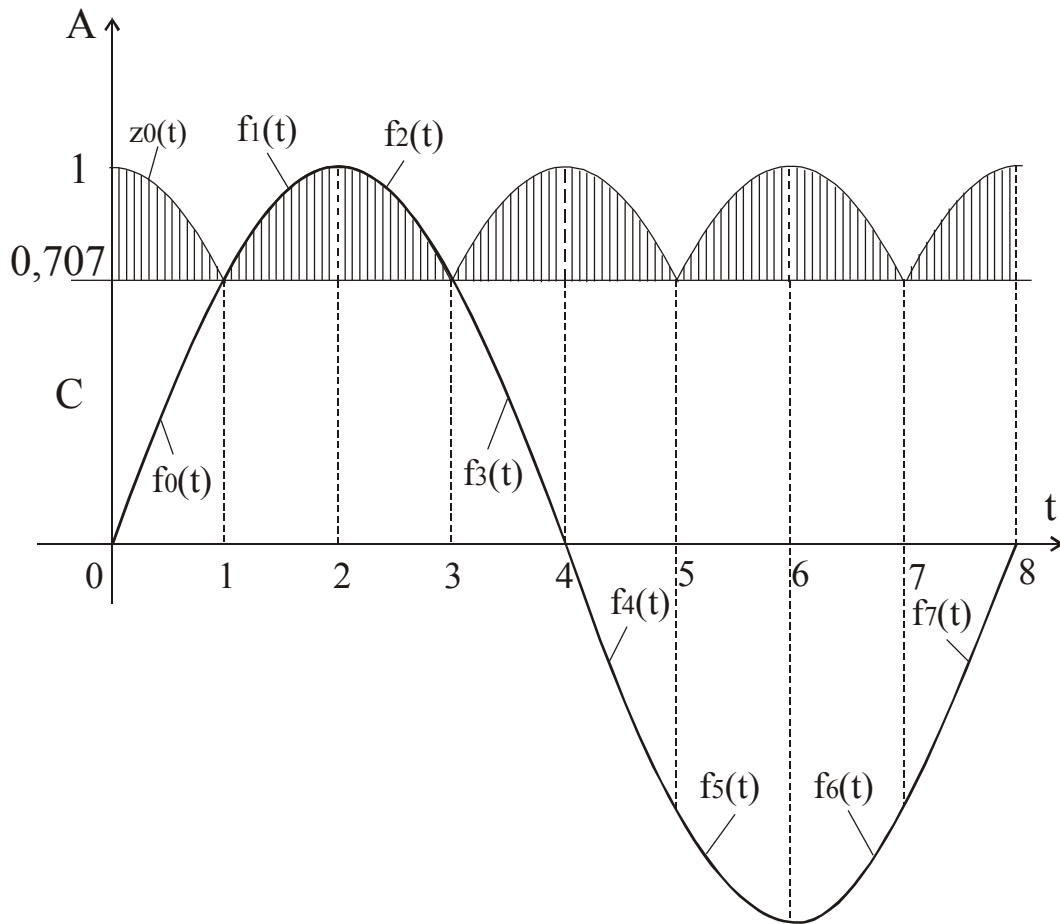


Рис. 3.5. Декодування синусоїди на основі фрагменту  $z_0(t)$ .

Інформація на виході кодера на інтервалі періоду синусоїди представляється двома масивами:  $A_{z_0}$  – що представляє амплітуду синусоїди, яка вважається незмінною на протязі періоду, і масив кодів  $\{t_i\}$ , що відповідають моментам зміни функцій в алгоритмі (3.1).

На основі системи рівнянь (3.2) однозначно відновлюється значення синусоїди, якщо в пам'яті записано наступний масив:

$$\{A_{z_0}\}, \{t_i\}.$$

Оцінимо об'єм даних джерела при описаному способі кодування:

$$I_x = \hat{E}[\log_2 z_{i\max}] + 7 \cdot \hat{E}[\log_2 T],$$

де  $z_{i\max}$  – максимальне значення фрагмента функцій  $x(t)$ ;

$T$  – максимальна тривалість одного періоду синусоїди.

Наприклад, при передаванні сигналів стандартами ІКМ  $A_{0x} = 256$ ,

частота дискретизації 8 кГц.

При кодуванні описаним способом  $z_{i\max} = (1 - C) \cdot A_{0x}$ ,

де  $C = \frac{\sqrt{2}}{2}$ ,  $z_{i\max} = 75$ .

Звідки  $\hat{E}[\log_2 z_{i\max}] = \hat{E}[\log_2 75] = 7$  (біт).

Якщо частота сигналу  $F_c$  змінюється від 10 Гц до 8000 Гц, відповідно період  $T$  змінюється від 100000 до 125 мкс.

Для кодування  $t_i$  необхідно 17 біт.

$$\hat{E}[\log_2 100000] = 17 \text{ (біт)}.$$

Таким чином, обсяг даних буде дорівнювати

$$I = 7 + 17 \cdot 8 = 143 \text{ (біт)}.$$

Аналіз запропонованого методу кодування показує, що реєструвати всі інтервали  $t_i$  недоцільно, їх можна буде обчислити в процесорі декодера, якщо прийняти допущення, що час  $t_i$  є базовим і на протязі одного періоду не змінюється, тобто  $t_1 = t_2 = t_3 = \dots = t_8$ .

В даному випадку код синусоїди буде представлений двома кодами:

$\{z_{i\max}, t_1\}$ , а обсяг даних, який однозначно описує параметри синусоїди, буде дорівнювати:

$$I_2 = 7 + 17 = 24 \text{ (біт)}.$$

Дане представлення має недоліки:

а) маніпуляція сигналу синусоїди чотирма фазами не дозволяє синхронізуватися (визначити перехід функції через 0);

б) кодування моментів часу  $t$  в базисі Радемахера приводить до значної надлишковості;

в) не дозволяє відслідковувати зміни сигналу на протязі періоду.

Тому пропонується метод кодування синусоїдальних сигналів, який базується на наступному: при переході синусоїди через 0 і додатній похідній, що відповідає рівнянням:

$$\begin{cases} x(t) = 0 \\ dx/dt > 0 \end{cases}$$

відбувається реєстрація значень сигналу і паралельно відбувається обчислення похідної. В момент рівності зареєстрованого значення сигналу і обчисленого значення похідної спрацьовує перший компаратор порогової схеми порівняння і реєструється або передається в канал зв'язку біт Галуа. Обчислені значення похідної до цього моменту формують фрагмент  $Z_i = \frac{dx(t)}{dt}$ , який передається в канал зв'язку після біту Галуа. Далі відбувається порівняння реальних значень сигналу з обчисленими згідно алгоритму декодування, що описується системою рівнянь (3.2). При спрацюванні кожного наступного компаратора порогової схеми в канал зв'язку поступає біт Галуа. Якщо не спрацював один із компараторів, то біт Галуа інвертується. Після інвертованого біту в канал зв'язку передають  $l$  реальних значень виміряного параметру. Величина  $l$  визначається частотою дискретизації  $l = \frac{l_d}{8}$ , де  $l_d$  – кількість точок дискретизації на період гармонічного сигналу. З зареєстрованих значень за допомогою перетворень кодування – декодування формується новий фрагмент, який подається на наступний компаратор схеми порівняння. Якщо даний компаратор спрацьовує, то зареєстрований фрагмент вважається базовим для всіх наступних етапів кодування і в канал подається прямиий біт Галуа. Якщо схема не спрацьовує, тобто форма сигналу відрізняється від форми гармонічного сигналу, то біт Галуа інвертується і за ним в канал зв'язку поступають  $k$  значень реального сигналу.

Експериментальні дослідження запропонованого методу кодування гармонічних сигналів проведено на основі даних, знятих цифровим реєстратором аналогових сигналів, розробленим в Інституті мікропроцесорних систем керування об'єктами електроенергетики (м. Львів), який забезпечує реєстрацію (осцилографування) миттєвих значень

електричних параметрів (струмів, напруг) на первинному електричному обладнанні в передаварійних і аварійних режимах.

Перетворення аналогового сигналу в цифровий здійснюється за допомогою 12-розрядного АЦП з частотою дискретизації  $f_d = 2000$  Гц.

Оцінимо об'єм даних джерела при заданих параметрах:

$$I = n \cdot l_d,$$

де  $n$  – розрядність АЦП;  $l_d$  – кількість вибірок сигналу,  $l_d = f_d \cdot T$ ;  $T$  – період сигналу в електричній мережі.

При  $n = 12$ ,  $l_d = 2000 \cdot 0,02 = 40$ .

Отже, для кодування одного періоду гармонічного сигналу необхідно:

$$I = 12 \cdot 40 = 480 \text{ біт.}$$

При кодуванні вказаних сигналів запропонованим методом для  $n = 12$ :

$$Z_i = \left(1 - \frac{\sqrt{2}}{2}\right) \cdot 2^n = 0,293 \cdot 4096 = 1200.$$

Звідки  $n_1 = \hat{E}[\log_2 Z_i] = \hat{E}[1200] = 11$  біт.

Отже, об'єм даних фрагмента (рис.3.5) буде дорівнювати:

$$I_1 = \left(\hat{E}\left[\frac{l_d}{8}\right] \cdot n_1\right) = 55 \text{ біт.}$$

При застосуванні запропонованого методу для потокового кодування, параметр часу  $t_i$  можна представити послідовністю Галуа типу 11101000.

Якщо на протязі одного періоду гармонічний сигнал не змінив свою форму, тобто спрацювали всі вісім компараторів, ми отримаємо наступний масив даних:

$$G Z_1 Z_2 Z_3 Z_4 Z_5 G G G G G G G G.$$

Перший прямиий біт Галуа  $G$  вказує на те, що далі слідує масив значень кодового фрагменту  $Z_i$ , за допомогою якого відбувається декодування сигналу. Наступні прями біти Галуа вказують на те, що декодування відбувається згідно алгоритму.

Об'єм даних для вищевказаних параметрів сигналу дорівнює:

$$I_x = 1+11+11+11+11+11+1+1+1+1+1+1+1 = 63 \text{ біт.}$$

Коефіцієнт стиснення дорівнює

$$k_c = \frac{I}{I_x} = \frac{480}{63}.$$

Застосування даного алгоритму дозволяє реєструвати гармонічний сигнал із зменшенням надлишковості в  $k_c = 7.6$  рази на протязі одного періоду порівняно з початковим об'ємом даних без втрат точності.

Осцилограми комплексного значення напруги  $x(t)$  (рис. 3.6 і рис. 3.7), описують реальний перехідний процес, що відбувається під час аварійних станів на об'єктах електроенергетики. Початок перехідного процесу представлено на рис. 3.6, а момент стабілізації перехідного процесу і переходу до нормального режиму роботи на рис. 3.7 .

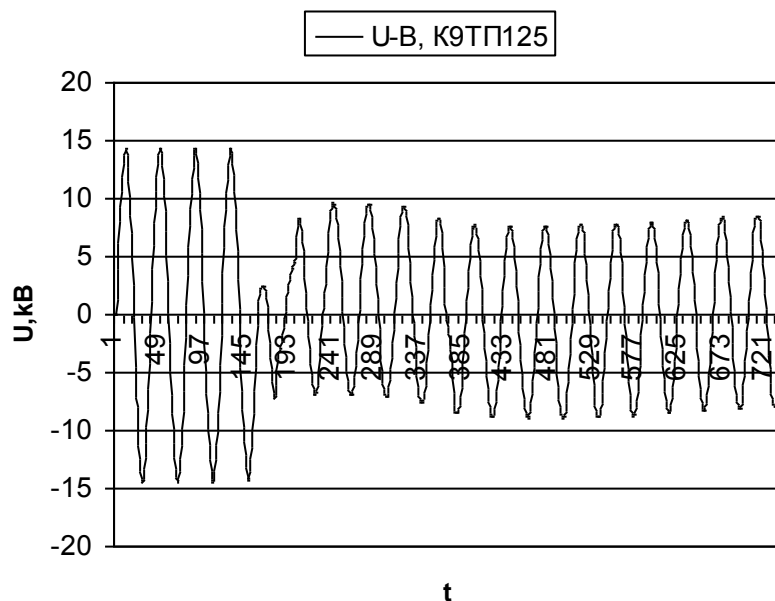


Рис. 3.6. Осцилограма сигналу в момент аварії на об'єкті електроенергетики.

В результаті кодування сигналу представленого на рис. 3.6. отримаємо потік даних:

$$G\{Z_i\}GGG\dots G\bar{G}\{y_i\}\bar{G}\{y_i\}\dots G\{Z_i\}GG\dots\bar{G}\{y_i\}.$$

Розрахуємо об'єм даних при кодуванні запропонованим методом.



Для кодування одного масиву  $\{Z_i\}$  використовується  $I_1 = 55$  біт, а одного масиву  $\{y_i\}$  –  $I_2 = n \cdot l = 12 \cdot 5 = 60$  біт.

Кількість бітів Галуа обчислюється за формулою:

$$I_G = \frac{t_r}{T} \cdot 8, \quad I_G = 152 \text{ біт},$$

де  $t_r$  – час реєстрації.

Об'єм даних для кодування перехідного процесу (рис.3.6) обчислюється за формулою:

$$I_\Sigma = c_1 \cdot I_1 + c_2 \cdot I_y + I_G,$$

де  $c_1$  – кількість фрагментів  $\{Z_i\}$  в закодованій послідовності,  $c_1 = 3$ ;

$c_2$  – кількість фрагментів  $\{y_i\}$  в закодованій послідовності,  $c_2 = 96$ .

$$I_\Sigma = 6076 \text{ біт}.$$

Початковий об'єм даних  $I_0 = 8892$  біт.

Коефіцієнт стиснення дорівнює:

$$k_c = \frac{I_0}{I_\Sigma}; \quad k_c = 1,46.$$

При кодуванні сигналу, представленого на рис. 3.7, отримаємо потік даних:

$$G\{Z_i\}GGGGGGG\dots GGGGGG\dots GGGGGG,$$

його об'єм обчислюється за формулою

$$I_\Sigma = c_1 \cdot I_1 + I_G, \quad I_\Sigma = 207,$$

при  $c_1 = 1$ .

Коефіцієнт стиснення дорівнює:

$$k_c = \frac{I_0}{I_\Sigma}, \quad k_c = 42,96.$$

В розробленому методі ми працюємо з масивом цифрових даних, але при цьому враховуємо структуру даних до процесу аналого-цифрового перетворення.

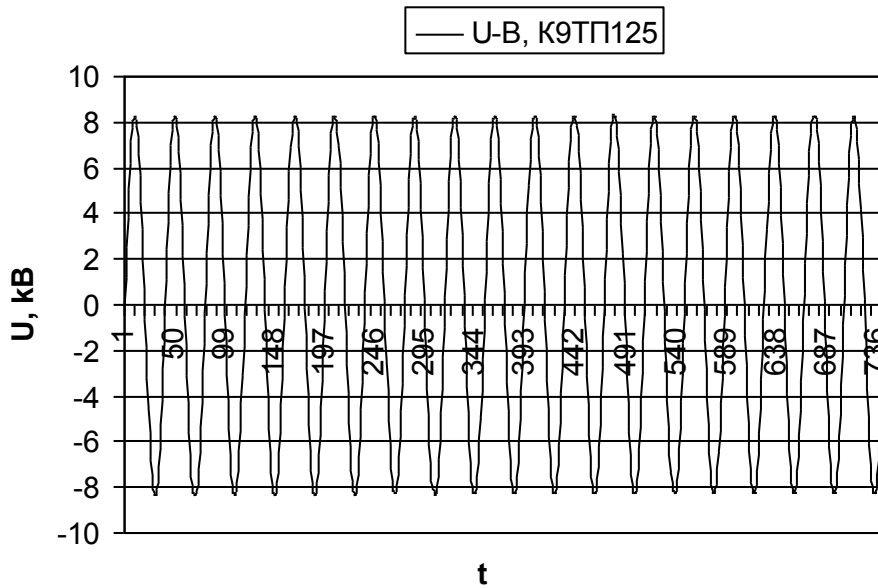


Рис. 3.7. Осцилограма сигналу після аварії на об'єкті електроенергетики.

Приведені розрахунки коефіцієнта стиснення для різних фрагментів зареєстрованого сигналу показали, що коефіцієнт стиснення залежить від кількості періодів синусоїди з однаковими параметрами, чим таких періодів більше, тим більший коефіцієнт стиснення.

Кодування бітами Галуа гарантує цілісність пакету даних і дає пікову синхронізацію синусоїди тому, що ця послідовність має особливу автокореляційну функцію.

При впливі завад типу “стирання” або вставок інформаційних бітів кодування часу розрядно-позиційним кодом є неефективним, в той час, як кодування шляхом інверсії бітів Галуа дозволяє однозначно прив'язати суттєві відліки  $Z_i$  до фактичних значень часу, незалежно від попередніх помилок.

### 3.2. Одновимірні джерела інформації

Швидко зростаючі обсяги інформації вимагають створення нових ефективних технологій зменшення надлишковості даних.

Широке застосування отримали адаптивні та неадаптивні методи стиснення даних. Адаптивні методи базуються на аналізі станів об'єктів управління і адаптивного кодування. Процедура кодування базується на визначенні активних і неактивних відліків, що можна продемонструвати на прикладі рис. 3.8.

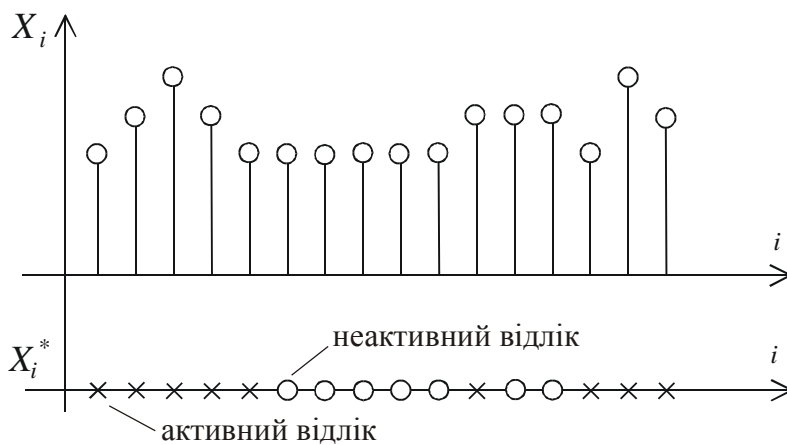


Рис. 3.8. Решітчаста функція та послідовність активних і неактивних відліків:

де  $x_i = x_i^*$ ,  $x_i \neq x_i - 1$ ,  $x_i^*$  – активний відлік,  $x_i$  – неактивний відлік.

Ефект стиснення даних досягається за рахунок кодування тільки активних відліків і їх номерів, розрядність коду визначається за формулою Хартлі [18]:

$$n = \hat{E}[\log_2 A],$$

де  $\hat{E}[\cdot]$  – цілочисельна функція;  $A$  – діапазон квантування;  $n$  – розрядність двійкового коду для представлення величини  $x_i$ ;  $0 \leq x_i \leq A$ .

В результаті такого адаптивного кодування формується наступний потік даних:  $\{i, x_i^*\}$ ,  $i = 1, 2, \dots, m$ ,

де  $m$  – число відліків.

Розрядність коду номера активного відліку дорівнює:

$$l = \hat{E}[\log_2 m].$$

Сумарна розрядність кодів  $S$ , що реєструються, визначається сумою:

$$S = n + l.$$

Метод адаптивного кодування в базисі Галуа полягає у тому, що кожному відліку у відповідність ставиться біт послідовності Галуа. Якщо відлік активний, то біт Галуа інвертується  $\bar{G}$  і після нього слідує значення контрольованого параметра, прямий біт Галуа  $G$  означає, що відлік неактивний.

Потік інформаційних даних набуде вигляду:

$$\bar{G}X^*GGG\dots\bar{G}X^*\bar{G}X^*GGG\dots G.$$

Для різних процесів число активних відліків  $f$  може змінюватись в діапазоні:

$$1 \leq f \leq m.$$

Можливі два варіанти слідування активних відліків  $f$ :

1) Активні відліки слідують один за одним на віддалі  $d \geq 1$ , де  $l$  розрядність кодону Галуа, яка рівна розрядності коду номера відліка, при  $l = 5$  послідовність має вигляд:

$$GGGGGG\bar{G}X^*GGGGGGGGGG\bar{G}X^*GGGGGGGGGG\bar{G}X^*GG.$$

2) Активні відліки, що слідують через кількість біт, що менша, ніж розрядність кодону  $d \leq l$ , тобто послідовність має вигляд:

$$GGG\bar{G}X^*GG\bar{G}X^*G\bar{G}X^*GGG\bar{G}X^*GG\bar{G}X^*GGG\bar{G}X^*\bar{G}X^*G.$$

Запропонований модифікований метод адаптивного кодування в базисі Галуа, який при появі активного відліка реєструє тільки його значення та кодон Галуа. Метод ефективно зменшує надлишковість даних при першому варіанті слідування активних відліків. У даному випадку одержуємо верхню оцінку коефіцієнта стиснення.

Початковий об'єм даних, коли кожне значення кодується максимальною кількістю біт, розраховується за формулою:

$$I_0 = m \cdot \hat{E}[\log_2 A].$$

Обчислимо об'єм даних при різних методах кодування:

1) метод однопараметричного адаптивного кодування:

$$I_1 = f \cdot \left( \hat{E}[\log_2 A] + \hat{E}[\log_2 m] \right),$$

2) метод адаптивного кодування в базисі Галуа [61]:

$$I_2 = m + f \cdot \hat{E}[\log_2 A];$$

3) модифікований метод адаптивного кодування в базисі Галуа [62]:

$$I_{3\max} = f \cdot \left( \hat{E}[\log_2 m] + \hat{E}[\log_2 A] \right) - \text{нижня оцінка},$$

$$I_{3\min} = \hat{E}[\log_2 A] + \hat{E}[\log_2 m] + (f - 1) \cdot \left( \hat{E}[\log_2 A] + 1 \right) =$$

– верхня оцінка.

$$= \hat{E}[\log_2 m] + f \cdot \hat{E}[\log_2 A] + f - 1$$

Нижня оцінка співпадає з 1-м методом, верхня дозволяє збільшити коефіцієнт стиснення в 1,6 – 2 рази. Тобто, якщо процес задовольняє умові, що активні відліки йдуть послідовно один за одним, то показники стиснення з використанням запропонованого методу є кращими, ніж методів 1 і 2.

Коефіцієнт стиснення розраховуємо відносно початкового – нульового методу, об'єм даних при якому рівний  $I_0$  (рис. 3.9):

– для першого методу:

$$K_{ct1} = \frac{m \cdot \hat{E}(\log_2 A)}{f \cdot \left( \hat{E}[\log_2 A] + \hat{E}[\log_2 m] \right)};$$

для другого методу:

$$K_{ct2} = \frac{m \cdot \hat{E}(\log_2 A)}{m + f \cdot \hat{E}[\log_2 A]};$$

для третього методу, при умові що всі активні відліки слідуєть один за одним:

$$K_{ct3} = \frac{m \cdot \hat{E}[\log_2 A]}{\hat{E}[\log_2 m] + f \cdot \hat{E}[\log_2 A] + (f - 1)}$$

Для інших випадків коефіцієнт стиснення при використанні третього методу визначається згідно аналітичного виразу:

$$K_{ct3} = \frac{m \cdot \hat{E}(\log_2 A)}{f \cdot \hat{E}[\log_2 A] + m - j}$$

де  $j$  – число бітів Галуа, які не враховуються.

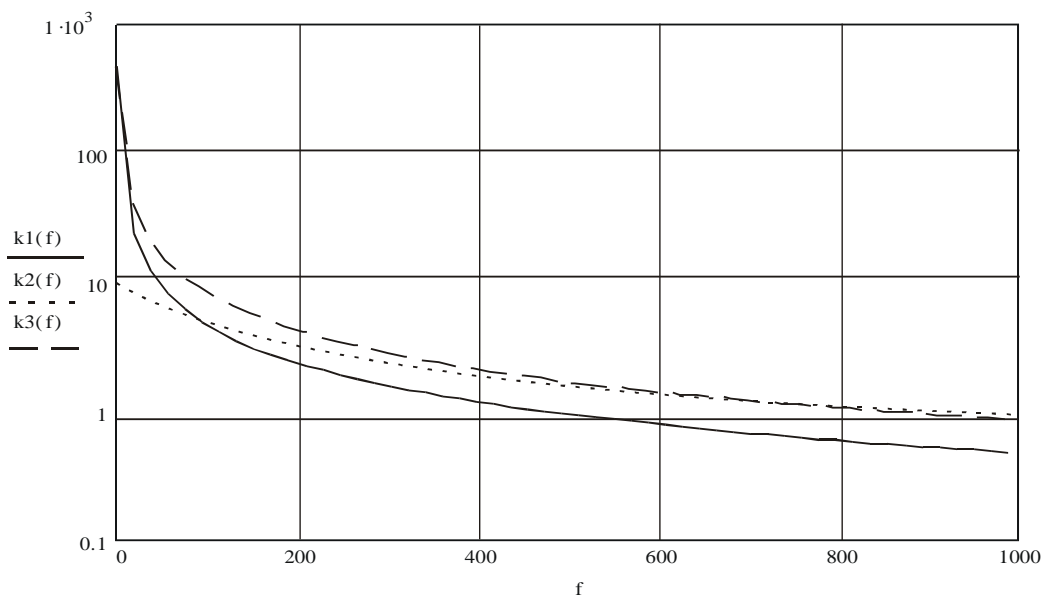


Рис. 3.9. Залежності коефіцієнта стиснення від кількості активних відліків.

Отже, запропонований метод доцільно використовувати для стиснення даних, що мають велику розрядність даних, і понад 50 % активних відліків.

Розроблене програмне забезпечення методу стиснення даних на основі адаптивного кодування в базисі Галуа приведено в додатках Б, В, Г.

За рахунок рекурентних властивостей бітів Галуа формується потік даних з блоковою синхронізацією. Це дає можливість використовувати даний метод у багатоканальних системах збору та обробки інформації.

### 3.3. Багатоканальні джерела інформації

Однією зі складних задач на рівні розподілених комп'ютерних систем (РКС) є впровадження стандартних протоколів зв'язку між давачами технологічних параметрів, контролерами і ПК [63, 64]. Високий рівень промислових завод, а також вимоги підвищеної надійності і живучості РКС ставлять високі вимоги до протоколів фізичного рівня та методів кодування.

Основним недоліком стандартних протоколів є необхідність здійснення процедури бітстафінгу за рахунок незалежності кожного біту службових даних при реалізації флагів. Другим недоліком більшості існуючих протоколів є відсутність методів виключення неактивних джерел інформації в полі даних [65, 66].

Властивості кодів Галуа, які визначають логічний зв'язок між різними елементами послідовності, відкривають нові можливості розробки методів стиснення даних та організації блокової і символної синхронізації в послідовних біт- та байт-орієнтованих потоках даних [65].

В багатоканальних системах контролю та управління на основі ВІТ (рис. 3.10) дані в каналах змінюються в різні моменти часу, що ускладнює використання стандартних комунікаційних пристроїв [67 – 69].

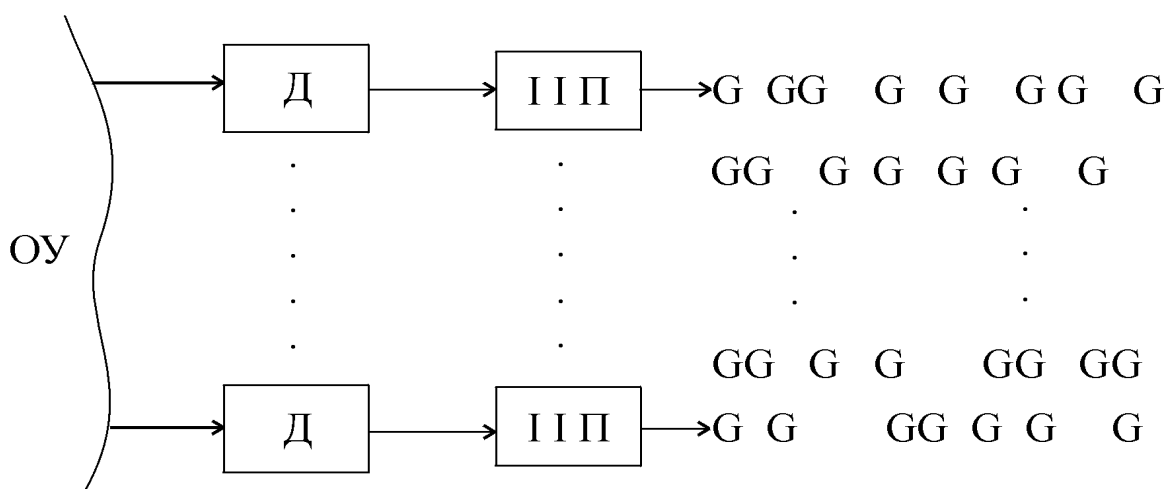


Рис. 3.10. Кодування даних в багатоканальних системах на основі вертикальної інформаційної технології: Д – давач; ІІП – інтегрально імпульсний перетворювач.

Інтегрально-імпульсний перетворювач – проміжний перетворювач, що виконує функції аналого-цифрового перетворення сигналів давача, має внутрішню пам'ять, автономне живлення і оптоелектронну гальванічну розв'язку електричних кіл.

Даний метод передбачає ефективне використання вертикальної інформаційної технології в автоматизованих розподілених системах, а також адаптивне та неадаптивне кодування інформаційних потоків.

Вирішити дану задачу можна використанням буферного запам'ятовуючого пристрою, в якому зберігаються останні  $n$  – розрядів послідовності Галуа кожного каналу і передаються в канал зв'язку в визначені моменти часу або по запиту з центральної ЕОМ, але при цьому відбувається старіння інформації.

Структуру повідомлень багатоканальної системи можна представити матрицею:

$$\begin{array}{c|cccc} N & & & & \\ 1 & X_{i,1}^* & X_{i+1,1} & \cdot & \cdot & \cdot & X_{m1}^* \\ 2 & X_{i,2}^* & X_{i+1,2} & \cdot & \cdot & \cdot & X_{m2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ j & X_{i,j}^* & X_{i+1,3} & \cdot & \cdot & \cdot & X_{mj} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ n & X_{i,n} & X_{i+1,n}^* & \cdot & \cdot & \cdot & X_{mn}^* \end{array},$$

де  $n$  – число каналів;  $m$  – число відліків;  $X_i^*$  – активний відлік;  $X_i$  – неактивний відлік.

Для зменшення надлишковості даних в багатоканальних системах, які необхідно передавати по каналах зв'язку, та формування послідовного біт-орієнтованого протоколу проводимо сканування паралельних каналів і вставляємо біт рекурентної послідовності Галуа  $G$  – якщо відлік неактивний,



або інвертований біт коду Галуа  $\bar{G}$  – якщо відлік активний.

В результаті такого кодування формується наступний потік даних:

$$CC\bar{G}X_{i1}\bar{G}X_{i2}\dots\bar{G}X_{ij}\dots GGG\dots\bar{G}X_{i+1,n}\bar{G}X_{m1}\dots G\dots\bar{G}X_{mn}CC,$$

де  $CC$  – символи блокової синхронізації.

Коефіцієнт стиснення для запропонованого методу визначається за формулою [70]:

$$k_c = \frac{n \cdot m \cdot \hat{E}[\log_2 A]}{n \cdot m + f_a \cdot \hat{E}[\log_2 A]},$$

де  $f_a$  – число активних відліків;  $n$  – номер каналу;  $m$  – число повідомлень  $n$ -го каналу.

Графічні залежності зміни коефіцієнта стиснення  $k_c$  від зміни розрядності відліків  $g_1 = 8$ ;  $g_2 = 16$ ;  $g_3 = 24$  та кількості активних відліків  $1 \leq f \leq 124$  представлено на рис. 3.11.

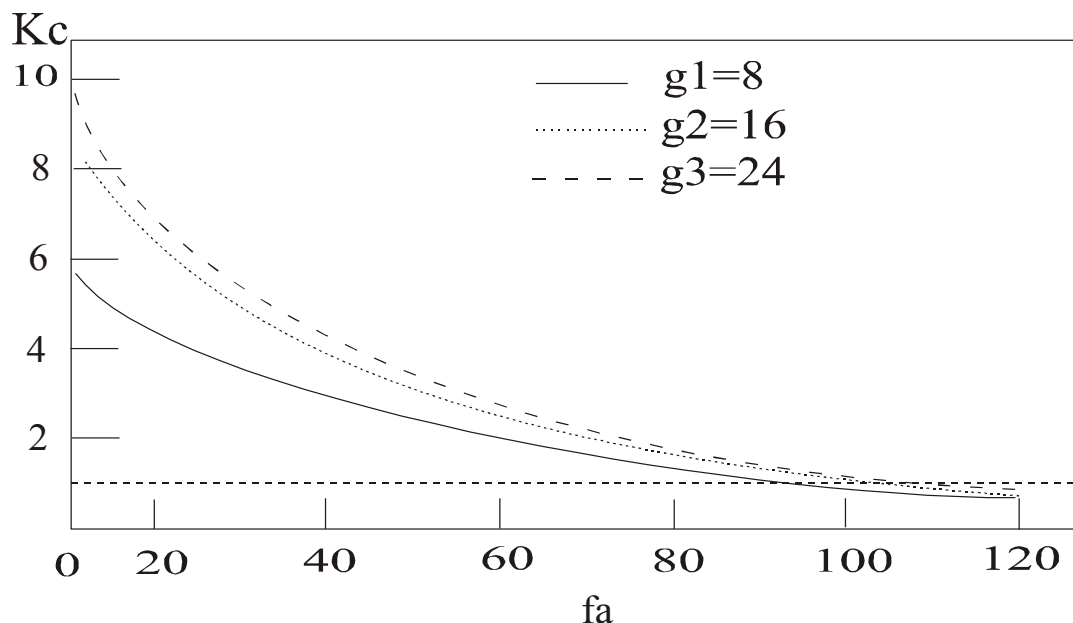


Рис. 3.11. Залежність стиснення даних від кількості активних відліків.

З графіків видно, що використання запропонованого методу є найбільш ефективним при кількості активних відліків менше 50%, тобто для об'єктів з невисокою динамікою.

Запропонований метод стиснення даних на основі кодів поля Галуа дає можливість адаптації інформаційної системи до зміни станів джерел інформації та зміни їх статистичних характеристик в часі.

### 3.4. Багатовимірні джерела інформації

Кодування багатомірних ДІ є складною теоретичною та актуальною технічною задачею [71 – 74].

В загальному вигляді задачу кодування багатомірного ДІ можна формалізувати наступним чином:

заданий вектор багатомірного простору  $N_k$  у вигляді

$$N_k = \{b_1, b_2, \dots, b_i, \dots, b_k\}, \quad (3.3)$$

де  $b_i$  – значення  $i$  – го параметру.

Існують різні методи перетворення багатовимірного простору в одновимірний на базі використання різних базисів чисел. При використанні базису Радемахера для такого перетворення вибирається система базисів:

$$\{a^0, a^1, a^2, \dots, a^n\}, \quad (3.4)$$

де  $a$  – модуль системи числення.

При використанні двійкового числення  $a_i = 2$ .

Таким чином кожне значення  $b_i$  може дорівнювати “0” або “1”.

При використанні інших систем числень, наприклад  $a = 8$ ,  $a = 10$ ,  $a = 16$  відповідно число значень  $b_i$  буде дорівнювати 8, 10, 16.

Такий спосіб кодування багатовимірного ДІ передбачає однакове число розрядів для кодування  $b_i$ , що рідко можна спостерігати на практиці.

При цьому може виникати суттєва надлишковість подання даних, якщо окремі ординати багатовимірного простору потребують

меншого числа розрядів, тобто  $b_i \ll a^i$ .

Даний недолік можна практично виключити при кодуванні станів ДІ в базисі Крестенсона.

При цьому вибирається така система взаємопростих модулів  $P_i$ , для яких виконуються умови:

$$b_i \leq P_i - 1.$$

В даному випадку, застосувавши пряме перетворення системи залишкових класів над послідовністю (3.3), отримаємо [47, 75]:

$$N_k = \text{res} \sum_{i=1}^n B_i \cdot b_i \pmod{\wp},$$

де  $\wp = \prod_{i=1}^n P_i$ ,  $0 \leq N_k \leq \wp - 1$ ,

$$B_i = \frac{\wp}{P_i} \cdot m_i \equiv 1 \pmod{P_i},$$

де  $m_i$  – задовольняють умові  $0 \leq m_i \leq P_i - 1$ .

Приклад кодування тривимірного джерела в базисі Крестенсона показаний на рис. 3.12.

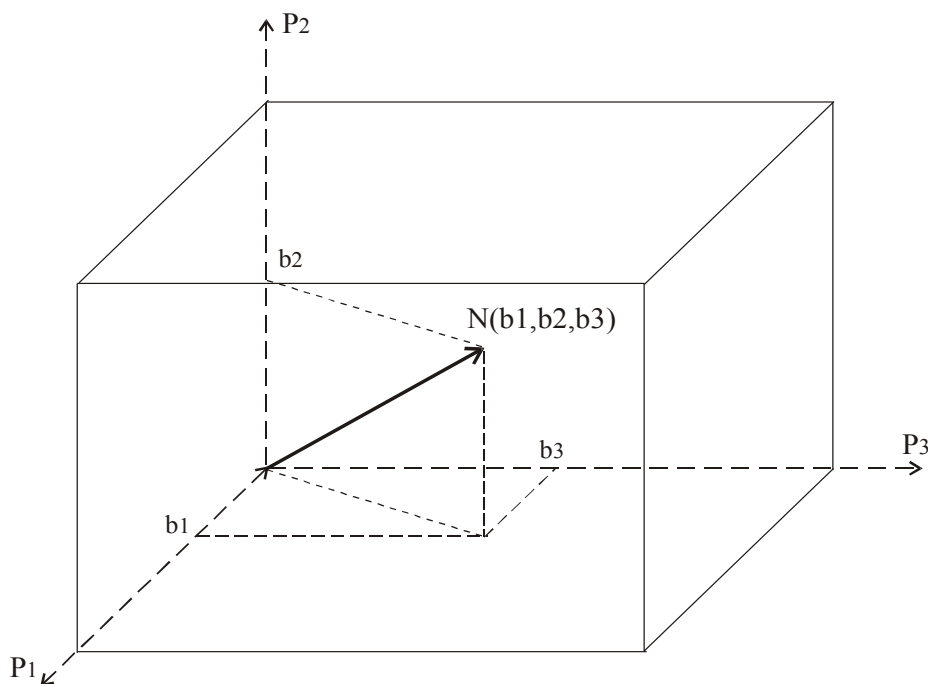


Рис. 3.12. Кодування тривимірного джерела інформації в базисі Крестенсона.

Даний метод кодування, на відміну від декартових координат, дозволяє виключити надлишковість кодування в тривимірному просторі, яка виникає при виконанні умови (3.4).

Структурна схема процесора обробки даних на основі перетворення Крестенсона – Галуа представлена на рис. 3.13 [76].

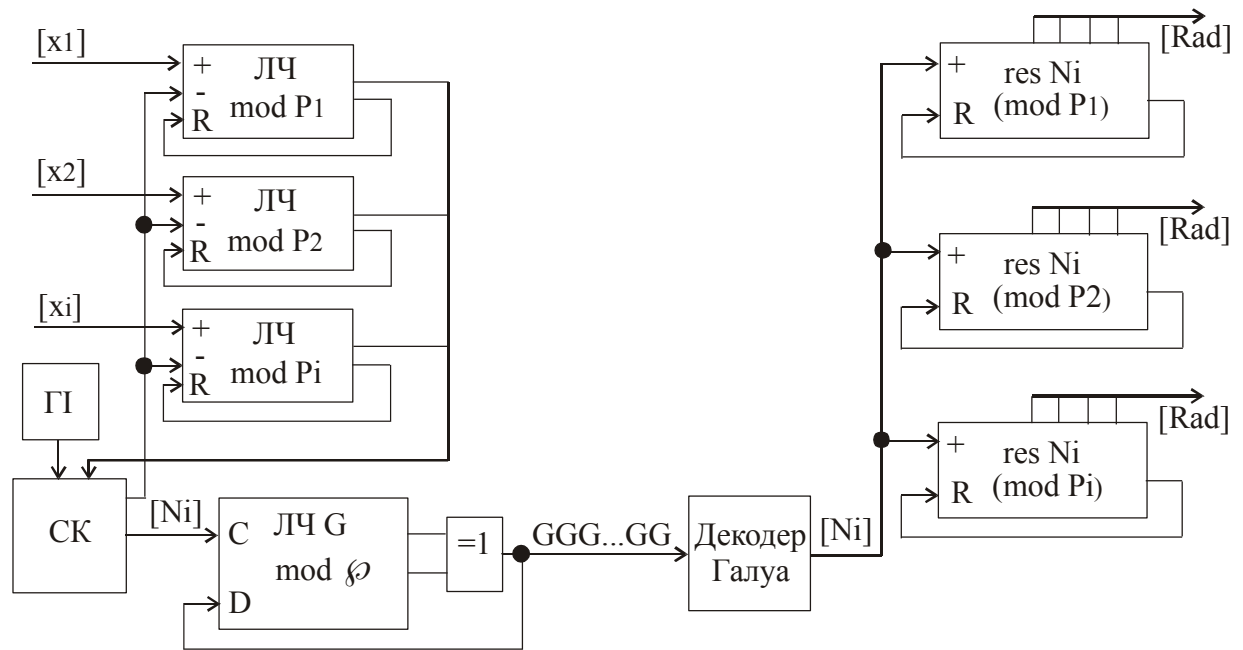


Рис. 3.13. Структурна схема системи кодування та передавання даних: ЛЧ mod  $P_i$  – лічильник по модулю  $P_i$ ; ЛЧ G – лічильник Галуа; ГІ – генератор тактових імпульсів; СК – схема керування.

Виміряні значення технологічних параметрів  $x_i$  поступають на вхід сумування лічильників. Після завершення циклу вимірювання на входи віднімання вказаних лічильників поступають імпульси з генератора тактових імпульсів через схему керування. Віднімання імпульсів відбувається до тих пір, поки всі лічильники одночасно не встановляться в "0". Паралельно така сама кількість імпульсів поступає на генератор Галуа по модулю  $\phi$ . На приймальній стороні відбувається зворотна операція одержання значень  $x_i$ . Використання прямого перетворення системи залишкових класів дозволяє перейти від багатовимірного до одновимірного подання значень технологічних параметрів на передавальній стороні та зворотного

перетворення на приймальній стороні, і, відповідно, спростити реалізацію протоколів передавання даних та захистити дані від несанкціонованого доступу.

Використання взаємопростих модулів розмірності  $P_i = 2^n - 1$  дозволяє передавати залишки послідовностями Галуа (рис. 3.13).

Коефіцієнт зменшення надлишковості даних розраховують за формулою:

$$k_c = \frac{\sum_{i=1}^k \hat{E}[\log_2 b_{i \max}]}{\hat{E} \left[ \log_2 \left( \prod_{i=1}^n P_i - 1 \right) \right]}.$$

Важливою задачею є кодування багатомірних джерел інформації, які представляють багатовимірні вектори в вузлах двовимірної площини (рис.3.14).

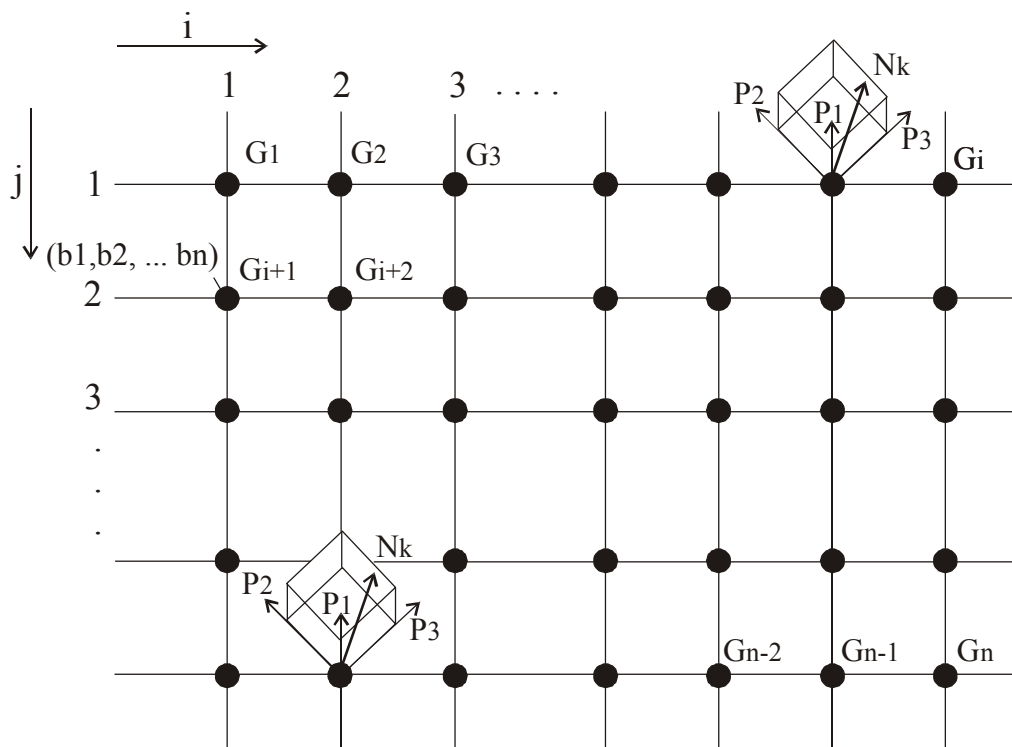


Рис. 3.14. Модель джерела інформації.

До таких задач належать задачі картографії, екологічного моніторингу, розподілу характеристик в перерізах нафтогазопроводів, метеорологічні

задачі.

В результаті отримуємо двовимірну площину з векторами  $N_{ijk}$ .

Недоліком такого методу кодування є необхідність доповнення значення  $N_k$  коефіцієнтами  $i$  та  $j$ .

Задача підвищення ефективності кодування таких ДІ може бути вирішена двома шляхами:

- а) розширенням системи модулів для внутрішнього кодування  $i$  та  $j$ ;
- б) кодуванням цих координат в базисі Галуа.

В першому випадку пряме перетворення залишкових класів в розширеній системі модулів має вигляд [75]:

$$N_{kij} = \text{res} \left( \sum_{i=1}^{k+2} b_i \cdot \beta_i \pmod{\wp^*} \right),$$

де  $\wp^*$  – розширений модуль, який включає систему модулів

$$P_1, P_2, \dots, P_i, \dots, P_k, P_{k+1}, P_{k+2},$$

$$\text{де } \begin{cases} 0 < P_{k+1} \leq \alpha - 1 \\ 0 < P_{k+2} \leq \beta - 1 \end{cases}, \quad i = 1, 2, \dots, \alpha, \quad j = 1, 2, \dots, \beta.$$

Коефіцієнт стиснення для даного методу розраховується за формулою:

$$K_{c1} = \frac{\sum_{i=1}^k \hat{E}[\log_2 b_{i\max}] + \hat{E}[\log_2 \alpha] + \hat{E}[\log_2 \beta]}{\hat{E}[\log_2 (\wp^* - 1)]}. \quad (3.5)$$

При застосуванні кодів Галуа вектор  $N_k$  доповнюється одним бітом Галуа, як показано в [62], і двовимірне джерело представляється у вигляді одновимірного масиву даних

$$\{G_l, N_k\},$$

де  $G_l$  –рекурентна послідовність бітів Галуа,

$$0 \leq G_l \leq \alpha \cdot \beta.$$

Коефіцієнт стиснення при цьому визначається за формулою

$$K_{c2} = \frac{\sum_{i=1}^k \hat{E}[\log_2 b_{i\max}] + \hat{E}[\log_2 \alpha] + \hat{E}[\log_2 \beta]}{1 + \hat{E}\left[\log_2 \left(\prod_{i=1}^n P_i - 1\right)\right]}. \quad (3.6)$$

Графічна залежність коефіцієнта стиснення на основі залежностей (3.5) і (3.6) при  $P1=17$ ,  $P2=53$ ,  $P3=153$ ,  $\alpha=10$ ,  $\beta=12$  подана на рис. 3.15.

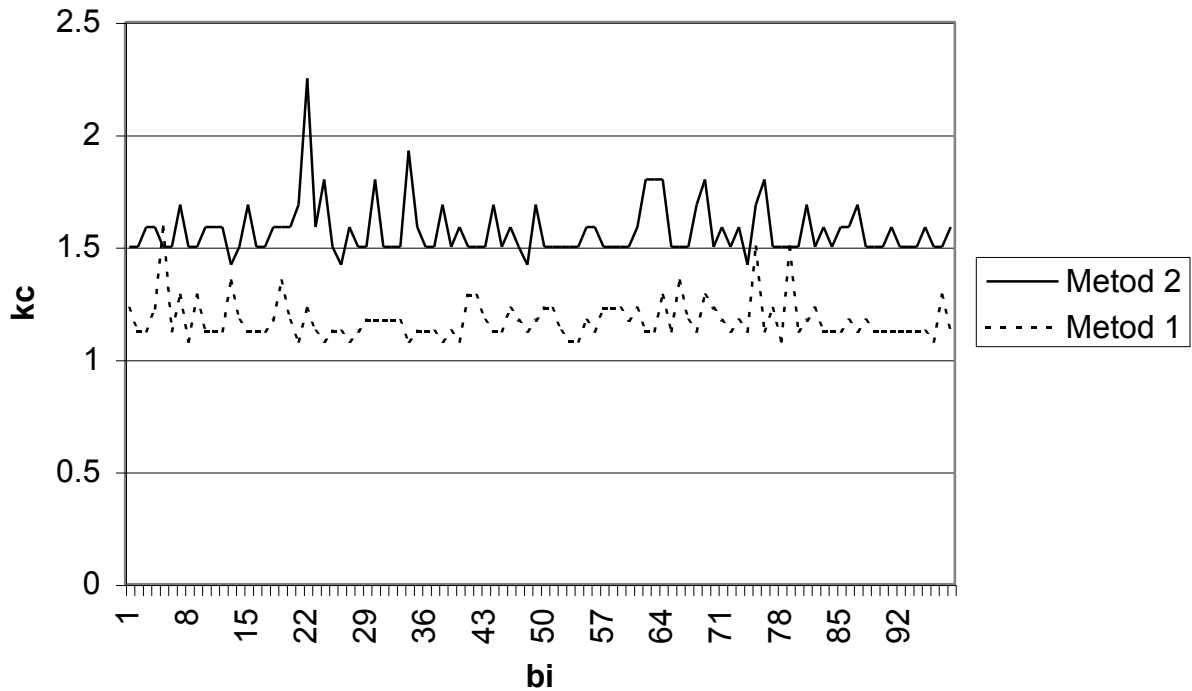


Рис. 3.15. Залежність коефіцієнта стиснення від значення вимірних параметрів.

Запропонований метод дозволяє перейти від багатовимірному представлення параметрів контрольованого об'єкту до одновимірному, що в свою чергу, суттєво спрощує протоколи передавання даних, захищає дані від несанкціонованого доступу та зменшує об'єми інформаційних та службових даних.

### 3.5. Дослідження методу інтегрально-імпульсного кодування у базисі Галуа

Методи інтегрально-імпульсного кодування (ІК) даних базуються на використанні вертикальної інформаційної технології в базисі Галуа [45, 61, 67, 77].

Відмінність вертикальної інформаційної технології (G – технології), реалізованої в базисі Галуа, від загальноприйнятої, що ґрунтується на використанні базису Радемахера (R – технологія) полягає не в паралельному, а в біт-орієнтованому представленні інкрементних кодів.

Наприклад, послідовність двійкових кодів:

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
·	·	·	·
·	·	·	·
·	·	·	·
1	1	1	1

базису Радемахера замінюється рекурентним біт - орієнтованим кодом базису Галуа:

$$[0000]101001101111.$$

При цьому досягається суттєве зменшення надлишковості інкрементних кодів за рахунок представлення  $i + 1$ -го коду зсунутим на один біт вправо кодом Галуа, який виділений дужками.

В [44] запропоновано використання ВІТ при розробці автоматизованої системи контролю і обліку енергоносіїв промислових підприємств.

Алгоритм кодування даних на основі ВІТ в системах обліку енергоносіїв наступний:



1) дискретизація по часу і квантування по рівню наданого сигналу (рис. 3.16);

2) інтегрування отриманого цифрового сигналу. Якщо вихідні сигнали давачів є знакозмінними, то перед інтегруванням необхідно додати постійну складову, яка дорівнює максимальному від'ємному значенню сигналу, тобто підняти сигнал в область додатних значень;

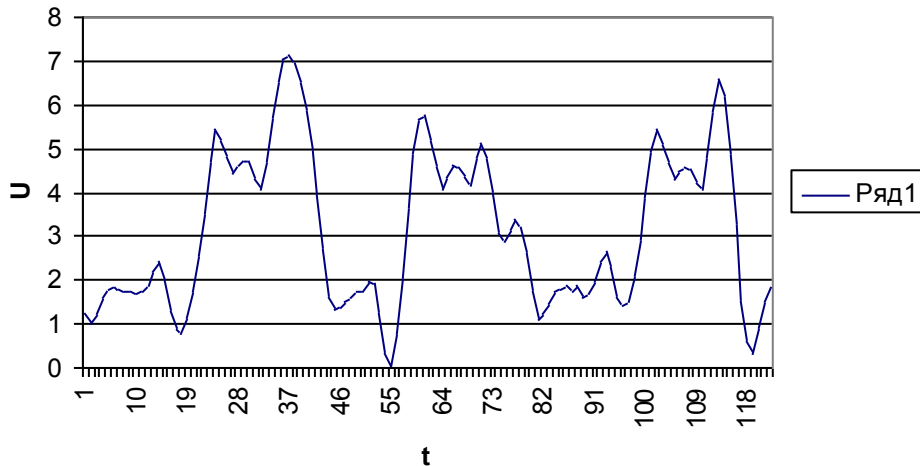


Рис. 3.16. Графічне представлення виміряного технологічного параметру.

3) розміщення вздовж осі ординат біт-орієнтованої послідовності Галуа;

4) при досягненні інтегрального значення параметру нового рівня амплітуди відбувається генерування та передавання в канал зв'язку нового біту Галуа (рис. 3.17);

5) отримаємо частотно – маніпульований сигнал, який залежить від крутизни інтегралу параметру.

В системах обліку електроенергії приведений алгоритм можна спростити, пропустивши крок 1 і 2, так як сигнал на виході імпульсних лічильників є інтегрованим.

При ДМ в канал зв'язку передається не абсолютне значення сигналу, а різниця між вихідним аналоговим сигналом і апроксимуючою напругою (сигнал помилки).

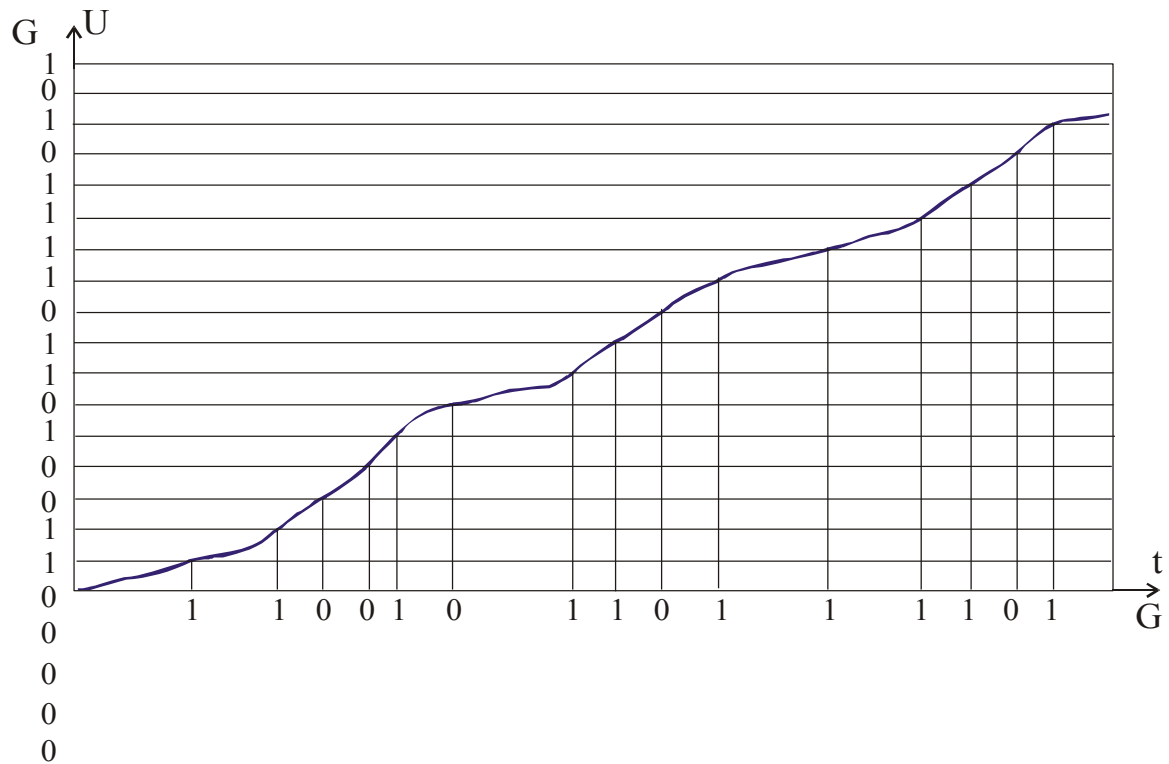


Рис. 3.17. Інтегральне представлення сигналу.

Суттєвим недоліком методу дельта-модуляції є спотворення інформації, як це показано на рис. 3.18, при наявності різких змін вхідної послідовності відліків  $X_i$ , а також втрата абсолютного значення сигналу при втраті або спотворенні одного біту.

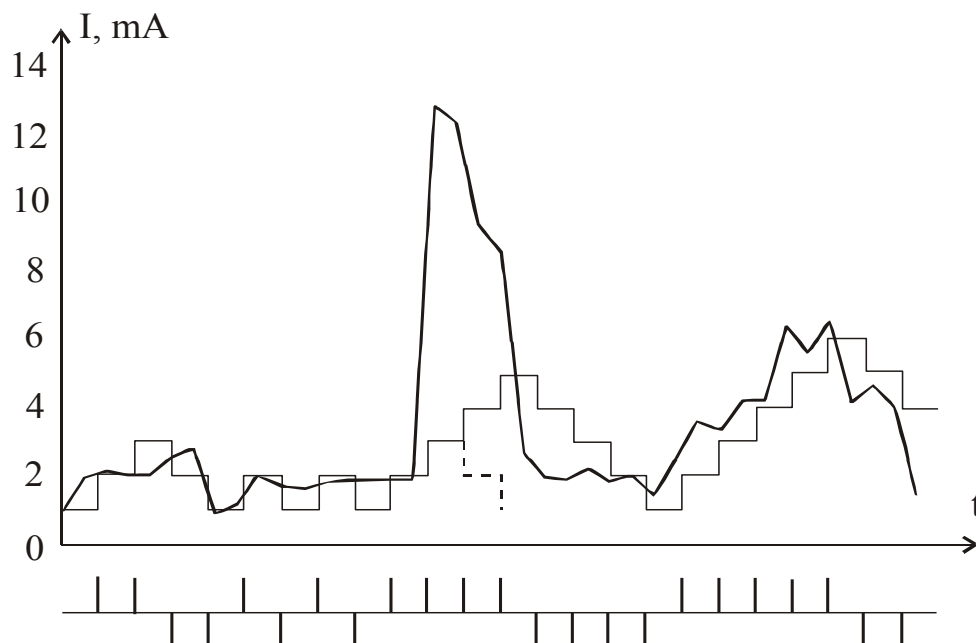
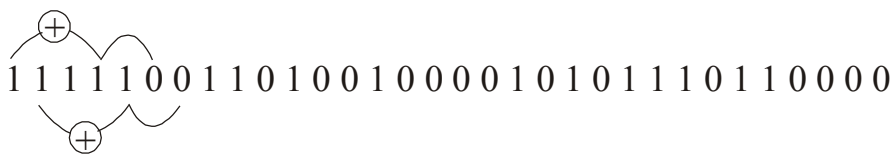


Рис. 3.18. Дельта-модульований сигнал.

Кодування інформації в базисі Галуа на основі інтегрально-імпульсної технології дозволяє в значній мірі покращити правильне відновлення інформації при наявності окремих різких стрибків  $X_i$  за рахунок можливості однозначного декодування дійсних значень  $X_i$  за допомогою  $n$  – розрядного кодону Галуа.

Можливість однозначного декодування даних при наявності стрибків характеризує модель послідовності  $X_i$  (рис. 3.19). Для кодування поданого на рис. 3.19 сигналу використовуємо 32– бітну послідовність Галуа:



При наявності стрибків сигналу (рис. 3.18) інтеграл також різко зростає (рис. 3.19), відповідно  $\Delta t \rightarrow 0$ .

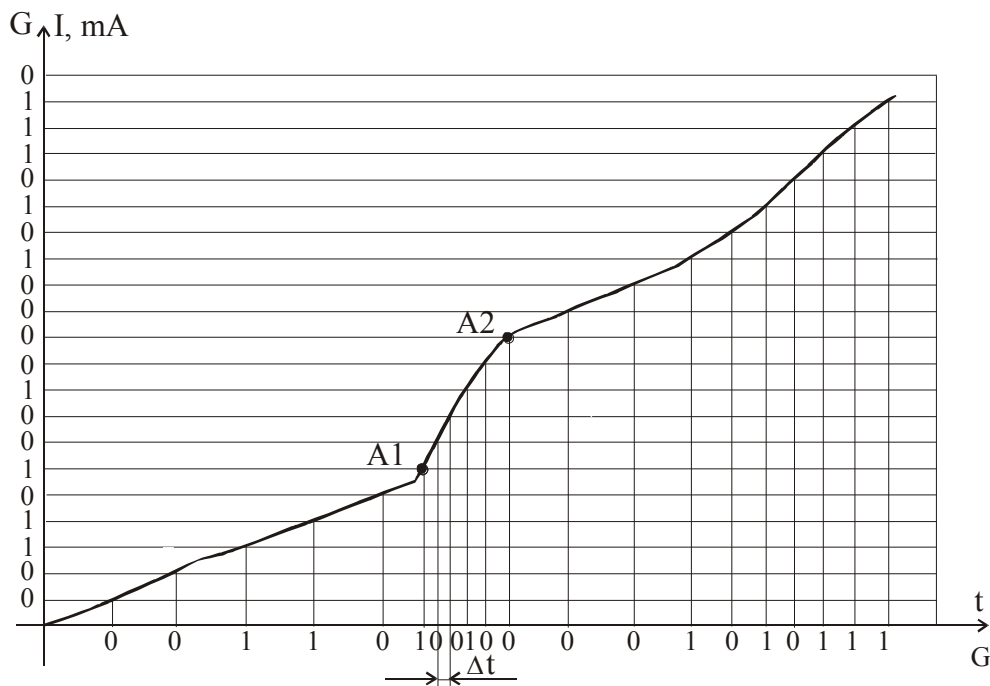


Рис. 3.19. Інтегральне представлення сигналу.

Для визначення абсолютного значення сигналу (при втраті декількох бітів, область  $A1 - A2$ ) (рис. 3.19) достатньо декодувати правильно прийняті наступні  $n_g$  біт, де  $n_g$  – розрядність кодону.

Миттєве значення контрольованого параметру визначається шляхом вимірювання часу між сусідніми бітами Галуа.

Дана технологія кодування дозволяє однозначно відновити абсолютне значення сигналу при впливі завад та пошкодженнях лінії зв'язку. Втрата даних може відбутися, якщо  $C > F_{зд}$ , де  $C = 1/\Delta t$  – швидкість створення повідомлень;  $F_{зд}$  – пропускна здатність лінії зв'язку.

## ВИСНОВКИ

1. Запропонований та досліджений метод стиснення гармонічних сигналів, який дозволяє закодувати період гармонічного сигналу по  $1/8$  частині і однозначно його відновити, а отже, зменшити надлишковість представлення гармонічних сигналів  $k_c = 7,6$  раз на період.

2. Розроблено адаптивний метод стиснення даних на основі кодів поля Галуа. Коефіцієнт стиснення запропонованого методу дорівнює  $k_c = 1,5-4$  в залежності від кількості активних відліків. За рахунок рекурентних властивостей бітів Галуа формується потік даних з блоковою синхронізацією, що дає можливість підвищити заводо захищеність закодованих даних.

3. Запропонований метод кодування багатовимірних ДІ на основі системи залишкових класів, який дозволяє перейти від багатовимірного до одновимірного представлення значень технологічних параметрів на передавальній стороні та зворотного перетворення на приймальній стороні. Проведені дослідження ефективності кодування номера ДІ в базисі Радемахера, Крестенсона, Галуа показали, що найменшу кількість біт для представлення номеру ДІ необхідно в базисі Галуа (1 біт) при послідовному опитуванні ДІ. Використання перетворення системи залишкових класів дозволяє захистити дані від несанкціонованого доступу. Виведені формули для розрахунку коефіцієнта стиснення запропонованого методу кодування даних.

4. Проведені дослідження методу інтегрально-імпульсного кодування в базисі Галуа показали його переваги над використанням методів дельта-модуляції та диференціальної ІКМ. Інтегрально-імпульсний метод кодування несе відомості про абсолютне значення виміряного параметру, а також дозволяє відновити абсолютне значення при впливі завад та відмовах каналу зв'язку.

## РОЗДІЛ 4

### РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ЗМЕНШЕННЯ НАДЛИШКОВОСТІ ІНФОРМАЦІЙНИХ ПОТОКІВ

#### 4.1. Інтегрально-імпульсний перетворювач з розширеними функціональними параметрами для систем обліку енергоносіїв

Інтегрально-імпульсний перетворювач (ІП) призначений для обробки та передавання даних про стан технологічних об'єктів. ІП був впроваджений в системі обліку енергоносіїв Alfa [78, 79].

ІП складається з перетворювача напруга – частота (ПНЧ), масштабуючого подільника частоти (МД), кодер–інтегратора Галуа (КІГ) і широтного модулятора (ШІМ) (рис. 4.1).

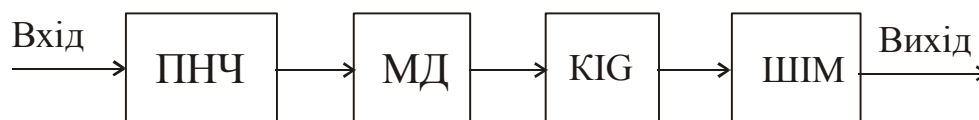


Рис. 4.1. Структурна схема ІП.

Досвід проектування та експлуатації вказаного типу ІП показав ряд функціональних та конструктивних недоліків:

- 1) функціональні недоліки:
  - немає можливості вибору ключів для формування кодової послідовності;
  - відсутній імпульсний вхід;
  - відсутня можливість зміни тривалості імпульсів 0 і 1 на виході ІП;
  - складний блок живлення за рахунок використання кристалу ПНЧ з двополярним живленням;
- 2) конструктивні недоліки:

- відсутній контроль напруги зовнішнього акумулятора;
- клемне підключення зовнішніх входів.

Враховуючи вказані недоліки, розроблено ІІП з розширеними функціональними можливостями та новій елементній базі (рис. 4.2).

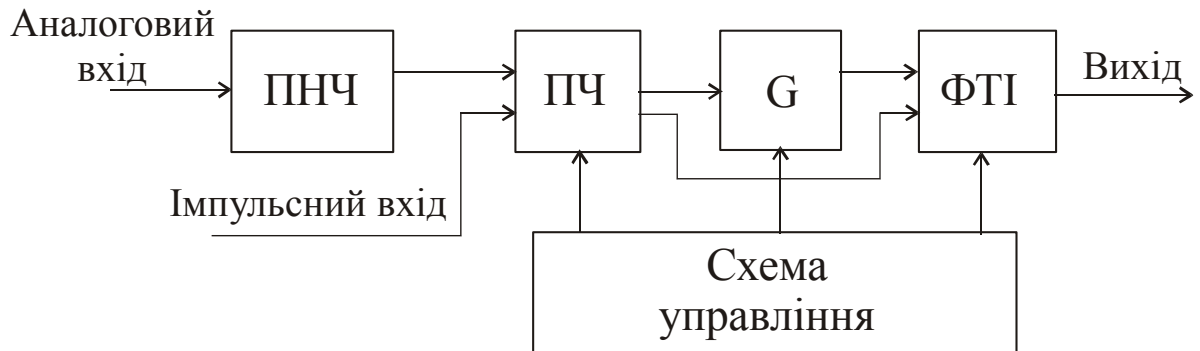


Рис. 4.2. Структурна схема вдосконаленого ІІП:

ПЧ – подільник частоти;

G – генератор коду Галуа;

ФТІ – формувач тривалості імпульсів.

Розроблений ІІП забезпечує наступні функціональні параметри (рис.4.3):

- зміну вхідної напруги 10 мкВ – 100мВ, 3 мВ – 30 В;
- зміну коефіцієнта ділення вхідної частоти:  $2^n$  (де  $n = \overline{1, 8}$ );
- вибір ключів  $2^{20} - 2^{30}$ ;
- тривалість імпульсів передавання (0,01 мс – 200 мс);
- акумулятор з контролем напруги;
- аналоговий та імпульсний вхід;
- однополярний ПНЧ.

ПНЧ призначений для перетворення вхідної напруги в частоту вихідних імпульсів, які можуть передаватися на великі відстані без спотворення інформаційного параметру – частоти. ПНЧ належить до класу інтегруючих перетворювачів, тому має ряд переваг: високу точність при мінімальній кількості необхідних прецизійних елементів, висока

завадостійкість, мала чутливість до зміни напруги живлення, відсутня диференціальна нелінійність, низька собівартість.

$f(V_{in}, C, k)$

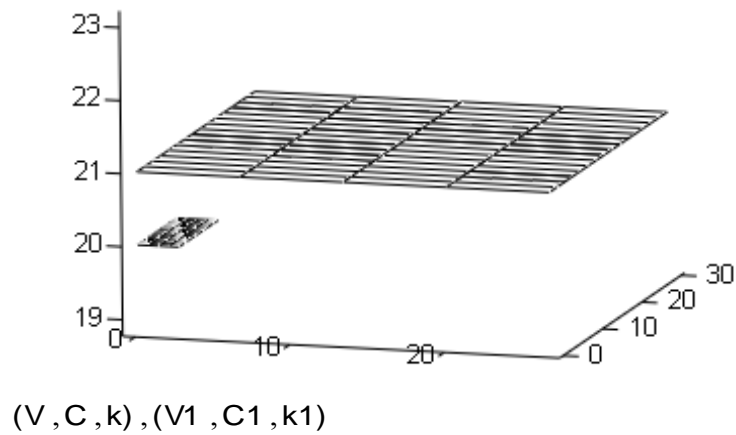


Рис. 4.3. Порівняння характеристик існуючого і вдосконаленого ІПП:

$V_{in}$  – діапазон зміни вхідної напруги;

$C$  – вибір тривалості вихідних імпульсів;

$k$  – вибір ключів.

При розробці ІПП використано ПНЧ – AD654 фірми Analog Devices, який має наступні характеристики: струм споживання 2 мА, високий вхідний опір (250 МОм), мале зміщення (1 мВ) і дрейф нуля (4 мкВ/°С), температурний дрейф коефіцієнта перетворення ( $50 \cdot 10^{-6}/^{\circ}\text{C}$ ), однополярне джерело живлення. Для роботи мікросхеми необхідні два зовнішні елементи: резистор  $R_t$  і конденсатор  $C_t$  для того, щоб задати характеристики перетворення.

За допомогою резистора  $R_t$  діапазон вхідної напруги можна встановлювати від 10 мкВ – 100 мВ до 3 мВ – 30 В.

Частота імпульсів на виході ПНЧ визначається за формулою

$$f = \frac{V_{in}}{10 \cdot R_t \cdot C_t},$$

де  $V_{in}$  – вхідна напруга.



Максимальна частота перетворювача до 500 кГц при динамічному діапазоні 80 дБ.

Основні блоки ІПП реалізовані на програмованій логічній інтегральній схемі (ПЛІС) фірми ALTERA EPM3064ALC44 (рис. 4.4) [80]:

- 8-розрядний двійковий лічильник count\_8;
- мультиплексор mul\_8\_1;
- генератора коду Галуа g\_generat;
- формувач тривалості імпульсів – fti.

Вимірний сигнал у вигляді частоти імпульсів з ПНЧ поступає на вхід  $F_{in}$  двійкового лічильника. Коефіцієнт ділення вхідної частоти на  $2^n$ ,  $n = \overline{1, 8}$ , задається адресними входами (A0, A1, A2) мультиплексора mul\_8\_1.

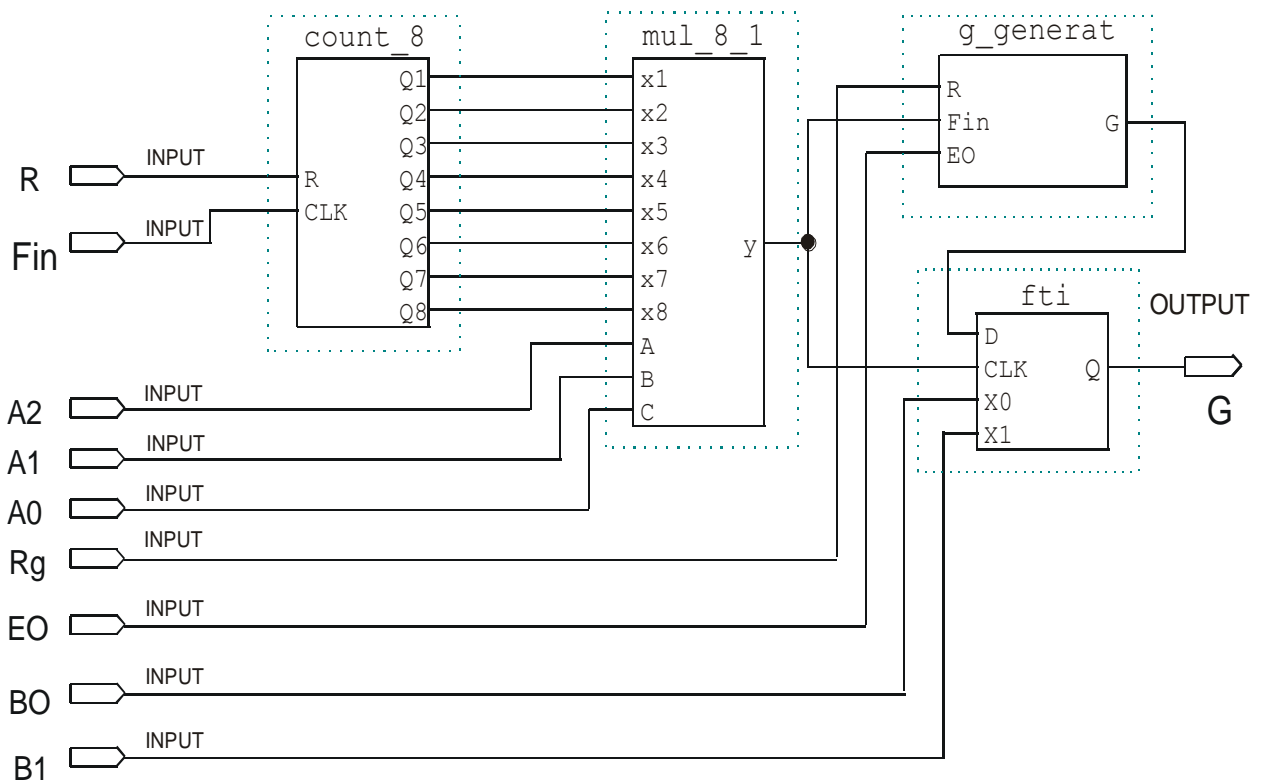


Рис. 4.4. Функціональна схема ІПП.

З мультиплексора сигнал поступає на генератор послідовності Галуа g\_generat і на тактовий вхід формувача тривалості імпульсів fti. Вибір тривалості вихідних імпульсів передавання здійснюється за допомогою

входів ( $B0$  і  $B1$ ) відповідно до табл. 4.1. Процедура генерування кода Галуа написана на AHDL у виді параметризованої макрофункції, яка описує пристрій, спрощена структурна схема якого приведена на рис. 4.5 [81].

Таблиця 4.1

Тривалість імпульсів передавання

B1	B0	Тривалість одиниці, мс	Тривалість нуля, мс
0	0	0,01	0,02
0	1	0,1	0,2
1	0	1	2
1	1	10	20

Параметрами макрофункції є довжина характеристичного многочлена і число, що описує початковий стан тригерів. Текстовий опис макрофункції G\_GENERAT приведений в додатку E. Результати моделювання роботи ІІП представлені на рис. 4.6.

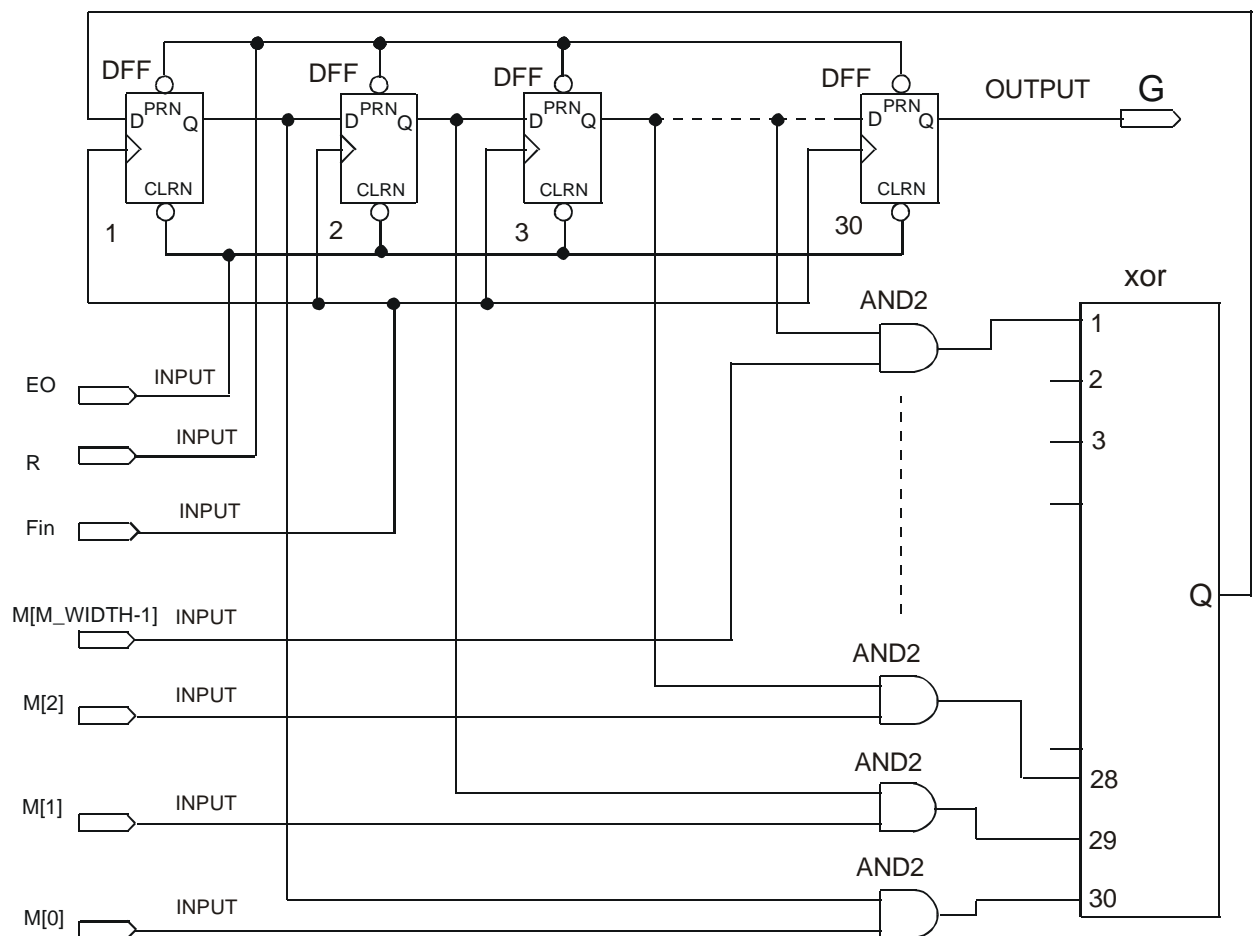


Рис. 4.5. Структурна схема блоку G\_GENERAT.

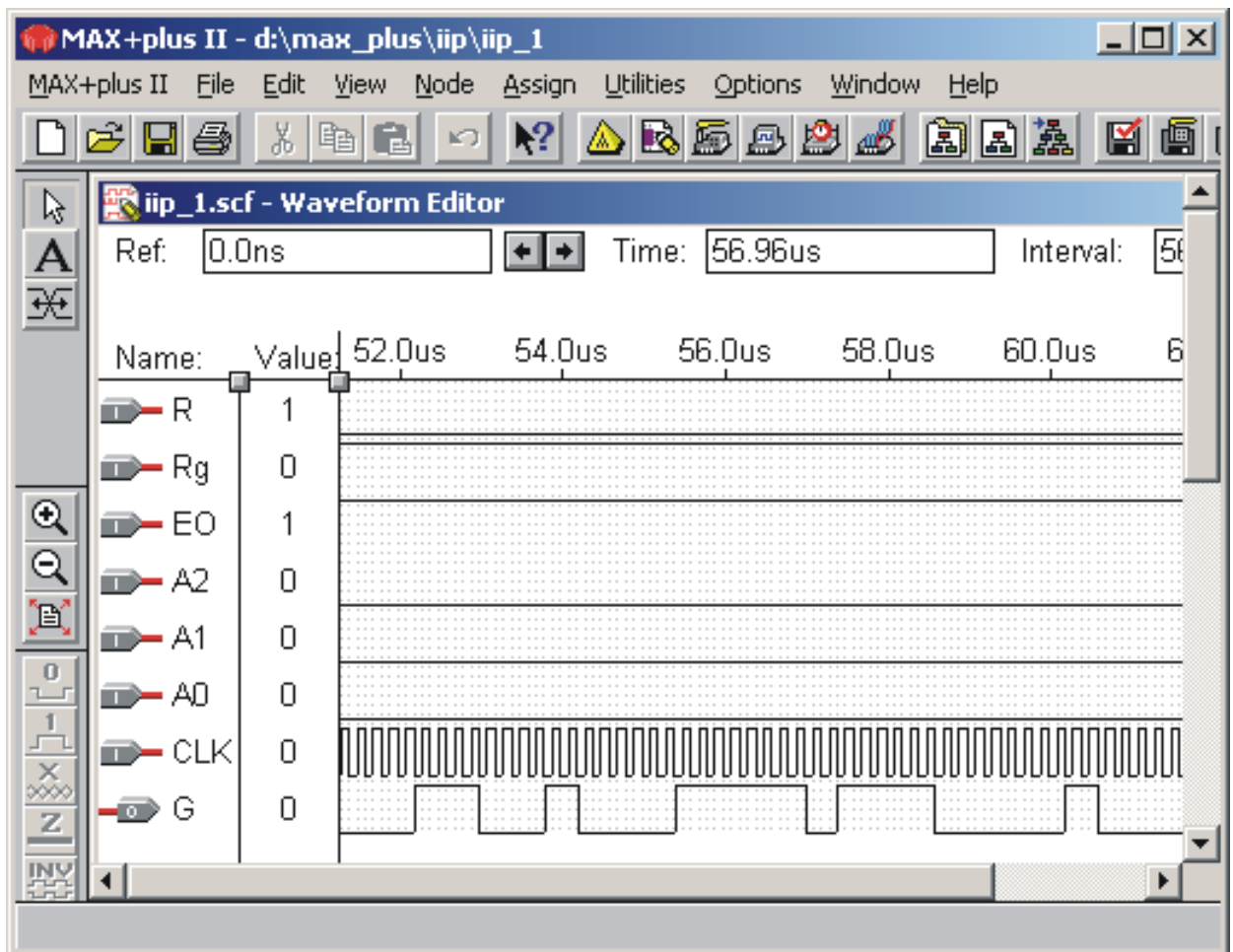


Рис. 4.6. Результати моделювання роботи ІІП.

Розробку ІІП виконано з використанням САПР MAX+PLUS II, яке дозволяє на рівні часових діаграм оцінити поведінку пристрою до програмування його в ПЛІС (рис.4.6). Часові діаграми відображають як логіку функціонування, так і реальні часові відношення сигналів. Моделювання здійснюється з високим ступенем адекватності, що значно спрощує процес відлагодження пристрою.

Цифрові блоки ІІП реалізовані на ПЛІС серії MAX3000A. Мікросхема EPM3064A має матричну архітектуру і виготовлена по технології EEPROM.

Мікросхеми серії MAX3000A мають наступні характеристики:

- напруга живлення ядра 3,3 В;
- сумісність по входах/виходах з рівнями 5,0В; 3,3В; 2,5В;
- мінімальна затримка поширення сигналу від входу до виходу – 4,5 нс;
- можливість гарячого включення;

- сумісність зі стандартами PCI;
- програмований біт секретності;
- програмований “turbo-bit” для кожної макроячейки, який дозволяє зменшити споживання до 50 %;
- максимальна частота лічильника 192 МГц.

Мікросхеми серії MAX3000A програмуються в системі через стандартний чотириконтактний JTAG інтерфейс. Програмне забезпечення MAX+PLUS II створює конфігураційну послідовність, яка завантажується в ПЛІС з допомогою спеціалізованого завантажувального кабелю Byte Blaster MV (рис. 4.7) [82].

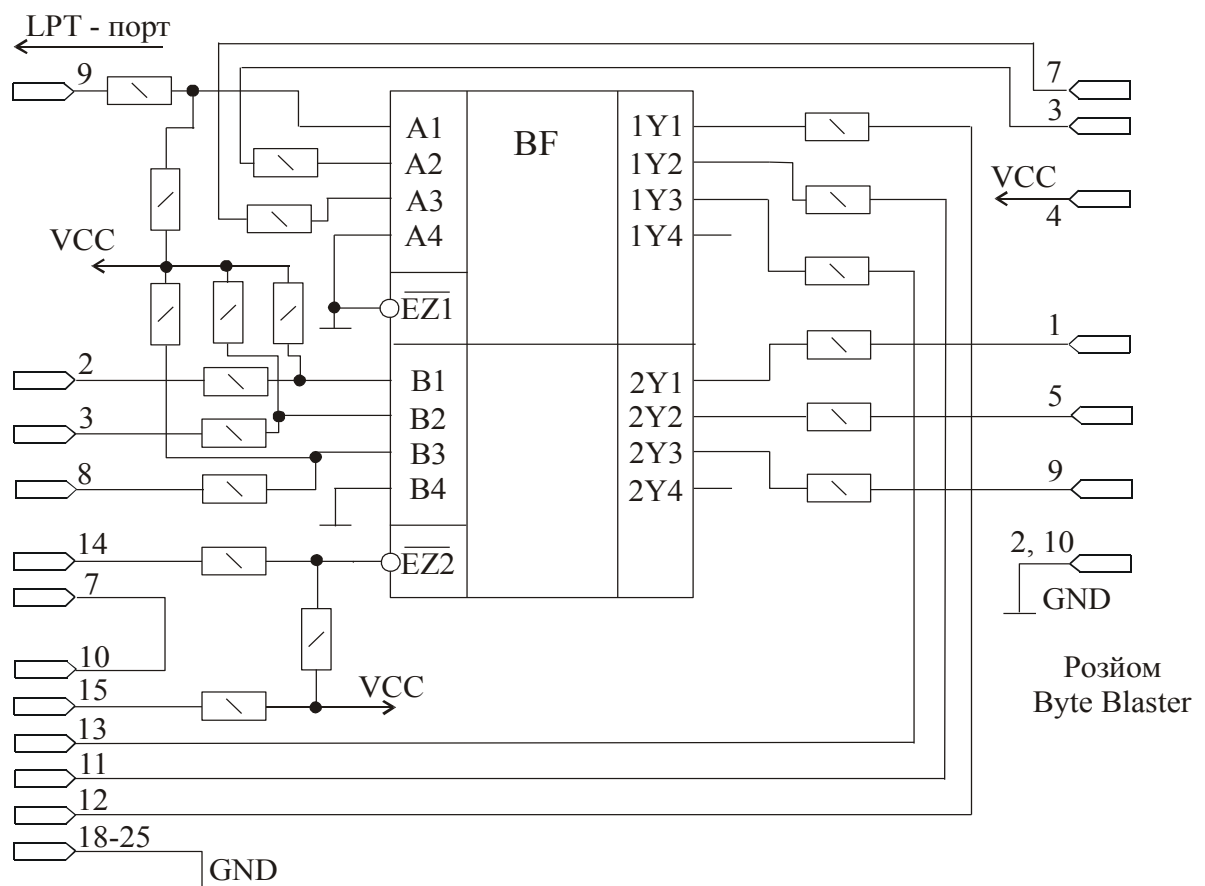


Рис. 4.7. Принципова схема завантажувального кабелю Byte Blaster MV.

Призначення контактів розніму Byte Blaster MV в режимі програмування приведено в табл. 4.2 [82].

Живлення завантажувального кабелю Byte Blaster MV здійснюється від джерела живлення ІІП.

## Призначення контактів розніму Byte Blaster MV

Контакт розніму Byte Blaster	Режим програмування через порт JTAG	
	Сигнал	Призначення
1	TCK	Тактовий сигнал
2	GND	Сигнальна земля
3	TDO	Дані з ПЛІС
4	VCC	Напруга живлення
5	TMS	Контроль автомату JTAG
6	-	Непідключений
7	-	Непідключений
8	-	Непідключений
9	TDI	Дані в ПЛІС
10	GND	Сигнальна земля

Розробка ІІП на базі ПЛІС (рис. 4.8) дозволила підвищити надійність пристрою, суттєво прискорити час проектування та виготовлення, а отже, знизити вартість пристрою.

Використання ІІП з розширеними функціональними параметрами дозволить значно розширити сферу їх застосування в розподілених комп'ютерних системах нафтогазової, енергетичної та інших галузях промисловості.

#### **4.2. Спецпроцесор обробки даних на основі перетворення Крестенсона-Галуа**

Однією із актуальних задач при розробці комп'ютерних систем контролю та управління розподіленими технологічними об'єктами є ефективне використання каналів зв'язку. Вирішити вказану задачу можна використовуючи методи розділення каналів зв'язку.

Основну задачу, при розділенні каналів зв'язку, виконують групові модулятори (ГМ), завданням яких є виключення конфліктів між абонентами при використанні колективного каналу зв'язку та забезпечення максимального завантаження каналу зв'язку. Якщо кожний абонент характеризується певною швидкістю передачі даних  $C_i$ , то, як правило

$$C_{\text{к.з.}} = \sum_{n=1}^i C_i.$$

Суть розділення каналу зв'язку полягає в реалізації можливості одночасного обміну інформацією багатьма абонентами. При цьому, задача виділення сигналів кожного абонента еквівалентна виділенню сигналів на фоні завад. Розрізняють два класи методів розділення каналів зв'язку:

- лінійне розділення;
- нелінійне розділення каналів зв'язку.

Методи лінійного розділення каналів зв'язку поділяються на: амплітудне розділення; частотне розділення; фазове розділення; часове

розділення; розподілення по формі сигналу.

У даний час найбільшого застосування набули методи лінійного розділення каналів зв'язку, однак їх використання має ряд недоліків, зокрема, ускладнює виключення неактивних джерел інформації, зростають об'єми службових даних, відбувається старіння інформації при великій кількості вхідних каналів.

Використання теоретико-числових базисів відкриває нові можливості розробки ефективних методів нелінійного розділення каналів зв'язку та розвитку сфери їх застосування в РКС.

Процесор кодування даних розроблений на базі структури (див. рис. 3.11) і призначений для кодування та передавання даних багатовимірних джерел інформації розподілених комп'ютерних систем контролю та керування технологічними процесами [76].

Перетворення багатовимірної вектора в одновимірний відбувається за наступним алгоритмом: унітарний код  $N_i$ , що відповідає значенню технологічного параметру поступає на вхід додавання UP лічильників Counter\_P1, Counter\_P2, Counter\_P3, які працюють по модулях  $P_1, P_2, \dots, P_n$  (рис. 4.9).

Після закінчення циклу вимірювання, на вхід віднімання DN вказаних лічильників, через ключ AND2, поступають імпульси з генератора синхроімпульсів (вхід CLK) ці ж імпульси надходять на лічильник Галуа Counter\_Galua. Лічильники працюють на віднімання до того часу, коли на всіх їхніх виходах TCD одночасно буде логічний "0". Сформований генератором Галуа код буде відповідати перетворенню СЗК:

$$N_i = \sum_{i=1}^n b_i \cdot B_i \pmod{\wp},$$

де  $b_i$  – залишки, що відповідають значенню технологічних параметрів;

$B_i$  – базисні числа;

$\wp = P_1 \cdot P_2 \cdot \dots \cdot P_n$  – модулі СЗК.

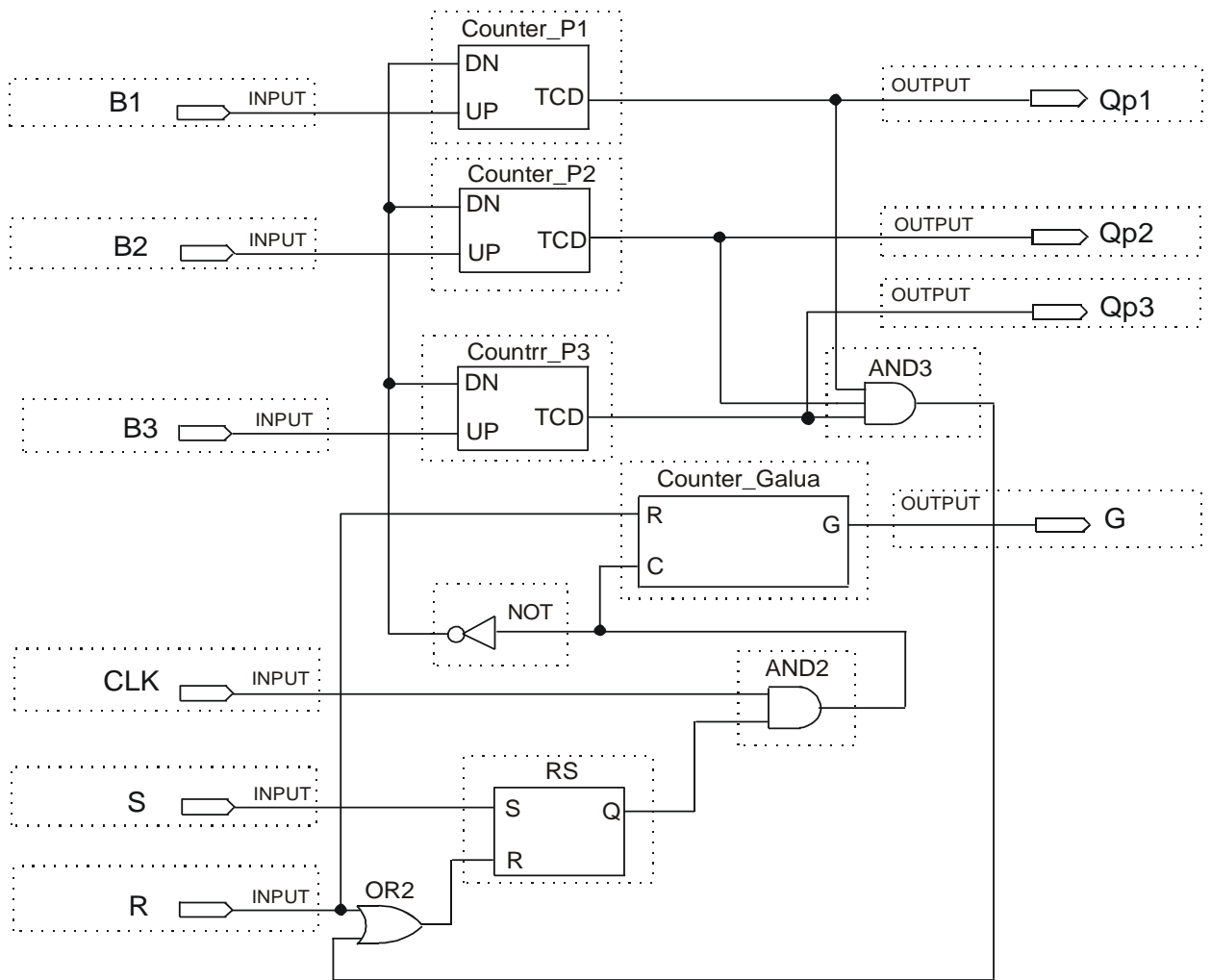


Рис. 4.9. Функціональна схема кодера.

В канал зв'язку передається останніх  $n$  – біт послідовності Галуа. На приймальній стороні код Галуа надходить на декодер Галуа (рис. 4.10), на виході якого отримуємо унітарний код, що поступає на лічильники, які ведуть підрахунок по модулях  $P_1, P_2, \dots, P_n$ . На виході лічильників отримаємо значення технологічних параметрів.

Декодер Галуа. З каналу зв'язку послідовний код Галуа поступає в регістр зсуву RG\_20, де відбувається перетворення послідовного коду в паралельний. З регістра RG\_20 код Галуа перезаписується в регістр зсуву із зворотнім зв'язком, в якому відбувається генерування послідовності Галуа до того часу, поки всі розряди регістра приймуть одиничні значення. При виконання цієї умови на виході ключа, який реалізований на логічному елементі I, буде логічна одиниця, яка встановить RS – тригер в положення “0”.



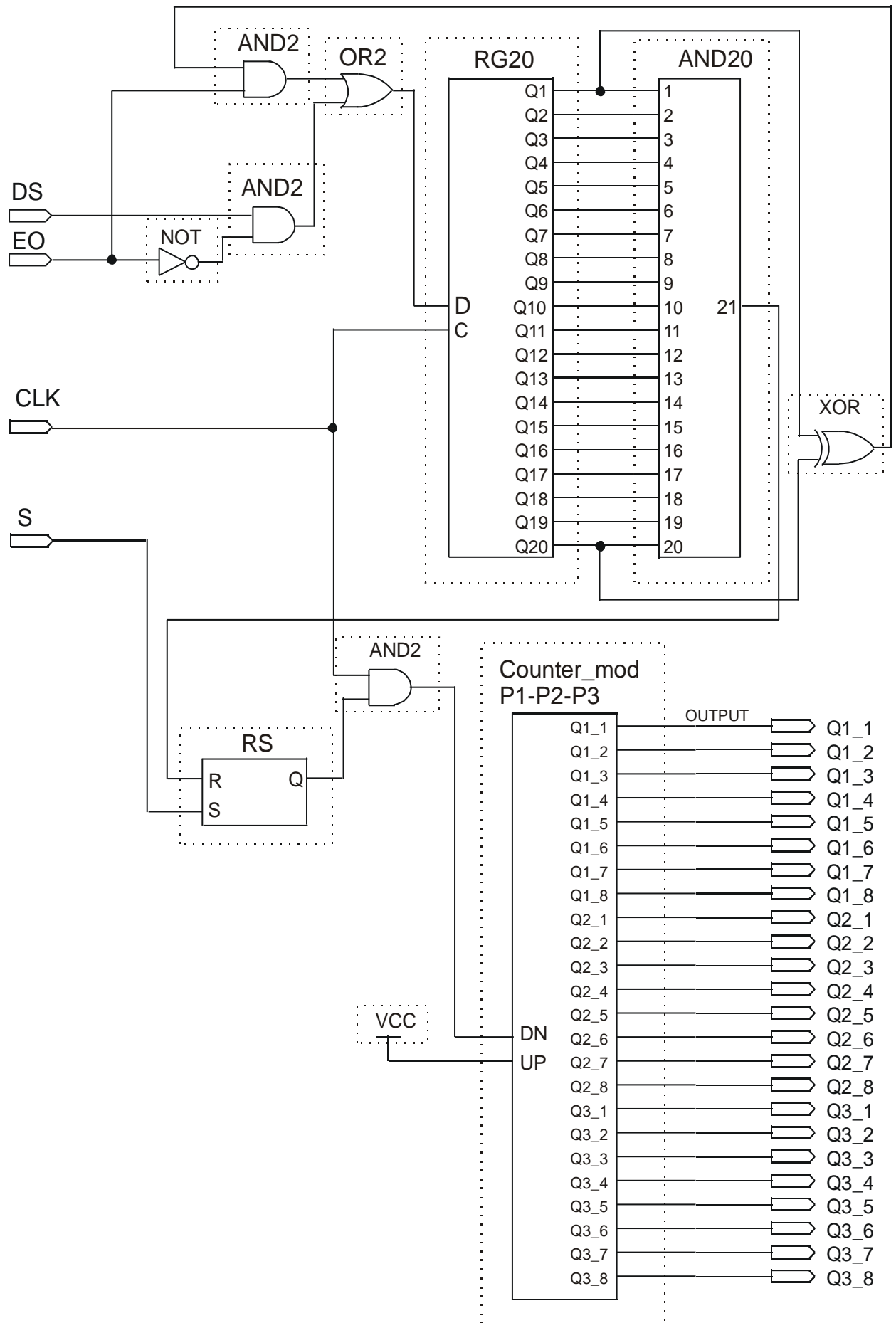


Рис. 4.10. Функціональна схема декодера.

В лічильнику СТ\_20, всі розряди якого попередньо встановлені в “1”, проходить віднімання імпульсів, кількість яких пропорційна кількості імпульсів, що поступають на вхід лічильника Галуа. Після приходу логічної одиниці з ключа на вхід R RS– тригера надходження імпульсів припиняється.

На виходах лічильника Counter P1-P2-P3 отримаємо двійковий код, що відповідає вимірним значенням технологічних параметрів.

Використання прямого перетворення СЗК дозволяє перейти від багатовимірного представлення вимірних значень технологічних параметрів до одновимірного, що забезпечує наступні переваги:

- зменшення об’єму даних за рахунок зменшення кількості допоміжних бітів, що кодують номер каналу та старт-стопові біти кожного параметру;
- спрощення інтерфейсних схем та комунікаційних протоколів передавання даних;
- захист від несанкціонованого доступу.

### **4.3. Процесор стиснення даних на основі базисних функцій Галуа**

Процесор призначений для реалізації методу стиснення даних на основі базисних функцій Галуа нульового то першого порядку, розробленого в пункті 2.5, і виконує наступні функції:

- 1) визначає номер рівня сигналу;
- 2) генерує послідовність бітів Галуа;
- 3) порівнює попередній і наступний рівень сигналу;
- 4) інвертує останній біт Галуа при зміні рівня сигналу;
- 5) генерує нову послідовність бітів Галуа.

Генератор базисних функцій Галуа (рис 4.11) складається з блока

ідентифікації стану (БІ), дешифратора (DC), генератора Галуа, цифрового компаратора (ЦК) і ключа (К).

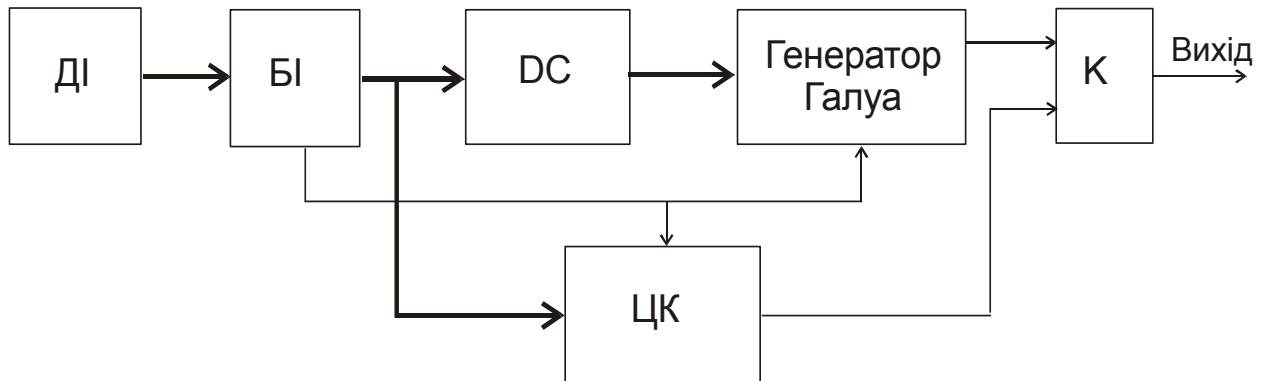


Рис. 4.11. Структурна схема процесора стиснення даних.

Блок індикації ставить у відповідність одному із станів ОК двійковий код, який поступає на DC і на цифровий компаратор. Дешифратор призначений для перетворення паралельного двійкового коду в код, який відповідає початковому значенню послідовності Галуа. В цифровому компараторі відбувається порівняння попереднього і наступного значення коду і формується на виході сигнал 0 або 1:

$$\begin{cases} 0, \text{ якщо } A_i = A_{i+1} \\ 1, \text{ якщо } A_i \neq A_{i+1} \end{cases},$$

де  $A_i$  – поточний рівень сигналу.

Одиничний сигнал на виході ЦК вказує на зміну стану об’єкту і це приводить до інвертування поточного біту послідовності Галуа. При нульовому значенні сигналу на виході ЦК поточний біт не інвертується. Цифровий компаратор (рис. 4.12) складається із регістрів зсуву RG1 і RG2, дешифратора і мультиплексора 16:1 [83]. Двійковий код поступає на паралельні входи регістра пам’яті RG1, з приходом тактового сигналу відбувається перезапис коду із RG1 в RG2 і запис нового значення коду в RG1. З виходу RG1 код поступає на входи дешифратора DC, а з виходу RG2 – на адресні входи мультиплексора MX. DC перетворює паралельний двійковий код в код “біжучий нуль”, який поступає на інформаційні входи

мультиплексора MX. При рівних значеннях кодів, що зберігаються в RG1 і RG2 на виході мультиплексора завжди буде нульовий рівень сигналу і тільки при різних значеннях кодів на виході отримаємо одиничний рівень сигналу.

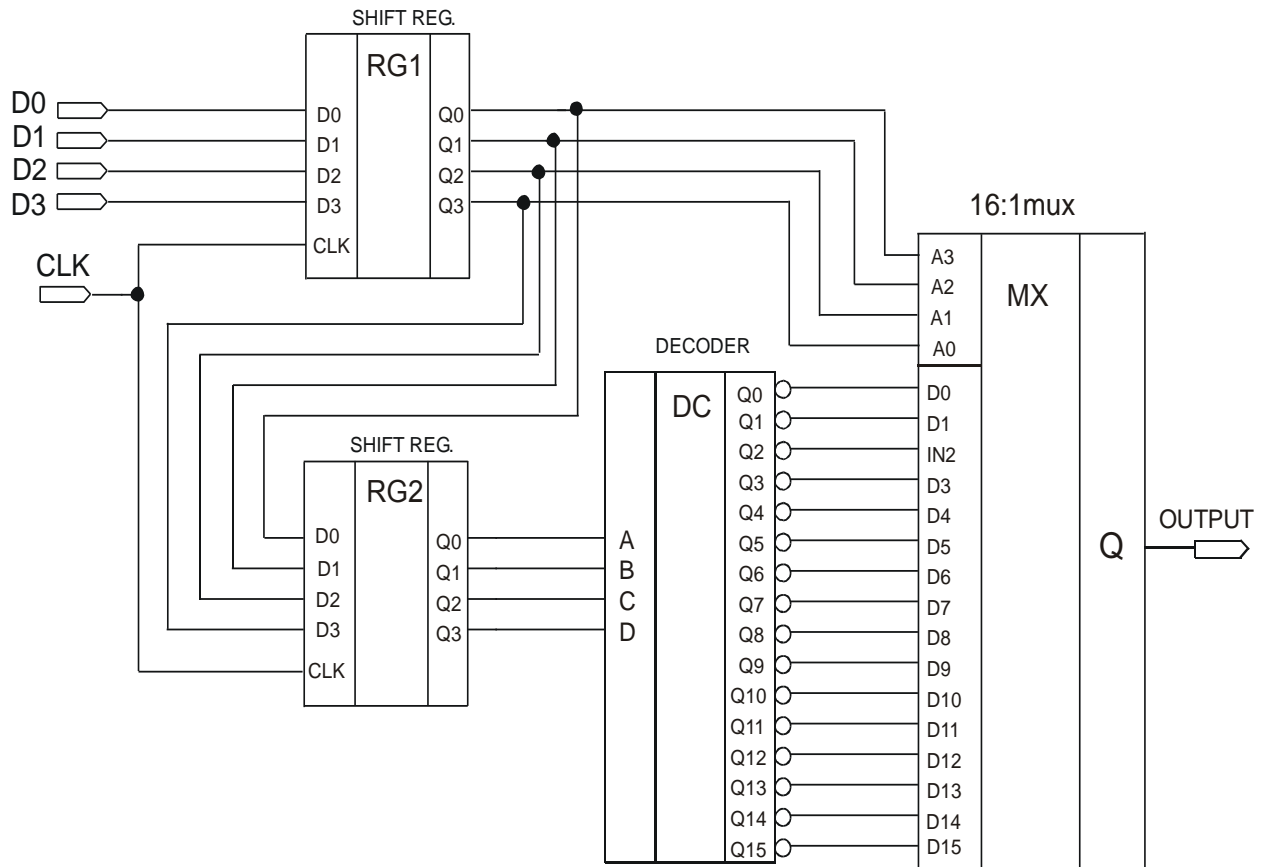


Рис. 4.12. Функціональна схема цифрового компаратора.

Синтезуємо дешифратор, який перетворює двійковий код в код, що відповідає кодону Галуа (табл. 4.3).

За таблицею істинності складаємо рівняння виходів дешифратора:

$$y_1 = \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4;$$

$$y_2 = x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4;$$

$$y_3 = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4;$$

$$y_4 = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4.$$

Таблиця 4.3

Таблиця істинності дешифратора

$x_4$	$x_3$	$x_2$	$x_1$	$y_4$	$y_3$	$y_2$	$y_1$
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	1	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1

Після мінімізації одержимо:

$$y_1 = \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge \overline{x_3};$$

$$y_2 = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_4} \vee \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \vee \overline{x_2} \wedge \overline{x_4} \wedge (\overline{x_1} \wedge \overline{x_3} \vee \overline{x_1} \wedge x_3);$$

$$y_3 = x_2 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3};$$

$$y_4 = x_1 \wedge \overline{x_3} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_4}.$$

За рівняннями виходів розробляємо функціональну схему дешифратора (рис. 4.13).

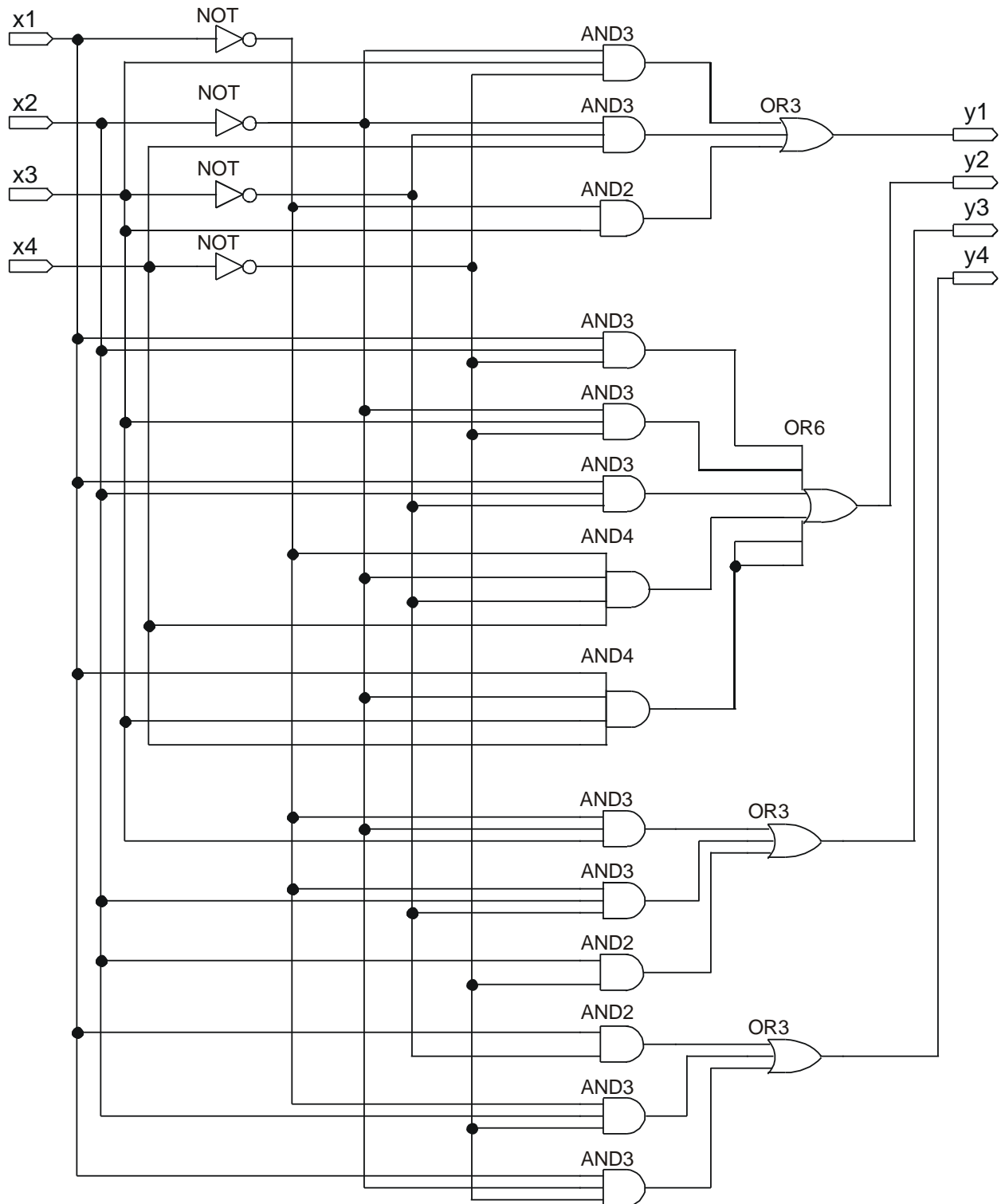


Рис. 4.13. Функціональна схема дешифратора.

Незважаючи на складний математичний апарат, спецпроцесори Галуа мають просту апаратну реалізацію.

Генератор Галуа реалізований на регістрі зсуву зі зворотнім зв'язком (рис. 4.14). На паралельні інформаційні входи D0–D4 із дешифратора DC поступає чотирирозрядний кодон, який записується синхроімпульсом на вході P/S. З приходом тактових імпульсів на вході C на виході одержуємо рекурентну послідовність Галуа з періодом  $N = 2^4 - 1$ .

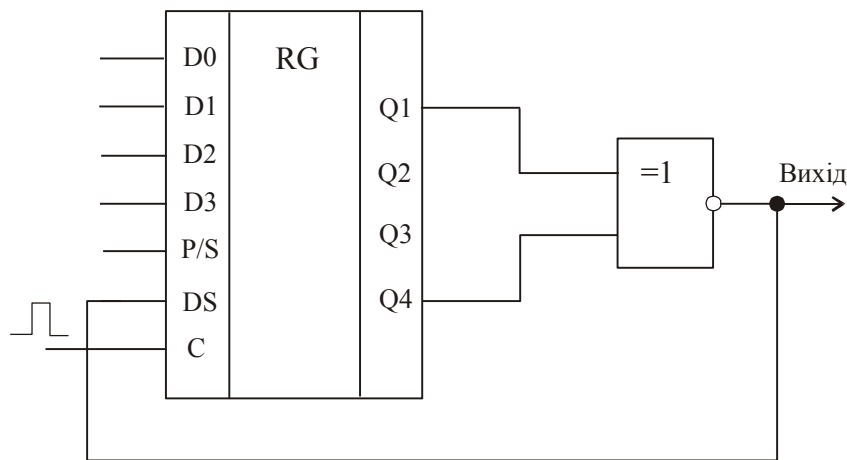


Рис. 4.14. Генератор базисних функцій.

При зміні рівня сигналу на паралельні входи D0–D4 генератора поступає новий кодон.

Розроблений процесор стиснення даних на основі базисних функцій Галуа забезпечує формування потоків даних з рівномірною швидкістю передавання, що дозволило розширити функціональні можливості вертикальної інформаційної технології

#### 4.4. Структура спецпроцесора формування потоків даних логіко-статистичних інформаційних моделей

Актуальною практичною задачею при розробці РКС контролю управління технологічними об'єктами є ефективне використання ліній зв'язку. Використання ЛСІМ дозволило зменшити об'єми технологічних даних, відповідно і вимоги до пропускної здатності ліній зв'язку.

Аналіз ДІ, які представляють технологічні об'єкти нафтогазової промисловості, показав, що їх активність залежить від стану, в якому перебуває ОК і в більшості випадків кількість активних каналів становить 25 % від їх загальної кількості [4].

Спецпроцесор формування потоків даних ЛСІМ призначений для комутації  $2^n$  вхідних каналів на  $2^{n-2}$  вихідних канали.

Робота спецпроцесора описується наступним алгоритмом (рис.4.15): з формувача ЛСІМ дані у вигляді "0" і "1", що відповідають стану ДІ і залежать від використаної ЛСІМ [3], поступають на логічний елемент "виключаюче АБО", на другий вхід якого надходять біти послідовності Галуа. На виході елемента "виключаюче АБО" одержуємо прямі або інвертовані біти послідовності Галуа згідно рівнянь:

$$\begin{aligned} 0 \oplus G &= G \\ 1 \oplus G &= \bar{G} \end{aligned}$$

які поступають на логічний комутатор (ЛК).

З кожного формувача ЛСІМ на ЛК поступають дані про стан ДІ (0 – активний, 1 – неактивний), а також інформаційні дані, якщо ДІ активне.

Логічний комутатор здійснює комутацію активних вхідних каналів на вільні вихідні канали. Генератори рекурентних послідовностей Галуа реалізовані на базі регістра зсуву із зворотнім зв'язком (див. рис. 2.12). Розрядність непривідних поліномів для генерування рекурентних послідовностей вибираємо  $2^r$ , де  $20 \leq r \leq 30$  і залежить від системних



характеристик ДІ. Швидкість генерування послідовностей Галуа дорівнює швидкості формування ЛСІМ, задається частотою тактових імпульсів і може бути різною для різних ДІ.

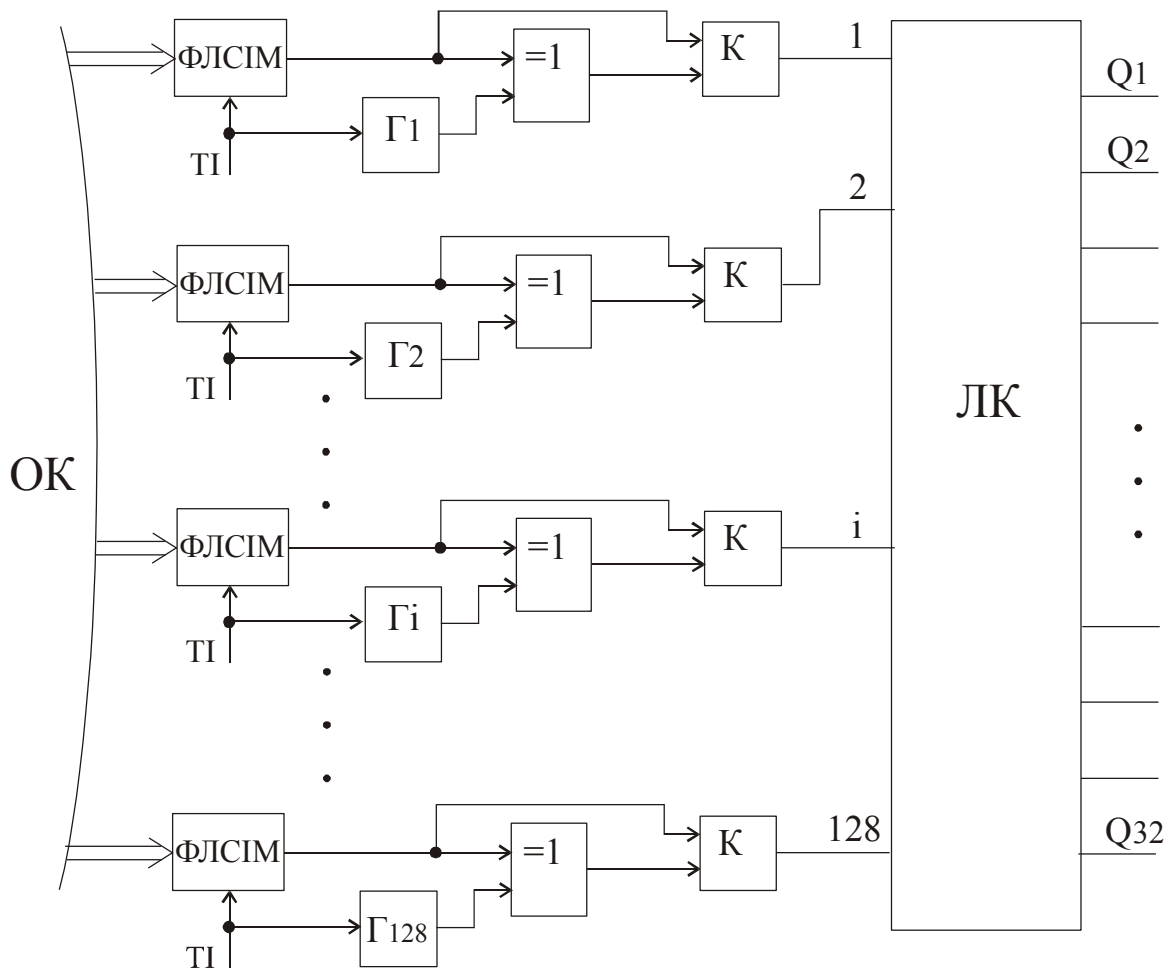


Рис. 4.15. Структурна схема спецпроцесора формування потоків даних ЛСІМ:

ФЛСІМ – формувач даних логіко-статистичних інформаційних моделей;  $\Gamma_i$  – генератор послідовності Галуа; К – ключ; ЛК – логічний комутатор; ТІ – тактові імпульси.

Декодування бітів Галуа відбувається програмно згідно блок-схеми приведеної в додатку Ж.

Розроблена структура спецпроцесора дозволить підключати нові ДІ при експлуатації існуючих систем збору та обробки даних без прокладання нових ліній зв'язку.

#### **4.5. Алгоритм та програма стиснення даних в контролерах енергетичних систем**

Важливим етапом написання програмного забезпечення методу стиснення даних, представлених гармонічними сигналами, є розробка алгоритму. Необхідно визначити структуру системи (модулі) (додаток З), деталізувати алгоритм кожного з модулів, визначити: взаємозв'язки і взаємодію між модулями програмного забезпечення; множини вхідних і вихідних результатів; спосіб формування результатів; точність обчислень. А також потрібно передбачити роботу системи у будь-якому випадку, можливішому в процесі розв'язку задачі.

Програмне забезпечення методу стиснення гармонічних сигналів складається з двох основних модулів:

- модуль архівування;
- модуль розархівування;

Модуль архівування (додаток К) виконує функції компресії даних сигналу і складається з наступних кроків:

- 1) зчитати дані з файлу у буфер – виконується операція читання даних з носія інформації у буфер даних;
- 2) ініціалізація змінних для диференціювання – виконується формування значень, які потрібні для виконання операції диференціювання, таких як крок дискретизації, значення циклічної частоти і ряд інших параметрів;
- 3) перевірка чи сигнал гармонічний – якщо сигнал гармонічний, то перейти до кроку 4, в іншому випадку перейти до кроку 10;
- 4) провести диференціювання сигналу – виконується процес диференціювання;
- 5) перевірка чи отримане значення рівне елементу даних сигналу – при виконанні умови перейти до кроку 6, при невиконанні перейти до кроку 14;
- 6) генерувати біт Галуа – виконується генерування біта Галуа;

7) отриманий біт записати у буфер – виконується запис біта у буфер. Цей біт служить міткою початку стиснутого періоду гармонічного коливання;

8) записати  $1/8$  значення періоду у буфер – виконується запис частини періоду гармонічного коливання, згідно якої відбувається декомпресія цілого періоду гармонічного коливання;

9) збільшити значення лічильника на розмір  $7/8$  періоду – виконується збільшення лічильника, щоб він вказував на наступний період. Перейти до кроку 14;

10) генерувати біт Галуа – виконується генерування біта Галуа;

11) інвертувати біт і записати у буфер - виконується інвертування і запис біта у буфер. Цей біт служить міткою початку даних, які не містять гармонічного коливання;

12) записати у буфер негармонічні дані – виконується запис реальних даних;

13) збільшити значення лічильника на розмір одного періоду – виконується збільшення значення лічильника, щоб він вказував на наступний період;

14) перевірка чи досягнуто кінця буфера – при невиконанні умови збільшити значення лічильника на 1 і перейти до кроку 3;

15) запис буфера у файл – виконати операцію запису стиснутих даних вмісту буфера на носій інформації;

16) перевірка чи досягнуто кінця файлу – при виконанні умови завершити виконання модуля, в іншому випадку виконати операцію читання даних з носія інформації у буфер даних і повернутися до кроку 3.

Модуль розархівування (додаток Л) виконує функції декомпресії даних і складається з наступних кроків:

1) зчитати дані з файлу у буфер – виконується операція читання даних з носія інформації у буфер даних;

2) ініціалізація змінних для інтегрування – виконується формування

значень, які потрібні для виконання операції інтегрування;

3) перевірка чи знайдено біт Галуа – при невиконанні умови перейти до кроку 11;

4) перевірка чи знайдений біт інвертований – при виконанні перейти до кроку 9, в іншому випадку перейти до кроку 5;

5) інтегрувати  $1/8$  періоду – виконується процес інтегрування значення  $1/8$  періоду гармонічного коливання;

6) формування  $7/8$  періоду – виконується формування  $7/8$  періоду гармонічного коливання з отриманого інтегрованого  $1/8$  періоду;

7) запис отриманого періоду у буфер – виконується запис отриманого періоду у буфер;

8) збільшити лічильник на розмір  $7/8$  періоду - виконується збільшення лічильника, щоб він вказував на наступний період;

9) записати без змін дані у буфер – виконується запис даних сигналу у буфер без змін;

10) збільшити значення лічильника на розмір одного періоду – виконується збільшення лічильника, щоб він вказував на наступний період;

11) перевірка чи досягнуто кінця буфера – при невиконанні умови збільшити значення лічильника на 1 і перейти до кроку 3;

12) записати буфер у файл – виконати операцію запису розпакованих даних вмісту буфера на носій інформації;

13) перевірка чи досягнуто кінця файлу – при виконанні умови завершити виконання модуля, в іншому випадку виконати операцію читання даних з носія інформації у буфер даних і повернутися до кроку 3.

#### **4.5.1. Функціональна структура програмного забезпечення**

Функціональна схема програми стиснення даних енергетичних об'єктів складається (додаток М):

головне меню – з нього користувач вибирає пункти меню файл, опції, довідка;

1) меню файл містить підменю:

- Запакувати – компресувати файл з даними сигналу у архівний файл;
- Розпакувати - декомпресувати архівний файл у файл з даними сигналу;
- Вихід – вийти з програми;

2) меню опції містить наступні параметри налаштування:

- кількість відліків на період – вказується значення кількості дискретних значень, які припадають на один період гармонічного коливання;
- крок дискретизації – вказується з яким кроком дискретизації дані сигналу збережені у файлі даних;
- розрядність генератора кодів Галуа – вказується з якою розрядністю мають генеруватися коди Галуа;

3) меню довідка – містить довідку про програму.

При виборі меню файл можна виконати підменю запакувати, розпакувати, вихід. Підменю запакувати викликає модулі компресії: аналіз даних компресії, компресор, генератор кодів Галуа. Модуль аналізу даних компресії зчитує з пам'яті значення кількості відліків, яке при запуску програми встановлюється у значення 40. Також модуль компресії зчитує з пам'яті значення кроку дискретизації, яке при запуску програми встановлюється у значення 0,02. Аналогічно модуль генерації кодів Галуа зчитує з пам'яті значення розрядності кодів, яке при запуску програми встановлюється у значення 20. Підменю розпакувати викликає модуль декомпресії, який складається з елементів: аналіз даних декомпресії, декомпресор, аналіз кодів Галуа. Підменю опції дає можливість змінювати значення параметрів, кількість відліків, крок дискретизації, розрядність кодів. Підменю довідка виводить діалогове вікно з додатковою інформацією про програму.

Програмне забезпечення методу стиснення даних енергетичних об'єктів розроблене в інтегрованому середовищі програмування Borland C++ Builder 5.0.

#### **4.5.2. Програмна реалізація методу стиснення даних енергетичних об'єктів**

Програма містить наступні модулі:

модуль “Архівування”, який призначений для компресії даних сигналу, який збережений у файлі даних.

- Форма виклику: модуль викликається серією внутрішніх процедур.
- Вхідні параметри: вказівник на масив даних. Масив обмежений розміром 4096 елементів.

- Вихідні параметри: вказівник на масиви даних.

Модуль “Розархівування” призначений для декомпресії даних, які збережені у файлі архіву.

- Форма виклику: модуль викликається серією внутрішніх процедур.
- Вхідні параметри: вказівник на масив даних. Масив обмежений розміром 4096 елементів.
- Вихідні параметри: вказівник на масиви даними. Масив обмежений розміром в 4096 елементів.

Модуль архівування (додаток К) являє собою об'єкт `sign_arch`, який містить наступні внутрішні процедури: `analisis_sig_arch()`, `arch_sig()`, `gen_Galua (unsigned int &g_zn, int roz)`:

- `analisis_sig_arch` – виконує аналіз даних компресії;
- `arch_sig` – виконує сам процес компресії;
- `gen_Galua (unsigned int & g_zn, int roz)` – виконує генерацію кодів Галуа, де `g_zn` – поточне значення коду Галуа, `roz` – значення розрядності коду Галуа.

Також в цьому об'єкті міститься вказівник `*buf1`, конструктор `sign_arch`, який присвоює вказівнику `*buf1` значення масиву даних.

Модуль розархівування являє собою об'єкт `sign_darch` який містить наступні внутрішні процедури: `analis_sig_darch()`, `darch_sig()`, `dconv_Galua (unsigned int &g_zn, int roz)`:

- `analis_sig_darch` – виконує аналіз даних декомпресії;
- `darch_sig` – виконує сам процес декомпресії;
- `dconv_Galua (unsigned int & g_zn, int roz)` – виконує аналіз кодів Галуа, де `g_zn` – поточне значення коду Галуа, `roz` – значення розрядності коду Галуа.

Також в цьому об'єкті міститься аналогічно вказівник `*buf1`, конструктор `sign_arch`, який присвоює вказівнику `*buf1` значення масиву даних.

В тексті програми містяться дві структури: `Sign_stat`, і `Sig`. Вони призначені: `Sig` – містить статистичну інформацію про процес компресії, яка записується у файл; `Sign_stat` – містить тимчасову інформацію про параметри компресії і декомпресії.

Як уже згадувалось вище, програма реалізована в інтегрованому середовищі програмування Borland C++ Builder 5.0, яке має не тільки надзвичайно великі можливості програмування, а й засоби для розробки швидкого і зрозумілого діалогу з користувачем (Object Inspector, панель Component Palette, Project Manger, Class Explorer та ін.).

Діалог користувача з програмою можна описати на основі функціональної схеми (додаток М). Будь-яку операцію користувач може виконати з головного меню програми: запакувати, розпакувати файл, змінити опції.

При запуску програми виводиться основне вікно програми (рис. 4.15). в цьому вікні міститься два вікна: дерево папок, файли у вибраній папці, а також меню – файл, опції, вигляд і довідка.

Меню файл містить підменю: запакувати – запакувати вибраний файл,

якщо він містить гармонічні коливання, розпакувати – розпакувати вибраний файл, якщо він містить архівні дані гармонічного сигналу.

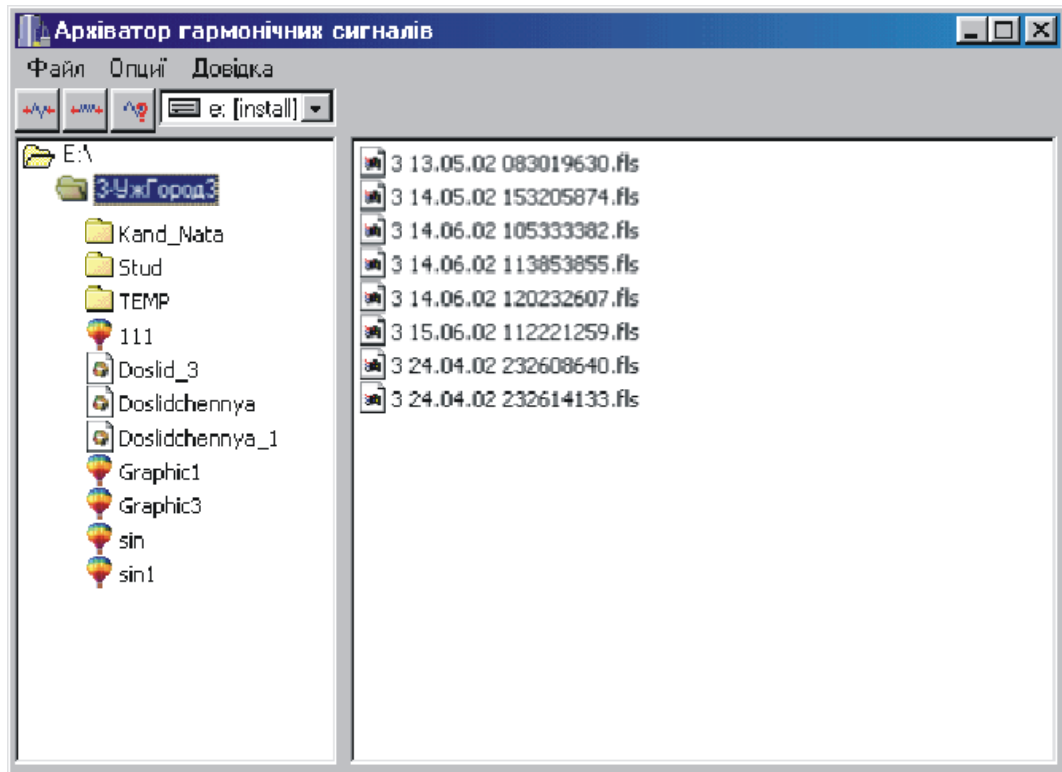


Рис. 4.15. Головне вікно програми.

Меню опції викликає вікно, яке містить параметри: кількість відліків на період – кількість дискретних значень елементів гармонічного коливання на один період, крок дискретизації, розрядність генератора кодів Галуа.



## ВИСНОВКИ

1. Розроблено інтегрально-імпульсний перетворювач з розширеними функціональними можливостями, який забезпечує:

- зміну вхідної напруги 10 мкВ – 100мВ, 3 мВ – 30 В;
- зміну коефіцієнта ділення вхідної частоти:  $2^n$  (де  $n = \overline{1, 8}$  );
- вибір ключів  $2^{20} - 2^{30}$ ;
- тривалість імпульсів передавання (0,01 мс – 200 мс).

Основні модулі ІІП реалізовані на ПЛІС, що підвищило надійність розробленого пристрою. Виготовлено дослідний взірець ІІП з розширеними функціональними можливостями.

2. Розроблено спецпроцесор в базисі Крестенсона, який призначений для збору та передавання даних з розподілених об'єктів низових комп'ютерних мереж.

3. На основі запропонованих базисних функцій Галуа розроблено процесор стиснення даних.

4. Запропонована структура спецпроцесора формування потоків даних логіко-статистичних інформаційних моделей, яка дозволить ефективніше використовувати лінії зв'язку розподілених комп'ютерних систем.

5. Розроблено алгоритм та програмне забезпечення методу стиснення даних, представлених гармонічними сигналами, для використання в системі реєстрації миттєвих значень електричних параметрів на первинному електричному обладнанні.

## ОСНОВНІ ВИСНОВКИ

У дисертаційній роботі розв'язано наукову задачу розробки та дослідження методів зменшення надлишковості цифрових даних у базисі Галуа, що забезпечують стиснення даних в процесі формування і передавання на низовому рівні РКС. При цьому отримано наступні результати.

1. Проаналізовано існуючі методи та інформаційні технології стиснення даних у комп'ютерних системах на основі представлення потоків повідомлень в різних теоретико-числових базисах і доведена перспектива використання базису Галуа для стиснення технологічних даних.

2. Розроблена архітектура системи ідентифікації станів ОК забезпечує на низовому рівні можливість зменшення обсягів даних на 1–2 порядки за рахунок кодування відповідного стану об'єкта та системи формальних параметрів, що описують стан об'єкта, замість інформації, що формується усіма сенсорами.

3. Проведений теоретичний аналіз методик оцінки коефіцієнта стиснення даних, на основі чого запропоновано визначення комплексного коефіцієнта ефективності методів стиснення.

4. Розроблений метод стиснення даних на основі запропонованих базисних функцій Галуа нульового і першого порядку, який забезпечує стабільний ефект зменшення обсягів даних процесів з різкими змінами амплітуди, для яких відомі методи неефективні, призводять до втрат інформації, збільшення обсягів даних стосовно первинних. Причому застосування базисних функцій Галуа першого порядку забезпечує рівномірність слідування бітів, що знижує навантаження і швидкісні вимоги до використання каналів зв'язку.

5. Запропонований і досліджений метод кодування гармонічних сигналів, що дає змогу закодувати період гармонічного сигналу по  $1/8$

частині і однозначно його відновити, а отже, зменшити надлишковість представлення гармонічних сигналів у 7,6 рази.

6. Досліджено метод кодування багатовимірних ДІ в базисі Крестенсона, що дає змогу перейти від багатовимірного до одновимірного представлення значень технологічних параметрів на передавальній стороні та зворотного перетворення на приймальній стороні. Запропоновано використання взаємопростих модулів розмірності  $P = 2^n - 1$ , що дало змогу передавати залишки послідовностями Галуа і реалізувати перетворення Крестенсона – Галуа. Проведені дослідження ефективності кодування номера ДІ в базисах Радемахера, Крестенсона, Галуа показали, що найменшу кількість біт для представлення номеру ДІ необхідно в базисі Галуа (1 біт) при послідовному опитуванні ДІ. Використання системи залишкових класів дає можливість захистити дані від несанкціонованого доступу. Одержані аналітичні залежності коефіцієнта стиснення розробленого методу.

7. Розроблено інтегрально-імпульсний перетворювач з розширеними функціональними можливостями, що дозволило розширити сферу його використання в РКС нафтогазової, енергетичної та інших галузей промисловості. Основні модулі ІІП реалізовано на ПЛІС, завдяки чому підвищена надійність і зменшено габарити пристрою.

8. На основі запропонованого методу стиснення гармонічних сигналів розроблено програмне забезпечення для використання у комп'ютерній системі реєстрації миттєвих значень струмів та напруг на первинному електричному обладнанні.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2000. – 672 с.
2. Николайчук Я.М. Низові обчислювальні мережі: Навчальний посібник. – К.: УМК ВО, 1990. – 55с.
3. Яцків Н.Г., Николайчук Я.М. Системні характеристики джерел інформації та оцінка їх ентропії // Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2002. – Т.3, №3. – С. 197–200.
4. Яцків Н.Г., Николайчук Я.М. Нафтогазова автоматика та нові інформаційні технології–важливий важіль підвищення техногенної екологічної безпеки в паливно–енергетичному комплексі // Вимірювальна та обчислювальна техніка в технологічних процесах. - 2001. – №4. – С. 173 – 176.
5. Ольховський Ю.Б., Новоселов О.Н., Мановцев А.Б. Сжатие данных при телеизмерениях. – М.: Советское радио, 1971. – 304 с.
6. Орищенко В.И. Санников. В.А., Свириденко В.А. Сжатие данных в системах сбора и передачи информации / Под ред. В.А. Свириденко. – М.: Радио и связь, 1985. –184 с.
7. Свириденко В. А. Анализ систем со сжатием данных. М.: Связь, 1977. – 744 с.
8. Ильин В.А. Телеуправление и телеизмерение: Учеб. пособие для вузов. - 3-е изд., перераб. и доп. - М.: Энергоиздат, 1982. - 560 с., ил.
9. Николайчук Я.М., Лучук М.А., Жуган Л. И., Шевчук Б.М. Идентификация информационных состояний объектов исследования и управления на основе системы логико-статистических информационных моделей. Ин-т кибернетики им. В.М.Глушкова АН УССР, Киев, Препринт 88-45 ИК АН УССР, 1988.

10. Николайчук Я.М., Сегін А.І. Моделі джерел інформації та методи їх представлення // Методи та прилади контролю якості. – 1998. – №2. – С.80 – 84.

11. Авдеев Б.Я., Антонюк Е.М., Долинов С.Н. Адаптивные телеизмерительные системы / Под ред. А.В.Фремке. – Л.: Энергоатомиздат, 1981. – 248 с.

12. Кузьмин И.В., Кедрус В.А. Основы информации и кодирования. – К.: Вища школа, 1986.– 238с.

13. Николайчук Я.М., Зевелев С.Я., Крикун З.Н. Выбор оптимальных параметров кодирования методом вычетов. Респ. сб. «Автоматизация и телемеханизация нефтяной промышленности». – 1975. – №2.

14. Николайчук Я.М., Божнев В.П., Зевелев С.Я., Применение методов теории чисел для сжатия измерительной информации в системах телеконтроля процессов бурения. Материалы Всесоюзной конференции молодых ученых нефтяных ВУЗов. – МИНХиГП.: – 1975.

15. Николайчук Я.М., Крикун З.Н., Божнев В.П., Представление измерительной информации в нормализованной системе исчисления остаточных классов. Известия ВУЗов «Нефть и газ». – 1976. – №6.

16. Николайчук Я.М., Петришин Л.Б., Шевчук Б.М. Сжатие данных на основе кодов поля Галуа. Проблемы создания преобразователей формы информации. Тезисы докладов 6 Всесоюзн. симпозиума. – Киев, ИК АН УССР. – 1988.

17. Шеннон К. Работы по теории информации и кибернетике. - М.: Изд-во иностр. лит, 1963. – 438 с.

18. Макс Ж. Методы и техника обработки сигналов при физических измерениях: Пер. с франц. - М.: Мир, 1983. – Т.1 – 311 с., Т.2 – 256 с.

19. Краус М., Кучбах Э., Вошни О.Г. Сбор данных в управляющих вычислительных системах. Пер. с нем. – М.: Мир, 1987. – 294 с.

20. Abramson D.M. An adaptive dependency source model for data compression. *Commun ACM* 32, 1(Jan.), – 1989. – P.77 – 83.

21. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: Диалог-Мифи, 2002. –384 с.

22. Huffman D.A. A method for the construction of minimum redundancy codes. In *Proceedings of the Institute of Electrical and Radio Engineers* 40,9 (Sept.), – 1952. – P. 1098–1101.

23. Jakobsson M. Compression of character string by an adaptive dictionary. *BIT* 25, 4, – 1985. – P. 593–603.

24. Langdon G.G. and Rissanen J.J. A doubly-adaptive file compression algorithms. *IEEE Trans.Commun.* COM-31, 11(Nov.), – 1983. – P. 1253–1255.

25. Lempel A., Ziv J. On the complexity of finite sequences. *IEEE Trans.Inf.Theory* IT-22,1 (Jan.), – 1976. – P.75–81.

26. Moffat A. Word based text compression. Res. Rept., Dept.of Computer Science, Univ.of Melbourne, Victoria, Australia. 1987. – P.115–134.

27. Shannon C.E. Prediction and entropy of printed English. *Bell Syst. Tech. J.* (Jan.), – 1951. – P. 50 – 64.

28. Ziv J. , Lempel A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory.* Vol. IT – 23, N. 3. – 1977, P. 337–343.

29. Ziv J., Lempel A. Compression of individual sequences via variable rate coding. *IEEE ransactions on Information Theory.* Vol. IT-24. N.5, September. – 1978, P. 530 – 535.

30. Burrows M., Wheeler D. J. A block-sorting Lossless Data Compression Algorithm. Digital Systems Research Center. SRC report 124. May 10. – 1994.

31. Хаусли Т. Системы передачи и телеобработки данных: Пер. с англ.- М.: Радио и связь, 1994. – 456 с.

32. Шувалов В.П., Захарченко Н.В., Шварцман В.О. и др. Передача дискретных сообщений: Учебник для вузов / Под ред. В.П. Шувалова. – М.: Радио и связь, – 1990 – 464 с.

33. Банкет В.Л., Дорофеев В.М. Цифровые методы в спутниковой связи. - М, 1988. – 348 с.

34. Беллами Дж. Цифровая телефония. – М: Мир, – 1986. – 258 с.

35. Стил Р. Принципы дельта-модуляции. – М: Мир, – 1979. – 219 с.

36. Вокодерная телефония. Методы и проблемы. /Под ред. А.А. Пирогова. - М: Связь, – 1974. – 352 с.

37. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. – М: Мир, – 1978. – 432 с.

38. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов. - М: Радио и связь, – 1981. – 364 с.

39. Сапожков М.А., Михайлов В.Г. Вокодерная связь. – М: Радио и связь, - 1983. – 256 с.

40. Залмазон Л.А. Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи, и других областях. - М.: Наука, 1989. – 496 с.

41. Вариченко Л.В., Лабунец В. Г., Раков М. А. / Абстрактные алгебраические системы и цифровая обработка сигналов – К.: Наук. думка, 1986. – 248 с.

42. Садихов Р.Х., Чеголин П.М., Шмерко В.П. Методы и средства обработки сигналов в дискретных базисах. – Мн.: Наука и техника, 1987. – 296 с., ил.

43. Хармут Х.Ф. Передача информации ортогональными функциями: Пер.с англ. Дядюнова Н.Г. и Сенина А. И. – М.: Связь, 1975. – 270 с

44. Левицький А.О. Метод формування повідомлень на основі інтегрально-імпульсних моделей. //BISTRO/96/052 Матеріали 2-ї

Міжнародної науково-практичної конференції “Управління енерговикористанням”. – Львів. - 1997. – С. 36 – 39.

45. Виттих В.А., Гинзбург А. Н. Некоторые общие вопросы теории сокращения представления измерительных сигналов // Автометрия. – 1968.– №3.

46. Воздушно-космическая телеметрия. Пер. с англ. /Под ред. Трофимова К.Н. – М.: Воениздат, 1968. – 476 с.

47. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. - М.: Сов. радио. – 1968. – 460 с.

48. Торгашев В.А. Система остаточных классов и надёжность ЦВМ. – М.: Сов. радио. – 1973. – 274 с.

49. Яцків Н.Г. Дослідження системних характеристик методів формування даних в різних теоретико-числових базисах. Вимірювальна та обчислювальна техніка в технологічних процесах: Збірник наукових праць. – Хмельницький: ТУП. – 2002.–№9 (том 2). – С. 132 – 136.

50. Муттер В.М. Основы помехоустойчивой телепередачи информации. – Л.: Энергоатомиздат. Ленингр. отд-ние, 1990. – 268 с.

51. Нейфах А.Э. Сверточные коды для передачи дискретной информации. – М.: Наука, 1979. – 222 с.

52. Зяблов В.В., Шавгулидзе С.А. Обобщенные каскадные помехоустойчивые конструкции на базе сверточных кодов. – М.: Москва, 1991. – 207 с.

53. Диксон Р.К., Широкополосные системы: Пер. с англ. / Под ред. В.И. Журавлева. – М.: Связь 1979. – 304 с.

54. Варакин Л.Е. Системы связи с шумоподобными сигналами. – М.: Радио и связь, 1985. –384 с.

55. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов исправляющих ошибки.: Пер. с англ. – М.: Связь, 1979. –744 с., ил.



56. Блейхут Р. Теория и практика кодов контролирующих ошибки. Пер. с англ. – М.: Мир, 1986. – 576с., ил.

57. Петрович И.Т., Размахнин М.К. Системы связи с шумоподобными сигналами. – М.: Советское радио, 1969. – 232 с.

58. Окунев Ю.Б., Яковлев А.А. Широкополосные системы связи с составными сигналами. Под ред. А.М. Заездного. - М.: Сов. радио, 1968 - 168 с.

59. Дудыкевич В.Б. Специализированные периферийные процессоры для первичной обработки информации, представленной число-импульсным кодом // Респ. науч.-техн. конф. «Диагностика и коррекция погрешностей преобразователей технологической информации», Киев. Тез. докл. – К., 1989. – С. 108–110.

60. Тугевич В.Н. Телемеханика: Учеб. пособие для студентов вузов спец. “Автоматика и телемеханика” – 2-е изд.; перераб. и доп. – М.: Высшая школа, 1985.– 423 с.

61. Николайчук Я.Н. Разработка теории и комплекса технических средств формирования, передачи и обработки цифровых сообщений в низовых вычислительных сетях автоматизированных систем: Дис. докт. техн. наук: 05.13.05.– Ивано-Франковск, 1991. – 573 с.

62. Яцків Н.Г. Методи стиснення даних в інформаційно-керуючих системах // Розвідка і розробка нафтових і газових родовищ. Серія: Технічна кібернетика та електрифікація об’єктів паливно-енергетичного комплексу. – Івано-Франківськ: ІФДТУНГ. – 2001. – №37 (том 6). – С.183–186.

63. Палагин А.В., Николайчук Я.Н. Опыт разработки микропроцессорных распределенных систем реального времени. - К.: Знание, 1988. – 19с.

64. Николайчук Я.Н. Методология формирования, передачи и обработки дискретных сообщений в НВС // Материалы VI Всесоюзной школы-семинара. – Л: ФМИ. – 1987.– С. 187–193.

65. Петришин Л.Б. Николайчук Я.Н., Ищеряков С.М., Цифровая обработка сигналов на основе преобразования кодов поля Галуа / Методы и микроэлектронные средства цифровой обработки и преобразования сигналов.– Рига: ИЭВТ АН Латвии. – 1989.– С.130 – 132.

66. Яцків Н.Г., Яцків В.В. Методи кодування та фізичного представлення сигналів в інформаційно-керуючих системах // Розвідка і розробка нафтових і газових родовищ. Серія: Методи і засоби технічної діагностики. – Івано-Франківськ: ІФДТУНГ. – 2001. – №38 (том 8). – С.140–144.

67. Николайчук Я.М., Петришин Л.Б. Вертикальна інформаційна технологія в кодових системах Галуа. Матеріали 2-ї Української конференції з автоматичного керування «Автоматика-95» Львів. – 1995. – С.131.

68. Николайчук Я.М., Кусик Я.Б. Коды поля Галуа та їх застосування в перетворювачах форм інформації. Тезисы докладов 7-го симпозиума: Проблемы создания преобразователей формы информации – Киев: ИКАН Украины. – 1992.

69. Петришин Л.Б. Теоретичні основи перетворення форми та цифрової обробки інформації в базисі Галуа: Навч. посібник. – Київ.: ІЗіМН МОУ, 1997. – 237 с.

70. Николайчук Я.М., Яцків Н.Г. Методи стиснення даних в багатоканальних системах на основі кодів Галуа // Вісник національного університету “Львівська політехніка”. Радіоелектроніка та телекомунікації. – Львів. – 2002. – №443. – С.135–138.

71. Николайчук Я.М., Ширмовский Г.Я., Рациональное кодирование и концентрация данных в низовой сети АСУ ТП бурение. Автоматизация и телемеханизация нефтяной промышленности. – М.: ВНИИОЭНГ.– 1983. – №3. – С.236–239.

72. Николайчук Я.М., Ширмовский Г.Я., Процюк В.Р. Компактное кодирование сообщений в многоуровневой системе баз данных// Управляющие системы и машины. – 1984. – №1.

73. Николайчук Я.М. О представлении информации в многоканальных информационно-измерительных системах. Респ. сб. «Изменение, контроль и автоматизация в нефтяной и газовой промышленности». – К.: “Техніка”, 1974.

74. Николайчук Я.М., Шевчук Б.М. Методы цифровой обработки шумоподобных сигналов на основе кодовых ключей. В книге «Технические средства обработки информации для высокопроизводительных ЭВМ и систем» - Киев, Сб. тр. ИКАН УССР, 1988.

75. Nykolaychuk Y., Yatskiv N., The coding of multichannel sources of information //Proc. of the VII<sup>th</sup> International Conf. The Experience of Designing and Application of CAD Systems in Microelectronics. CADSM 2003. – Lviv-Slavske (Ukraine).–2003. – P. 249 – 250.

76. Яцків Н.Г., Король Р.І., Яцків В.В., Федчишин Т.Г. Спецпроцесор обробки даних на основі перетворення Крестенсона – Галуа // Вісник Технологічного університету Поділля. – 2003. –Т1, №3. – С. 105 – 108.

77. Николайчук Я.М., Стус С.М. Методи вертикальної інформаційної технології в базисі Галуа. 3-я українська конференція “Автоматика 96” Севастополь. - 1996. – С. 77.

78. Николайчук Я. М. Основи побудови обчислювальних систем на базі вертикальної інформаційної технології // Тези науково-практичної конференції професорсько-викладацького складу. Івано-Франківськ. – 1999. – С. 90–92.

79. Левицький А.О. Алгоритмічне забезпечення і моделі об’єктів контролю розподілених систем енергообліку // Розробка нафтових і газових родовищ. Серія: Техн. кібернетика та електрифікація ОПЕК. - Івано-Франківськ.: ІФДТУНГ.– 1998. – №35. – С. 15 – 24.

80. Стешенко В. Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС. // Chip News. – 1999. – №8.

81. Стешенко В. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов – М.: «Додека», 2000. – 224с.

82. ByteBlaster MV Parallel Port Download Cable, Data Sheet, Altera corporation, ver.1, April, 1998.

83. Угрюмов Е. П. Цифровая схемотехника. – СПб.: БХВ – Санкт-Петербург, 2002. – 528 с.

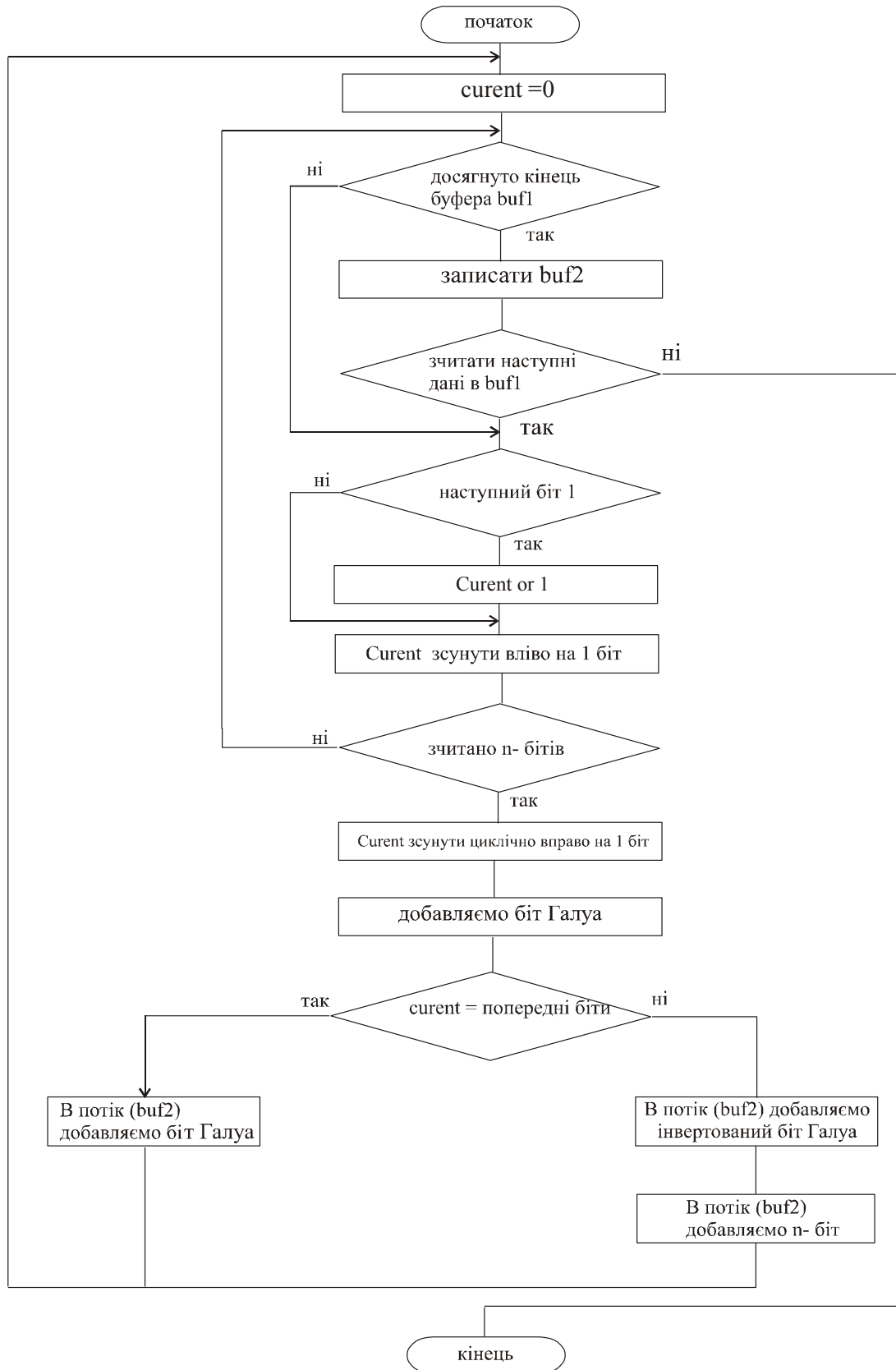
## Додаток А

## Об'єкти низових комп'ютерних мереж та їх інформаційні характеристики

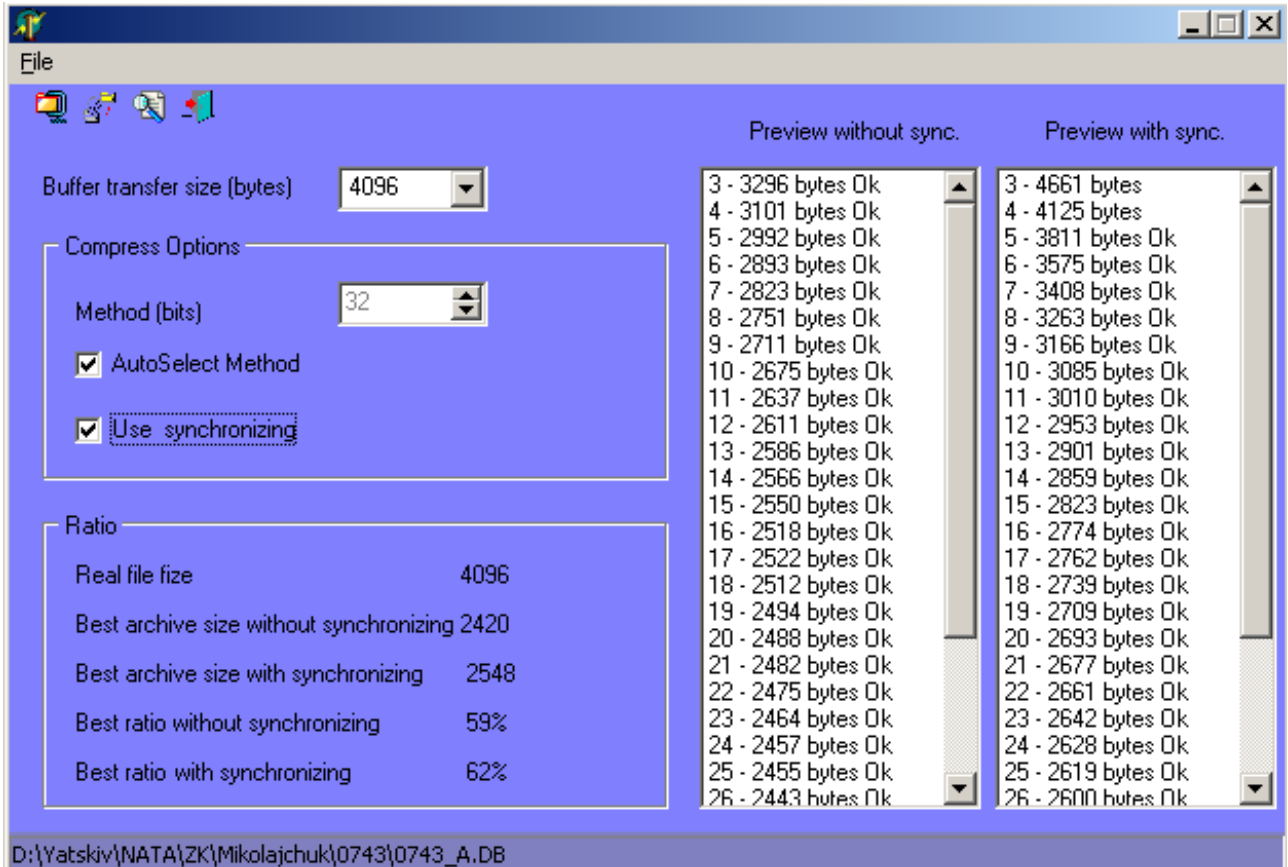
Типи об'єктів керування	Число каналів	Об'єм даних, біт	Швидкість створення повідомлень	Способи представлення даних
Стаціонарні: 1. Технологічна установка підготовки нафти	16 – 32	160 – 320	1,6 – 3,2 кбіт/с	Радемахера, унітарний
2. Газокомпресорна установка	128-256	~ 2560	2,4 кбіт/с	Радемахера, Галуа
Квазістаціонарні: 1. Промислові резервуари	8 – 16	128	128 біт/с	Унітарний, Радемахера, Грея, Галуа
2. Верстатигойдалки	8	80 біт	80 біт/с	Радемахера, унітарний, Галуа, Крейга
Нестаціонарні: 1. Бурова установка	4 – 32	32 – 436 біт	200 біт/с	Радемахера, Крестенсона
Розподілені	Промислові резервуари, станки - гойдалки			
Нерозподілені	Технологічна установка підготовки нафти			
Магістральні	Газокомпресорна установка			

## Додаток Б

## Блок-схема програми стиснення даних на основі адаптивного кодування в базисі Галуа



## Додаток В

Інтерфейс програми стиснення даних на основі  
адаптивного кодування в базисі Галуа

Додаток Д  
Текст програми стиснення даних на основі адаптивного  
кодування в базисі Галуа

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Spin, Menus, Buttons, ComCtrls;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Compress1: TMenuItem;
    Decompress1: TMenuItem;
    N1: TMenuItem;
    Exit1: TMenuItem;
    Preview1: TMenuItem;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    OpenFileDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    GroupBox1: TGroupBox;
    ComboBox1: TComboBox;
    CheckBox1: TCheckBox;
    Label2: TLabel;
    CheckBox2: TCheckBox;
    SpinEdit1: TSpinEdit;
    Label1: TLabel;
    ListBox1: TListBox;
    Label3: TLabel;
    StatusBar1: TStatusBar;
    GroupBox2: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    ListBox2: TListBox;
    Label9: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Compress1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
```



```

procedure SpeedButton4Click(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure Preview1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
  procedure do_arch(file1,file2:string);
  procedure do_unarch(file1,file2:string);
  procedure preview(file1:string);
  procedure prev_arch;
end;

var
  Form1: TForm1;
  handle1,handle2:file;
  {$O-}
  buf1:array[0..16383] of byte;
  buf2:array[0..32767] of byte;
  n_buf,cmpr: array[0..29] of longword;
  fsize,curfs,current,last,n_mnog:longword;
  bit1,bit2,what,corect,procn:byte;
  writ,readen:integer;
  n_flag,nb:word;
  bufsize:word;

implementation

uses Unit2;
{$R *.DFM}

procedure TForm1.prev_arch;
begin
  Form2.Caption:='Preview in process ...';
  Preview(Form1.OpenDialog1.FileName);
  if Terminated then exit;
  nb:=byte(SpinEdit1.Value);
  bufsize:=StrToInt(ComboBox1.Text);
  Form2.Caption:='Archiving in process ...';
  do_arch(Form1.OpenDialog1.FileName,Form1.SaveDialog1.FileName);
  Form2.Caption:='Processing ...';
end;

procedure write_read;
begin
  blockwrite(handle2,buf2,writ);
  asm movzx eax,word ptr writ;
      mov ah,byte ptr buf2[eax];

```

```

    mov byte ptr buf2,ah;
end;
Form2.ProgressBar1.StepIt;
blockread(handle1,buf1,bufsize,readen);
if Terminated then asm mov esi,0; end
else asm mov esi,1 end;
end;

procedure TForm1.do_arch(file1,file2:string);
begin
  assignfile(handle1,file1); reset(handle1,1);
  if IOResult<>0 then begin
    ShowMessage('Unable to Open File '+file1);
    exit; end;
  assignfile(handle2,file2); rewrite(handle2,1);
  if IOResult<>0 then begin
    ShowMessage('Unable to Create File '+file2);
    exit; end;
  fsize:=filesize(handle1);
  Form2.ProgressBar1.Position:=0;
  Form2.ProgressBar1.Max:=(fsize div bufsize)+1;
  blockwrite(handle2,fsize,4);
  blockwrite(handle2,nb,2);
  blockwrite(handle2,n_flag,1);
  blockread(handle1,buf1,bufsize,readen);
  bit1:=8;bit2:=7;what:=0;current:=0;last:=0;curfs:=0;
  n_mnog:=$ffffff;
asm
  xor   esi,esi
  xor   edi,edi
@a:
  mov   current,0
  mov   cx,nb
@1:
  dec   bit1
  cmp   bit1,-1
  jne   @2
  mov   bit1,7
  inc   esi
  inc   curfs
  mov   eax,fsize
  cmp   curfs,eax
  jna   @20
@2a: jmp   @exit
@20:
  cmp   si,word ptr readen
  jne   @2
  push  cx
  mov   word ptr writ,di

```

```

    call  write_read
    pop   cx
    cmp   esi,0
    je    @2a
    xor   esi,esi
    xor   edi,edi
@2:
    mov   al,byte ptr buf1[esi]
    movzx dx,bit 1
    bt    ax,dx
    jnc   @3
    or    current,1
@3:
    shl   current,1
    loop  @1
    rcr   current,1
    cmp   n_flag,1
    jne   @b0
    mov   what,0
    shl   n_mnog,1
    jnc   @g3
    bt    n_mnog,1
    jc    @g2
@g1:
    or    n_mnog,1
    mov   what,1
@g2:
    jmp   @g4
@g3:
    bt    n_mnog,1
    jnc   @g2
    jmp   @g1
@g4:
    call  @badd
@b0:
    mov   eax,current
    cmp   last,eax
    je    @b
    mov   what,1
    call  @badd
    call  @addcur
    jmp   @a
@addcur:
    push  ebx
    mov   ebx,current
    mov   last,ebx
    movzx ecx,nb
@@@1:
    dec   cx
    mov   what,0
    bt    ebx,ecx
    jnc   @@@2

```

```

        mov     what,1
@@@2:   call    @badd
        inc     cx
        loop   @@@1
    pop     ebx
    ret
@badd:
    movzx dx,bit2
    mov     al,byte ptr buf2[edi]
    btr     ax,dx
    cmp     what,1
    jne     @@1
    bts     ax,dx
@@@1:   mov     byte ptr buf2[edi],al
    cmp     bit2,0
    jne     @@2
    mov     bit2,8
    inc     di
@@@2:   dec     bit2
    ret

@b:
    mov     what,0
    call    @badd
    jmp     @a
@exit:
    mov     al,byte ptr buf1[esi]
    movzx dx,bit1
    bt      ax,dx
    jnc     @e1
    or      current,1
@e1:
    dec     cx
    shl     current,cl
    mov     what,1
    call    @badd
    call    @addcur
    mov     word ptr writ,di
    call    write_read
    mov     esi,ebx
end;
closefile(handle1);
closefile(handle2);
if Terminated then begin
    assignfile(handle2,Form1.SaveDialog1.FileName);
    erase(handle2);
end;
end;
end;

```

```

procedure TForm1.do_unarch(file1,file2:string);
  procedure errmess;
  begin

    ShowMessage('Synchronizing wrong. Press Ok to stop');
  end;
begin
  assignfile(handle1,file1); reset(handle1,1);
  if IOResult<>0 then begin
    ShowMessage('Unable to Open File '+file1);
    exit; end;
  assignfile(handle2,file2); rewrite(handle2,1);
  if IOResult<>0 then begin
    ShowMessage('Unable to Create File '+file2);
    exit; end;
  blockread(handle1,fsize,4);
  Form2.ProgressBar1.Position:=0;
  Form2.ProgressBar1.Max:=(fsize div bufsize)+1;
  blockread(handle1,nb,2);
  blockread(handle1,n_flag,1);
  blockread(handle1,buf1,bufsize,readen);
  bit1:=8;bit2:=7;what:=0;current:=0;last:=0;curfs:=0;
  n_mnog:=$ffffff;
  asm
    xor   esi,esi
    xor   edi,edi
@a:
    cmp   n_flag,1
    jne   @b0
    mov   what,0
    shl   n_mnog,1
    jnc   @g3
    bt    n_mnog,1
    jc    @g2
@g1:  or    n_mnog,1
    mov   what,1
@g2:  jmp   @g4
@bcmp:
    dec   bit1
    cmp   bit1,-1
    jne   @1
    inc   si
    mov   bit1,7
    cmp   si,word ptr readen
    jne   @1
    push  cx
    mov   word ptr writ,di
    call  write_read

```

```

pop    cx
    cmp    esi,0
    je     @ex
        xor    esi,esi
        xor    edi,edi
@1:    mov    al,byte ptr buf1[esi]
        movzx dx,bit1
        bt     ax,dx
        ret
@badd:
        movzx dx,bit2
        mov    al,byte ptr buf2[edi]
        btr    ax,dx
        cmp    what,1
        jne   @@@1
        bts    ax,dx
@@@1:  mov    byte ptr buf2[edi],al
        cmp    bit2,0
        jne   @@@2
        mov    bit2,8
        inc    di
        inc    curfs
        mov    eax,fsize
        cmp    curfs,eax
        jne   @@@2
        mov    word ptr writ,di
        call   write_read
        pop    eax
        jmp    @exit
@ex:   pop    ax
        jmp    @exit
@@@2:  dec    bit2
        ret
@g3:   bt     n_mnog,1
        jnc   @g2
        jmp   @g1
@g4:   call   @bcmp
        rcl   what,1
        cmp    what,3
        jne   @error
@b0:   call   @bcmp
        jnc   @b
        mov    last,0
        mov    cx,NB
@@a:   mov    what,0
        call   @bcmp

```

```

        jnc    @@b
        mov   what,1
        or    last,1
@@b:   call   @badd
        shl   last,1
        loop  @@a
        rcr   last,1
        jmp   @a
@b:
        push  ebx
        mov   ebx,last
        movzx ecx,nb
@@@@1:
        dec   cx
        mov   what,0
        bt    ebx,ecx
        jnc   @@@@2
        mov   what,1
@@@@2:  call   @badd
        inc   cx
        loop  @@@@1
        pop   ebx
        jmp   @a
@error:
        cmp   what,0
        je    @b0
        mov   esi,ebx
        call  errmess
@exit:
        mov   esi,ebx
end;
closefile(handle1);
closefile(handle2);
if Terminated then begin
    assignfile(handle2,Form1.SaveDialog1.FileName);
    erase(handle2);
end;
end;

procedure TForm1.preview(file1:string);
procedure bread;
begin
if corect=1 then blockread(handle1,buf1[4],bufsize,readen)
    else blockread(handle1,buf1,bufsize,readen);
Form2.ProgressBar1.StepIt;
    if Terminated then asm mov esi,0; end
    else asm mov esi,1 end;
end;
var

```

```

s_bit,loopl:byte;
lastb: array[0..29] of longword;
bits,beg: array[0..29] of byte;
i,readn:word;
n:string[2];
begin
  assignfile(handle1,file1); reset(handle1,1);
  if IOResult<>0 then begin
    ShowMessage('Unable to Open File '+file1);
    exit; end;
  fsize:=filesize(handle1);
  Form2.ProgressBar1.Position:=0;
  Form2.ProgressBar1.Max:=(fsize div 4096)+1;
  if fsize>$1ffffff then begin
    ShowMessage('File '+file1+'too large'); exit;end;
  current:=0;
  for i:=0 to 29 do begin
    bits[i]:=7; cmpr[i]:=0; lastb[i]:=0;
    beg[i]:=0; n_buf[i]:=0;
  end;
  asm
    mov  last,esi
    mov  corect,0
    mov  bufsize,4100
  @read:
    call  bread
    cmp  esi,0
    je   @exit
    mov  loopl,3
  @d:
    call  @compr
    inc  loopl
    cmp  loopl,33
    jne  @d
    mov  corect,1
    mov  eax,dword ptr buf1[eax-4]
    mov  dword ptr buf1[0],eax
    mov  ax,word ptr bufsize
    mov  dword ptr bufsize,4096
    cmp  word ptr readn,ax
    je   @read
    jmp  @exit

  @compr:
    mov  si,word ptr readn
    mov  word ptr readn,si
    movzx si,byte ptr corect
    add  word ptr readn,si
    movzx esi,byte ptr loopl

```



```

movzx esi,byte ptr beg[esi-3]
    add    word ptr readn,si
@a:
    call   @n_read
    jc    @out
    mov   eax,dword ptr current
    cmp   dword ptr lastb[ebx*4],eax
    je    @b
    mov   dword ptr lastb[ebx*4],eax
    movzx eax,byte ptr loopl
    add   dword ptr cmpr[ebx*4],eax
@b:    inc   dword ptr cmpr[ebx*4]
    inc   dword ptr n_buf[ebx*4]
    jmp   @a
@out:
    mov   cl,byte ptr s_bit
    mov   byte ptr bits[ebx],cl
    mov   eax,esi
    sub   esi,edi
    mov   ecx,4
    sub   ecx,esi
    mov   byte ptr beg[ebx],cl
    ret

@n_read:
    movzx ebx,byte ptr loopl
    mov   ecx,ebx
    sub   ebx,3
    mov   dword ptr current,0
    mov   edi,esi
    mov   al,byte ptr bits[ebx]
    mov   byte ptr s_bit,al
@1:
    mov   al,byte ptr buf1[esi]
    movzx dx,byte ptr bits[ebx]
    cmp   byte ptr bits[ebx],0
    jne   @2
    mov   byte ptr bits[ebx],7
    inc   si
    cmp   si,word ptr readn
    jne   @3
    stc
    ret
@2:    dec   byte ptr bits[ebx]
@3:    bt    ax,dx
    jnc   @4
    or    dword ptr current,1
@4:    shl   dword ptr current,1
    loop  @1

```

```

        rcr    current,1
        clc
        ret
@exit:
        mov   ebx,last
        mov   esi,last
end;
closefile (handle1);
current:=cmpr[0];
for i:=3 to 32 do begin
    if cmpr[i-3]<current then begin
        current:=cmpr[i-3];
        nb:=i;
    end;
end;
SpinEdit1.Value:=nb;
if nb<>0 then Form1.SpinEdit1.Value:=nb else exit;
Form1.ListBox1.Clear; Form1.ListBox2.Clear;
for i:=3 to 32 do begin
    if fsize>(cmpr[i-3]div 8)+1 then n:='Ok'
    else n:="";
    Form1.ListBox1.Items.Add (IntToStr(i)+' - '+
        IntToStr ((cmpr[i-3]div 8)+1)+' bytes '+n);
    if fsize>(((cmpr[i-3]+n_buf[i-3])div 8)+1) then n:='Ok'
    else n:="";
    Form1.ListBox2.Items.Add (IntToStr(i)+' - '+
        IntToStr(((cmpr[i-3]+n_buf[i-3]) div 8)+1)+' bytes '+n);
end;
Form1.Label4.Caption:='Real file fize'+#9#9#9+'
    '+IntToStr(fsize);
Form1.Label5.Caption:='Best archive size without synchronizing '
    '+IntToStr((cmpr [nb-3]div 8)+1);
Form1.Label6.Caption:='Best archive size with synchronizing
    '+IntToStr(((cmpr[nb-3]+n_buf[nb-3])div 8)+1);
Form1.Label7.Caption:='Best ratio without synchronizing'+#9+'
    '+IntToStr((((cmpr[nb-3]div 8)+1)*100)div fsize)+'%';
Form1.Label8.Caption:='Best ratio without synchronizing'+#9+'
    '+IntToStr((((cmpr[nb-3]+n_buf[nb-3])div 8)+1)*100)div fsize)+'%
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    ComboBox1.ItemIndex:=3;
    Terminated:=false;
end;

procedure TForm1.CheckBox2Click(Sender: TObject);
begin
    if CheckBox2.Checked then SpinEdit1.Enabled:=false

```

```

        else SpinEdit1.Enabled:=true;
    end;
    procedure TForm1.SpeedButton1Click(Sender: TObject);
    begin
        if not OpenFileDialog1.Execute then exit;
        if not SaveDialog1.Execute then exit;
        if FileGetAttr(SaveDialog1.FileName)<>-1 then
            if MessageDlg(SaveDialog1.FileName+' exist. Replace
it?',mtWarning,[mbYes,mbCancel],0)=mrCancel then exit;
            if (FileGetAttr(OpenDialog1.FileName)and faReadOnly)<>0 then begin ShowMessage('Read
Only file can't be processed');exit; end;
            StatusBar1.SimpleText:=OpenDialog1.FileName;
            n_flag:=byte(CheckBox1.Checked);
            Terminated:=false;
            if CheckBox2.Checked then begin
                procn:=4;
                Thrd1:=TMyThrd.CreateThread(2);
                Form1.Hide;
                Form2.Show;
                exit;
            end;
            nb:=byte(SpinEdit1.Value);
            bufsize:=StrToInt(ComboBox1.Text);
            procn:=1;
            Thrd1:=TMyThrd. CreateThread (2);
            Form1.Hide;
            Form2.Show;
        end;

    procedure TForm1.Compress1Click (Sender: TObject);
    begin
        SpeedButton1Click (Sender);
    end;

    procedure TForm1.SpeedButton2Click (Sender: TObject);
    begin
        if not OpenFileDialog1.Execute then exit;
        if not SaveDialog1.Execute then exit;
        Terminated:=false;
        if FileGetAttr(SaveDialog1.FileName)<>-1 then
            if MessageDlg(SaveDialog1.FileName+' exist. Replace it?',
mtWarning,[mbYes,mbCancel],0)=mrCancel then exit;
            if (FileGetAttr(OpenDialog1.FileName)and faReadOnly)<>0 then begin ShowMessage ('Read
Only file can't be processed');exit; end;
            StatusBar1.SimpleText:=OpenDialog1.FileName;
            bufsize:=StrToInt(ComboBox1.Text);
            procn:=2;
            Thrd1:=TMyThrd.CreateThread(2);

```

```
    Form1.Hide;
    Form2.Show;
end;
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
    halt;
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
    halt;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    if not OpenFileDialog1.Execute then exit;
    if (FileGetAttr(OpenDialog1.FileName)and faReadOnly)<>0 then
        begin ShowMessage('ReadOnly file can"t be processed');exit; end;
    StatusBar1.SimpleText:=OpenDialog1.FileName;
    procn:=3;
    Terminated:=false;
    Thrd1:=TMyThrd.CreateThread(2);
    Form1.Hide;
    Form2.Show;
end;

procedure TForm1.Preview1Click(Sender: TObject);
begin
    SpeedButton3Click(Sender);
end;

end.
```

## Додаток Е

## Текстовий опис макрофункції G\_GENERAT

```

%=====
% Функція G_GENERAT          %
% Генерує послідовність Галуа, задану %
% характеристичним багаточленом M      %
% Входи:                          %
% M   - вхід багаточлена;           %
% EO  - дозвіл генерації (default=GND);%
% Fin - тактовий вхід;              %
% R   - вхід скидання в 0 (default=GND); %
% Виходи:                          %
% G - вихід послідовності Галуа     %
% Параметри:                        %
% M_WIDTH – довжина полінома;       %
% M_BEGIN – початковий стан         %
%          (default=»100000..00»);  %
%=====
INCLUDE «lpm_xor.inc»;
INCLUDE «lpm_constant.inc»;
PARAMETERS
(
    M_WIDTH,
    M_BEGIN = 0
);
SUBDESIGN g_generat
(

```

```

M[M_WIDTH-1..0]      : INPUT = GND;
Fin                   : INPUT;
EO                    : INPUT = GND;
R                     : INPUT = GND;
G                     : OUTPUT;
)
VARIABLE
  dffs[M_WIDTH-2..0]  : DFFE; - тригери регістра зсуву
  shift_node[M_WIDTH-2..0] : NODE;
  xor_node[M_WIDTH-2..0] : NODE;
  shiftin, shiftout   : NODE; - вхід і вихід регістра зсуву
  IF (USED(M_BEGIN)) GENERATE
    ac : lpm_constant
    WITH (LPM_WIDTH = M_WIDTH, LPM_CVALUE = M_BEGIN);
  END GENERATE;
BEGIN
  ASSERT (M_WIDTH>0)
  REPORT «Значення параметру M_WIDTH повинно бути більше нуля»
    SEVERITY ERROR;

% ----- загальні виводи тригерів -----%
  dffs[].ena = EO;
  dffs[].Fin = Fin;
% ----- асинхронні операції ----- %
  IF (USED(M_BEGIN)) GENERATE
  dffs[].clrn = !R # ac.result[M_WIDTH-2..0];
%   встановлення початкового стану           %
  dffs[].prn = !R # !ac.result[M_WIDTH-2..0];

```

```

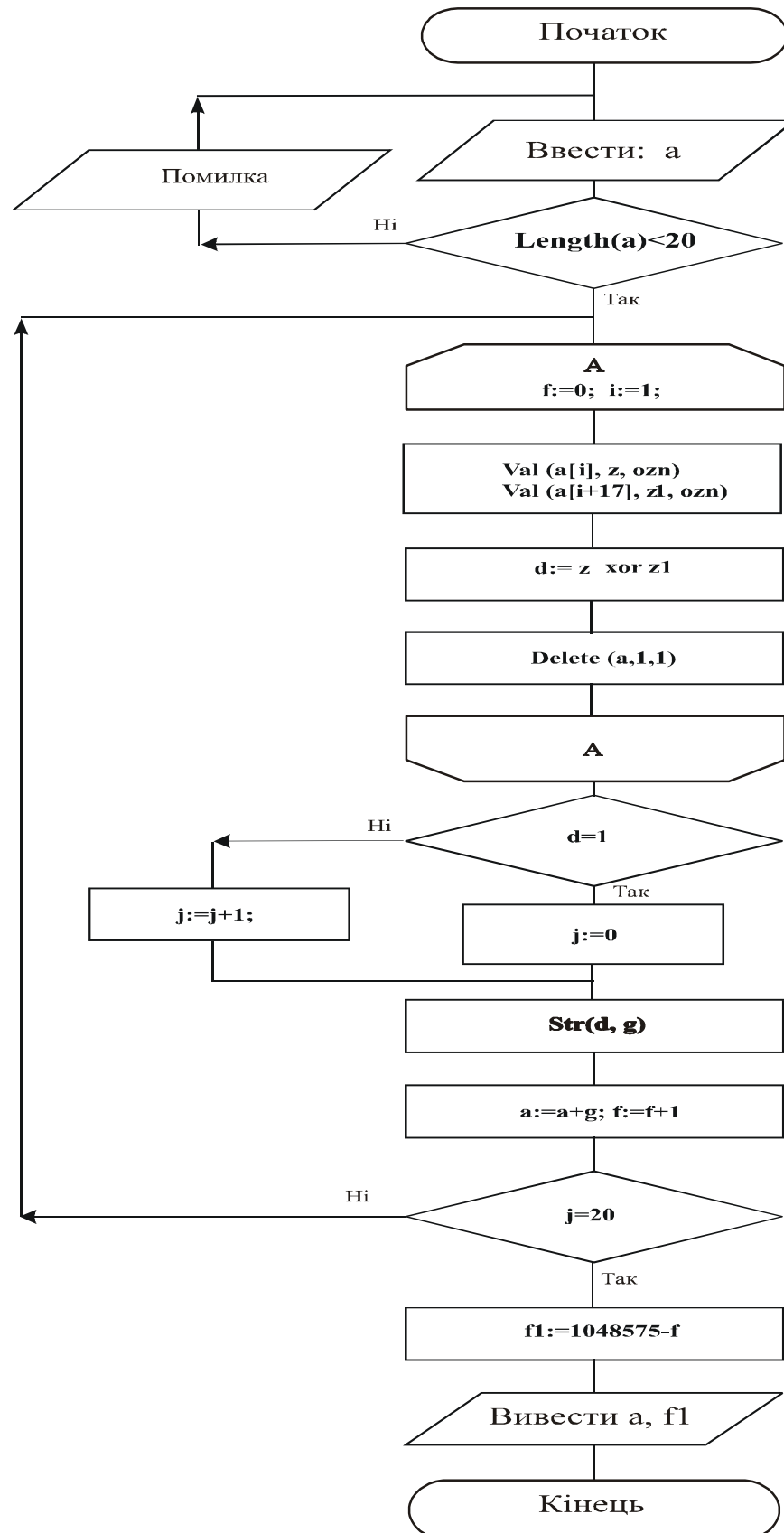
%   тригерів, заданих M_BEGIN           %
    ELSE GENERATE
    dffs[M_WIDTH-3..0].clrn = !R;
%   установка початкового стану         %
    dffs [M_WIDTH-2].prn = !R;
%   тригерів «10000..00                 %
    END GENERATE;

% ----- зворотні зв'язки ----- %
    xor_node[] = dffs[] & M[M_WIDTH-2..0];
    shiftin = lpm_xor(xor_node[])
    WITH (LPM_SIZE = M_WIDTH-1, LPM_WIDTH=1);
    shift_node[] = (shiftin, dffs[M_WIDTH-2..1]);
    shiftout = dffs[0];
    dffs[].d = !R & shift_node[];
    G = shiftout;
END

```

## Додаток Ж

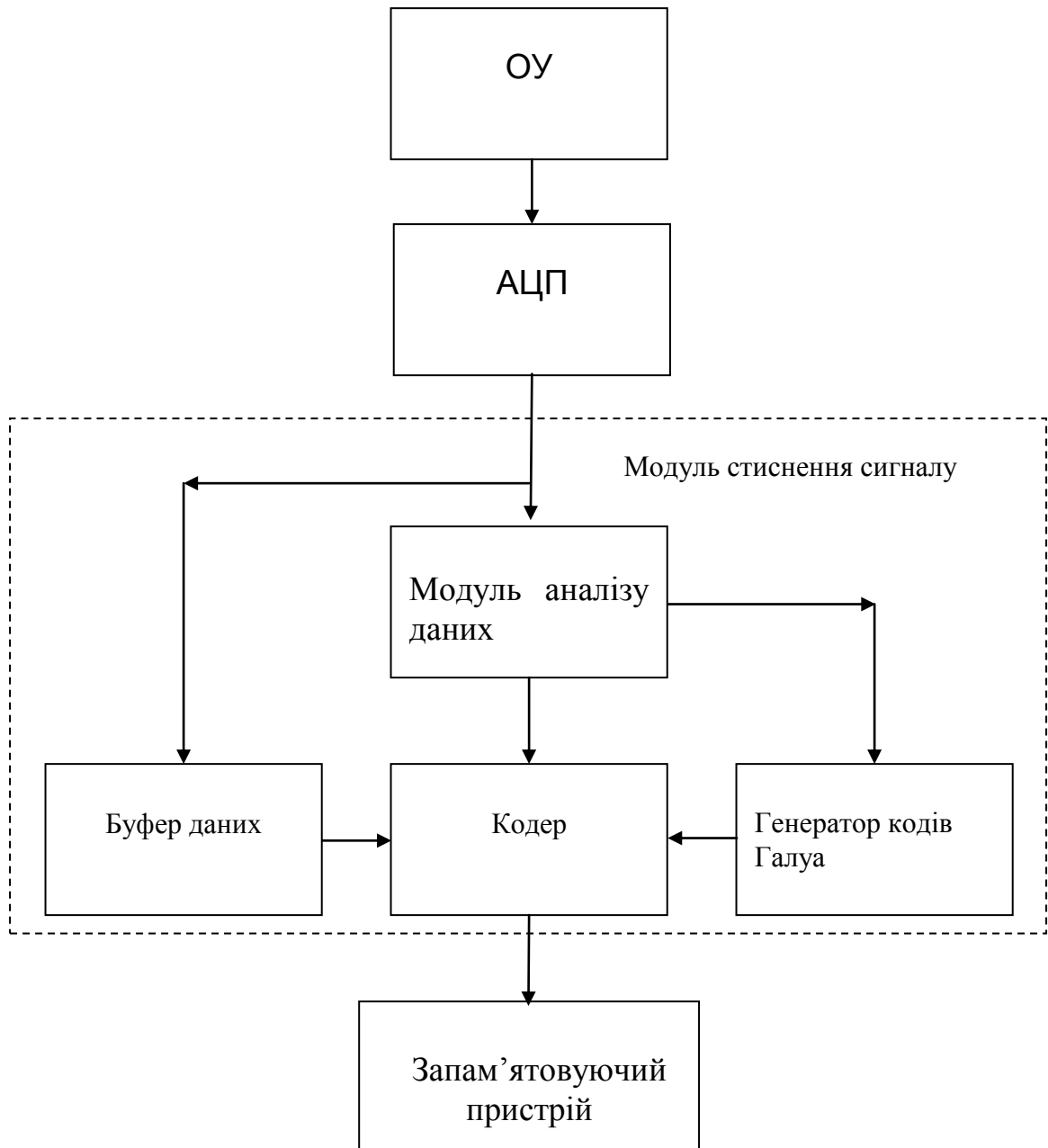
## Блок-схема програми декодування бітів Галуа





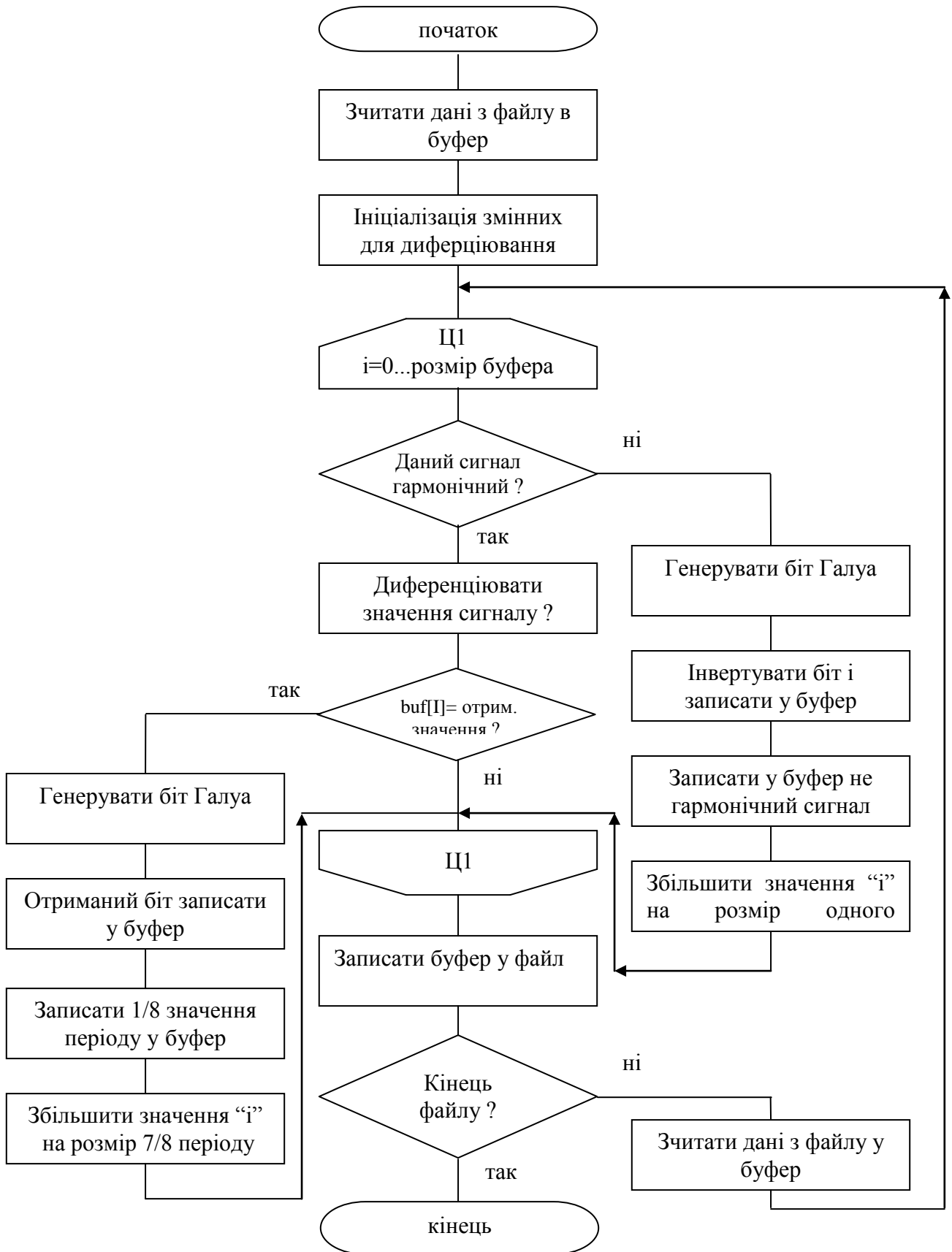
## Додаток 3

## Структурна схема системи реєстрації та зберігання даних



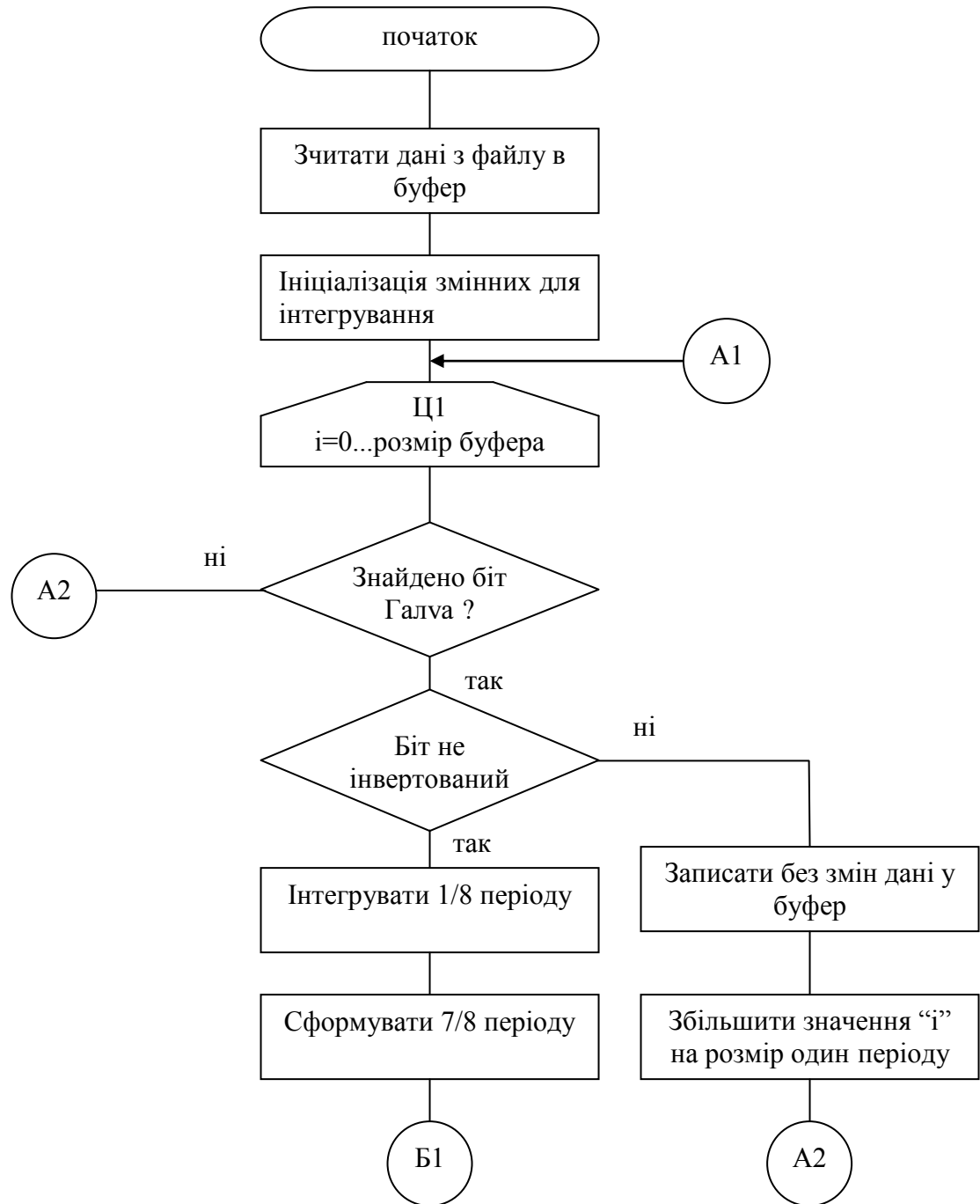
## Додаток К

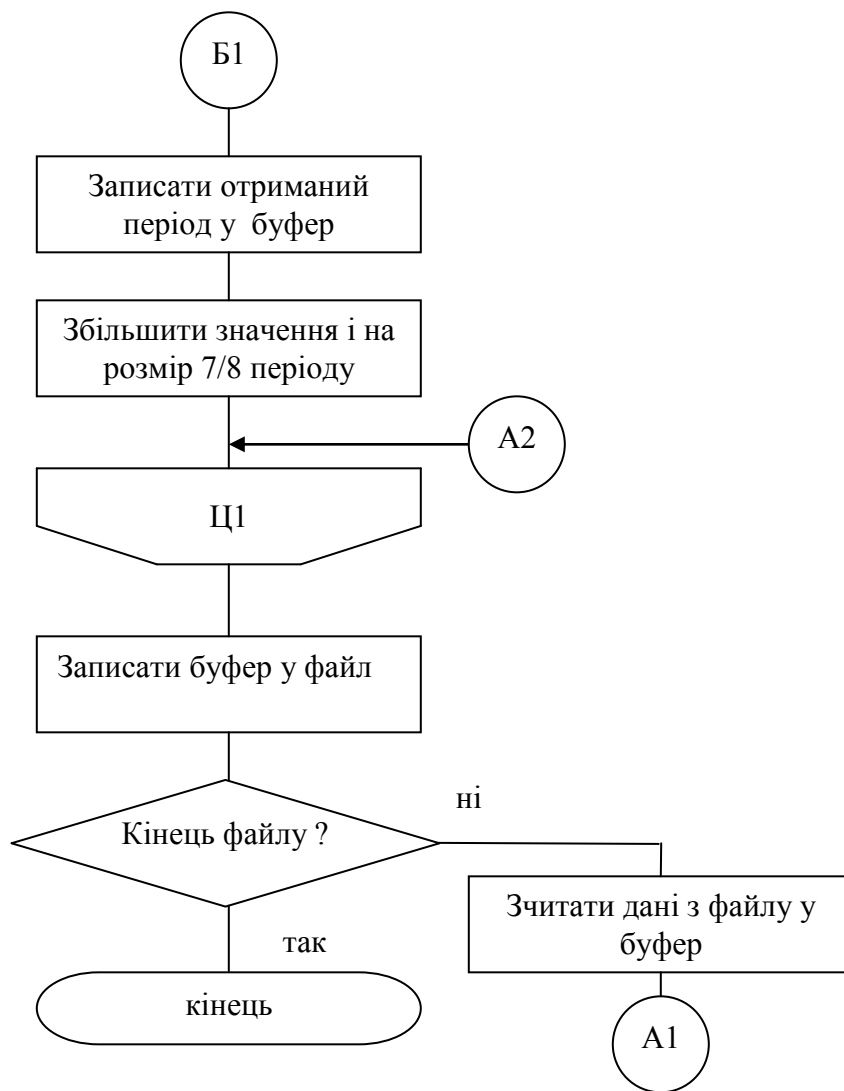
## Схема алгоритму архіватора



## Додаток Л

## Схема алгоритму розархіватора





**Затверджую**

Директор інституту мікропроцесорних

систем керування об'єктами

електроенергетики

І.О. Сабадаш



I.O. Сабадаш

м. Львів

2002 р

**АКТ**

впровадження результатів дисертаційної роботи

**Яцків Наталії Георгіївни**

**“Методи та засоби стиснення даних в розподілених комп'ютерних системах на основі кодів поля Галуа”**

У відповідності до науково-технічної програми “Створення концепції інформаційно-керуючих систем контролю та обліку енергоносіїв, систем керування технологічними процесами електроенергетичних об'єктів та захисту їх обладнання”. Інституту мікропроцесорних систем керування об'єктами електроенергетики НАН України (ІМСКОЕ) передано пакет програм, який розроблено за результатами дисертаційних досліджень Яцків Н.Г., який забезпечує стиснення даних знятих цифровим реєстратором аналогових сигналів, що забезпечує реєстрацію (осцилографування) миттєвих значень електричних параметрів (струмів, напруг) на первинному електричному обладнанні в передаварійних і аварійних режимах:

При розробці програмного забезпечення використано запропонований в дисертаційній роботі Яцків Н.Г. метод стиснення даних представлених гармонічними сигналами. Розроблений пакет програм забезпечує коефіцієнт стиснення в середньому в 3-4 рази а в найкращому випадку до 10 разів, що розширює часовий діапазон реєстрації аварійних, передаварійних та перехідних процесів в електричних мережах.

ІМСКОЕ передані принципові схеми та конструкторська документація цифрового процесора перетворення аналогових сигналів в інтегрально-імпульсний код Галуа, реалізований на ПЛІС фірми ALTERA MAX3000A EPM3064ALC44.

Головний інженер

 Я. Постолук

Старший науковий співробітник

 О.Равлик

Затверджую

Заступник голови правління-  
технічний директор

ВАТ "Івано-Франківськцемент"

Б. Й. Заяць

2003р.

АКТ

впровадження результатів дисертаційної роботи

Яцків Наталії Георгіївни "Методи та засоби стиснення даних в  
розподілених комп'ютерних системах на основі кодів поля Галуа"

На ВАТ "Івано-Франківськцемент" Карпатським державним центром інформаційних засобів і технологій НАН України сумісно з СКБ засобів автоматизації та фірмою "Енеркон" впроваджена комп'ютерна система контролю та обліку витрати енергоносіїв.

В плані вдосконалення названої системи та зменшення об'ємів інформаційних даних, які передаються та реєструються комп'ютерною системою, розроблено програмне забезпечення по результатах дисертаційних досліджень Яцків Н. Г.

При розробці вказаного програмного забезпечення використаний запропонований в дисертаційній роботі метод стиснення даних на основі кодів поля Галуа, в основу якого покладена процедура біт-орієнтованого кодування номерів активних відліків, які характеризують зміни значень витрати енергоносія в певні моменти часу.

В склад програмного забезпечення входить пакет програм: інтерфейс користувача; програма стиснення даних в реальному масштабі часу; програма стиснення даних в пакетному режимі.

Використання розробленого програмного забезпечення в комп'ютерній системі обліку витрати енергоносіїв дало можливість в 2-5 разів зменшити об'єми масивів архівних даних про витрату енергоносіїв в окремих цехах.

Головний метролог-начальник

служби автоматизації і метрології  В. М. Мандрик

Інженер метролог

 В. В. Калин

Затверджую

Проректор з навчальної роботи



Журавель Г.П.

03 2003р.

використання в навчальному процесі результатів дисертаційної роботи Яцків Наталії Георгіївни **“Методи та засоби стиснення даних в розподілених комп’ютерних системах на основі кодів поля Галуа”**

1. Дані про використання наукових результатів дисертаційної роботи Яцків Н.Г. в навчальному процесі Тернопільської академії народного господарства, для спеціальностей “Спеціалізовані комп’ютерні системи” та “Комп’ютерні системи та мережі” приведені в таблиці.

Таблиця

Дисципліна	Розділ в програмі з курсу	Основні результати отримані в дисертаційній роботі
1	2	3
1. Компресія даних	<p>1. Стиснення даних в базисі Галуа. Адаптація в базисі Галуа. Основи вертикальної інформаційної технології в базисі Галуа.</p> <p>2. Система залишкових класів. Метод залишків. Одно каскадне та двокаскадне стиснення даних на основі системи залишкових класів. Кодування багатомірних джерел інформації.</p> <p>3. Методи стиснення даних представлених гармонічними сигналами.</p>	<p>1.1 Дослідження методів стиснення даних в базисі Галуа.</p> <p>1.2. Стиснення даних на основі вертикальної інформаційної технології.</p> <p>2.1. Метод кодування та стиснення звукових повідомлень на основі залишків, що дозволяє зменшити об’єм та захистити дані від несанкціонованого доступу.</p> <p>2.2. Розроблений процесор формування та передавання даних багатомірних об’єктів на основі перетворення Крестенсона – Галуа, який дозволяє зменшити надлишковість кодування в трьохмірному просторі, збільшити швидкість обробки та передавання даних в комп’ютерних системах.</p> <p>3.1. Розроблений метод стиснення даних представлених гармонійними сигналами, який по 1/8 періоду дозволяє відновити сигнал та забезпечує коефіцієнт стиснення <math>k = 5 - 8</math> разів.</p>

1	2	3
2. Низові комп'ютерні мережі	1. Системні властивості об'єктів управління (ОУ). Стационарні, квазістационарні, нестационарні (ОУ).	1.1. Проведено класифікацію станів ОУ: семантичні, технологічні, інформаційні. 1.2. Розроблена система кодування станів джерел інформації.

2. Як додаткова література використані такі роботи Яцків Н.Г.

2.1. Яцків Н.Г. Методи стиснення даних в інформаційно-керуючих системах // Розвідка і розробка нафтових і газових родовищ. Серія: Технічна кібернетика та електрифікація об'єктів паливно-енергетичного комплексу. – Івано-Франківськ: ІФДТУНГ. – 2001. – №37 (том 6). – С.183-186.

2.2. Яцків Н.Г., Яцків В.В. Методи кодування та фізичного представлення сигналів в інформаційно-керуючих системах // Розвідка і розробка нафтових і газових родовищ. Серія: Методи і засоби технічної діагностики. – Івано-Франківськ: ІФДТУНГ. – 2001. – №38 (том 8). – С.140-144.

2.3. Николайчук Я.М., Яцків Н.Г. Методи стиснення даних в багатоканальних системах на основі кодів Галуа // Вісник національного університету "Львівська політехніка". Радіоелектроніка та телекомунікації. – Львів. – 2002. – №443. – С.135-138.

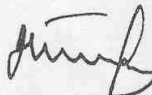
2.4. Яцків Н.Г., Николайчук Я.М. Системні характеристики джерел інформації та оцінка їх ентропії // Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2002. – №3 (том 1). – 197-200.

2.5. Яцків Н.Г. Дослідження системних характеристик методів формування даних в різних теоретико-числових базисах. Вимірювальна та обчислювальна техніка в технологічних процесах: Збірник наукових праць. – Хмельницький: ТУП. – 2002. – №9 (том 2). – С. 132-136.

3. Наукові результати роботи Яцків Н.Г. використано в курсовому та дипломному проектуванні в студентських наукових семінарах та конференціях.

Таким чином, використання в навчальному процесі ТАНГ основних наукових результатів дисертаційної роботи Яцків Н.Г. дозволяє познайомити майбутніх інженерів (спеціалістів, магістрів) з найсучаснішими досягненнями в галузі зменшення надлишковості цифрових даних в комп'ютерних системах та мережах, що підвищує рівень професійної підготовки студентів спеціальності: 7.091501 "Комп'ютерні системи і мережі", 7.091503 "Спеціалізовані комп'ютерні системи".

В.о. директора інституту  
комп'ютерних інформаційних технологій  
к.т.н., доцент



10.02.03

Білоусов І.А.